

**SISTEMA DE INFORMACIÓN WEB PARA APOYAR LA GESTIÓN DE
PROYECTOS EN LA EMPRESA: PROYECTOS MANTENIMIENTO INDUSTRIAL
M&R**

HERNÁN ANTONIO RAMÍREZ OSORIO
hernan.ramirez@correounivalle.edu.co

Universidad del Valle Sede Tuluá
Facultad de Ingeniería Sistemas
Escuela de Ingeniería de Sistemas y Computación
Ingeniería de Sistemas
Tuluá
2016

**SISTEMA DE INFORMACIÓN WEB PARA APOYAR LA GESTIÓN DE
PROYECTOS EN LA EMPRESA: PROYECTOS MANTENIMIENTO INDUSTRIAL
M&R**

HERNÁN ANTONIO RAMÍREZ OSORIO
hernan.ramirez@correounivalle.edu.co

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director: DIANA LORENA VELANDIA VANEGAS. MSc.

Universidad del Valle Sede Tuluá
Facultad de Ingeniería Sistemas
Escuela de Ingeniería de Sistemas y Computación
Ingeniería de Sistemas
Tuluá
2016

AGRADECIMIENTOS

A mi directora de trabajo de grado, MSc. Diana Lorena Velandia Vanegas por su dedicación y esfuerzo, quien con sus conocimientos y experiencia me guió y acompañó en todo este proceso.

A la Universidad del Valle y profesores de la sede, quienes siempre están dispuestos a compartir sus conocimientos, en especial a la coordinadora de programa Yuri Mercedes Bermúdez Mazuera por sus innumerables consejos y constante apoyo al estudiantado.

DEDICATORIA

A mi madre por infundir en mí la lucha y el deseo de superación, quien con su esfuerzo, dedicación y amor me ha impulsado a crecer como ser humano.

A mis hermanos, quienes siempre tuvieron una voz de aliento en los momentos difíciles, a mi familia, a mi novia y demás personas que me brindaron sus palabras de aliento y me ayudaron a alcanzar esta meta tan importante en mi vida.

CONTENIDO

<u>AGRADECIMIENTOS</u>	<u>III</u>
<u>DEDICATORIA.....</u>	<u>IV</u>
<u>GLOSARIO.....</u>	<u>XI</u>
<u>1 PLANTEAMIENTO DEL PROBLEMA</u>	<u>16</u>
<u>2 OBJETIVOS</u>	<u>22</u>
2.1 OBJETIVO GENERAL	22
2.2 OBJETIVOS ESPECÍFICOS	22
<u>3 ORGANIZACION DEL DOCUMENTO</u>	<u>23</u>
<u>CAPÍTULO 1 – MARCO DE REFERENCIA.....</u>	<u>24</u>
<u>4 EMPRESA M&R.....</u>	<u>24</u>
<u>5 GESTIÓN DE PROYECTOS</u>	<u>26</u>
<u>6 CONCEPTOS Y HERRAMIENTAS DE INGENIERIA DE SOFTWARE.....</u>	<u>27</u>
6.1 SISTEMAS DE INFORMACIÓN	27
6.1.1 TIPOS DE SISTEMAS DE INFORMACIÓN	28
6.2 INGENIERÍA DE REQUISITOS	28
6.3 METODOLOGÍA ÁGIL XP (EXTREME PROGRAMMING)	30
6.3.1 FASES DE LA METODOLOGÍA XP	32
6.4 PSP (PERSONAL SOFTWARE PROCESS)	33
6.4.1 PRINCIPIOS Y OBJETIVOS DE PSP	34

6.4.2	ESTRATEGIA DE PSP	34
6.5	PLATAFORMA DE DESARROLLO JAVA.....	35
	APLICACIONES JAVA J2EE.....	35
6.5.1	ORM (MAPEO OBJETO RELACIONAL)	36
6.5.2	DAO (DATA ACCESS OBJECT)	36
6.5.3	JPA Y JPQL	37
6.5.4	EJB	37
6.5.5	MANAGED BEAN.....	37
6.5.6	JSF (JAVA SERVER FACES)	37
6.5.7	PRIMEFACES	38
6.5.8	SERVIDOR DE APLICACIONES GLASSFISH	38
6.5.9	POSTGRESQL	39
7	<u>DESARROLLO DEL PROYECTO</u>	<u>41</u>
7.1	DESARROLLO DE METODOLOGÍA XP	41
7.1.1	FASE EXPLORACIÓN.....	41
7.1.2	PLANIFICACIÓN	43
7.1.3	ITERACIONES	47
7.1.4	PUESTA EN PRODUCCIÓN	53
7.2	HERRAMIENTAS USADAS.....	53
7.3	IMPLEMENTACIÓN DEL PROYECTO	56
	<u>CAPÍTULO 3 – RESULTADOS Y ANÁLISIS</u>	<u>57</u>
8	<u>RESULTADOS</u>	<u>57</u>
8.1	RESULTADOS IMPLEMENTACIÓN PROYECTO EN EMPRESA M&R.....	58
8.2	RESULTADOS USO DE METODOLOGÍA XP CON PSP	62
8.2.1	RESULTADOS CON RESPECTO A LA PRECISIÓN EN LAS ESTIMACIONES DE TIEMPO.....	62
8.2.2	RESULTADOS CON RESPECTO A LA CANTIDAD DE ERRORES POR ITERACIÓN.....	64

<u>CAPÍTULO 4 – CONCLUSIONES Y TRABAJOS FUTUROS.....</u>	<u>66</u>
<u>13 CONCLUSIONES.....</u>	<u>66</u>
<u>14 TRABAJOS FUTUROS.....</u>	<u>67</u>
<u>15 REFERENCIAS BIBLOGRAFICAS</u>	<u>68</u>

LISTA DE FIGURAS

<i>Ilustración 1 Consolidado de horas extras.....</i>	<i>18</i>
<i>Ilustración 2 Organigrama.....</i>	<i>25</i>
<i>Ilustración 3 Metodología XP usada</i>	<i>31</i>
<i>Ilustración 4 Arquitectura de las Aplicaciones Web.....</i>	<i>36</i>
<i>Ilustración 5 Versiones de PSP</i>	<i>45</i>
<i>Ilustración 6 Diagrama de Base de Datos</i>	<i>51</i>
<i>Ilustración 7 Conjunto de Tecnologías Java EE</i>	<i>54</i>
<i>Ilustración 8 Consulta de datos. Anterior vs Actual</i>	<i>59</i>
<i>Ilustración 9 Consolidado de tiempos registros en proyecto</i>	<i>60</i>
<i>Ilustración 10 Detalle ejecución de proyecto</i>	<i>61</i>
<i>Ilustración 11 Detalle de esfuerzo por parte del personal en un proyecto</i>	<i>61</i>
<i>Ilustración 12 Detalle de material usado en ejecución de proyectos.....</i>	<i>62</i>
<i>Ilustración 13 estimación de esfuerzo vs el tiempo real iteración 1</i>	<i>63</i>
<i>Ilustración 14 Estimación de esfuerzo vs el tiempo real iteración 4</i>	<i>64</i>
<i>Ilustración 15 Errores y mejoras en funcionalidades</i>	<i>65</i>

LISTA DE TABLAS

<i>Tabla 1 Proceso de gestión de proyectos.....</i>	<i>21</i>
<i>Tabla 2 Límites de postgresSQL</i>	<i>40</i>
<i>Tabla 3 Plantilla Historia de Usuario para Crear Cotización.....</i>	<i>42</i>
<i>Tabla 4 Plan de Entregas Iteración 1.....</i>	<i>44</i>
<i>Tabla 5 Estimación de tiempo por Historia</i>	<i>46</i>
<i>Tabla 6 PSP 0. Estimación de Esfuerzo Iteración 1</i>	<i>47</i>
<i>Tabla 7 Tareas de Ingeniería para Iteración 1 - Historia Crear Proveedor</i>	<i>49</i>
<i>Tabla 8 Tarjeta CRC para Proveedor Vista</i>	<i>50</i>
<i>Tabla 9 Plantilla Pruebas Funcionales por Historia</i>	<i>52</i>
<i>Tabla 10 Objetivo vs Resultados Obtenidos.....</i>	<i>58</i>

LISTA DE ANEXOS

- Anexo 1 – Fase de exploración.
- Anexo 2 – Fase de planeación.
- Anexo 3 – Estimación de esfuerzo.
- Anexo 4 – Estándar de codificación.
- Anexo 5 – Estándar de base de datos.
- Anexo 6 – Estándar de mensajes.
- Anexo 7 – Tareas de ingeniería.
- Anexo 8 – Tarjetas CRC
- Anexo 9 – Modelo base de datos.
- Anexo 10 – Pruebas funcionales.
- Anexo 11 – Manual de instalación.
- Anexo 12 – Carta de aceptación.
- Anexo 13 – Pruebas de aceptación.

GLOSARIO

STAKEHOLDERS

Cualquier grupo o individuo que pueda afectar o ser afectado por el logro de los propósitos de una corporación. Stakeholders incluye a empleados, clientes, proveedores, accionistas, bancos, ambientalistas, gobierno u otros grupos que puedan ayudar o perjudicar la corporación. [1]

ELICITACIÓN DE REQUISITOS

La Elicitación de Requisitos –ER– es la ciencia y la disciplina que se ocupa de establecer y documentar los requisitos del software. La versión 1.2 del Capability Maturity Model Integration-Development –CMMI– divide los requisitos del software en dos áreas de proceso: la primera es el Desarrollo de Requisitos, que incluye la elicitación, la definición, el análisis, la especificación y la validación; la segunda es la Gestión de Requisitos, que implica la gestión de requisitos que se han desarrollado, incluyendo el control de cambios y la verificación.[2]

La Elicitación de Requisitos –ER– se considera como la primera etapa en el proceso de abstraer una comprensión del problema que se quiere resolver con el producto software. Se trata, esencialmente, de una actividad humana donde se identifican las partes interesadas y se establecen las relaciones entre el comprador, el cliente, los usuarios y el equipo de desarrollado. [2]

MANIFIESTO ÁGIL

El manifiesto ágil es un documento que resume en cuatro valores y doce principios las mejores prácticas para el desarrollo de software, basados en la experiencia de 17 industriales del software, en procura de desarrollos más rápidos conservando su calidad. [3]

FRAMEWORK

Un framework es un conjunto de bibliotecas, herramientas y normas a seguir que ayudan a desarrollar aplicaciones. Los frameworks los desarrollan los programadores de sistemas. Un framework está compuesto por varios segmentos/componentes que interactúan los unos con los otros. Las aplicaciones

pueden escribirse de manera más eficaz si utilizamos un framework adaptado al proyecto. Un framework java proporciona un conjunto de características a partir de una implementación de objeto.[4]

JAVA SERVER FACES

JavaServer Faces (JSF) es un framework para la creación de aplicaciones java enfocadas a la Web. Simplifica el desarrollo de interfaces de usuario mediante un enfoque de componentes centralizados. [5]

Servidor de Aplicaciones Glassfish

GlassFish es un servidor de aplicaciones desarrollado por Sun Microsystems que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación[6].

POSTGRESQL

PostgreSQL es un sistema de gestión de base de datos relacional de objetos (ORDBMS), Desarrollado en la Universidad de California en Berkeley Departamento de Ciencias de la Computación.[7]

ORDEN DE TRABAJO ODT

Una orden de trabajo ODT es una solicitud que se genera para la solución de una incidencia, la prestación de un servicio, la ejecución de un mantenimiento o la realización de un proyecto. Las órdenes de trabajo serán recibidas por los técnicos encargados de realizar dichas actividades.[8]

FINE TUNING

Hacer cambios muy pequeños a algo con el fin de hacer que funcione lo mejor posible.

RESUMEN

El presente trabajo de grado, tuvo como propósito analizar la condición en que se encontraba la empresa *Proyectos – Mantenimiento Industrial M&R* e implementar una solución tecnológica a las necesidades encontradas en cuanto a la gestión de proyectos. El trabajo inicia con un diagnóstico general de los procesos por los cuales pasa un proyecto en la empresa, detectando puntos críticos y planteando soluciones adecuadas para estos.

A continuación se define una herramienta tecnológica, la cual apoya todo el proceso de gestión de proyectos en la empresa, aportando de esta manera una ventaja competitiva y mejores bases para proyectarse de una mejor manera a un mercado más amplio como el nacional.

Finalmente, y para dar cumplimiento a los objetivos planteados, se opta por el uso de la metodología ágil XP, complementado con PSP el cual aporta métodos que muestran como planificar, medir y administrar el trabajo.

ABSTRACT

This thesis, was aimed to analyze the condition was *Proyectos – Mantenimiento Industrial M & R* and implement a technological solution to the competing needs in terms of project management. The work begins with a general diagnosis of the processes through which passes a project in the company, identifying critical points and suggesting suitable for these solutions.

Below is a technological tool is defined, which supports the entire process of project management in the company, thus providing a competitive advantage and better base to project a better way to a broader market as the national.

Finally, to comply with the objectives, you opt for the use of agile methodology XP, supplemented with PSP which provides methods that show how to plan, measure and manage the work.

INTRODUCCIÓN

El aumento de los proyectos es una realidad que cada día se hace más evidente en escalas mundiales y que actualmente se refleja a nivel nacional, mostrando un crecimiento continuo en los últimos años, como lo describe el artículo de la revista Portafolio en el 2014 “La industria, el comercio y las empresas de hidrocarburos aportaron el 91,9% de las ventas.” [9], lo anterior complementado con el periódico el país, el cual hace énfasis en resultados obtenidos en encuestas recientes realizadas por el DANE: “En enero de 2016 la producción real de la industria colombiana presentó un aumento de 8,2 % con relación al mismo mes de 2015, y mantuvo la tendencia positiva por octavo mes consecutivo.”[10], cifras que resaltan el buen momento por el que pasa la industria a nivel nacional.

Es por esto que se hace necesario que grandes, medianas y pequeñas empresas replanteen la manera de ejecutar los proyectos que se desarrollan actualmente y hagan uso de herramientas que apoyen los procesos involucrados, pues el constante flujo de actividades, mayores requerimientos y la necesidad de controlar cada fase en el desarrollo de un proyecto lo hace indispensable.

De acuerdo con lo anterior, son evidentes las bondades que trae consigo implementar un sistema que apoye la gestión de proyectos, pues aporta directamente a incrementar las ventajas competitivas de una empresa, agilizando procesos, garantizando una adecuada ejecución y minimizando los tiempos de muchas tareas que se realizan de manera sistemática.

En este proyecto se tomó como referencia una empresa del municipio de Tuluá llamada *Proyectos – Mantenimiento industrial M&R*, la cual se encuentra en un proceso de constante crecimiento, proyectándose así en mercados más extensos

y competitivos como el nacional. Así pues, se debió analizar su condición en cuanto al manejo dado a la realización de proyectos, abordando procesos como: La planificación y control con respecto a los empleados; planificación de materiales y posterior a esto el control del proyecto donde se efectúa la integración de los procesos anteriores y se lleva a cabo la ejecución iniciando con la cotización respectiva y finalizando con el seguimiento a las actividades realizadas. Teniendo en cuenta que en dicho momento la empresa no contaba con ningún tipo de apoyo tecnológico que gestionara cada proceso involucrado en el desarrollo de un proyecto, se debió plantear una solución adecuada a sus necesidades, para lo cual se abordaron diferentes puntos en el desarrollo de este proyecto, iniciando con el planteamiento al problema donde se analizó la condición de la empresa, seguido a esto, se definieron objetivos esperando resultados puntuales que conllevaron a la solución de dicho problema; además, se tomó como base algunos estudios realizados que demuestran el incremento reportado en los proyectos a nivel mundial, justificando así todo el desarrollo.

Finalmente, se especificaron claramente las funcionalidades de dicho sistema, el cual se desarrolló bajo la metodología de desarrollo ágil Xp (extreme Programming) debido a los requisitos cambiantes previstos inicialmente y tomando como referencia también el uso de PSP (Personal Software Process).

1 PLANTEAMIENTO DEL PROBLEMA

Proyectos – Mantenimiento industrial M&R es una empresa Tulueña con más de 10 años en el mercado, dedicada al diseño, fabricación, montaje y mantenimiento en sectores industriales de la región; la cual, gracias a los buenos resultados obtenidos se encuentra en un proceso de constante crecimiento que la proyecta en el mercado a nivel nacional.

Dada la proyección que ha tomado la empresa actualmente, es razonable pensar que se ven enfrentados a un mercado cada vez más competitivo, y al desarrollo de una cantidad más amplia de proyectos, los cuales cuentan con mayores exigencias y menos tiempo de entrega; representando para la empresa, innumerables situaciones de riesgo en cuanto a retrasos de actividades y posibles costos adicionales como pago de horas extras y multas impuestas por el cliente.

Dicho lo anterior, se pueden destacar diferentes procesos o áreas de la empresa que se ven involucrados en el desarrollo de cada proyecto, como son:

Planificación y control de personal, para lo cual es necesario contar con datos importantes como: Habilidades, desempeño en la empresa, proyectos en los que ha participado, información básica, entre otros. Datos que son utilizados para nuevas contrataciones o asignación de proyectos al personal actual. Llevar a cabo una mala planificación en este punto, significa para la empresa asimilar gastos adicionales en diferentes momentos como son: Falta de contrataciones, exceso de contrataciones y mal asignación de tareas al personal actual.

Planificación de materiales, el material es un elemento esencial en el desarrollo de un proyecto, pues no se puede llevar a cabo si no se cuenta con el requerido; por lo que es necesario conocer con que dispone la empresa evitando desaprovechamiento de materiales, como también, conocer que se debe comprar evitando posibles retrasos en una fase futura, teniendo en cuenta que algunos pedidos pueden tardar en ser entregados.

Control de proyectos, siendo fundamental para llegar a un proyecto exitoso, es necesario planificar y controlar cada una de las fases por las que cruza un

proyecto: Iniciando con la elaboración de la cotización, verificando materiales necesarios y proveedores; posteriormente, se inicia la ejecución donde se debe tener en cuenta: los tiempos, el estado actual y las tareas pendientes con su responsable. Y la terminación de éste, donde se cierra documentalmente dicho proyecto, almacenando un respectivo historial con información importante como: planos, cotizaciones, entre otros; que son necesarios para futuros proyectos.

Inicialmente, los procesos nombrados en el punto anterior eran manejados manualmente por el gerente de la empresa, el cual estaba obligado a tener presente cada uno de los proyectos que se estaban elaborando y los requerimientos de estos; asumiendo altos riesgos en la planificación, ejecución y seguimiento, que sumado al constante crecimiento de la empresa significaba una carga laboral muy alta para dicha persona. Asimismo, el análisis de datos en el desarrollo de un proyecto se hacía complejo, pues no se contaba con ningún control o herramienta que ayudara a realizarlo fácilmente. Se realizó manualmente un consolidado de horas extras trabajadas en la empresa en donde se obtuvieron los datos que se muestran en la ilustración 1.

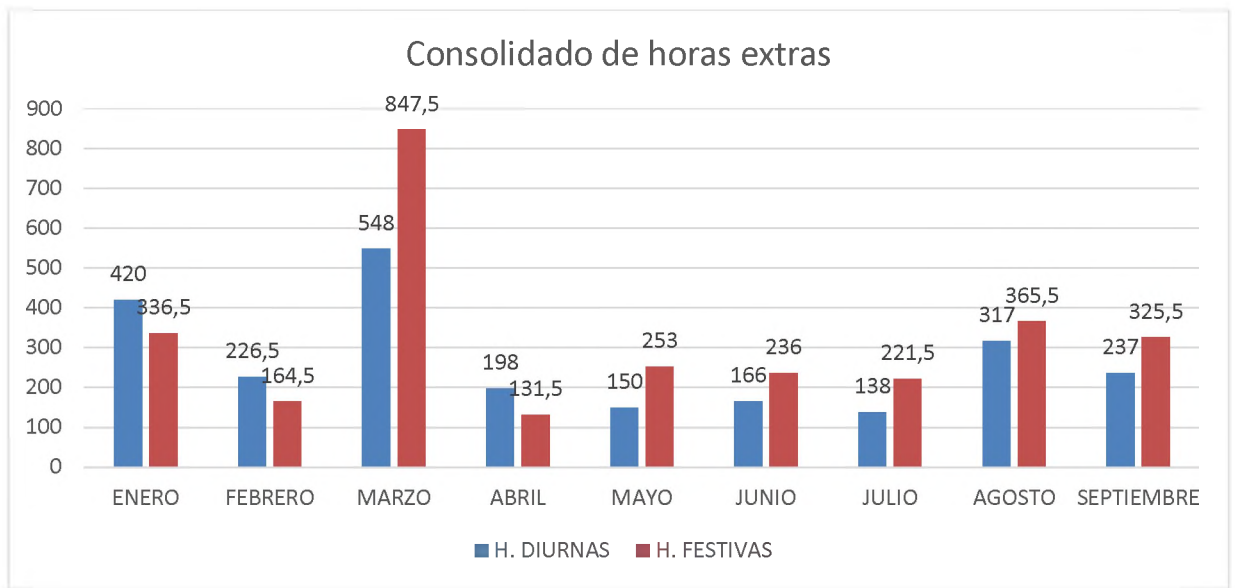


Ilustración 1 Consolidado de horas extras

Fuente: Elaboración propia

Es evidente el exceso de horas trabajadas en la empresa y la dificultad de detectar el proyecto al que se le dedico más esfuerzo laboral, datos importantes al momento de tomar medidas correctivas y ejecutar una mejor planeación en proyectos futuros.

Adicional a esto, era necesario llevar un control sobre los proyectos que se desarrollan en la empresa y las posibles acciones que se deben tomar al respecto, teniendo en cuenta que era complejo determinar acertadamente los responsables de un defecto de fabricación, pues no se contaba con una adecuada trazabilidad que integrase cada una de las fases en el desarrollo, y de esta manera poder analizar en qué punto ocurre el fallo y que lo ocasiono: Un mal proceso realizado por parte del empleado, material defectuoso, mal proceso empleado por parte del encargado, etc.

Es evidente que debido al crecimiento de la empresa, se materializa un aumento significativo de proyectos, lo cual resultaba en una situación crítica pues no se

contaba con herramientas para llevar un alto flujo de trabajo; herramientas que llevaran un proceso organizado y sistematizado controlando cada proyecto y sus respectivos lineamientos, además, llevar a cabo algunas actividades definidas, como generación de formatos, remisiones, ordenes de trabajo, manejo de material, etc. Dicho lo anterior, es cierto que la empresa estaba expuesta a riesgos económicos asumiendo gastos adicionales como se nombró.

Finalmente, y teniendo en cuenta los aspectos nombrados, se implementa una solución informática que apoye la gestión de proyectos con un debido seguimiento, agilizando procesos y llevando un control total sobre la historia de los proyectos realizados, aliviando cargas en el personal e impulsando a la empresa a un mejor mercado. En la tabla 1 se muestran los principales procesos manejados por la empresa y como se desarrollaban antes de la implementación del sistema propuesto.

Proceso	Desarrollo
Cotización	La elaboración de la cotización se realizaba en un documento de Excel basado en un formato propio de la empresa, diligenciándose manualmente por el encargado del proyecto. En el documento se debía especificar: 1. Materiales: Cantidad y precio de los materiales necesarios, se calculaba el precio manualmente tomando como referencia el valor de la compra, pero si no se tenía presente dicho valor, se debía calcular según la experiencia del cotizante. Además, en algunos casos había inseguridad en la existencia de dicho material. 2. Estimación de tiempo: Se tiene en cuenta el oficio a realizar, el riesgo del trabajo y la cantidad de

	horas requeridas. 3. Gastos adicionales: Se estiman gastos extra como: Transporte, consumibles, entre otros. Finalmente, con la suma de todos los elementos se obtiene el valor total de la cotización.
Planificación de personal	Según los requerimientos y necesidades del proyecto, el encargado debe verificar que empleados son más aptos para desenvolverse en las actividades correspondientes; decisión que era tomada según su perspectiva, lo cual traía inconvenientes si el encargado fuese una persona que no conociese el personal. El personal se tiene en cuenta para la ejecución del proyecto y si fuese necesario se contratan nuevos empleados para apoyar el desarrollo.
Planificación de material.	El encargado del proyecto debe tener en cuenta la disponibilidad de los materiales necesarios en el desarrollo, realizando los pedidos correspondientes a los proveedores y evitando posible desabastecimiento en fases críticas.
Seguimiento de proyecto.	Se debe realizar un seguimiento sobre el desarrollo del proyecto, analizando factores como: Estado del proyecto, días restantes para la entrega, horas de trabajo invertidas y empleados que han trabajado en el proyecto. Comprobando de esta manera la correcta ejecución del proyecto lo que proporciona fundamentos para tomar decisiones correctivas y garantizar una terminación exitosa.
Almacenamiento de documentos de proyecto.	Al finalizar el proyecto se debe almacenar documentos propios, como: Cotización, planos, ODT, empleados y sus actividades dentro del proyecto, además de tener en cuenta

	el desarrollo del mismo para proyectos similares en un futuro.

Tabla 1 Proceso de gestión de proyectos

Fuente: Elaboración propia.

2 OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar un sistema de información que apoye la gestión de proyectos integrando módulos para la gestión de personal y planificación de materiales, teniendo en cuenta procedimientos y formatos propios en la empresa: Proyectos-Mantenimiento industrial M&R.

2.2 OBJETIVOS ESPECÍFICOS

- Analizar los procesos que se llevan a cabo en el desarrollo de un proyecto en la empresa M&R.
- Diseñar los procesos que permitan la gestión de personal, materiales y control en la empresa de acuerdo a sus necesidades específicas.
- Desarrollar el sistema que permita apoyar la gestión de proyectos en la empresa.
- Implantar el sistema en la empresa M&R, validando su correcto funcionamiento.

3 ORGANIZACION DEL DOCUMENTO

A continuación, en el Capítulo I se describen los conceptos teóricos que aportan al trabajo: ingeniería de requisitos, metodología ágil XP, PSP Personal Software Process, Framework JSF Prime Faces, Arquitectura de Aplicaciones Multinivel, Glassfish. También se describen algunos sistemas y/o trabajos relacionados y sus diferencias y aportes con respecto a este trabajo. En el Capítulo 2 se presenta el desarrollo del proyecto, donde se describe la metodología utilizada incluyendo la fase de pruebas e implementación. En el Capítulo 3 se describen los resultados obtenidos en cada uno de los objetivos propuestos. En el Capítulo 4 se presentan las conclusiones a las que llega el trabajo, además de los trabajos futuros que se pueden emprender. Finalmente, se presenta la lista de referencias bibliográficas que soportan el desarrollo del trabajo.

CAPÍTULO 1 – MARCO DE REFERENCIA

En esta sección se presenta el marco teórico del trabajo, la cual aborda 4 temas: en primer lugar se lleva a cabo la revisión acerca del estado actual de los procesos en la empresa M&R. En segundo lugar se abordan las bases conceptuales del proceso de gestión de proyectos. En tercer lugar se aborda las bases conceptuales de los modelos y herramientas usados para el desarrollo del proyecto como son, los sistemas de información, la ingeniería de requisitos y su importancia en el desarrollo de software, la metodología ágil Xp y el proceso personal de software (PSP) y finalmente el uso de plataforma Java con el Framework JSF PrimeFaces.

4 EMPRESA M&R

Proyectos – Mantenimiento industrial M&R es una empresa nacida en la ciudad de Tuluá, que inicia sus operaciones el 31 de Agosto del 2005, contando actualmente con más de 10 años en el mercado local y proyectándose en los últimos años a un mercado más amplio como el nacional.

Desde sus inicios, se empeñaron por ser especialistas en el área del diseño, fabricación, montaje y mantenimiento en las zonas industriales de la región, destacando clientes como: Nestle S.A, comestibles la rosa, Harinas del Valle, Levapan, entre otras, para los cuales se ha hecho un proveedor de primera mano en cuanto al desarrollo de proyectos.

Actualmente, se ubican en la calle 19 # 29-20 de Tuluá, donde cuentan con un local con más de 400 m², en el cual centralizan sus operaciones apoyadas en un equipo de profesionales garantizando así la calidad en el desarrollo de las actividades.

En la ilustración 2 se muestra el organigrama de la empresa, en donde se destaca el jefe de taller, que es el principal responsable de los proyectos en donde están involucrados los operarios.

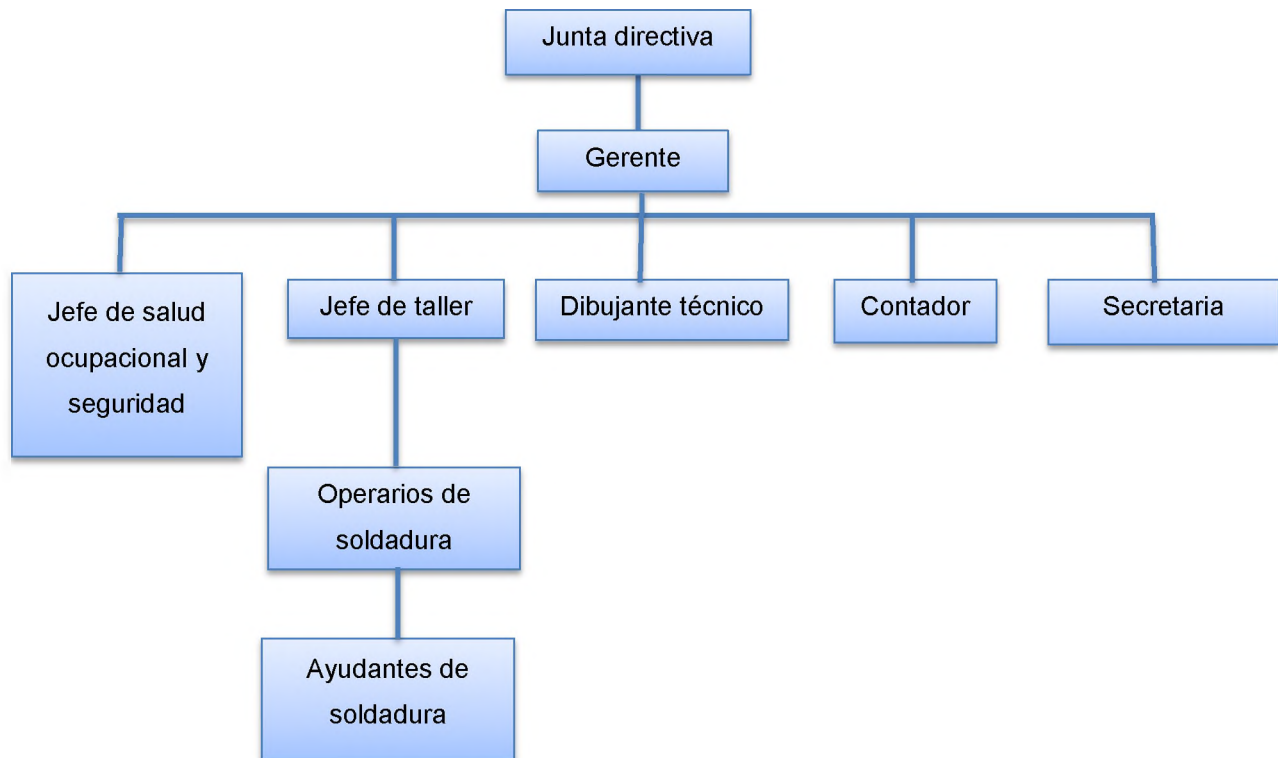


Ilustración 2 Organigrama

Fuente: Elaboración propia.

5 GESTIÓN DE PROYECTOS

La gestión de proyectos es un enfoque metódico para planificar y orientar los procesos del proyecto de principio a fin. Según el Instituto de Gestión de Proyectos (Project Management Institute, PMI), los procesos se guían por cinco etapas: iniciación, planificación, ejecución, control y cierre.[11]

Así pues, se entiende la gestión de proyectos como un método de gran importancia en las empresas para llegar a proyectos exitosos y rentables encaminándolos a ser más competitivos en un mercado que se mueve constantemente.

Internamente, la empresa lleva un proceso para el desarrollo de proyectos, que aunque no lo tengan estandarizado siempre cumple con los mismos pasos:

- **Iniciación:** Teniendo el proyecto, el encargado analiza y diseña una solución al problema o necesidad que plantea el cliente, para lo cual debe estimar un tiempo para el desarrollo y montaje del mismo, teniendo en cuenta los empleados con los cuales dispone.
- **Planeación:** Se debe analizar el tiempo estimado con los recursos disponibles (espacio, tiempo, mano de obra y materia prima). Si no se cuenta con lo necesario para cumplir con la meta propuesta, se debe iniciar la búsqueda según corresponda, materiales: Analizar el proveedor con los mejores precios y tiempos de entrega, para realizar la compra pertinente; mano de obra: Analizar los empleados cuyas habilidades se ajusten mejor a los requerimientos del proyecto para posibles contrataciones. Anteriormente, este proceso se realizaba de manera manual, por lo cual el encargado debía conocer las habilidades de cada empleado, además de buscar en la empresa que materiales estaban disponibles, pues no se contaba con un inventario.
- **Ejecución:** Partiendo de los recursos asignados al proyecto, se inicia su desarrollo asignando actividades a los empleados, y disponiendo de la suficiente materia prima existente en la empresa. Dichos empleados deben

realizar diferentes trabajos supervisados por el encargado del proyecto o encargado de taller si es necesario, quienes proveerán de planos e información suficiente para un buen desarrollo del proyecto. Inicialmente, no se contaba con una herramienta de ayuda para registrar cada actividad realizada en la ejecución del proyecto, por lo que el encargado debía tener en cuenta cada uno de los pasos que se realizaron.

- **Control:** El encargado del proyecto o jefe de taller, realiza constantes revisiones a las actividades desarrollados por los empleados garantizando de esta manera la calidad y cumplimiento de los requerimientos del cliente.
- **Cierre:** El encargado del proyecto se encarga de almacenar la información disponible del desarrollo del proyecto. Anteriormente se almacenaban planos físicos en un archivo del cual dispone la empresa para futuros desarrollos, además de almacenar la cotización en el historial de correos electrónicos enviados.

6 CONCEPTOS Y HERRAMIENTAS DE INGENIERIA DE SOFTWARE

6.1 SISTEMAS DE INFORMACIÓN

Es el conjunto formal de procesos que operando sobre una colección de datos estructurada de acuerdo con las necesidades de una empresa, recopila, elabora y distribuye la información necesaria para la operación de dicha empresa y para las actividades de dirección y control correspondientes, apoyando, al menos en parte, los procesos de toma de decisiones necesarios para desempeñar las funciones de negocio de la empresa de acuerdo con su estrategia [12]. Existen varios tipos de sistemas de información como: sistemas de procesamiento de transacciones, sistemas de información gerencial, sistema de soporte a decisiones, sistema de información ejecutiva [13].

6.1.1 Tipos de sistemas de información

- **Sistema de información gerencial:** este sistema resulta de la interacción entre los diferentes involucrados en procesos dentro de una organización tales como: personas, tecnologías y procedimientos; estos sistemas son orientados a dar solución a problemas empresariales.
- **Sistema de procesamiento de transacciones:** su función es recolectar, almacenar y recuperar la información generada por transacción producida dentro de una organización, una transacción es considerado como un evento que genera o actualiza datos que están almacenados.
- **Sistema de soporte a la toma de decisiones:** es un sistema de apoyo, más no que automatiza la toma de decisiones, una decisión es la elección entre varias opciones basadas en estimaciones de los valores de las mismas, un sistema de soporte a la toma de decisiones pretende ayudar a un grupo de personas con un objetivo común a reunir inteligencia, generar alternativas o caminos y de esta manera tomar decisiones.
- **Sistema de información ejecutiva:** es una herramienta que está orientada para usuarios de nivel gerencias, permite monitorear el estado de las variables de un área de una empresa, partiendo de la información interna y externa de la misma. [14]

6.2 INGENIERÍA DE REQUISITOS

Con el fin de precisar y llegar a un entendimiento más preciso de lo que es y representa la ingeniería de requisitos, a continuación se definirá que es un requisito. Existen varias definiciones, entre las que se pueden citar:

- Una condición o capacidad requerida por un usuario para resolver un problema.

- Una condición o capacidad que debe alcanzar o poseer un sistema o componente para satisfacer un contrato, un estándar, una especificación o un documento formalmente impuesto.
- Una forma documentada de 1 o 2 [15]

Las actividades en Ingeniería de Requisitos (IR) son altamente dependientes de las decisiones humanas y esto hace difícil automatizarlas. Sin embargo los analistas de *software* están siendo cada vez más conscientes de los beneficios que se pueden obtener del soporte automatizado en la Ingeniería de Requisitos [16].

Al comienzo de un proyecto de *software*, la información debe capturarse rápidamente y de manera comprensiva. Es vital que el ingeniero pueda captar la mayor cantidad de información ofrecida por el representante de los clientes que sea posible en periodos cortos de tiempo [17]. Esta actividad se conoce como especificación de requisitos. Existen especificaciones de requisitos que varían en estructura, contenido y en formatos de representación. Una de las razones para esto es la existencia de muchos estándares y guías que definen el contenido y la estructura de una “buena” especificación de requisitos en diferentes formas. Otra razón es la influencia del método o métodos usados en el proceso de Ingeniería de Requisitos [18].

Otra alternativa para capturar requisitos rápidamente y con más claridad es XP, que es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo. En cuanto a la fase de requisitos en XP se trabajan las historias de usuario. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que debe poseer el sistema, sean requisitos funcionales o de calidad [19].

6.3 METODOLOGÍA ÁGIL XP (eXtreme Programming)

Las metodologías ágiles nacen con el objetivo de permitir a los equipos desarrollar *software* rápidamente, además responder a los cambios que se presentan a lo largo del proyecto. Las metodologías ágiles se basan en el manifiesto ágil teniendo siempre como características la inclusión de los clientes en el proceso de desarrollo. Permite centrarse más en los procesos de desarrollo por medio de cronogramas flexibles, sin dar mucha relevancia a la documentación. Estas metodologías son impuestas por el equipo de desarrollo, es decir son amoldadas a la organización con menos control. El cliente se define como un miembro más del equipo de desarrollo, los cuales son pequeños, no superiores a 10 integrantes, trabajando todos en el mismo sitio. Existen pocos roles y no se enfatiza tanto en la arquitectura del *software* [20]. Las metodologías ágiles están centradas en potenciar las relaciones interpersonales como punto clave para el éxito en el desarrollo de *software*, estas promueven el trabajo en equipo y en la fase de requisitos trabajan con historias de usuario [21]. Lo cual implica que la comunicación constante entre usuario y desarrollador es vital para describir claramente las necesidades y objetivos del proyecto [22].

Las metodologías ágiles buscan el beneficio común tanto del equipo de desarrollo como del cliente. El cliente es un miembro más del equipo en estas metodologías, ya que es quien sabe cómo funcionan los procesos y qué es lo que necesita. La documentación en las metodologías ágiles es más dinámica, está basada en la comunicación cara a cara con el cliente, evitando intermediarios. Estas metodologías ágiles buscan aumentar la productividad al generar entregables al cliente en menor tiempo. El avance del proyecto se mide con mayor confiabilidad en el número de requisitos aprobados por el *stakeholder* e implementados. En este tipo de metodologías, se habla de actividades en lugar de etapas [23].

La metodología ágil XP propone un ciclo de vida dinámico soportado en ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo [24]. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas (ver ilustración 3).



Ilustración 3 Metodología XP usada

Fuente: Elaboración Propia

XP incluye como estrategia de elicitación de requisitos las historias de usuario, las cuales describen una funcionalidad que es de valor para el usuario o cliente de un *software*. La historia de usuario solamente debe describir comportamientos externos del sistema que puedan ser entendidos por el cliente. Las historias de usuario se componen de tres aspectos [25]:

- Una descripción de la historia, utilizada para realizar la planificación o, como recordatorio de funcionalidades pendientes.

- Conversaciones acerca de la historia que sirven para profundizar en los detalles de los procesos.
- Pruebas que especifican detalles del documento y que se puede utilizar para determinar cuándo una historia ha sido completada.

En este trabajo se abordó la metodología ágil XP, por ser una metodología de desarrollo de *software* liviana, de bajo riesgo y flexible. Se destaca ante las demás por ser concreta, ofreciendo retroalimentación en ciclos cortos. Es de enfoque incremental, permite cronogramas flexibles respondiendo a las necesidades cambiantes del negocio. Maneja una comunicación directa con el cliente y los *stakeholders*. Está diseñada para trabajar con proyectos que serán desarrollados por equipos pequeños de programadores donde se efectúan desarrollos que agregan valor y que pueden ser probados incluso en un día [26].

6.3.1 Fases de la Metodología XP

- **Exploración:** en esta fase, el cliente define lo que necesita mediante la redacción de “historias de usuarios” a alto nivel. Los programadores estiman los tiempos de desarrollo de cada actividad definida en la historia, con base en esta información [27]. En esta fase las estimaciones realizadas deben ser preliminares, dado que la especificación es de muy alto nivel y podrían variar cuando se analicen más en detalle para cada iteración.
- **Planificación:** En esta fase los *stakeholder* y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a estas, las entregas. Estas entregas se especifican en un documento denominado Plan de Entregas, o *Release Plan*. Además en esta fase, cuando el desarrollador no tiene claro el requisito, realiza pequeños programas de prueba denominados spikes, para validar con el cliente la funcionalidad con el fin de reducir riesgos y facilitar la estimación.

Estos spikes son desechados en la fase de iteración, por lo que se dedica tiempo a la implementación de los entregables.

- **Iteraciones:** es la fase donde cada historia de usuario definida, se debe traducir en tareas específicas de programación de funcionalidades, que generan los entregables. Debido a que las historias de usuario no se definen con suficiente detalle, en la fase de exploración, entonces se realiza un proceso de análisis con el cliente para especificar elementos como: restricciones o datos necesarios para las funcionalidades requeridas.
- **Puesta en Producción:** implementación de las funcionalidades y realización de pruebas que puedan generar modificaciones o ajustes, denominados en esta metodología como *fine tuning*.

En la metodología XP se utiliza la práctica *Release Plan*, la cual es un proceso que crea un plan de alto nivel que cubre un periodo de más de una iteración de desarrollo. Este plan permite al cliente y al equipo de desarrollo decidir qué será desarrollado y cuánto tiempo tardará para tener una versión demostrable del producto. El *Release Plan* agrupa un conjunto de historias de usuario que dan valor a una parte del negocio y que deberán ser analizadas cada una por tareas [28].

6.4 PSP (Personal Software Process)

La disciplina del PSP provee un marco estructurado para desarrollar habilidades personales y métodos que se necesitarán más adelante para ir forjando al ingeniero de software. Es importante que la calidad del software desarrollado abarque hasta el más mínimo detalle, por muy pequeño que éste sea, ya que si no se hace así, puede dañar el sistema entero [29].

El Personal Software Process, PSP, indica a los profesionales de software cómo:

- Administrar la calidad de los proyectos de desarrollo de software.
- Reducir defectos en los productos.
- Estimar y planear el trabajo.

PSP, fue diseñado para ayudar al ingeniero de software a hacer bien el trabajo, aplicar métodos avanzados de ingeniería en su labor diaria, a utilizar métodos detallados de planificación, estimación y control del rendimiento frente a los tiempos planeados.

PSP es la disciplina del trabajo con alta calidad. El trabajo del ingeniero de software según PSP, se puede resumir en planificar el trabajo, hacer el trabajo de acuerdo al plan y producir productos de máxima calidad [30].

6.4.1 Principios y Objetivos de PSP

El PSP es un conjunto ordenado de procesos definidos que orientan a los ingenieros de software a medir, evaluar y monitorear la manera de hacer sus tareas. Los principales objetivos del PSP son [30]:

- Mejorar las estimaciones.
- Mejorar la planeación y acompañamiento de cronogramas.
- Proteger contra el exceso de compromisos.
- Crear un compromiso personal con la calidad.
- Compromiso del desarrollador en la mejora continua del proceso de desarrollo.
- Aumento de la calidad a través de la reducción de la incidencia de errores.
- Mayor precisión en las estimaciones de tamaño del software y tiempo de desarrollo.

6.4.2 Estrategia de PSP

El modelo PSP está dividido en niveles, implantados de manera incremental. Los niveles superiores adicionan características a los niveles ya implantados lo que

minimiza el impacto de los cambios en los hábitos del desarrollador. Este deberá tan sólo adaptar nuevas técnicas a las ya existentes y conocidas. Lo más importante en el proceso de aprendizaje son los datos recogidos después de cada fase, pues con base en los resultados obtenidos en la fase actual se propone mejorar el desempeño personal para la siguiente fase. [30].

Teniendo en cuenta que este trabajo se desarrolló bajo una metodología de desarrollo ágil que exige tiempos de desarrollo muy estrechos para lo cual propone programación en parejas como herramienta para cumplir con objetivos planeados de manera efectiva; es por esto que se decide adoptar como complemento PSP aprovechando los elementos para el constante mejoramiento personal en el desarrollo de software de calidad trazando mejoras continuas en los procesos empleados.

6.5 PLATAFORMA DE DESARROLLO JAVA

Aplicaciones Java J2EE

Java Enterprise Edition, define un conjunto de estándares que habilita soluciones para manejo y desarrollo de aplicaciones empresariales multicapa que son centralizadas en el servidor. Su fin principalmente es ser utilizado en desarrollo de aplicaciones empresariales, ya que sus especificaciones y funcionalidades son orientadas a negocios. Sus grandes ventajas son: robustez, fiabilidad, estabilidad y seguridad. El objetivo principal es simplificar las aplicaciones empresariales, basándolas en componentes modulares y estandarizados, para proveer un completo conjunto de servicios[31].

En la ilustración 4 se muestra la arquitectura de las aplicaciones web, el primer nivel que corresponde al cliente y la vista, es donde está la interfaz de usuario y se realiza una iteración con el mismo (en aplicaciones o en páginas web). El segundo nivel que es la lógica, está referido habitualmente a algún tipo de programa o

script y es el encargado de realizar todas las validaciones desde los procesos manejados en el sistema, según reglas de negocio establecidas por los stakeholder. El tercer nivel, denominado de persistencia o de objetos de acceso a datos, son los encargados de permitir la interacción con la base de datos a través del mapeo de clases generadas en la capa de modelo.

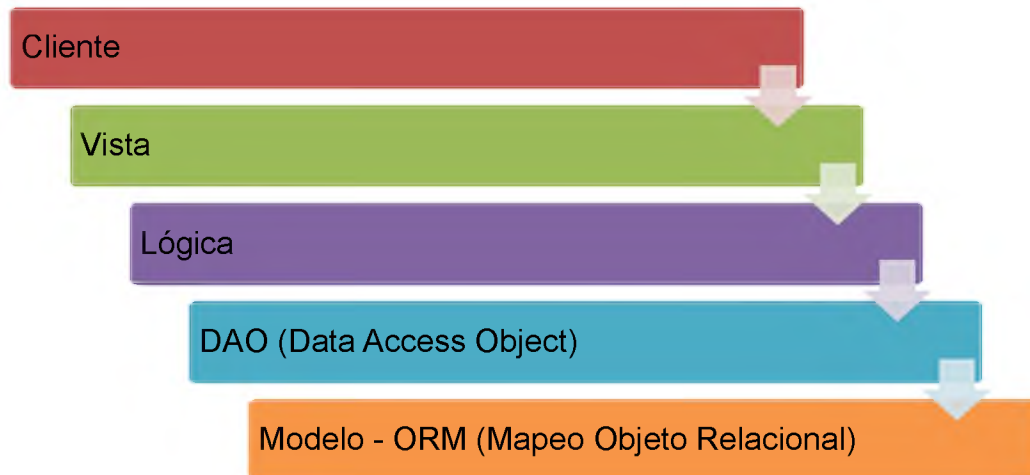


Ilustración 4 Arquitectura de las Aplicaciones Web

Fuente: Elaboración Propia.

6.5.1 ORM (Mapeo Objeto Relacional)

El mapeo objeto-relacional es una técnica de programación para convertir datos del sistema de tipos utilizado en un lenguaje de programación orientado a objetos al utilizado en una base de datos relacional. En la práctica esto crea una base de datos virtual orientada a objetos sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (esencialmente la herencia y el polimorfismo)[32].

6.5.2 DAO (Data Access Object)

El DAO implementa el mecanismo de acceso requerido para trabajar con la base de datos. Se utiliza para abstraer y la encapsular todos los accesos a la base de datos. El DAO gestiona la conexión con la fuente de datos para obtener y

almacenar datos. Esencialmente, la DAO actúa como un adaptador entre el componente y la base de datos[33].

6.5.3 JPA y JPQL

La API de persistencia de Java (JPA) es una solución basada en estándares Java para la persistencia. La persistencia utiliza un enfoque de mapeo objeto / relacional para cerrar la brecha entre un orientado a objetos modelo y una base de datos relacional. La API de persistencia de Java también se puede utilizar en Java SE aplicaciones, fuera del entorno Java EE. Java Persistence consiste en lo siguiente áreas[34]:

- La API Java Persistence
- El lenguaje de consulta (JPQL)
- Objeto / metadatos de mapeo relacional.

6.5.4 EJB

Los Enterprise Java Beans son componentes del lado del servidor que encapsulan la lógica del negocio de una aplicación[35].

6.5.5 Managed Bean

Son objetos gestionados por contenedor de peso ligero (POJO) con requisitos mínimos, para apoyar a un pequeño conjunto de servicios básicos, como la inyección de recursos, las devoluciones de llamada de ciclo de vida, y los interceptores. Beans gestionados representan una generalización de los beans gestionados especificados por la tecnología JavaServer Faces y se puede utilizar en cualquier lugar de una aplicación Java EE, no sólo en la Módulos Web [36].

6.5.6 JSF (Java Server Faces)

La tecnología Java Server Faces establece el estándar para la construcción de interfaces de usuario del lado del servidor. Con los aportes de un grupo de

expertos, la API Java Server Faces fue diseñados para que puedan ser aprovechados por las herramientas que harán que el desarrollo de aplicaciones web sea incluso más fácil[37].

6.5.7 PrimeFaces

Es una librería de componentes visuales de código abierto para el conjunto Java Server Faces 2.0 desarrollada y mantenida por Prime Technology. Su objetivo principal es ofrecer un conjunto de componentes para facilitar la creación y diseño de aplicaciones web[38].

6.5.8 Servidor de Aplicaciones Glassfish

GlassFish es un servidor de aplicaciones desarrollado por Sun Microsystems destacado por su fiabilidad y estabilidad. Sus principales características son:

- Compatibilidad con las tecnologías de interoperabilidad de Web Services (WSIT) – Sun trabaja estrechamente con Microsoft para garantizar la interoperabilidad de las tecnologías de empresa de los servicios web, como la optimización de mensajes, la mensajería fiable y la seguridad.[39]
- Compatibilidad con JBI: Java Business Integration (JBI) amplía Java EE con Service Provider Interfaces (SPI) de integración empresarial. Estas interfaces (SPI) permiten a los desarrolladores crear o implementar el entorno de integración empresarial Java para especificaciones como WSCI, BPEL4WS y W3C Choreography Working Group.[39]
- Admite la replicación "en memoria" – La replicación "en memoria" en otros servidores proporciona un almacenamiento ligero de los datos del estado de la sesión sin necesidad de obtener una base de datos aparte. Este tipo de duplicación utiliza memoria de otros servidores para el almacenamiento de alta disponibilidad de datos de sesión HTTP y de beans de estado de la sesión.[39]
- Mejoras en el equilibrio de la carga – Se han añadido algunas mejoras en el plugin del equilibrio de la carga.[39]
- Comunidad Open Source y GlassFish : En junio de 2005, Sun estrenó la comunidad GlassFish cuyo propósito es desarrollar un servidor de aplicaciones

libre, de código abierto y de nivel comercial que implemente las funciones más recientes de la plataforma Java EE 5 y de las tecnologías de empresa relacionadas.[39]

6.5.9 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.[40]

Algunas de las características más importantes y soportadas por PostgreSQL:

- Generales
 - Base de datos 100% ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
 - Integridad referencial.
 - Tablespaces.
 - Copias de seguridad en caliente (Online/hot backups).
 - Unicode.
 - Regionalización por columna.
 - Completa documentación.
 - Licencia BSD.
 - Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
- Programación/Desarrollo
 - Funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de Oracle), PL/Perl, PL/Python y PL/Tcl.
 - Bloques anónimos de código de procedimientos (sentencias DO).
 - Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido, entre otros).

- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.
- SQL
 - SQL92, SQL99, SQL2003, SQL2008
 - Llaves primarias (primary keys) y foráneas (foreign keys)
 - Check, Unique y Not null constraints
 - Restricciones de unicidad postergables (deferrable constraints)
 - Columnas auto-incrementales
 - Índices compuestos, únicos, parciales y funcio
 - Consultas recursivas
 - Disparadores (triggers) comunes, por columna, condicionales.

Algunos de los límites de postgresQL se muestran en la tabla 2.

Limite	Valor
Máximo tamaño base de datos	Ilimitado (Depende del sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1,6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250-1600 (depende del tipo)
Máximo número de índices por tabla	Ilimitado

Tabla 2 Límites de postgresQL

Fuente: Elaboración propia

CAPÍTULO 2. DESARROLLO DEL PROYECTO - METODOLOGÍA

En esta sección se presenta el desarrollo del trabajo, a través de la descripción de cada una de las fases de la metodología XP y el uso de PSP para la fase de desarrollo en cada iteración. Se presenta también las herramientas usadas para el desarrollo y como se realizó la fase de pruebas e implementación del sistema.

7 DESARROLLO DEL PROYECTO

XP es una de las llamadas metodologías ágiles de desarrollo de software más exitosas de los tiempos recientes. La metodología propuesta en XP está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan. XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo [24]. Por lo anterior, se hizo uso de esta metodología, la cual se detalla a continuación y en la cual se agregó el uso de PSP como herramienta para el desarrollo en la fase de iteraciones. El proceso personal de software brinda la posibilidad de mejorar el desarrollo del sistema estableciendo un marco de disciplina personal. El proceso PSP consiste, como se mencionó en el marco de referencia, en un conjunto de métodos y formas que muestran como planificar, medir y administrar el trabajo [29].

7.1 DESARROLLO DE METODOLOGÍA XP

7.1.1 Fase Exploración

En esta fase se hizo uso de historias de usuario, estimando los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que están basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración. Esta fase dura típicamente un par de semanas, y el resultado es una visión general del

sistema, y un plazo total estimado [24]. Por lo tanto, en esta fase se mantuvo un nivel de detalle bajo como lo propone la metodología. Las historias de usuario, debieron poder ser programadas en un tiempo de entre una y tres semanas. Si el tiempo era mayor fueron divididas en dos o más historias, con el propósito de implementar las funcionalidades para la iteración que se modela [41]. Para la elicitación de requisitos bajo historias de usuario se usó la plantilla que se muestra en la tabla 3, en donde se muestra la funcionalidad de crear cotización, que hizo parte de la segunda iteración. En este caso el administrador será el responsable de registrar la cotización en el sistema, teniendo en cuenta los materiales, las horas de trabajo y gastos adicionales.

Historia de Usuario	
Número: 14	Usuario: Administrador
Nombre: Crear cotización	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Luego de crear el proyecto se inicia con la cotización, para lo cual el sistema debe mostrar campos para que el usuario diligencie la siguiente información: • Materiales necesarios, para lo cual el sistema debe facilitar el diligenciamiento, mostrando la disponibilidad de este en la empresa, si se encuentra disponible mostrar el costo de este, de lo contrario avisar que el material no está disponible y mostrar un promedio de los costos anteriores. • Horas de trabajo necesarias según cargo. • Gastos adicionales (Transporte, etc).	
Observaciones: Ninguna	

Tabla 3 Plantilla Historia de Usuario para Crear Cotización

Fuente: Elaboración propia

En el anexo 1 se encuentran el total de historias generadas para el proyecto.

7.1.2 Planificación

En esta fase, el cliente y el desarrollador acordaron el orden en que deberían implementarse las historias de usuario, y, asociadas a éstas, las entregas. Esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas, o “Release Plan” [24]. Dentro de la planificación, según las reglas y prácticas de Xp [42], si alguna de las historias de usuario tiene “riesgos” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba (“spikes”), para reducir estos riesgos. Sin embargo, como en el desarrollo del proyecto se tuvo de manera constante al cliente validando los requerimientos, la estimación no generó inconvenientes, lo que no generó tiempo adicional en la generación de prototipos para la validación. En lugar de esto, después de la estimación y la codificación, al hacer las pruebas unitarias, si se tenían cambios o ajustes, se realizaban pequeñas iteraciones para hacer las modificaciones directamente sobre el código fuente. La tabla 4, muestra la plantilla de reléase plan para la iteración 1, en la cual se programaron 6 historias de usuario de las especificadas en la fase de exploración. Para este caso, se tomó la implementación de las funcionalidades del proveedor y los empleados y se definieron las tareas de ingeniería para cada historia de usuario definiendo el tiempo estimado y la fecha de entrega estimada. El anexo 2, muestra el plan de entregas generado para cada una de las iteraciones. Según Martín Fowler (uno de los firmantes del “Agile Manifesto”) [43], los planes en XP se diferencian de las metodologías tradicionales en tres aspectos: Simplicidad del plan, los planes son realizados por las mismas personas que realizarán el trabajo y los planes no son predicciones del futuro, sino simplemente la mejor estimación de cómo saldrán las cosas. Después de la primera iteración o de algunas iteraciones se realizaba nuevamente una reunión con los stakeholders, para evaluar nuevamente el plan de entregas y ajustarlo si era necesario.

Iteración	Hist. De Usuario	Tarea de ingeniería		Tiempo estimado	Fecha de entrega
1	4 – Crear proveedor	4.1	Realizar método para funcionalidad	4 horas	Marzo 21 de 2016
		4.2	Realizar validaciones de datos.	4 horas	
		4.3	Crear interfaz gráfica	1 día	
		4.4	Pruebas unitarias.	2 horas	
	5 – Listar proveedores	5.1	Realizar métodos funcionales	4 horas	
		5.2	Realizar interfaz grafica	1 día	
		5.3	Pruebas unitarias	2 horas	
	6 – Consultar/Modificar proveedores	6.1	Crear métodos para funcionalidades modificar y consultar	4 horas	
		6.2	Crear interfaz grafica	1 día	
		6.3	Pruebas unitarias	2 horas	
	10 - Crear empleado	10.1	Crear métodos funcionales	1 día	
		10.2	Realizar validaciones de datos	4 horas	
		10.3	Crear interfaz grafica	4 horas	
		10.4	Pruebas unitarias	2 horas	
	11 - Listar empleados	11.1	Crear métodos funcionales	1 día	
		11.2	Crear interfaz grafica	1 día	
		11.3	Pruebas unitarias.	2 horas	
	12- Consultar/Modificar empleado	12.1	Crear métodos para funcionalidades Consultar/Modificar	1 día	
		12.2	Crear interfaz gráfica.	1 día	
		12.3	Pruebas unitarias.	2 horas	

Tabla 4 Plan de Entregas Iteración 1

Fuente: Elaboración propia.

En esta fase de Xp, en el presente trabajo se realizó la inclusión de PSP. El PSP permitió en esta etapa, administrar el tiempo en función de las actividades y a su vez conocer la importancia de realizar un seguimiento del mismo. Para llegar a este propósito fue importante lograr identificar las actividades y el tiempo involucrado de cada una de ellas [29]. De acuerdo a los principios de planificación del PSP, para que los ingenieros sean eficaces deben seguir procesos definidos que sean medibles así como planificar su trabajo. Por otra parte, los principios de calidad del PSP promueven que cada ingeniero realice trabajo de calidad. Para alcanzar esta calidad los ingenieros son responsables de la calidad de los productos que producen, previniendo defectos y haciendo su trabajo de manera correcta [44]. Para la fase de exploración desarrollada en el proyecto, se tomó como referencia la progresión de las distintas versiones del PSP como se muestra en la ilustración 5.



Ilustración 5 Versiones de PSP

Fuente: Elaboración propia.

Para ello, en esta fase se trabajó con la versión PSP0 en donde se diligenció la plantilla de estimación de esfuerzo que se muestra en la tabla 5, en donde por cada historia de usuario y las tareas generadas del reléase plan o plan de entregas, se realizó estimación de tiempo vs el tiempo real que llevo la implementación de cada tarea.

Historia de usuario	Tiempo Estimado	Tiempo real
4 - Crear un proveedor en el sistema	2 días, 2 horas	4 horas
5 - Consultar/modificar proveedor en el sistema	1 día, 6 horas	1 hora, 45 minutos
10 - Crear empleado	2 días, 2 horas	2 horas 50 minutos
11 - Listar empleados	2 días, 2 horas	1 hora 35 minutos
12 - Consultar/modificar empleado	2 días, 2 horas	2 horas 10 minutos

Tabla 5 Estimación de tiempo por Historia

Fuente: Elaboración propia.

Finalmente, se realizó el detalle de la estimación del esfuerzo por cada tarea en cada iteración del Proyecto, indicando el tiempo en horas, el número de errores y el estado de cada tarea. La plantilla usada para llevar control de esta información se muestra en la tabla 6, que incluye las tareas y estimación para la iteración 1. El anexo 3 muestra la estimación de tiempo y reporte de errores para el total de iteraciones del proyecto.

N°	Tarea de ingeniería	Tiempo Estimado	Hora Inicio	Hora fin	Errores	Estado
4	4.1 Realizar método para funcionalidad	4 horas	9:00am	10:00am	Ninguno	Finalizado
	4.2 Realizar validaciones de datos	4 Horas	10:00am	11: 00am	Ninguno	Finalizado
	4.3 Crear interfaz gráfica	1 día	11:00am	13:00pm	Ninguno	Finalizado
	4.4 pruebas unitarias	2 horas	14:00pm	15:00pm	Error con codificación UTF-8	
5	5.1 Realizar métodos funcionales	4 horas	15:00pm	15:30pm	Ninguno	Finalizado

	5.2 Realizar interfaz grafica	1 día	15:30pm	16:30pm	Ninguno	Finalizado
	5.3 Pruebas unitarias	2 horas	16:30pm	16:45pm	Ninguno	Finalizado
6	6.1 Crear métodos para funcionalidades modificar y consultar	4 horas	17:00pm	17:50pm	Ninguno	Finalizado
	6.2 Crear interfaz grafica	1 día	18:00pm	20:00pm	Ninguno	Finalizado
	6.3 Pruebas unitarias	2 horas	20:00pm	20:30pm	Ninguno	Finalizado
10	10.1 Crear métodos funcionales	1 día	8:30am	9:20am	Ninguno	Finalizado
	10.2 Realizar validaciones de datos	4 horas	9:30am	11:00am	Ninguno	Finalizado
	10.3 Crear interfaz grafica	4 horas	11:00am	13:00pm	Ninguno	Finalizado
	10.4 Pruebas unitarias	2 horas	13:00pm	13:30pm	Ninguno	Finalizado
11	11.1 Crear métodos funcionales	1 día	14:00pm	14:30pm	Ninguno	Finalizado
	11.2 Crear interfaz grafica	1 día	14:40pm	15:30pm	Ninguno	Finalizado
	11.3 Pruebas unitarias	2 horas	15:30pm	15:45pm	Ninguno	Finalizado
12	12.1 Crear métodos para funcionalidades Consultar/Modificar	1 día	16:00pm	16:30pm	Ninguno	Finalizado
	12.2 Crear interfaz gráfica.	1 día	16:40pm	17:50pm	Ninguno	Finalizado
	12.3 Pruebas unitarias.	2 horas	18:00pm	18:30pm	Ninguno	Finalizado

Tabla 6 PSP 0. Estimación de Esfuerzo Iteración 1

Fuente: Elaboración propia.

7.1.3 Iteraciones

Una vez acordado el plan de entregas o reléase plan se comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala las historias de usuarios. Esta es la fase principal en el ciclo de desarrollo de XP [24]. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a

la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, validando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también participó activamente durante esta fase del ciclo. Las iteraciones también se utilizaron para medir el progreso del proyecto, ya que una iteración terminada sin errores es una medida clara de avance.

El uso de PSP en esta fase fue de mucha relevancia, ya que es donde se aplica completamente los elementos propuestos en cada versión propuesta en la ilustración 5 – PSP0 y PSP0.1, en donde se define el estándar de codificación y se mide el tiempo real para ser comparado con el tiempo estimado en la fase anterior. En esta fase de desarrollo se realizaron varias actividades como son: diseño, codificación, compilación y pruebas. PSP0.1 extiende el proceso mediante la adición de un estándar de codificación y de una propuesta de mejora al proceso (En Inglés, Process Improvement Proposal o PIP) que le facilita al ingeniero identificar acciones de mejora para obtener mayor desempeño en su proceso software personal [45]. En este sentido se creó el documento de estándar de codificación basado en gran parte por los estándares de codificación del lenguaje Java presentados en Java Language Specification, de Sun Microsystems, Inc [46] (ver anexo 4), el estándar de base de datos (ver anexo 5) y el estándar de mensajes (ver anexo 6).

El proyecto generó un total de 4 iteraciones, en donde según el reléase plan se codificaron y se realizaron las pruebas, pasando así a la versión PSP1 y PSP1.1 (estimación y planificación), en donde, de acuerdo a las mediciones recolectadas en PSP0 y PSP0.1, se estimaron los casos de prueba (PSP1). Las mediciones acumuladas en la primera iteración se emplearon para estimar el tiempo requerido para la codificación de las tareas del reléase plan de la iteración 2 y así con las siguientes.

Las historias de usuarios seleccionadas para cada entrega fueron desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada iteración se generó las tareas de ingeniería y las tarjetas CRC (Clase-Responsabilidad-Colaboración), como se muestra en las tablas 7 y 8 respectivamente. Estas plantillas permitieron describir el proceso de desarrollo planteado en la fase de diseño y codificación propuestos en PSP1. Las plantillas para el total de iteraciones se encuentran en el Anexo 7 (Tareas de Ingeniería) y el Anexo 8 (Tarjetas CRC).

Tarea de Ingeniería	
Número de tarea: 4.1	Historia de usuario: 4
Nombre tarea: Realizar método para funcionalidad	
Tipo de tarea: Desarrollo	Puntos Estimados: 4 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método que reciba por parámetro un objeto de tipo empleado, con el cual se realizaran validaciones previas como: verificar que los datos obligatorios se encuentren diligenciados, si se encuentra un error se mostrara un mensaje en la parte superior del formulario especificado en la tarea de ingeniería 4.3, detallando que campo contiene error, si por el contrario no encuentra ningún error procede a guardar el objeto en la capa de datos. Si guarda exitosamente redirecciona la aplicación a la interfaz principal con un mensaje de aprobación, si por el contrario ocurre algún error se mostrara un mensaje al usuario notificando el error.	

Tabla 7 Tareas de Ingeniería para Iteración 1 - Historia Crear Proveedor

ProveedorVista	
Funcionalidades	Colaboraciones
Controlar el estado de la interfaz gráfica de Proveedor en crear y	ProveedorLogicaLocal

listar. • Almacenar toda la información diligenciada por el usuario. • Administrar las funcionalidades para crear y listar proveedores. • Verificar valides de datos.	

Tabla 8 Tarjeta CRC para Proveedor Vista

Fuente: Elaboración propia.

Adicional a las tareas de ingeniería y tarjetas CRC, en esta fase, se creó el modelo de base datos general del sistema (ver ilustración 6). En el anexo 9, se encuentra el modelo en más detalle.

Teniendo en cuenta que no se iba a codificar todas las historias de usuario, el modelo se iba perfeccionando en cada iteración, de acuerdo a las historias programadas.

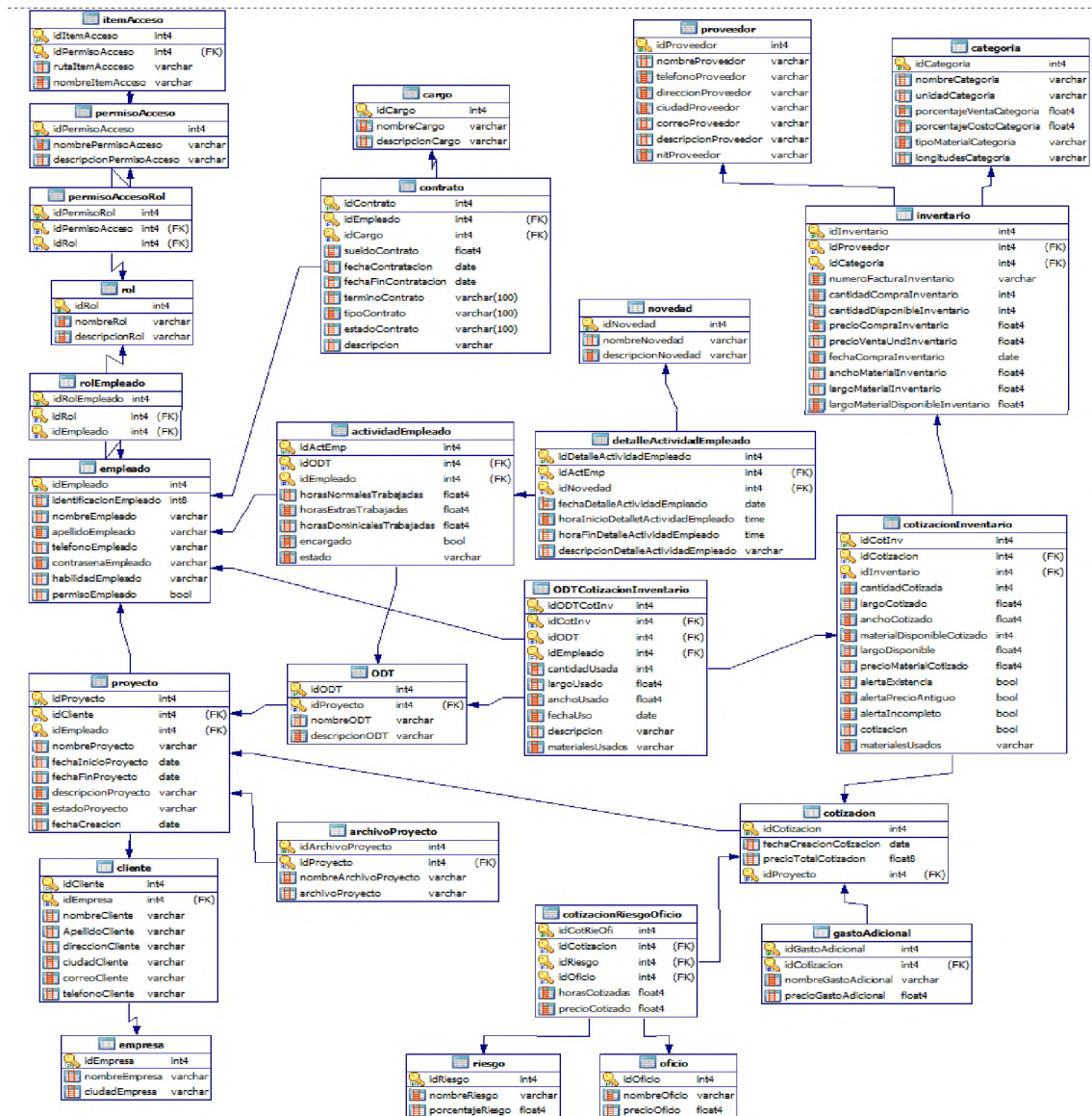


Ilustración 6 Diagrama de Base de Datos

Fuente: Elaboración propia

Finalmente, para cada historia de usuario se establecieron las pruebas de aceptación. Estas pruebas se realizaron al final de cada iteración, pero también al final de cada una de las siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior fueron analizadas para evaluar su corrección, así como para prever

que no vuelvan a ocurrir. La tabla 9 muestra la plantilla usada de pruebas, en donde se establecen elementos como las condiciones de ejecución y los pasos para la misma, identificado el resultado esperado y si fue positiva o no. El anexo 10 muestra el total de pruebas realizadas por iteración.

Prueba funcional	
Número de prueba: 1	Historia de usuario: 4
Nombre de la prueba: Crear proveedor	
Descripción: Se probará que se registren exitosamente nuevos proveedores en el sistema, validando correctamente su información.	
Condiciones de ejecución: El proveedor a ser creado no existe en el sistema.	
Entrada/Pasos de ejecución: • Se selecciona la opción proveedor -> crear proveedor. • Se diligencia el formulario según los campos especificados, con datos validos e inválidos según sea el campo. ○ Nombre – Alfanumérico obligatorio. ○ Teléfono – Alfanumérico obligatorio. ○ Correo – Alfanumérico, debe cumplir un formato. ○ Dirección – Alfanumérico. ○ Ciudad – Alfanumérico. ○ Descripción – Alfanumérico. • Se pulsa la opción Guardar.	
Resultado esperado: Registro creado exitosamente, mensaje de información especificando la operación exitosa.	
Evaluación de la prueba: Positiva.	

Tabla 9 Plantilla Pruebas Funcionales por Historia

Fuente: Elaboración propia.

7.1.4 Puesta en Producción

En esta fase, la metodología Xp, indica que si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa [24]. Por lo tanto, solo al final de las 4 iteraciones se puso el sistema en producción. En esta fase no se realizaron más desarrollos funcionales, pero en algunos casos si fue necesarias tareas de ajuste (“fine tuning”) [24].

Ahora, respecto a PSP, en esta fase se aplica la fase de postmortem, que es donde se complementa el plan realizado en la primera fase de acuerdo a las mediciones obtenidas tras construir el producto. Esta fase propuesta en la versión de PSP0 se aplica en cada iteración de la metodología Xp, permitiendo en cada una mejorar la estimación. Por otra parte, aquí se aplica PSP2 y PSP2.1 (gestión de calidad y diseño), en donde se añaden dos actividades a la fase de desarrollo: revisión de diseño y revisión de código. La prevención y eliminación de defectos es la parte central de PSP2 [45]. En esta versión del proceso, al pasar por cada iteración se aprende a evaluar y mejorar las estimaciones, así como a mejorar la calidad del producto.

7.2 HERRAMIENTAS USADAS

La ilustración 7 muestra el conjunto de tecnologías de Java EE que se utilizaron para el desarrollo del sistema de información. Todas estas tecnologías serán descritas a continuación.

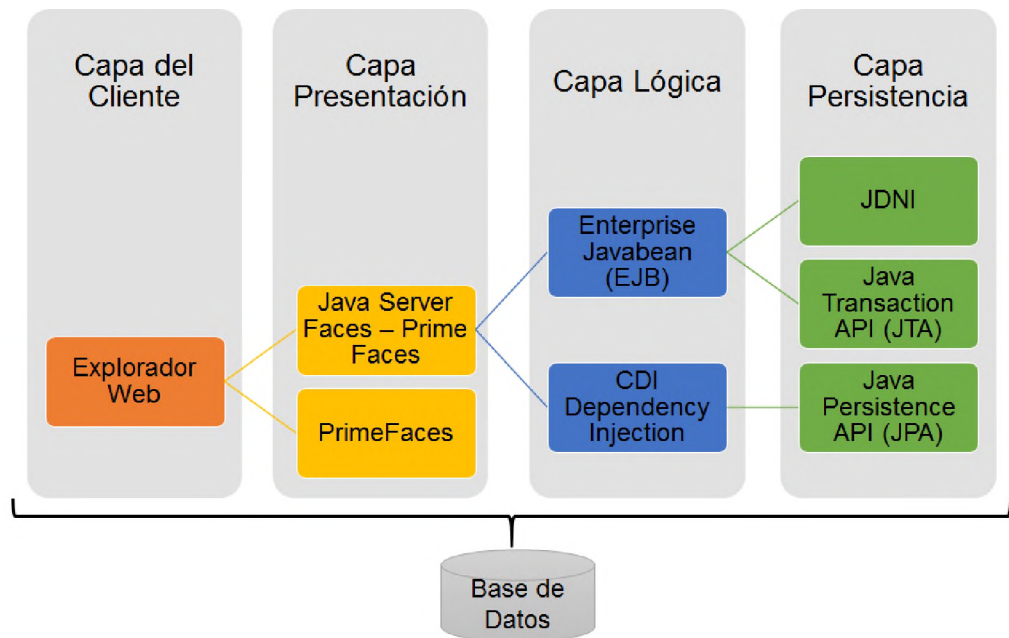


Ilustración 7 Conjunto de Tecnologías Java EE

Fuente: Elaboración Propia.

En primer lugar, desde la capa del cliente y capa presentación, se hizo uso de JSF que es un estándar de Java hacia la construcción de interfaces de usuario para aplicaciones web que simplifican el desarrollo de aplicaciones web del lado del cliente. En las páginas JSF se utilizó la extensión *.xhtml, es decir, una combinación de XML con HTML, en donde se pudo incluir componentes como CSS, JavaScript, entre otros.

La especificación de JSF define seis fases distintas en su ciclo de vida:

1. Restauración de la vista: Crea un árbol de componentes en el servidor para representar la información de un cliente.
2. Aplicar valores de la petición: Actualiza los valores del servidor con datos del cliente.
3. Proceso de validación: Valida los datos del usuario y hace la conversión.
4. Actualización de valores del modelo: Actualiza el modelo del servidor con nuevos datos.

5. Invocar la aplicación: Ejecutar cualquier lógica de aplicación para cumplir con la solicitud.

6. Procesar la respuesta: Guarda un estado y da una respuesta al cliente.

Para facilitar el uso de los componentes se hizo uso de PrimeFaces que es una librería de componentes visuales de código abierto para el conjunto Java Server Faces 2.0 desarrollada y mantenida por Prime Technology. Su objetivo principal es ofrecer un conjunto de componentes para facilitar la creación y diseño de aplicaciones web [47]. Los componentes de PrimeFaces fueron conectados con el servidor a través de los ManagedBean que no son más que beans que reciben la información del html para enviarlos a la capa de lógica.

En segundo lugar, para la capa de lógica, se hizo uso de los JavaBeans empresariales (Enterprise JavaBeans [1, 6], EJB), que son una tecnología (API) que forma parte del estándar de Java EE. Están diseñados para desarrollo y despliegue de aplicaciones (distribuidas) de negocio basadas en componentes del lado del servidor. Una vez que se desarrolla una aplicación, ésta puede ser desplegada en cualquier servidor que soporte la especificación de EJB [38].

En tercer lugar, para la capa de persistencia se hizo uso de JPA que es, una abstracción que está por encima de JDBC lo que permitió ser independiente de SQL. Todas las clases y anotaciones de esta API se encuentran en el paquete `javax.persistence`.

Los principales componentes de JPA son:

- Mapeo de base de datos relacionales (ORM). Es el mecanismo para mapear objetos a los datos almacenados en una base de datos relacional.
- Un API administrador de entidad para realizar operaciones en la base de datos tales como crear, leer, actualizar, eliminar (CRUD).

- El Java Persistence Query Language (JPQL) que permite recuperar datos con un lenguaje de consultas orientado a objetos.
- Las transacciones y mecanismos de bloqueo cuando se accedan a los datos concurrentemente, la API Java Transaction (JTA).

El motor de base de datos usado fue PostgreSQL. Que sin dudas es uno de los mejores gestores en el mercado, destacando características como: *Open souce* lo cual descarta gastos adicionales en la implantación del sistema, *Multiplataforma* dando libertad de instalación en cualquier servidor sin importar su SO, además de su *robustez y estabilidad* lo que ofrece seguridad y disponibilidad constante de los datos para ser accedidos en cualquier momento.

7.3 IMPLEMENTACIÓN DEL PROYECTO

Finalizadas las iteraciones en la metodología, el siguiente paso fue la implementación del proyecto. Para ello, se hizo uso de un servidor el cual tenía instalado el motor de base de datos POSTGRESQL y el servidor Glassfish. Al servidor se puede acceder a través de la siguiente dirección: www.proyectosmyr.com/testing.

Los pasos para montar la aplicación se muestran en el Anexo 11. Se utilizó el servidor de aplicaciones Glassfish que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.

CAPÍTULO 3 – RESULTADOS Y ANÁLISIS

En esta sección se presentan los resultados obtenidos por cada uno de los objetivos.

8 RESULTADOS

En la tabla 10 se muestra por cada objetivo los resultados obtenidos.

Objetivo	Resultado Obtenidos
Analizar los procesos que se llevan a cabo en el desarrollo de un proyecto en la empresa M&R.	Historias de usuario según requisitos del cliente asignadas por cada iteración.
Diseñar los procesos que permitan la gestión de personal, materiales y control en la empresa de acuerdo a sus necesidades específicas.	Generación del reléase plan por iteración. Modelo de base de datos del proyecto y estimación de tiempos.
Desarrollar el sistema que permita apoyar la gestión de proyectos en la empresa.	Planeación de tareas de ingeniería, tarjetas CRC, codificación y desarrollo de pruebas. Se agregan funcionalidades adicionales, como la generación de documentos XLSX en diferentes puntos de la aplicación, facilitando de esta manera trabajos de análisis de datos por parte de los encargados. Se agrega un módulo de seguridad, donde se asignan permisos de ingreso a funcionalidades a los usuarios, garantizando así la total confidencialidad

	de datos valiosos para la empresa. Además, la aplicación cuenta con estilos Responsive lo cual permite ser accedido desde cualquier tipo de dispositivo adaptándose al tamaño de la pantalla.
Implantar el sistema en la empresa M&R, validando su correcto funcionamiento.	Realización de pruebas en todos los módulos de la aplicación que permitieron verificar el funcionamiento idóneo del software y permitieron generar el entregable final del sistema desarrollado. Capacitación a los usuarios finales del sistema. Ver Anexo 12, donde se muestra la carta de constancia de implementación del sistema.

Tabla 10 Objetivo vs Resultados Obtenidos

Fuente: Elaboración Propia.

Los resultados del proyecto fueron medidos desde dos puntos de vista, el primero es en cuanto a la mejora en la ejecución de los procesos en la empresa M&R a partir de la implementación del proyecto de gestión de los proyectos. Y el segundo, desde el aporte de la metodología Xp y el uso de PSP en el desarrollo de sistemas de información, cuando el desarrollo es personal.

8.1 RESULTADOS IMPLEMENTACIÓN PROYECTO EN EMPRESA M&R

Las mejoras en la empresa M&R fueron evaluadas en base al análisis del tiempo que involucraba el ingreso y consulta de información en el sistema, y en base a la estimación de los proyectos en tiempo, costo, personal y materiales. Este análisis involucra lo siguiente:

- *Tiempo de registro y consulta de información actual vs tiempo de registro consulta antes de implementado el sistema:* La consulta de información es el primer factor que se ve impactado con la implantación del sistema, pues se hace evidente la diferencia de tiempos al consultar información manualmente o por medio de la aplicación. La ilustración 8 compara el tiempo tomado para realizar las diferentes actividades de los dos modos.

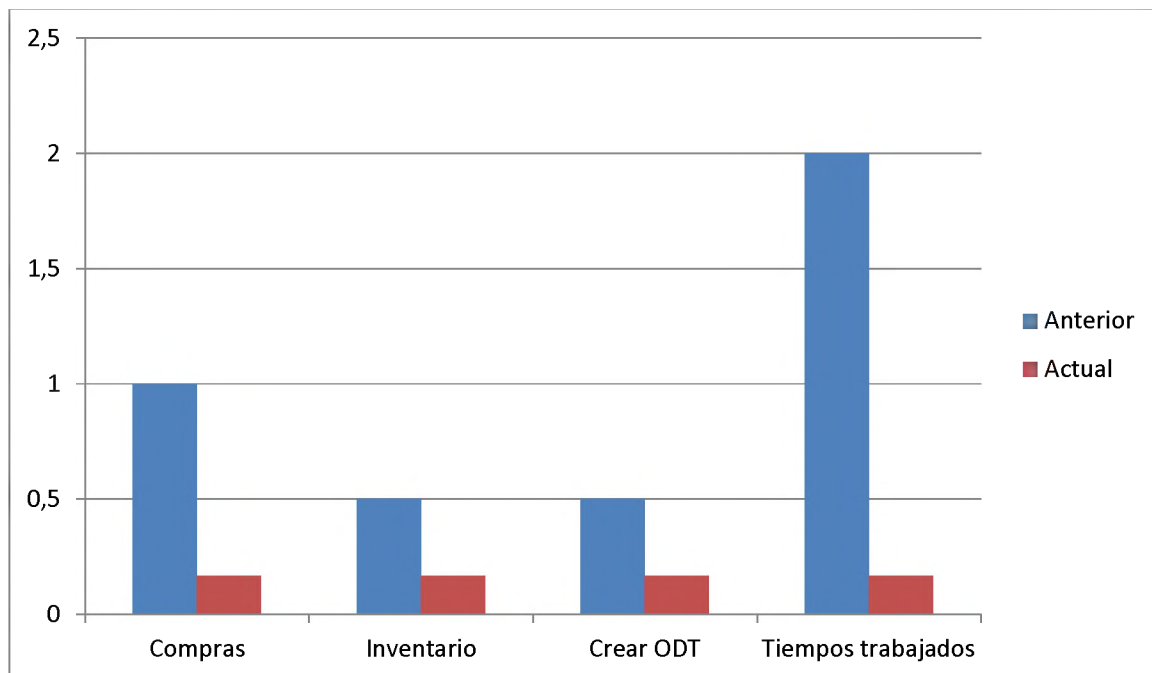


Ilustración 8 Consulta de datos. Anterior vs Actual

Fuente: Elaboración propia.

- *Tiempo estimado del proyecto con el sistema vs tiempo real de ejecución:* Inicialmente, este tipo análisis no era posible de realizar pues no se llevaba ningún control sobre el tiempo de esfuerzo que dedicaban los empleados a un proyecto o actividad. Actualmente, la aplicación ofrece la opción de mostrar el tiempo de esfuerzo dedicado a un proyecto en particular desde dos puntos de vista: Mostrar el detalle de actividades realizadas por cada empleado especificando hora y descripción de la tarea realizada, como también mostrar el consolidado de horas dedicadas a un proyecto,

mostrando la suma total de horas normales, extras y festivas que se han trabajado. La ilustración 9 muestra los datos obtenidos por la aplicación, donde se evidencia la facilidad al momento de analizar el tiempo de esfuerzo para un proyecto en particular.

	GESTIÓN Y CONTROL DE PROYECTOS · PROYECTOS MANTENIMIENTO INDUSTRIAL M&R			
	Documento:	Consolidado de actividades		
	Fecha generación:	25/10/2016	Versión:	1.0
ODT	Horas Normales	Horas Extras	Horas Festivas	
Construcción de transportador sección milo	80	25	10	

Ilustración 9 Consolidado de tiempos registros en proyecto

Fuente: Elaboración propia.

- *Costo estimado del proyecto vs costo final del proyecto:* El sistema ofrece la opción de generar el detalle del desarrollo del proyecto, mostrando datos como: Esfuerzo de tiempo dedicado, materiales usados y el cálculo total de estos elementos. Esta actividad no era posible realizarse anteriormente, pues no se contaba con el registro adecuado para la realización del informe. La ilustración 10 muestra el informe generado por la aplicación, donde se especifica el detalle de la ejecución del proyecto y el costo acumulado.

	GESTIÓN Y CONTROL DE PROYECTOS - PROYECTOS MANTENIMIENTO INDUSTRIAL M&R				
	Documento:	Detalle desarrollo de proyecto			
	Fecha generación:	26/10/2016	Versión:	1.0	
Proyecto de prueba	504650,4238				
Empleado	H. normales	H. Extras	H. festivas	Precio	
Jhon Jairo Ramírez Osorio	1	1,166666627	1,5	\$7.638,89	
Hernán Ramírez Castañeda	8,5	3,5	0	\$16.666,67	
			TOTAL	\$24.305,56	
Empleado	Material	Fecha	Usado	Dimensión	Precio
Jhon Jairo Ramírez Osorio	Angulo 3/16 x 1"	01/10/2016	2	120.0cm	\$124.137,94
Juan Pablo Ramírez Osorio	Angulo 3/16 x 1"	05/10/2016	1	30.0cm	\$15.517,24
Juan Pablo Ramírez Osorio	Rodamientos	04/10/2016	4	4	\$320.000,03
Jhon Jairo Ramírez Osorio	Angulo 3/16 x 1"	06/10/2016	1	30.0cm	\$15.517,24
Hernán Ramírez Castañeda	Angulo 3/16 x 1"	06/10/2016	1	10.0cm	\$5.172,41
				TOTAL	480344,8672

Ilustración 10 Detalle ejecución de proyecto

Fuente: Elaboración propia

- *Personal estimado del proyecto vs personal final en la ejecución del proyecto:* El sistema ofrece la opción de generar en detalle el esfuerzo de tiempo dedicado por parte de los empleados, documento en el cual se especifica claramente las actividades realizadas y su ejecutor, ofreciendo una serie de filtros para analizar los datos desde diferentes puntos de vista. La ilustración 11 muestra el documento generado por el sistema y las posibles opciones de filtrado.

	GESTIÓN Y CONTROL DE PROYECTOS - PROYECTOS MANTENIMIENTO INDUSTRIAL M&R						
	Documento:	Detalle actividades					
	Fecha generación:	25/10/2016	Versión:	1.0			
Fecha	Empleado	ODT	Hora inicio	Hora Fin	Total horas	Novedad	Descripción
28/10/2016	Hernán Ramírez Castañeda	Construcción de transportador sección milo	13:00	16:00	3	ND	Descripción 1
27/10/2016	Hernán Ramírez Castañeda	Construcción de transportador sección milo	09:00	11:00	2	ND	Descripción 2
15/10/2016	Hernán Ramírez Castañeda	Construcción de transportador sección milo	10:00	12:00	2	ND	Descripción 3
15/10/2016	Hernán Ramírez Castañeda	Construcción de transportador sección milo	13:00	16:00	3	ND	Descripción 4
15/10/2016	Hernán Ramírez Castañeda	Construcción de transportador sección milo	08:00	10:00	2	ND	Descripción 5
09/10/2016	Jhon Jairo Ramírez Osorio	Construcción de transportador sección milo	11:30	13:00	1,5	ND	Descripción 6
04/10/2016	Jhon Jairo Ramírez Osorio	Construcción de transportador sección milo	07:30	08:00	0,5	ND	Descripción 7
04/10/2016	Jhon Jairo Ramírez Osorio	Construcción de transportador sección milo	11:30	13:10	1,666666627	ND	Descripción 8

Ilustración 11 Detalle de esfuerzo por parte del personal en un proyecto

- *Materiales cotizados del proyecto vs material real utilizado:* Otra de las características que ofrece la aplicación es la generación de un informe con

el detalle de los materiales usados en el desarrollo de un proyecto, especificando en detalle datos como: El día, el empleado responsable y una descripción breve sobre el uso de este material. Inicialmente, este tipo de informes no era posible de realizarse, pues no se llevaba el control sobre el uso de materiales en un proyecto. La ilustración 12 muestra los resultados.

	GESTIÓN Y CONTROL DE PROYECTOS - PROYECTOS MANTENIMIENTO INDUSTRIAL M&R			
	Documento:	Material usado		
	Fecha generación:	26/10/2016	Versión:	1.0
ODT	Material	Empleado	Fecha	Usado
Proyecto de prueba	Angulo 3/16 x 1"	Jhon Jairo Ramírez Osorio	06/10/2016	30.0cm
Proyecto de prueba	Angulo 3/16 x 1"	Hernán Ramírez Castañeda	06/10/2016	10.0cm
Proyecto de prueba	Angulo 3/16 x 1"	Juan Pablo Ramírez Osorio	05/10/2016	30.0cm
Proyecto de prueba	Rodamientos	Juan Pablo Ramírez Osorio	04/10/2016	4
Proyecto de prueba	Angulo 3/16 x 1"	Jhon Jairo Ramírez Osorio	01/10/2016	120.0cm

Ilustración 12 Detalle de material usado en ejecución de proyectos

Fuente: Elaboración propia

8.2 RESULTADOS USO DE METODOLOGÍA XP CON PSP

El uso de la metodología Xp permitió generar una programación más organizada, dado que el proceso de planificación y pruebas fue eficiente. Esta forma de programación por iteraciones cortas generó una menor tasa de errores, dado que la comunicación con los stakeholders fue permanente y las pruebas eran continuas. Además, cuando se generaba un cambio, en una iteración, agregarlo o modificarlo fue más sencillo, facilitando los cambios y permitiendo ahorrar mucho tiempo. Por otro lado, el uso de PSP dentro de esta metodología permitió evaluar y mejorar el proceso de desarrollo en la fase de iteraciones.

8.2.1 Resultados con respecto a la precisión en las estimaciones de tiempo

La ilustración 13 muestra la relación de estimación de esfuerzo vs el tiempo real para la primera iteración. En este caso se pudo encontrar que en la primera iteración, como programador no se tiene la experiencia en estimar el tiempo de desarrollo de cada funcionalidad, lo que puede generar desfase de tiempo con el

real. Ahora, la ilustración 14 muestra la estimación de la iteración final 4, en donde la relación con el tiempo real es más aproximada en la mayoría de casos, a excepción de unos casos particulares donde se presentaron cambios e inconvenientes no previstos en la funcionalidad, lo que indica que la medición en cada iteración, genera en el programador el registro de su experiencia, lo que le permite establecer con más claridad el tiempo, de acuerdo a las funcionalidades a implementar.

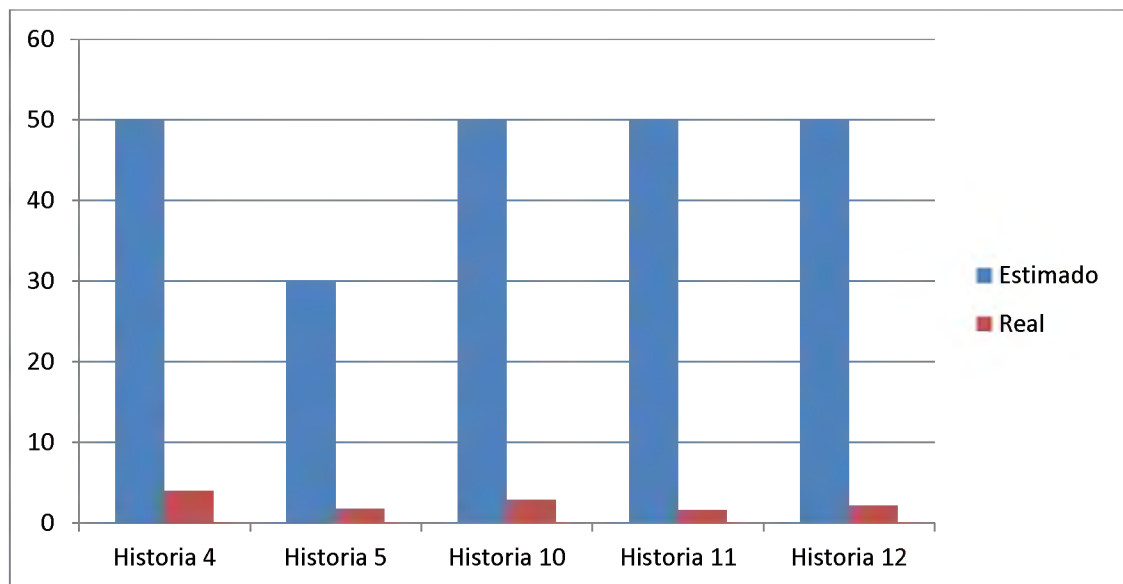


Ilustración 13 estimación de esfuerzo vs el tiempo real iteración 1

Fuente: Elaboración propia.

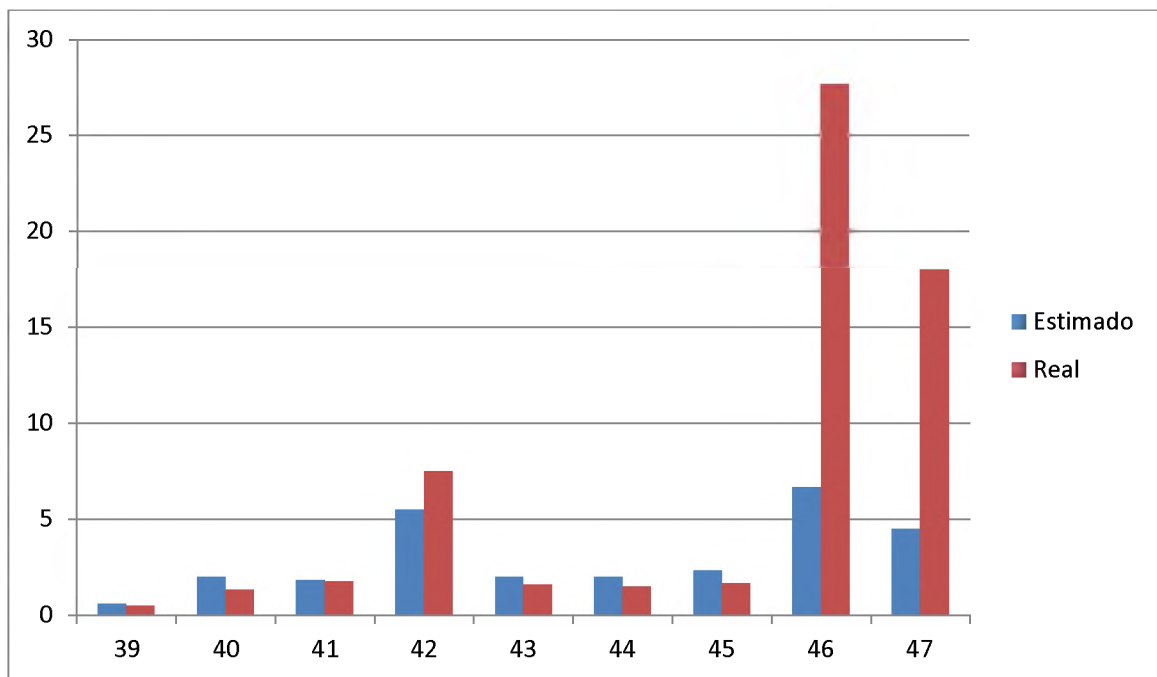


Ilustración 14 Estimación de esfuerzo vs el tiempo real iteración 4

Fuente: Elaboración propia.

8.2.2 Resultados con respecto a la cantidad de errores por iteración

El otro componente principal del PSP es el referente a la calidad, donde se promueve la habilidad de encontrar y remover defectos en etapas tempranas del proceso de desarrollo de software. Las revisiones de diseño y de código son actividades que se incorporan en el PSP a partir de la versión 2 del proceso. La ilustración 15, muestra la cantidad de errores y mejoras en las funcionalidades por iteración en donde se encontró que en las primeras iteraciones, donde las funcionalidades eran básicas se hicieron diferentes sugerencias por parte de la empresa para realizar mejoras en la visualización de datos como: Ordenamientos de tabla, filtrado, etc. También se puede observar, que en medio de la ejecución del desarrollo donde se realizaban funcionalidades críticas para el funcionamiento de la aplicación, se encontraron funcionalidades que fueron rechazadas por parte de la empresa, esto debido a los requerimientos cambiantes que se evidenciaban

en el transcurso de la ejecución. Finalmente, las etapas finales del desarrollo fueron menos impactadas por mejoras o errores propuestos por parte de la empresa, esto debido a que ya se tenían requerimientos más objetivos y se conocían las mejoras aplicadas anteriormente.

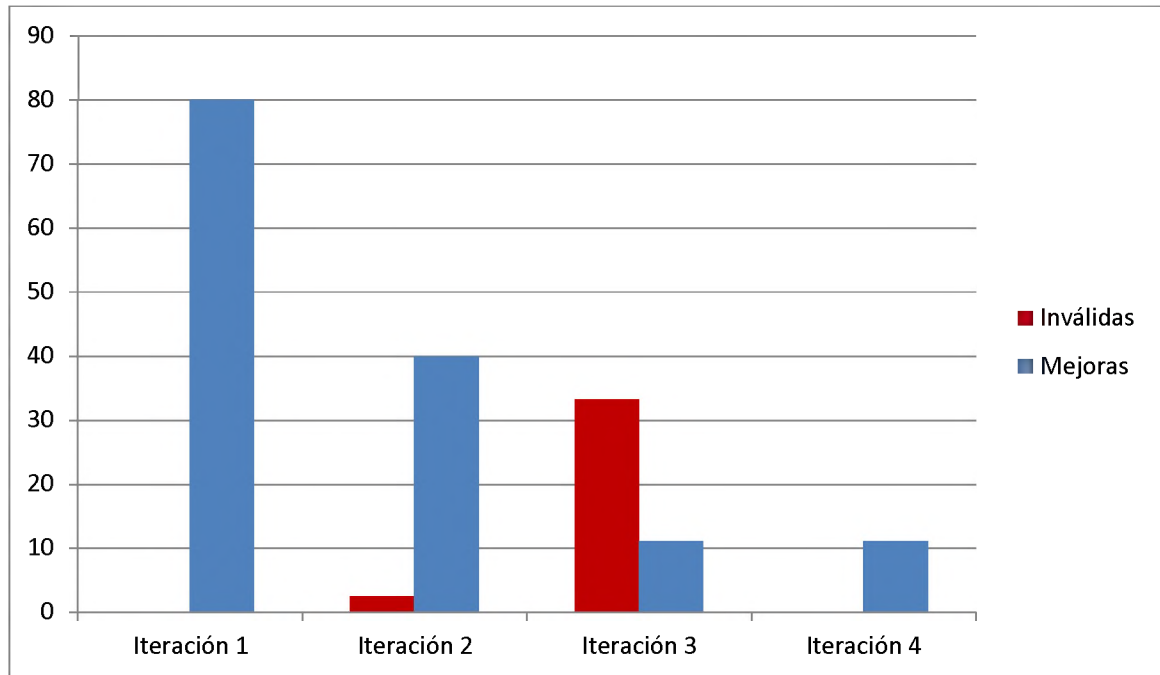


Ilustración 15 Errores y mejoras en funcionalidades

Fuente: Elaboración propia

CAPÍTULO 4 – CONCLUSIONES Y TRABAJOS FUTUROS

En esta sección se describen las conclusiones del trabajo realizado y se plantean algunos trabajos futuros que pueden ser abordados.

13 CONCLUSIONES

El uso de una metodología ágil como XP encamina los objetivos del proyecto por un buen camino, garantizando en gran medida la calidad del producto y satisfacción del cliente en cuanto a los resultados esperados debido a la amplia participación que tuvo en el transcurso del desarrollo.

Usar PSP como complemento de una metodología ágil resulta muy productivo para el proyecto y para quien lo aplica, pues exige constante mejoramiento personal en el desarrollo de software de calidad, trazando mejoras continuas en los procesos empleados por medio de planificación, estimación y control de rendimiento frente a los tiempos planeados.

El uso de una herramienta tecnológica para apoyar cualquier tipo de proceso, ofrece grandes ventajas competitivas en un mercado en constante cambio. Así pues, se determina el impacto positivo que trae consigo la implantación de este sistema en la empresa *Proyectos – Mantenimiento Industrial M&R* el cual le ofrece mejores condiciones para controlar e impulsar el constante crecimiento que se tiene en el mercado nacional.

14 TRABAJOS FUTUROS

Dentro de los trabajos futuros se incluye:

- Un módulo que permita apoyar procesos contables en la empresa en cuanto a compras y ventas realizadas en la empresa.
- Un módulo de alertas que avise al encargado del proyecto cuando el proyecto se encuentre en un estado crítico como: Se sobrepasan los recursos estimados (Tiempo o materiales), el tiempo de entrega del proyecto es próximo, entre otros.
- Un módulo de auditoria que registre todos los cambios realizados en la aplicación con datos como: Responsable, descripción de la acción y fecha.
- Desarrollar una aplicación móvil que permita la consulta de datos importantes como: Disponibilidad de materiales en inventario con su respectivo precio, estado de los proyectos, generación de gráficos que muestren la condición en que se encuentra el proyecto, además de administrar la seguridad de la aplicación por parte del administrador.
- Desarrollar una aplicación móvil que permita el registro de esfuerzo de tiempo invertido por los empleados, en la cual cada uno de estos pueda realizar su registro personalmente.

CAPÍTULO 6 – REFERENCIAS BIBLIOGRÁFICAS

En esta sección se presenta una lista de referencias bibliográficas que soportan este trabajo.

15 REFERENCIAS BIBLOGRAFICAS

- [1] Iese, “La evolución del concepto stakeholders en los escritos de Ed Freeman,” *Newsletter*, vol. 5, p. 4, 2009.
- [2] U. Nikual and M. Manies, “La Elicitación De Requisitos En El Contexto De Un Proyecto Software,” *USB Medellín*, vol. 2, no. 2, pp. 25–29, 2011.
- [3] E. H. Uribe and L. E. V. Ayala, “Del Manifiesto Ágil, sus valores y sus principios,” *Redalyc.Uaemex.Mx*, no. 34, pp. 381–385, 2007.
- [4] J. Lafosse, *Struts 2: el framework de desarrollo de aplicaciones Java EE*, Barcelona. 2010.
- [5] O. C. Chris Schalk, “Introduction to Javaserer Faces - What is JSF?,” 2005.
- [6] D. S. Manchado and D. F. Puntos, “Estudio del servidor de aplicaciones Glassfish y de las aplicaciones J2EE,” 2010.
- [7] T. Postgresql and G. Development, “PostgreSQL 9.1.23 Documentation.”
- [8] Concatel, “Creación, gestión y consulta de órdenes de trabajo,” 2010. .
- [9] E. más grandes crecieron el doble de la Economía, “Empresas más grandes crecieron el doble de la economía,” 2015. .
- [10] Elpais.com.co, “La industria colombiana empezó el año con pie derecho,” *Marzo 15, 2016*, 2016. .
- [11] M. Rouse, “Gestión de proyectos,” 2015.
- [12] Escuela Neijing, A. (Bell L. Silberschatz, H. F. (Bell L. Korth, B. Sudarshan, S. (Instituto Indio de Tecnología, J. Rodríguez, M. Daureo, J. Martínez, M.

Marqués, R. Lapiedra, C. Devece, and J. Guiral, *Introducción a la gestión de sistemas de información en la empresa*, vol. 2010. 2011.

- [13] I. Mendoza Vázquez, "Definicion De Un Framework Para Aplicaciones Web Con Navegacion Sensible a Concerns."
- [14] E. R. Bravo, "Information systems and performance: the role of technology, the task and the individual," *Behav. Inf. Technol.*, pp. 1–7, 2013.
- [15] IEEE and S. 610.121990, "IEEE Standard Glossary of Software Engineering Terminology," vol. 121990, p. 138, 1990.
- [16] J. M. Carrillo, D. Gea, J. Nicolás, J. L. Fernández, A. Toval, C. Ebert, and A. Vizcaíno, "Requirements engineering tools: Capabilities , survey and assessment," *Inf. Softw. Technol.*, vol. 54, no. 10, pp. 1142–1157, 2012.
- [17] F. Schneider, H. Naughton, and B. Berenbach, "Procedia Computer Science A modeling language to support early lifecycle requirements modeling for systems engineering," vol. 0, pp. 201–206, 2012.
- [18] K. Pohl, "Requirements Engineering : An Overview," vol. 36, pp. 1–40, 1995.
- [19] C. Jeffries, R., Anderson, A., Hendrickson, *Extreme Programming Installed*. 2001.
- [20] P. Letelier and C. Penadés, "Metodologías ágiles para el desarrollo de software : eXtreme Programming (XP)," 2003.
- [21] M. Blom, "Is Scrum and XP suitable for CSE Development?," *Procedia Comput. Sci.*, vol. 1, pp. 1511–1517, 2010.
- [22] J. Newkirk, "Introduction to agile processes and extreme programming," *Proc. 24th Int. Conf. Softw. Eng. - ICSE '02*, p. 695, 2002.
- [23] E. Uribe Herrera and L. E. Valencia Ayala, "Del manifiesto ágil sus valores y principios," *Sci. Tech. Año XIII*, vol. 34, no. 34, pp. 381–385, 2007.
- [24] I. J. Joskowicz and C. Mingus, "Reglas y Prácticas en eXtreme

Programming,” pp. 1–22, 2008.

- [25] M. Cohn, *User Stories Applied – For agile software Development*. 2004.
- [26] K. Beck, “Extreme Programming Explained,” 2002.
- [27] A.-W. Kent Beck, “Extreme Programming Explained: Embrace Change,” 2000.
- [28] M. Cohn, *Agile Estimating and Planning*. 2006.
- [29] A. Chávez and E. Durán, “Aplicación de PSP (Personal Software Process) para el desarrollo de un sistema administrador de códigos de barra a partir de la evaluación de Procesos de Reingeniería,” p. 8, 2010.
- [30] D. Duran and A. Gamboa, “Introduciendo Psp (Procesos Personal De Software) En El Aula,” vol. 2, pp. 1–5, 2010.
- [31] J. A. Rodríguez, J. González Moreno, and T. Murillas Herrera, “Software basado en arquitectura de Java EE y estándar SOAP para la validación en línea de listas de sancionados,” *J. Technol.*, no. 1, pp. 21–26, 2008.
- [32] H. del Busto and O. Enriquez, “Mapeo Objeto/Relacional (ORM).,” *Rev. Telem@tica*, vol. 10, no. 3, pp. 1–7, 2012.
- [33] O. Corporation, “Core J2EE Patterns - Data Access Object.” .
- [34] O. Corporation, “The Java EE 6 Tutorial,” no. January, 2013.
- [35] H. del Busto and O. Enriquez, “Mapeo Objeto/Relacional (ORM).,” *Rev. Telem@tica*, vol. 10, no. 3, pp. 1–7, 2012.
- [36] O. Corporation, “The Java EE 6 Tutorial,” no. January, 2013.
- [37] O. Corporation, “JavaServer Faces Technology.”
- [38] F. Pech-May, M. A. Gomez-Rodriguez, L. A. de la Cruz-Diaz, and S. U. Lara-Jeronimo, “Desarrollo de Aplicaciones web con JPA, EJB, JSF y PrimeFaces,” p. 9.
- [39] Oracle, “Funciones de Sun GlassFish Enterprise Server.” .

- [40] Rafaelma, "Sobre PostgreSQL," 2010. .
- [41] H. Ordoñez, A. Escobar, D. Velandia, C. Cobos, A. Ordoñez, and R. Segovia, "An Impact Study of Business Process Models for Requirements Elicitation in XP," in *Computational Science and Its Applications - ICCSA 2015*, 2015, pp. 298–312.
- [42] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifiesto por el Desarrollo Ágil de Software," 2001. [Online]. Available: <http://agilemanifesto.org/iso/es/>.
- [43] M. Fowler, "Is Design Dead ? Growing an Architecture," pp. 1–14, 2006.
- [44] W. R. Nichols and R. Salazar, "Deploying TSP on a National Scale: An Experience Report from Pilot Projects in Mexico," no. March, p. 177, 2009.
- [45] O. Gómez, "Estudio del Proceso Software Personal (PSP) en un entorno académico," *ReCibe*, vol. 3, no. 2, p. 28, 2014.
- [46] J. Gosling, B. Joy, G. Steele, and G. Bracha, "The Java Language Specification - Java SE 8 Edition," *Addison-Wesley*, p. 684, 2005.
- [47] "Prime Faces." [Online]. Available: <http://primefaces.org/>.

ANEXOS

ANEXO 1



PROYECTO:	Gestión de proyectos			Página 1 de 13
DOCUMENTO:	Historias de usuario			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Historias de usuario

Gestión de proyectos



PROYECTO:	Gestión de proyectos		Página 2 de 13
DOCUMENTO:	Historias de usuario		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

Tabla de contenido

Historia de usuario 1: Autenticar usuario	3
Historia de usuario 2: Listar usuarios	3
Historia de usuario 3: Consultar/Modificar usuario.....	4
Historia de usuario 4: Crear proveedor.....	4
Historia de usuario 5: Listar proveedores	5
Historia de usuario 6: Consultar/Modificar proveedor.....	5
Historia de usuario 7: Ingresar material a inventario	6
Historia de usuario 8: Listar material en inventario.....	6
Historia de usuario 9: Consultar/Modificar material	7
Historia de usuario 10: Crear empleado	7
Historia de usuario 11: Listar empleados.....	8
Historia de usuario 12: Consultar/Modificar empleado	8
Historia de usuario 13: Crear proyecto	9
Historia de Usuario 14: Crear cotización.....	9
Historia de usuario 15: Crear actividades	10
Historia de usuario 16: Listar proyectos	10
Historia de usuario 17: Consultar/Modificar proyectos.....	11
Historia de usuario 18: Crear cliente.....	11
Historia de usuario 19: Listar clientes	12
Historia de usuario 20: Consultar/Modificar cliente.....	12

Historia de usuario 1: Autenticar usuario

Historia de Usuario	
Número: 1	Usuario: Administrador
Nombre: Autenticar usuario	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema inicialmente debe mostrar un formulario para iniciar sesión en el sistema, el cual contará con dos campos de texto (Usuario y contraseña) y un botón (Iniciar sesión). si la autenticación es válida se permite el acceso a la aplicación, de lo contrario se mantendrá en el formulario de autenticación. Si el usuario intenta ingresar a una funcionalidad del sistema sin haberse autenticado anteriormente, el sistema debe redireccionar el flujo de la aplicación al formulario inicial.	
Observaciones:	

Historia de usuario 2: Listar usuarios

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre: Listar usuarios en el sistema.	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para listar los usuarios con permisos de acceso al sistema, para posteriormente ser consultados o modificados. Se debe mostrar en forma de tabla, en el cual se mostrara información básica para cada usuario, como nombre, teléfono, usuario, además de las opciones para modificar o consultar cada registro.	
Observaciones:	

Historia de usuario 3: Consultar/Modificar usuario

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre: Consultar/modificar usuarios en el sistema	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Luego de listar los usuarios, se deben mostrar opciones para consultar o modificar un usuario determinado. Al ser seleccionado se desplegará un formulario con campos diligenciados con información respectiva: Nombre completo, teléfono, dirección, número de identificación, última conexión al sistema. Además de botones que permitan guardar un cambio (si se seleccionó modificar) y un botón para volver atrás.	
Observaciones:	

Historia de usuario 4: Crear proveedor

Historia de Usuario	
Número: 4	Usuario: Administrador
Nombre: Crear un proveedor en el sistema	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer al usuario la opción de crear un nuevo proveedor, al seleccionar esta opción se debe mostrar en pantalla los campos necesarios para diligenciar información básica como: Nombre, dirección, teléfono, correo electrónico y una descripción. Además, se debe contar con una opción para volver a la página anterior y una opción para guardar dicho proveedor en el sistema.	
Observaciones:	

Historia de usuario 5: Listar proveedores

Historia de Usuario	
Número: 5	Usuario: Administrador
Nombre: Consultar/modificar proveedor en el sistema	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada: 1
Descripción: El usuario podrá listar los proveedores existentes en el sistema para así consultar o modificar la información de este. La información se debe mostrar en forma de lista, en la cual se mostrará: Nombre, teléfono, dirección, ciudad y descripción. La información podrá se filtrada según nombre, ciudad ó descripción. Además se contara con una opción para seleccionar un registro en particular, ya sea para consultar o modificar dicho proveedor.	
Observaciones:	

Historia de usuario 6: Consultar/Modificar proveedor

Historia de Usuario	
Número: 6	Usuario: Administrador
Nombre: Consultar/modificar proveedor en el sistema	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada: 1
Descripción: Al seleccionar un registro a consultar o modificar en la historia de usuario 5. Se debe desplegar un formulario con campos diligenciados según la información de cada proveedor, siendo estos editables o no, según la opción seleccionada, consultar=no editables o modificar=editables. Adicional al formulario, se debe mostrar una lista con los productos adquiridos con este proveedor, organizados según la fecha de compra más actual y permitiendo filtrar según nombre de material, tipo de material.	
Observaciones:	

Historia de usuario 7: Ingresar material a inventario

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre: Ingresar materiales al inventario	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al seleccionar la opción de crear un nuevo material, el sistema debe mostrar en pantalla los campos necesarios para diligenciar la información básica del material, como: Nombre, cantidad, tipo de material (Inoxidable, hierro, galvanizado, aluminio), proveedor, precio de compra. Además de los botones para guardar y volver atrás.	
Observaciones: Si selecciona volver atrás, los cambios se perderán.	

Historia de usuario 8: Listar material en inventario

Historia de Usuario	
Número: 8	Usuario: Administrador
Nombre: Consultar/modificar materiales en inventario	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe mostrar una lista de los materiales disponibles en la empresa, posteriormente el usuario podrá seleccionar uno de estos materiales para Consultar o modificar. La información se podrá filtrar por el nombre, tipo de material o proveedor. La lista inicialmente esta ordenada descendientemente según la fecha de compra.	
Observaciones:	

Historia de usuario 9: Consultar/Modificar material

Historia de Usuario	
Número: 9	Usuario: Administrador
Nombre: Consultar/modificar materiales en inventario	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al seleccionar un registro el sistema debe desplegar un formulario diligenciado con la información básica de cada material. En el formulario se debe desplegar la información básica para dicho material, en la cual los campos serán editables o no editables según sea el caso: modificar o consultar respectivamente.	
Observaciones: Si el usuario selecciona volver los cambios realizados se perderán.	

Historia de usuario 10: Crear empleado

Historia de Usuario	
Número: 10	Usuario: Administrador
Nombre: Crear empleado	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema tendrá la opción de crear un nuevo empleado, al ser seleccionada se mostraran campos para diligenciar la siguiente información: Nombre, dirección, número de identificación, teléfono, cargo; además permitirá agregar habilidades relacionadas al empleado. Se mostraran opciones para guardar y volver atrás, si selecciona volver atrás se perderán los cambios. No se permite crear un empleado que ya a sido creado anteriormente en el sistema.	
Observaciones:	

Historia de usuario 11: Listar empleados

Historia de Usuario	
Número: 11	Usuario: Administrador
Nombre: Listar empleados	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada: 1
Descripción: El sistema mostrará la opción para listar los empleados existentes en el sistema, para posteriormente ser consultados/modificados (Historia de usuario 12). Se mostrara en forma de tabla información como: Identificación, nombre, apellido, teléfono, fecha de contratación, además de dos opciones para consultar o modificar respectivamente.	
Observaciones:	

Historia de usuario 12: Consultar/Modificar empleado

Historia de Usuario	
Número: 12	Usuario: Administrador
Nombre: Consultar/modificar empleado	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada: 1
Descripción: Al seleccionar un empleado se mostrará un formulario diligenciado con la información básica correspondiente. Los campos serán editables o no, según la opción seleccionada consultar=no editable y modificar=editable. En la parte inferior se mostraran los proyectos a los que a estado vinculado este empleado. Si selecciono modificar se renderizara un botón para guardar los cambios, al ser seleccionado dicho botón guardará los cambios efectuados y posteriormente se mostrara un mensaje en la parte superior donde se especifique si el proceso fue exitoso.	

Si selecciona volver atrás antes de guardar, los cambios se perderán.

Observaciones:

Historia de usuario 13: Crear proyecto

Historia de Usuario	
Número: 13	Usuario: Administrador
Nombre: Crear proyecto	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El usuario administrador presiona en la opción de crear nuevo proyecto, el sistema muestra campos en los cuales se podrá diligenciar información básica como: Nombre, cliente, fecha de inicio, fecha de fin, encargado del proyecto, estado. Posterior a esto, se podrá realizar la cotización de dicho proyecto (Historia de usuario 14).	
Observaciones: Los proyectos tienen estados según el proceso en que se encuentren (cotización, ejecución, terminado, cancelado).	

Historia de Usuario 14: Crear cotización

Historia de Usuario	
Número: 14	Usuario: Administrador
Nombre: Crear cotización	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Luego de crear el proyecto se inicia con la cotización, para lo cual el sistema debe mostrar campos para que el usuario diligencie la siguiente información: • Materiales necesarios, para lo cual el sistema debe facilitar el diligenciamiento, mostrando la disponibilidad de este en la empresa, si se encuentra disponible mostrar el costo de este, de lo contrario avisar que el material no está disponible y mostrar un promedio de los costos anteriores. • Horas de trabajo necesarias según cargo.	

- Gastos adicionales (Transporte, etc).

Observaciones:

Historia de usuario 15: Crear actividades

Historia de Usuario	
Número: 15	Usuario: Administrador
Nombre: Crear actividades	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema ofrecerá la opción para crear actividades, las cuales serán tareas específicas para el desarrollo del proyecto, estas tendrán la siguiente información: Nombre, fecha de inicio, fecha de fin, empleado responsable, riesgo y oficio de la actividad. Se mostrará la opción para guardar la actividad o para volver atrás.	
Observaciones:	

Historia de usuario 16: Listar proyectos

Historia de Usuario	
Número: 16	Usuario: Administrador
Nombre: listar proyectos en el sistema	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema ofrecerá la opción para consultar proyectos, al seleccionarla se listaran los proyectos según especifique un filtro (En cotización, en ejecución, terminados, cancelados), desplegando una lista que tendrá información como: Nombre del proyecto, empleado responsable, cliente, estado. Además de ofrecer la opción para seleccionar un proyecto en específico para ser consultado o modificado.	

Observaciones:

Historia de usuario 17: Consultar/Modificar proyectos

Historia de Usuario	
Número: 17	Usuario: Administrador
Nombre: Consultar/modificar proyectos	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al seleccionar un proyecto para ser modificado o consultado, se despliega un formulario diligenciado con toda la información correspondiente al proyecto, además en la parte inferior se listan las actividades correspondientes al proyecto. Los campos serán editables según la selección, consultar=no editables y modificar=editables. Si se selecciona modificar se podrán crear nuevas actividades para el proyecto o eliminar las creadas. Se muestra un botón para guardar los cambios y para volver atrás, al guardar los cambios se muestra un mensaje donde se especifica si el proceso fue exitoso. Al presionar volver atrás se perderán los cambios que no se han guardado.	
Observaciones:	

Historia de usuario 18: Crear cliente

Historia de Usuario	
Número: 18	Usuario: Administrador
Nombre: Crear cliente	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para crear un nuevo cliente en el sistema, al ser seleccionada se desplegarán campos suficientes para diligenciar la información necesaria del cliente, como es: Nombre, dirección, ciudad, correo, teléfono, empresa.	

Si la operación de guardar es exitosa se redirecciona a la página principal y se mostrará un mensaje donde se especifique que el proceso fue exitoso. De lo contrario permanece en la misma página y mostrara un mensaje de error.

Observaciones:

Historia de usuario 19: Listar clientes

Historia de Usuario	
Número: 19	Usuario: Administrador
Nombre: listar clientes	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para consultar todos los clientes que se encuentran en el sistema, ofreciendo la opción para filtrar según nombre, ciudad o empresa. Además cada cliente tendrá la opción para ser modificado o consultado.	
Observaciones:	

Historia de usuario 20: Consultar/Modificar cliente

Historia de Usuario	
Número: 20	Usuario: Administrador
Nombre: Consultar/modificar clientes	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al seleccionar un cliente para consultar o modificar, se desplegará un formulario diligenciado con la información básica del cliente, además en la parte inferior se mostraran los proyectos que se han realizado con dicho cliente. Los campos serán editables según la opción escogida, consultar=no editables y modificar=editables. Se ofrecen opciones para guardar los cambios realizados o volver atrás, si selecciona volver atrás se perderán los cambios no guardados.	



PROYECTO:	Gestión de proyectos			Página 13 de 13
DOCUMENTO:	Historias de usuario			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Observaciones:



PROYECTO:	Gestión de proyectos			Página 1 de 13
DOCUMENTO:	Historias de usuario			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Historias de usuario

Gestión de proyectos



PROYECTO:	Gestión de proyectos			Página 2 de 13
DOCUMENTO:	Historias de usuario			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Historia de usuario 7: Ingresar material a inventario	3
Historia de usuario 8: Listar material en inventario.....	3
Historia de usuario 9: Consultar/Modificar inventario	4
Historia de usuario 13: Crear proyecto	4
Historia de usuario 14: Listar proyectos	5
Historia de usuario 15: Consultar/Modificar proyectos.....	5
Historia de Usuario 16: Crear cotización.....	6
Historia de usuario 17: Crear actividades	6
Historia de usuario 18: Crear cliente.....	7
Historia de usuario 19: Listar clientes	7
Historia de usuario 20: Consultar/Modificar cliente.....	8
Historia de usuario 21: Crear empresa	9
Historia de usuario 22: Listar empresas	9
Historia de usuario 23: Consultar/Modificar empresa.....	10
Historia de usuario 24: Crear riesgo.....	10
Historia de usuario 25: Listar riesgos	11
Historia de usuario 26: Consultar/Modificar riesgo.....	11
Historia de usuario 27: Crear oficio.....	12
Historia de usuario 28: Listar oficio.....	12
Historia de usuario 29: Consultar/Modificar oficio.....	13

Historia de usuario 7: Ingresar material a inventario

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre: Ingresar materiales al inventario	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al seleccionar la opción de crear un nuevo material, el sistema debe mostrar en pantalla los campos necesarios para diligenciar la información básica del material, como: Nombre, cantidad, tipo de material (Inoxidable, hierro, galvanizado, aluminio), proveedor, precio de compra. Además de los botones para guardar y volver atrás.	
Observaciones: Si selecciona volver atrás, los cambios se perderán.	

Historia de usuario 8: Listar material en inventario

Historia de Usuario	
Número: 8	Usuario: Administrador
Nombre: Consultar/modificar materiales en inventario	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe mostrar una lista de los materiales disponibles en la empresa, posteriormente el usuario podrá seleccionar uno de estos materiales para Consultar o modificar. La información se podrá filtrar por el nombre, tipo de material o proveedor. La lista inicialmente esta ordenada descendientemente según la fecha de compra.	
Observaciones:	

Historia de usuario 9: Consultar/Modificar inventario

Historia de Usuario	
Número: 9	Usuario: Administrador
Nombre: Consultar/modificar materiales en inventario	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al seleccionar un registro el sistema debe desplegar un formulario diligenciado con la información básica de cada material. En el formulario se debe desplegar la información básica para dicho material, en la cual los campos serán editables o no editables según sea el caso: modificar o consultar respectivamente.	
Observaciones: Si el usuario selecciona volver los cambios realizados se perderán.	

Historia de usuario 13: Crear proyecto

Historia de Usuario	
Número: 13	Usuario: Administrador
Nombre: Crear proyecto	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El usuario administrador presiona en la opción de crear nuevo proyecto, el sistema muestra campos en los cuales se podrá diligenciar información básica como: Nombre, cliente, fecha de inicio, fecha de fin, encargado del proyecto, estado. Posterior a esto, se podrá realizar la cotización de dicho proyecto (Historia de usuario 14).	
Observaciones: Los proyectos tienen estados según el proceso en que se encuentren (cotización, ejecución, terminado, cancelado).	

Historia de usuario 14: Listar proyectos

Historia de Usuario	
Número: 16	Usuario: Administrador
Nombre: listar proyectos en el sistema	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema ofrecerá la opción para consultar proyectos, al seleccionarla se listaran los proyectos según especifique un filtro (En cotización, en ejecución, terminados, cancelados), desplegando una lista que tendrá información como: Nombre del proyecto, empleado responsable, cliente, estado. Además de ofrecer la opción para seleccionar un proyecto en específico para ser consultado o modificado.	
Observaciones:	

Historia de usuario 15: Consultar/Modificar proyectos

Historia de Usuario	
Número: 17	Usuario: Administrador
Nombre: Consultar/modificar proyectos	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al seleccionar un proyecto para ser modificado o consultado, se despliega un formulario diligenciado con toda la información correspondiente al proyecto, además en la parte inferior se listan las actividades correspondientes al proyecto. Los campos serán editables según la selección, consultar=no editables y modificar=editables. Si se selecciona modificar se podrán crear nuevas actividades para el proyecto o eliminar las creadas. Se muestra un botón para guardar los cambios y para volver atrás, al guardar los cambios se muestra un mensaje donde se especifica si el proceso fue exitoso. Al presionar volver atrás se perderán los cambios que no se han guardado.	
Observaciones:	

Historia de Usuario 16: Crear cotización

Historia de Usuario	
Número: 14	Usuario: Administrador
Nombre: Crear cotización	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Luego de crear el proyecto se inicia con la cotización, para lo cual el sistema debe mostrar campos para que el usuario diligencie la siguiente información: • Materiales necesarios, para lo cual el sistema debe facilitar el diligenciamiento, mostrando la disponibilidad de este en la empresa, si se encuentra disponible mostrar el costo de este, de lo contrario avisar que el material no está disponible y mostrar un promedio de los costos anteriores. • Horas de trabajo necesarias según cargo. • Gastos adicionales (Transporte, etc).	
Observaciones:	

Historia de usuario 17: Crear actividades

Historia de Usuario	
Número: 15	Usuario: Administrador
Nombre: Crear actividades	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema ofrecerá la opción para crear actividades, las cuales serán tareas específicas para el desarrollo del proyecto, estas tendrán la siguiente información: Nombre, fecha de inicio, fecha de fin, empleado responsable, riesgo y oficio de la actividad. Se mostrará la opción para guardar la actividad o para volver atrás.	
Observaciones:	

Historia de usuario 18: Crear cliente

Historia de Usuario	
Número: 18	Usuario: Administrador
Nombre: Crear cliente	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para crear un nuevo cliente en el sistema, al ser seleccionada se desplegaran campos suficientes para diligenciar la información necesaria del cliente, como es: Nombre, dirección, ciudad, correo, teléfono, empresa. Si la operación de guardar es exitosa se redirecciona a la página principal y se mostrará un mensaje donde se especifique que el proceso fue exitoso. De lo contrario permanece en la misma página y mostrara un mensaje de error.	
Observaciones:	

Historia de usuario 19: Listar clientes

Historia de Usuario	
Número: 19	Usuario: Administrador
Nombre: listar clientes	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para consultar todos los clientes que se encuentran en el sistema, ofreciendo la opción para filtrar según nombre, ciudad o empresa. Además cada cliente tendrá la opción para ser modificado o consultado.	
Observaciones:	

Historia de usuario 20: Consultar/Modificar cliente

Historia de Usuario	
Número: 20	Usuario: Administrador
Nombre: Consultar/modificar clientes	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al seleccionar un cliente para consultar o modificar, se desplegará un formulario diligenciado con la información básica del cliente, además en la parte inferior se mostraran los proyectos que se han realizado con dicho cliente. Los campos serán editables según la opción escogida, consultar=no editables y modificar=editables. Se ofrecen opciones para guardar los cambios realizados o volver atrás, si selecciona volver atrás se perderán los cambios no guardados.	
Observaciones:	

Historia de usuario 21: Crear empresa

Historia de Usuario	
Número: 21	Usuario: Administrador
Nombre: Crear empresa	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para crear una empresa nueva, por lo que al ser seleccionada se debe desplegar una interfaz gráfica con campos suficientes para diligenciar la siguiente información: Nombre de la empresa, ciudad de la empresa. Si la operación de guardar es exitosa se redirecciona a la página principal y se mostrará un mensaje donde se especifique que el proceso fue exitoso. De lo contrario permanece en la misma página y mostrara un mensaje de error.	
Observaciones:	

Historia de usuario 22: Listar empresas

Historia de Usuario	
Número: 22	Usuario: Administrador
Nombre: Listar empresas	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para listar todas las empresas existentes en el sistema, las cuales se deben mostrar en forma de tabla, mostrando información como: Nombre de empresa y Ciudad de empresa. Ordenados alfabéticamente según el nombre y con la opción de ser filtrados según los campos.	
Observaciones:	

Historia de usuario 23: Consultar/Modificar empresa

Historia de Usuario	
Número: 23	Usuario: Administrador
Nombre: Consultar/Modificar empresa	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción de seleccionar una empresa en específico de una lista para ser consultada o modificada. Si se selecciona consultar los campos deben ser no editables y si por el contrario selecciono modificar los campos serán editables. Además se mostrara en forma de tabla los proyectos que se han desarrollado para dicha empresa.	
Observaciones:	

Historia de usuario 24: Crear riesgo

Historia de Usuario	
Número: 24	Usuario: Administrador
Nombre: Crear riesgo	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción de crear nuevos riesgos laborales, para lo cual se mostrará un formulario con campos suficientes para diligenciar la siguiente información: Nombre y Porcentaje de riesgo. Si la operación de guardar es exitosa se redirecciona a la página principal y se mostrará un mensaje donde se especifique que el proceso fue exitoso. De lo contrario permanece en la misma página y mostrara un mensaje de error.	
Observaciones:	

Historia de usuario 25: Listar riesgos

Historia de Usuario	
Número: 25	Usuario: Administrador
Nombre: Listar riesgos	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para listar los riesgos laborales existentes en el sistema, dicha información debe ser mostrada en forma de tabla con la opción de seleccionar un registro en específico para ser consultada o modificada posteriormente (Historia de usuario 26).	
Observaciones:	

Historia de usuario 26: Consultar/Modificar riesgo

Historia de Usuario	
Número: 26	Usuario: Administrador
Nombre: Consultar/Modificar riesgo	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para seleccionar un registro en específico para ser consultado o modificado. Si se selecciona consultar los campos aparecen no editables, de lo contrario aparecerán editables para su posterior modificación.	
Observaciones:	

Historia de usuario 27: Crear oficio

Historia de Usuario	
Número: 27	Usuario: Administrador
Nombre: Crear oficio	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para crear un nuevo oficio en el sistema, para lo cual se debe desplegar un formulario con campos suficientes para diligenciar la siguiente información: Nombre y precio. Si la operación de guardar es exitosa se redirecciona a la página principal y se mostrará un mensaje donde se especifique que el proceso fue exitoso. De lo contrario permanece en la misma página y mostrara un mensaje de error.	
Observaciones:	

Historia de usuario 28: Listar oficio

Historia de Usuario	
Número: 28	Usuario: Administrador
Nombre: Listar oficio	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para listar los oficios en el sistema, la información se mostrara en forma de tabla, además se debe ofrecer la opción para seleccionar un registro en específico para ser consultado o modificado posteriormente (Historia de usuario 29).	
Observaciones:	



PROYECTO:	Gestión de proyectos			Página 13 de 13	
DOCUMENTO:	Historias de usuario				
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio		

Historia de usuario 29: Consultar/Modificar oficio

Historia de Usuario	
Número: 29	Usuario: Administrador
Nombre: Consultar/Modificar oficio	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para consultar o modificar un oficio en específico, si se selecciona consultar, los campos serán deshabilitados, si por el contrario selecciona modificar los campos serán habilitados para modificaciones.	
Observaciones:	



PROYECTO:	Gestión de proyectos			Página 1 de 7
DOCUMENTO:	Historias de usuario			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Historias de usuario

Gestión de proyectos



PROYECTO:	Gestión de proyectos			Página 2 de 7
DOCUMENTO:	Historias de usuario			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Historia de usuario 30: Registrar cargos.	3
Historia de usuario 31: Listar cargos.	3
Historia de usuario 32: Consultar/Modificar cargo.....	4
Historia de usuario 33: Registrar tiempo trabajado en actividad.	4
Historia de usuario 34: Consultar tiempo trabajado en actividad.	5
Historia de usuario 35: Modificar tiempo trabajado en actividad.	5
Historia de Usuario 36: Guardar archivo de proyecto.	6
Historia de usuario 37: Listar archivos de proyecto.....	6
Historia de usuario 38: Eliminar archivo de proyecto.....	7

Historia de usuario 30: Registrar cargos.

Historia de Usuario	
Número: 30	Usuario: Administrador
Nombre: Registrar cargo	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para crear un nuevo cargo en el sistema, al ser seleccionada se desplegaran campos suficientes para diligenciar la información necesaria del cargo, como es: Nombre, descripción. Si la operación de guardar es exitosa se limpia el formulario y se mostrará un mensaje donde se especifique que el proceso fue exitoso. De lo contrario no se limpia el formulario y se muestra un mensaje de error especificando lo ocurrido. En la parte inferior del formulario se ofrecen botones para guardar y volver atrás.	
Observaciones: Si selecciona volver atrás, los cambios se perderán.	

Historia de usuario 31: Listar cargos.

Historia de Usuario	
Número: 31	Usuario: Administrador
Nombre: Listar cargos	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para mostrar una lista con los cargos registrados, lista en la cual se mostrara información respectiva a cada cargo (Nombre y descripción). Posteriormente el usuario podrá seleccionar un cargo para Consultar o modificar. La información se podrá filtrar por el nombre, o descripción. La lista inicialmente esta ordenada descendientemente según el nombre.	
Observaciones:	

Historia de usuario 32: Consultar/Modificar cargo.

Historia de Usuario	
Número: 9	Usuario: Administrador
Nombre: Consultar/modificar cargo	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al listar los cargos (Historia de usuario 31) se ofrecerá la opción para seleccionar un registro en particular y ser modificado o consultado. Al seleccionar un registro el sistema debe desplegar un formulario diligenciado con la información respectiva. Los campos estarán habilitados o deshabilitados según a opción seleccionada por el usuario, Consultar: Deshabilitados, Modificar: Habilitados. En la parte inferior del formulario se ubicaran botones para Guardar cambios y Cancelar. Si el usuario presiona cancelar los cambios se perderán.	
Observaciones:	

Historia de usuario 33: Registrar tiempo trabajado en actividad.

Historia de Usuario	
Número: 13	Usuario: Administrador
Nombre: Registrar tiempo trabajado en actividad.	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para registrar tiempos trabajados en una actividad específica, al ser seleccionada se desplegaran campos suficientes para diligenciar información como: Proyecto, Actividad (se carga según el proyecto seleccionado), Fecha a registrar, hora de inicio y hora de finalización. Si la operación de guardar es exitosa se limpia el formulario y se mostrará un mensaje donde se especifique que el proceso fue exitoso. De lo contrario permanece la información diligenciada y se muestra un mensaje de error especificando lo ocurrido.	

En la parte inferior del formulario se ofrecen botones para guardar y volver atrás.

Observaciones:

Al registrar el tiempo, se debe calcular las horas trabajadas que serán cargadas a la actividad seleccionada. Se debe tener en cuenta si la hora trabajada es Normal, Extra o dominical.

Historia de usuario 34: Consultar tiempo trabajado en actividad.

Historia de Usuario	
Número: 34	Usuario: Administrador
Nombre: Consultar tiempo trabajado en actividad	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción para consultar los tiempos trabajados en una actividad, pudiendo consultar de manera general los tiempos totales (Horas normales, extras y dominicales), además de poder consultar en detalle los tiempos trabajados por los empleados mostrando la fecha y hora en que se trabajó.	
Observaciones:	

Historia de usuario 35: Modificar tiempo trabajado en actividad.

Historia de Usuario	
Número: 35	Usuario: Administrador
Nombre: Modificar tiempo trabajado en actividad.	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al mostrar el detalle de los tiempos se debe ofrecer la opción para eliminar o modificar un registro, al seleccionar <i>modificar</i> se cargará la información respectiva en un formulario en el que se podrá editar lo siguientes campos: Fecha registrada, hora de inicio, hora de fin, descripción. Se debe mostrar un botón para Guardar cambios , y Cancelar . Al realizar la modificación se debe recalcular la suma de las horas trabajadas.	

Observaciones:

Historia de Usuario 36: Guardar archivo de proyecto.

Historia de Usuario	
Número: 36	Usuario: Administrador
Nombre: Guardar archivo de proyecto	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Se debe permitir la opción para guardar archivos de un proyecto. Al seleccionar esta opción se desplegará un formulario en el cual se debe diligenciar los siguientes campos: - Nombre del archivo, campo alfanumérico obligatorio. - Archivo, campo de selección obligatorio. Para cargar el archivo se debe ofrecer una ventana de navegación en la cual se pueda buscar el archivo en el equipo en que se accede. Se debe ofrecer las opciones para Guardar y Cancelar . Al finalizar se mostrará un mensaje de información especificando si el proceso fue exitoso o si ocurrió un error.	
Observaciones:	

Historia de usuario 37: Listar archivos de proyecto.

Historia de Usuario	
Número: 37	Usuario: Administrador
Nombre: Listar archivos de proyecto	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Se debe permitir la opción de listar todos los archivos que se han guardado de un proyecto en particular, para esto, se debe seleccionar un proyecto para ser consultado o modificado, en una pestaña se debe mostrar la opción de Archivos en la cual se mostrará en forma de lista todos los archivos que se han guardado, estos deben estar ordenados de	



PROYECTO:	Gestión de proyectos		Página 7 de 7
DOCUMENTO:	Historias de usuario		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

manera alfabética según el nombre y podrán ser filtrados por palabras clave.

Observaciones:

Historia de usuario 38: Eliminar archivo de proyecto.

Historia de Usuario	
Número: 18	Usuario: Administrador
Nombre: Eliminar archivo de proyecto	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al listar los archivos de un proyecto en particular, se debe ofrecer la opción para eliminar un archivo en específico. Al seleccionar eliminar, se debe mostrar un mensaje de confirmación para verificar si realmente se desea eliminar el archivo. Si el usuario acepta se procede a ejecutar la acción correspondiente a la eliminación, de lo contrario continua el flujo normal.	
Observaciones:	



PROYECTO:	Gestión de proyectos			Página 1 de 8
DOCUMENTO:	Historias de usuario			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Historias de usuario

Gestión de proyectos



PROYECTO:	Gestión de proyectos			Página 2 de 8
DOCUMENTO:	Historias de usuario			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Historia de usuario 39: Crear ODT.	3
Historia de usuario 40: Consultar/Modificar ODT.....	4
Historia de usuario 41: Exportar ODT.	4
Historia de usuario 42: Exportar cotización.	5
Historia de usuario 43: Modificar Usuarios.....	5
Historia de usuario 44: Configurar días no laborables.....	6
Historia de usuario 45: Configurar jornada laboral.....	6
Historia de usuario 46: Registro de tiempo trabajado.....	7
Historia de usuario 47: Registro de materiales usados.....	8

Historia de usuario 39: Crear ODT.

Historia de Usuario	
Número: 39	Usuario: Administrador
Nombre: Crear ODT	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe crear una orden de trabajo ODT automáticamente al registrar un proyecto, dicha ODT debe contar con los siguientes campos: • Proyecto al que pertenece. • Cliente. • Responsable del proyecto en la empresa. • Nombre, campo alfanumérico. Inicialmente tendrá el mismo nombre del proyecto. • Descripción, campo alfanumérico. Inicialmente tendrá la misma descripción del proyecto. • Fecha de inicio y de entrega. • Empleados encargados en la ejecución. • Materiales cotizados en el proyecto. Además, tendrá asociado el tiempo trabajado de los empleados y los materiales cotizados en dicho proyecto.	
Observaciones:	

Historia de usuario 40: Consultar/Modificar ODT.

Historia de Usuario	
Número: 40	Usuario: Administrador
Nombre: Consultar/Modificar ODT	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al seleccionar un proyecto para ser consultado o modificado, se debe ofrecer la opción de mostrar la ODT del proyecto con todos sus datos correspondientes. Será permitido la edición solo de los siguientes campos: • Nombre o detalle. • Descripción de ODT. • Empleados encargados de la ejecución del proyecto.	
Observaciones:	

Historia de usuario 41: Exportar ODT.

Historia de Usuario	
Número: 41	Usuario: Administrador
Nombre: Exportar ODT	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Al mostrar la ODT (Historia de usuario 39) se debe ofrecer la opción de exportar en formato PDF. Se mostrarán todos los datos especificados en la historia de usuario 39 .	
Observaciones:	

Historia de usuario 42: Exportar cotización.

Historia de Usuario	
Número: 42	Usuario: Administrador
Nombre: Exportar cotización.	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: En la sección de cotización de un proyecto seleccionado, se debe ofrecer la opción de “Exportar cotización”, al ser seleccionada se descargará automáticamente un archivo en formato XLSX con el detalle de la cotización realizada en dicho proyecto, se deben mostrar por separado las secciones de: Mano de obra, Materiales cotizados y gastos adicionales, con sus respectivos subtotales.	
Observaciones:	

Historia de usuario 43: Modificar Usuarios.

Historia de Usuario	
Número: 43	Usuario: Administrador
Nombre: Listar Usuarios.	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: En la sección de configuraciones se debe ofrecer la opción para listar todos los usuarios con permisos de acceso al sistema. La información debe ser mostrada en forma de tabla con los siguientes datos: • Nombre de usuario, podrá ser reordenado por este campo. • Estado de acceso al sistema. • Opción para activar o desactivar permisos de acceso al sistema.	
Observaciones: El estado de acceso al sistema estará denotado por activo o inactivo.	

Historia de usuario 44: Configurar días no laborables.

Historia de Usuario	
Número: 44	Usuario: Administrador
Nombre: Configurar días no laborables.	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: En la sección de configuración se debe ofrecer la opción para agregar fechas no laborables (fechas definidas por el gerente o fechas festivas) las cuales serán tenidas en cuenta para calcular horas extras o festivas trabajadas. Se debe tener en cuenta la siguiente información para cada fecha: • Fecha, campo seleccionable obligatorio. • Descripción, campo alfanumérico que describirá el porqué de la fecha. • Día festivo, especifica si es un día festivo o día normal no laborable.	
Observaciones:	

Historia de usuario 45: Configurar jornada laboral.

Historia de Usuario	
Número: 45	Usuario: Administrador
Nombre: Configurar jornada laboral.	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: El sistema debe ofrecer la opción de configurar la jornada laboral de la empresa, configurando para cada día por separado la siguiente información: • Día de la semana. • Hora de entrada en la mañana. • Hora de salida en la mañana. • Hora de entrada en la tarde. • Hora de salida en la mañana. Estos datos, serán tenidos en cuenta al momento de registrar una actividad desarrollada por parte de un empleado.	

Observaciones:

Historia de usuario 46: Registro de tiempo trabajado.

Historia de Usuario	
Número: 46	Usuario: Administrador
Nombre: registro de tiempo trabajado.	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Se debe ofrecer la opción de realizar registros de actividades de los empleados, teniendo en cuenta información como: • ODT de la actividad desarrollada. • Nombre del empleado quien realiza la actividad. • Fecha en que se realiza la actividad. • Indicar si existió una novedad en la ejecución de la actividad. • Hora de inicio y hora de fin de la actividad. • Descripción de la actividad desarrollada. Al realizar el registro de la actividad, se debe verificar si el intervalo de tiempo se encuentra dentro de la jornada laboral del día, de lo contrario calcular cuantas horas extras o festivas se laboraron. Luego del registro de la actividad, se debe actualizar los datos del consolidado de horas de la ODT, la cual puede ser consultada por el encargado.	
Observaciones: Se entiende por novedad a un indicador con el cual el encargado puede identificar un acontecimiento anormal en este intervalo de tiempo.	

Historia de usuario 47: Registro de materiales usados.

Historia de Usuario	
Número: 47	Usuario: Administrador
Nombre: Registro de materiales usados.	
Prioridad en negocio: Media (Alta/Media/Baja)	Puntos estimados:
Prioridad en desarrollo: Media (Alta/Media/Baja)	Iteración Asignada:
Descripción: Se debe ofrecer la opción de realizar registros de materiales usados en un proyecto en particular, teniendo en cuenta la siguiente información: • ODT donde se usa el material. • Empleado quien hace la petición del material. • Material usado. • Cantidad de unidades. • Largo y ancho, estos campos se habilitan según el material seleccionado. • Fecha en que se usa el material. • Descripción del uso que se le dará al material.	
Observaciones:	

ANEXO 2



PROYECTO:	Gestión de proyectos			Página 1 de 2
DOCUMENTO:	Plan de Entregas			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Plan de entregas

Gestión de proyectos

Plan de entregas:

Iteración	Hist. De Usuario	Tarea de ingeniería		Tiempo estimado	Fecha de entrega
1	4 – Crear proveedor	4.1	Realizar método para funcionalidad	4 horas	Marzo 21 de 2016
		4.2	Realizar validaciones de datos.	4 horas	
		4.3	Crear interfaz gráfica	1 día	
		4.4	Pruebas unitarias.	2 horas	
	5 – Listar proveedores	5.1	Realizar métodos funcionales	4 horas	
		5.2	Realizar interfaz grafica	1 día	
		5.3	Pruebas unitarias	2 horas	
	6 – Consultar/Modificar proveedores	6.1	Crear métodos para funcionalidades modificar y consultar	4 horas	
		6.2	Crear interfaz grafica	1 día	
		6.3	Pruebas unitarias	2 horas	
	10 - Crear empleado	10.1	Crear métodos funcionales	1 día	
		10.2	Realizar validaciones de datos	4 horas	
		10.3	Crear interfaz grafica	4 horas	
		10.4	Pruebas unitarias	2 horas	
	11 - Listar empleados	11.1	Crear métodos funcionales	1 día	
		11.2	Crear interfaz grafica	1 día	
		11.3	Pruebas unitarias.	2 horas	
	12- Consultar/Modificar empleado	12.1	Crear métodos para funcionalidades Consultar/Modificar	1 día	
		12.2	Crear interfaz gráfica.	1 día	
		12.3	Pruebas unitarias.	2 horas	



PROYECTO:	Gestión de proyectos			Página 1 de 4
DOCUMENTO:	Plan de Entregas			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Plan de entregas

Gestión de proyectos

Plan de entregas:

Iteración	Hist. De Usuario	Tarea de ingeniería		Tiempo estimado	Fecha de entrega
2	7 - Ingresar material a inventario.	7.1	Realizar métodos funcionales.	2 horas	Marzo 29 2016
		7.2	Realizar validaciones.	2 horas	
		7.3	Crear interfaz gráfica.	2 horas	
		7.4	Pruebas unitarias.	1 hora	
	8 – Listar material en inventario.	8.1	Realizar métodos funcionales.	1 hora	
		8.2	Crear interfaz gráfica.	1 hora	
		8.3	Pruebas unitarias.	1 hora	
	9 – Consultar/Modificar inventario.	9.1	Realizar métodos funcionales.	2 horas	
		9.2	Realizar validaciones.	2 horas	
		9.3	Crear interfaz gráfica.	2 horas	
		9.4	Pruebas unitarias.	1 hora	
	13 – Crear proyecto.	13.1	Crear métodos funcionales.	2 horas	
		13.2	Realizar validaciones.	2 horas	
		13.3	Crear interfaz gráfica.	2 horas	
		13.4	Pruebas unitarias.	1 hora	
	14 – Listar proyectos	14.1	Crear métodos funcionales.	1 hora	
		14.2	Crear interfaz gráfica.	1 hora	
		14.3	Pruebas unitarias.	1 hora	
	15 – Consultar/Modificar proyecto.	15.1	Realizar métodos funcionales.	2 horas	
		15.2	Realizar validaciones.	2 horas	
		15.3	Crear interfaz gráfica.	2 horas	
		15.4	Pruebas unitarias.	1 hora	
	16- Crear cotización	16.1	Crear métodos funcionales.	2 horas	
		16.2	Realizar validaciones.	2 horas	
		16.3	Crear interfaz gráfica.	2 horas	
		16.4	Pruebas unitarias.	1 hora	
	17- Crear actividades	17.1	Crear métodos funcionales.	2 horas	
		17.2	Realizar validaciones.	2 horas	
		17.3	Crear interfaz gráfica.	2 horas	
		17.4	Pruebas unitarias.	1 hora	
	18 – Crear clientes	18.1	Crear métodos funcionales.	2 horas	
		18.2	Realizar validaciones.	2 horas	

		18.3	Crear interfaz gráfica.	2 horas	
		18.4	Pruebas unitarias.	1 hora	
19 – Listar clientes		19.1	Crear métodos funcionales.	1 hora	
		19.2	Crear interfaz gráfica.	1 hora	
		19.3	Pruebas unitarias.	1 hora	
20 – Consultar/Modificar Cliente		20.1	Crear métodos funcionales.	2 horas	
		20.2	Realizar validaciones.	2 horas	
		20.3	Crear interfaz gráfica.	2 horas	
		20.4	Pruebas unitarias.	1 hora	
21 – Crear empresa		21.1	Crear métodos funcionales.	30 minutos	
		21.2	Realizar validaciones.	30 minutos	
		21.3	Crear interfaz gráfica.	30 minutos	
		21.4	Pruebas unitarias.	10 minutos	
22 – Listar empresa		22.1	Crear métodos funcionales.	20 minutos	
		22.2	Crear interfaz gráfica.	20 minutos	
		22.3	Pruebas unitarias.	10 minutos	
23 – Consultar/Modificar Empresa		23.1	Crear métodos funcionales.	20 minutos	
		23.2	Realizar validaciones.	15 minutos	
		23.3	Crear interfaz gráfica.	20 minutos	
		23.4	Pruebas unitarias.	10 minutos	
24 – Crear riesgo		24.1	Crear métodos funcionales.	20 minutos	
		24.2	Realizar validaciones.	20 minutos	
		24.3	Crear interfaz gráfica.	30 minutos	
		24.4	Pruebas unitarias.	15 minutos	
25 – Listar riesgo		25.1	Crear métodos funcionales.	20 minutos	
		25.2	Crear interfaz gráfica.	20 minutos	
		25.3	Pruebas unitarias.	10 minutos	
26 – Consultar/Modificar riesgo		26.1	Crear métodos funcionales.	20 minutos	
		26.2	Realizar validaciones.	30 minutos	
		26.3	Crear interfaz gráfica.	30 minutos	
		26.4	Pruebas unitarias.	15 minutos	
27 – Crear oficio		27.1	Crear métodos funcionales.	20 minutos	
		27.2	Realizar validaciones.	20 minutos	
		27.3	Crear interfaz gráfica.	30 minutos	
		27.4	Pruebas unitarias.	15 minutos	
28 – Listar oficio		28.1	Crear métodos	20 minutos	



PROYECTO:	Gestión de proyectos			Página 4 de 4
DOCUMENTO:	Plan de Entregas			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

			funcionales.		
		28.2	Crear interfaz gráfica.	20 minutos	
		28.3	Pruebas unitarias.	10 minutos	
	29 – Consultar/Modificar oficio	29.1	Crear métodos funcionales.	20 minutos	
		29.2	Realizar validaciones.	20 minutos	
		29.3	Crear interfaz gráfica.	30 minutos	
		29.4	Pruebas unitarias.	10 minutos	



PROYECTO:	Gestión de proyectos			Página 1 de 2
DOCUMENTO:	Plan de Entregas			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Plan de entregas

Gestión de proyectos

Plan de entregas:

Iteración	Hist. De Usuario	Tarea de ingeniería		Tiempo estimado	Fecha de entrega
3	30 – Registrar cargos.	30.1	Realizar métodos funcionales.	1 hora	Abril 29 de 2016
		30.2	Realizar validaciones.	40 minutos	
		30.3	Crear interfaz gráfica.	1 hora	
		30.4	Pruebas unitarias.	30 minutos	
	31 – Listar cargos.	31.1	Realizar métodos funcionales.	30 minutos	
		31.2	Crear interfaz gráfica.	40 minutos	
		31.3	Pruebas unitarias.	20 minutos	
	32 – Consultar/Modificar cargo.	32.1	Realizar métodos funcionales.	40 minutos	
		32.2	Realizar validaciones.	40 minutos	
		32.3	Crear interfaz gráfica.	1 hora	
		32.4	Pruebas unitarias.	20 minutos	
	33 – Registrar tiempo trabajado en actividad.	33.1	Crear métodos funcionales.	2 horas	
		33.2	Realizar validaciones.	1 hora	
		33.3	Crear interfaz gráfica.	1 hora	
		33.4	Pruebas unitarias.	30 minutos	
	34 – Consultar tiempo trabajado en actividad.	34.1	Crear métodos funcionales.	30 minutos	
		34.2	Crear interfaz gráfica.	1 hora	
		34.3	Pruebas unitarias.	30 minutos	
	35 – Modificar tiempo trabajado en actividad.	35.1	Realizar métodos funcionales.	1 hora	
		35.2	Realizar validaciones.	1 hora	
		35.3	Crear interfaz gráfica.	1 hora	
		35.4	Pruebas unitarias.	1 hora	
	36 – Guardar archivo de proyecto.	36.1	Crear métodos funcionales.	1 hora	
		36.2	Realizar validaciones.	1 hora	
		36.3	Crear interfaz gráfica.	40 minutos	
		36.4	Pruebas unitarias.	30 minutos.	
	37 – Listar archivos de proyecto.	37.1	Crear interfaz gráfica.	40 minutos	
	38 – Eliminar archivos de proyecto.	38.1	Crear métodos funcionales.	1 hora	
		38.2	Realizar validaciones.	1 hora	
		38.3	Pruebas unitarias.	30 minutos	



PROYECTO:	Gestión de proyectos			Página 1 de 2
DOCUMENTO:	Plan de Entregas			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Plan de entregas

Gestión de proyectos

Plan de entregas:

Iteración	Hist. De Usuario	Tarea de ingeniería		Tiempo estimado	Fecha de entrega
4	39. Crear ODT	39.1	Realizar métodos funcionales.	20 minutos.	Septiembre 13 de 2016
		39.2	Pruebas unitarias.	15 minutos.	
	40. Consultar/Modificar ODT.	40.1	Realizar métodos funcionales.	20 minutos.	
		40.2	Realizar Validaciones.	20 minutos.	
		40.3	Crear interfaz gráfica.	1 hora.	
		40.4	Pruebas unitarias.	20 minutos.	
	41. Exportar ODT.	41.1	Realizar métodos funcionales.	30 minutos.	
		41.2	Realizar reporte.	1 hora	
		41.3	Pruebas unitarias.	20 minutos.	
	42. Exportar cotización.	42.1	Realizar métodos funcionales.	4 horas.	
		42.2	Realizar reporte.	1 hora.	
		42.3	Pruebas unitarias.	30 minutos.	
	43. Modificar usuarios.	43.1	Realizar métodos funcionales.	40 minutos.	
		43.2	Realizar interfaz gráfica.	1 hora.	
		43.3	Pruebas unitarias.	20 minutos.	
	44. Configurar días no laborables.	44.1	Realizar métodos funcionales.	40 minutos	
		44.2	Realizar validaciones.	30 minutos	
		44.3	Crear interfaz gráfica.	30 minutos	
		44.4	Pruebas unitarias.	20 minutos	
	45. Configurar jornada laboral	46.1	Realizar métodos funcionales.	1 hora	
		46.2	Crear interfaz gráfica.	1 hora	
		46.3	Pruebas unitarias.	20 minutos	
	46. Registrar tiempo trabajado.	47.1	Realizar métodos funcionales.	4 horas	
		47.2	Realizar validaciones.	1 hora	
		47.3	Crear interfaz gráfica.	1 hora	
		47.4	Pruebas unitarias.	40 minutos	
	47. Registro de material usado	48.1	Realizar métodos funcionales.	2 horas	
		48.2	Realizar validaciones.	1 hora	
		48.3	Crear interfaz gráfica.	1 hora	
		48.4	Pruebas unitarias.	30 minutos	

ANEXO 3



PROYECTO:	Gestión de proyectos			Página 1 de 5
DOCUMENTO:	Estimación de esfuerzo			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Estimación de esfuerzo

Iteración 1



PROYECTO:	Gestión de proyectos			Página 2 de 5
DOCUMENTO:	Estimación de esfuerzo			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Historias de usuario – Estimación de esfuerzo	3
Tareas de ingeniería – Detalle esfuerzo	4
Grafica – Tiempo estimado vs real	5

Historias de usuario – Estimación de esfuerzo

Historia de usuario	Tiempo Estimado	Tiempo real
4 - Crear un proveedor en el sistema	2 días, 2 horas	4 horas
5 - Consultar/modificar proveedor en el sistema	1 día, 6 horas	1 hora, 45 minutos
10 - Crear empleado	2 días, 2 horas	2 horas 50 minutos
11 - Listar empleados	2 días, 2 horas	1 hora 35 minutos
12 - Consultar/modificar empleado	2 días, 2 horas	2 horas 10 minutos

Fecha estimada para entrega: 21 de Marzo de 2016

Fecha real de la entrega: 16 de Marzo de 2016

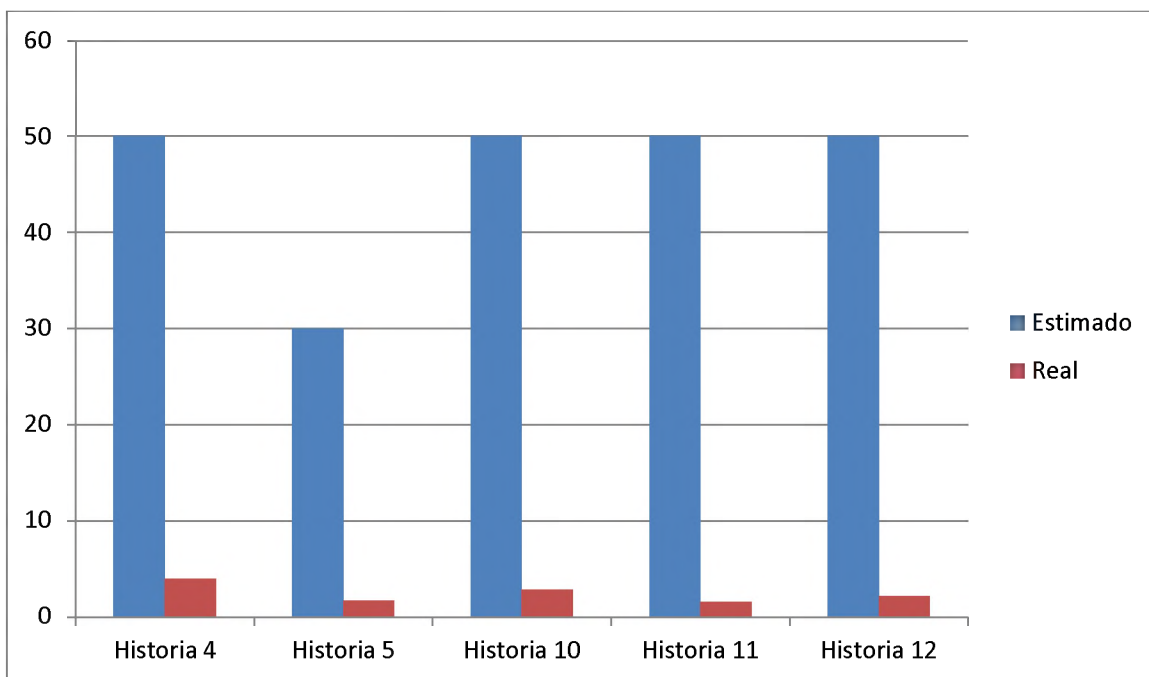
Tareas de ingeniería – Detalle esfuerzo

N°	Tarea de ingeniería	Tiempo Estimado	Hora Inicio	Hora fin	Errores	Estado
4	4.1 Realizar método para funcionalidad	4 horas	9:00am	10:00am	Ninguno	Finalizado
	4.2 Realizar validaciones de datos	4 Horas	10:00am	11: 00am	Ninguno	Finalizado
	4.3 Crear interfaz gráfica	1 día	11:00am	13:00pm	Ninguno	Finalizado
	4.4 pruebas unitarias	2 horas	14:00pm	15:00pm	Error con codificación UTF-8	Finalizado
5	5.1 Realizar métodos funcionales	4 horas	15:00pm	15:30pm	Ninguno	Finalizado
	5.2 Realizar interfaz grafica	1 día	15:30pm	16:30pm	Ninguno	Finalizado
	5.3 Pruebas unitarias	2 horas	16:30pm	16:45pm	Ninguno	Finalizado
6	6.1 Crear métodos para funcionalidades modificar y consultar	4 horas	17:00pm	17:50pm	Ninguno	Finalizado
	6.2 Crear interfaz grafica	1 día	18:00pm	20:00pm	Ninguno	Finalizado
	6.3 Pruebas unitarias	2 horas	20:00pm	20:30pm	Ninguno	Finalizado
10	10.1 Crear métodos funcionales	1 día	8:30am	9:20am	Ninguno	Finalizado
	10.2 Realizar validaciones de datos	4 horas	9:30am	11:00am	Ninguno	Finalizado
	10.3 Crear interfaz grafica	4 horas	11:00am	13:00pm	Ninguno	Finalizado
	10.4 Pruebas unitarias	2 horas	13:00pm	13:30pm	Ninguno	Finalizado
11	11.1 Crear métodos funcionales	1 día	14:00pm	14:30pm	Ninguno	Finalizado
	11.2 Crear interfaz grafica	1 día	14:40pm	15:30pm	Ninguno	Finalizado
	11.3 Pruebas unitarias	2 horas	15:30pm	15:45pm	Ninguno	Finalizado
12	12.1 Crear métodos	1 día	16:00pm	16:30pm	Ninguno	Finalizado

	para funcionalidades Consultar/Modificar					
	12.2 Crear interfaz gráfica.	1 día	16:40pm	17:50pm	Ninguno	Finalizado
	12.3 Pruebas unitarias.	2 horas	18:00pm	18:30pm	Ninguno	Finalizado

Grafica - Tiempo estimado vs real

Los valores están dados en horas.





PROYECTO:	Gestión de proyectos			Página 1 de 7
DOCUMENTO:	Estimación de esfuerzo			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Estimación de esfuerzo

Iteración 2



PROYECTO:	Gestión de proyectos			Página 2 de 7
DOCUMENTO:	Estimación de esfuerzo			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Historias de usuario – Estimación de esfuerzo	3
Tareas de ingeniería – Detalle esfuerzo	4
Grafica – Tiempo estimado vs real	7

Historias de usuario – Estimación de esfuerzo

Historia de usuario	Tiempo Estimado	Tiempo real
7- Ingresar material a inventario.	7 Horas	2 Horas
8- Listar material en inventario.	3 Horas	1 Hora
9- Consultar/Modificar inventario.	7 Horas	1,83 Horas
13- Crear proyecto.	7 Horas	1,41 Horas
14- Listar proyectos.	3 Horas	1 Hora
15- Consultar/Modificar proyectos.	7 Horas	0.75 Horas
16- Crear cotización.	7 Horas	8,66 Horas
17- Crear actividades.	7 Horas	8,83 Horas
18- Crear cliente.	7 Horas	1 Hora
19- Listar clientes.	3 Horas	0,41 Horas
20- Consultar/Modificar cliente.	7 Horas	1,25 Horas
21- Crear empresa.	1,6 Horas	0,58 Horas
22- Listar empresas.	0,83 Horas	0,33 Horas
23- Consultar/Modificar empresa.	1,08 Horas	0,41 Horas
24- Crear riesgo.	1,41 Horas	1,16 Horas
25- Listar riesgos.	0,83 Horas	0,58 Horas
26- Consultar/Modificar riesgo.	1,58 Horas	0,75 Horas
27- Crear oficio.	1,41 Horas	0,66 Horas
28- Listar oficio.	0,83 Horas	0,33 Horas
29- Consultar/Modificar oficio.	1,3 Horas	0,66 Horas

Fecha estimada para entrega: Marzo 29 de 2016

Fecha real de la entrega: Abril 2 de 2016

Tareas de ingeniería - Detalle esfuerzo

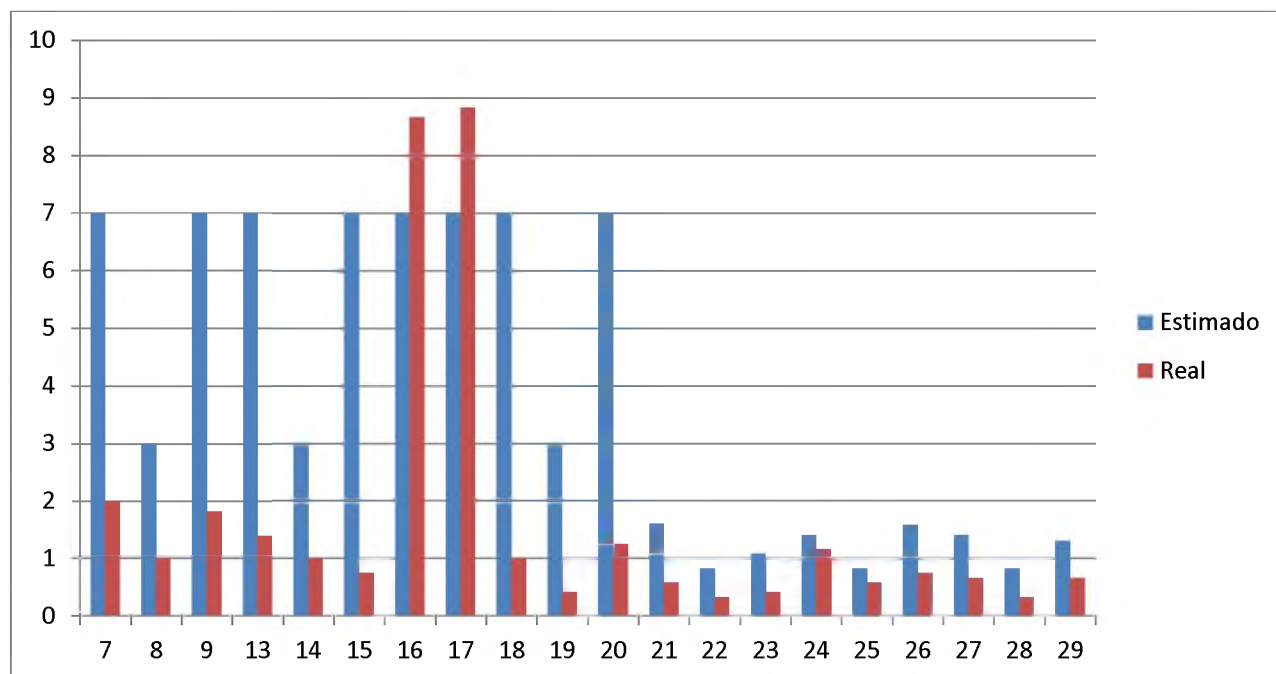
N°	Tarea de ingeniería	Tiempo Estimado	Hora Inicio	Hora fin	Errores	Estado
7	Realizar métodos funcionales.	2 horas	9:00	9:20	Ninguno	Finalizado
	Realizar validaciones.	2 horas	9:45	10:30	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	10:35	11:20	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	11:20	11:30	Ninguno	Finalizado
8	Realizar métodos funcionales.	1 hora	11:30	11:50	Ninguno	Finalizado
	Crear interfaz gráfica.	1 hora	11:50	12:20	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	12:20	12:30	Ninguno	Finalizado
9	Realizar métodos funcionales.	2 horas	14:00	14:20	Ninguno	Finalizado
	Realizar validaciones.	2 horas	14:20	14:50	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	14:50	15:30	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	15:30	15:50	Ninguno	Finalizado
13	Crear métodos funcionales.	2 horas	15:50	16:00	Ninguno	Finalizado
	Realizar validaciones.	2 horas	16:00	16:15	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	16:20	17:10	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	17:15	17:25	Ninguno	Finalizado
14	Crear métodos funcionales.	1 hora	17:30	17:40	Ninguno	Finalizado
	Crear interfaz gráfica.	1 hora	17:40	18:20	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	18:20	18:30	Ninguno	Finalizado
15	Realizar métodos funcionales.	2 horas	19:30	19:35	Ninguno	Finalizado
	Realizar validaciones.	2 horas	19:40	19:50	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	19:50	20:10	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	20:10	20:20	Ninguno	Finalizado
16	Crear métodos funcionales.	2 horas	20:30	20:40	Ninguno	Finalizado
	Realizar validaciones.	2 horas	21:00	00:00	Cargue de datos, identificación de detalles agregados (Material, gasto	Finalizado
			9:00	14:00		

					adicional, oficio- riesgo)	
	Crear interfaz gráfica.	2 horas	15:00	17:00	Ajax, renderizaci on, tabular informació n.	Finalizado
	Pruebas unitarias.	1 hora	17:20	17:50	Ninguno	Finalizado
17	Crear métodos funcionales.	2 horas	18:30	18:40	Ninguno	Finalizado
	Realizar validaciones.	2 horas	19:00	01:00	Seleccionar actividad, modificar elementos seleccionad os.	Finalizado
	Crear interfaz gráfica.	2 horas	9:30	11:30	Renderizaci on de component es, tabular informació n,	Finalizado
	Pruebas unitarias.	1 hora	11:40	12:20	Ninguno	Finalizado
	Crear métodos funcionales.	2 horas	16:00	16:15	Ninguno	Finalizado
18	Realizar validaciones.	2 horas	16:15	16:35	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	16:35	16:55	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	16:55	17:00	Ninguno	Finalizado
	Crear métodos funcionales.	1 hora	17:00	17:05	Ninguno	Finalizado
19	Crear interfaz gráfica.	1 hora	17:05	17:20	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	17:20	17:25	Ninguno	Finalizado
	Crear métodos funcionales.	2 horas	17:30	17:40	Ninguno	Finalizado
20	Realizar validaciones.	2 horas	17:45	18:00	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	18:00	18:40	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	18:40	18:50	Ninguno	Finalizado
	Crear métodos funcionales.	2 horas	22:30	22:35	Ninguno	Finalizado
	Realizar validaciones.	2 horas	22:35	22:45	Ninguno	Finalizado
21	Crear interfaz gráfica.	2 horas	22:45	23:00	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	23:00	23:05	Ninguno	Finalizado
	Crear métodos	1 hora	23:05	23:10	Ninguno	Finalizado
22	Crear métodos	1 hora	23:05	23:10	Ninguno	Finalizado

	funcionales.					
	Crear interfaz gráfica.	1 hora	23:10	23:20	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	23:20	23:25	Ninguno	Finalizado
23	Crear métodos funcionales.	2 horas	23:25	23:30	Ninguno	Finalizado
	Realizar validaciones.	2 horas	23:30	23:35	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	23:35	23:45	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	23:45	23:50	Ninguno	Finalizado
24	Crear métodos funcionales.	2 horas	10:00	10:30	Ninguno	Finalizado
	Realizar validaciones.	2 horas	10:30	10:40	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	10:40	11:00	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	11:00	11:10	Ninguno	Finalizado
25	Crear métodos funcionales.	1 hora	11:15	11:20	Ninguno	Finalizado
	Crear interfaz gráfica.	1 hora	11:20	11:40	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	11:40	11:50	Ninguno	Finalizado
26	Crear métodos funcionales.	2 horas	11:55	12:10	Ninguno	Finalizado
	Realizar validaciones.	2 horas	12:10	12:25	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	12:25	12:35	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	12:35	12:40	Ninguno	Finalizado
27	Crear métodos funcionales.	2 horas	14:00	14:10	Ninguno	Finalizado
	Realizar validaciones.	2 horas	14:10	14:20	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	14:20	14:35	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	14:35	14:40	Ninguno	Finalizado
28	Crear métodos funcionales.	1 hora	14:40	14:45	Ninguno	Finalizado
	Crear interfaz gráfica.	1 hora	14:45	14:55	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	14:55	15:00	Ninguno	Finalizado
29	Crear métodos funcionales.	2 horas	15:05	15:15	Ninguno	Finalizado
	Realizar validaciones.	2 horas	15:15	15:25	Ninguno	Finalizado
	Crear interfaz gráfica.	2 horas	15:25	15:40	Ninguno	Finalizado
	Pruebas unitarias.	1 hora	15:40	15:45	Ninguno	Finalizado

Grafica - Tiempo estimado vs real

Los valores están dados en horas.





PROYECTO:	Gestión de proyectos			Página 1 de 6
DOCUMENTO:	Estimación de esfuerzo			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Estimación de esfuerzo

Iteración 3



PROYECTO:	Gestión de proyectos			Página 2 de 6
DOCUMENTO:	Estimación de esfuerzo			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Historias de usuario – Estimación de esfuerzo	3
Tareas de ingeniería – Detalle esfuerzo	4
Grafica – Tiempo estimado vs real	6

Historias de usuario – Estimación de esfuerzo

Historia de usuario	Tiempo Estimado	Tiempo real
30 – Registrar cargos.	3,16 horas	1,6 horas
31 – Listas Cargos.	1, 5 horas	0,7 horas
32 – Consultar/Modificar cargo.	2,6 horas	2,17 horas
33 – Registrar tiempo trabajado en actividad.	4,5 horas	7,17 horas
34 – Consultar tiempo trabajado en actividad.	2 horas	2 horas
35 – Modificar tiempo trabajado en actividad.	4 horas	3 horas
36 – Guardar archivo de proyecto.	3,16 horas	2,5 horas
37 – Listar archivos de proyecto.	0,3 horas	0,17 horas
38 – Eliminar archivo de proyecto.	2,5 horas	2,5 horas

Fecha estimada para entrega: 31 de Mayo de 2016

Fecha real de la entrega: 31 de Mayo de 2016

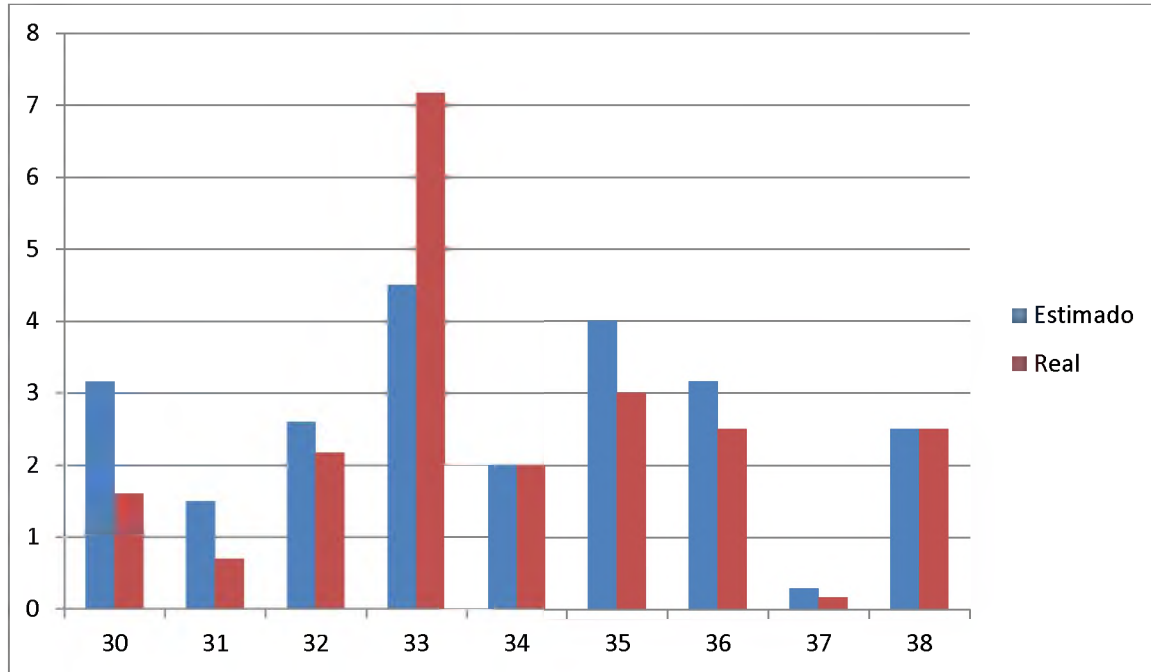
Tareas de ingeniería - Detalle esfuerzo

N°	Tarea de ingeniería	Tiempo Estimado	Hora Inicio	Hora fin	Errores	Estado
30	30.1 Realizar métodos funcionales.	1 hora	8:30	9:00	Ninguno.	Terminado.
	30.2 Realizar validaciones	40 minutos	9:00	9:20	Ninguno.	Terminado.
	30.3 Crear interfaz gráfica.	1 hora	9:25	10:00	Ninguno.	Terminado.
	30.4 Pruebas unitarias.	30 minutos	10:00	10:15	Ninguno.	Terminado.
31	31.1 Realizar métodos funcionales.	30 minutos	10:30	10:40	Ninguno.	Terminado.
	31.2 Crear interfaz gráfica.	40 minutos	10:40	11:00	Ninguno.	Terminado.
	31.3 Pruebas unitarias.	20 minutos	11:10	11:20	Ninguno.	Terminado.
32	32.1 Realizar métodos funcionales.	40 minutos	11:30	12:00	Ninguno.	Terminado.
	32.2 Realizar validaciones.	40 minutos	14:00	14:30	Ninguno.	Terminado.
	32.3 Crear interfaz gráfica.	1 hora	14:30	15:20	Ninguno.	Terminado.
	32.4 Pruebas unitarias.	20 minutos	15:20	15:40	Ninguno.	Terminado.
33	33.1 Realizar métodos funcionales.	2 horas	15:40	19:30	Ninguno.	Terminado.
	33.2 Realizar validaciones.	1 hora	8:00	9:30	Ninguno.	Terminado.
	33.3 Crear interfaz gráfica.	1 hora	9:30	10:20	Ninguno.	Terminado.
	33.4 Pruebas unitarias.	30 minutos	10:20	11:00	Ninguno.	Terminado.
34	34.1 Realizar métodos funcionales.	30 minutos	11:00	11:20	Ninguno.	Terminado.
	34.2 Crear interfaz gráfica.	1 hora	11:20	12:30	Ninguno.	Terminado.
	34.3 Pruebas unitarias.	30 minutos	12:30	13:00	Ninguno.	Terminado.
35	35.1 Realizar métodos funcionales.	1 hora	14:00	15:00	Ninguno.	Terminado.
	35.2 Realizar validaciones.	1 hora	15:00	15:40	Ninguno.	Terminado.

	35.3 Crear interfaz gráfica.	1 hora	15:40	16:30	Ninguno.	Terminado.
	35.4 Pruebas unitarias.	1 hora	16:30	17:00	Ninguno.	Terminado.
36	36.1 Realizar métodos funcionales.	1 hora	17:30	18:10	Ninguno.	Terminado.
	36.2 Realizar validaciones.	1 hora	18:10	19:00	Ninguno.	Terminado.
	36.3 Crear interfaz gráfica.	40 minutos	19:00	19:40	Ninguno.	Terminado.
	36.4 Pruebas unitarias.	30 minutos	19:40	20:00	Ninguno.	Terminado.
37	37.3 Crear interfaz gráfica.	20 minutos	20:20	20:30	Ninguno.	Terminado.
38	38.1 Realizar métodos funcionales.	1 hora	21:00	21:30	Ninguno.	Terminado.
	38.2 Realizar validaciones.	1 hora	21:30	22:10	Ninguno.	Terminado.
	38.3 Pruebas unitarias.	30 minutos	22:20	23:40	No se elimina correctamente el archivo del disco, sigue apareciendo en listar archivos después de eliminar. Problema con cache solucionado.	Terminado.

Grafica - Tiempo estimado vs real

Los valores están dados en horas.





PROYECTO:	Gestión de proyectos			Página 1 de 6
DOCUMENTO:	Estimación de esfuerzo			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Estimación de esfuerzo

Iteración 4



PROYECTO:	Gestión de proyectos			Página 2 de 6
DOCUMENTO:	Estimación de esfuerzo			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Historias de usuario – Estimación de esfuerzo	3
Tareas de ingeniería – Detalle esfuerzo	4
Grafica – Tiempo estimado vs real	6

Historias de usuario – Estimación de esfuerzo

Historia de usuario	Tiempo Estimado	Tiempo real
39 - Crear ODT.	35 minutos	30 minutos
40 - Consultar/Modificar ODT.	2 horas	1 hora 20 minutos
41 - Exportar ODT	1 hora 50 minutos	1 hora 45 minutos
42 - Exportar cotización	5 horas 30 minutos	7 horas 30 minutos
43 - Modificar usuarios.	2 horas	1 hora 35 minutos
44 - Configurar días no laborables.	2 horas	1 hora 30 minutos
45 - Configurar jornada laboral.	2 horas 20 minutos	1 hora 40 minutos
46 - Registro de tiempo trabajado	6 horas 40 minutos	27 horas 40 minutos
47 - Registro de materiales usados	4 horas 30 minutos	18 horas 10 minutos

Fecha estimada para entrega: Septiembre 13 de 2016

Fecha real de la entrega: Septiembre 18 de 2016

Tareas de ingeniería - Detalle esfuerzo

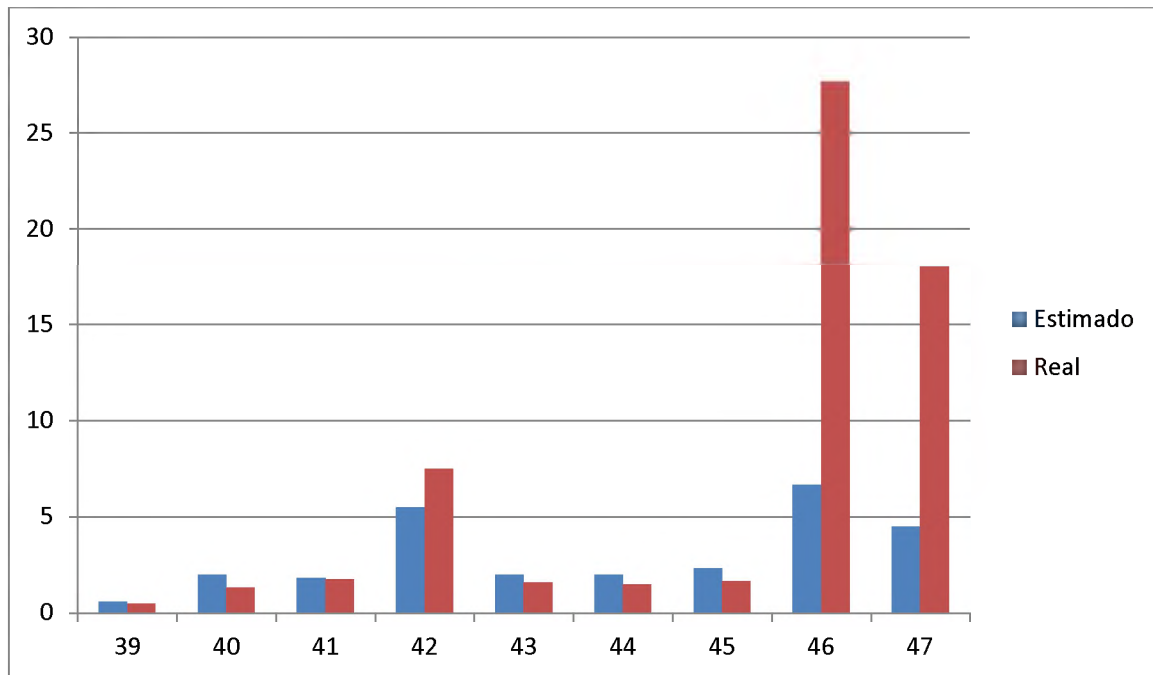
N°	Tarea de ingeniería	Tiempo Estimado	Hora Inicio	Hora fin	Errores	Estado
39	Realizar métodos funcionales.	20 minutos	8:00	8:20	Ninguno	Terminado
	Pruebas unitarias.	15 minutos	8:50	9:00	Ninguno	Terminado
40	Realizar métodos funcionales.	20 minutos	9:00	9:15	Ninguno	Terminado
	Realizar Validaciones.	20 minutos	9:15	9:30	Ninguno	Terminado
	Crear interfaz gráfica.	1 hora	9:30	10:00	Ninguno	Terminado
	Pruebas unitarias.	20 minutos	10:00	10:20	Ninguno	Terminado
41	Realizar métodos funcionales.	30 minutos	10:30	11:00	Ninguno	Terminado
	Realizar reporte.	1 hora	11:00	11:30	Ninguno	Terminado
	Pruebas unitarias.	20 minutos	11:30	11:45	Ninguno	Terminado
42	Realizar métodos funcionales.	4 horas	14:00	20:00	Problemas con librerías. Problemas con resultados al aplicar fórmulas de Excel.	Terminado
	Realizar reporte.	1 hora	8:00	9:00	Ninguno	Terminado
	Pruebas unitarias.	30 minutos	9:00	9:30	Ninguno	Terminado
43	Realizar métodos funcionales.	40 minutos	10:00	10:30	Ninguno	Terminado
	Realizar interfaz grafica	1 hora	10:40	12:00	Ninguno	Terminado
	Pruebas unitarias.	20 minutos	12:00	12:15	Ninguno	Terminado
44	Realizar métodos funcionales.	40 minutos	14:00	14:30	Ninguno	Terminado
	Realizar validaciones.	30 minutos	15:00	15:20	Ninguno	Terminado
	Crear interfaz gráfica.	30 minutos	15:20	15:40	Ninguno	Terminado
	Pruebas unitarias.	20 minutos	15:40	16:00	Ninguno	Terminado
45	Realizar métodos funcionales	1 hora	16:30	17:10	Ninguno	Terminado
	Crear interfaz grafica	1 hora	17:30	18:30	Ninguno	Terminado
	Pruebas unitarias.	20 minutos	18:30	19:00	Ninguno	Terminado
46	Realizar métodos funcionales.	4 horas	8:00	12:00	Ninguno	Terminado
			13:00	17:00	Ninguno	Terminado

47			18:00	22:00	Problemas en cálculo de horas laboradas.	Terminado
			8:00	12:00	Problema con horas extras laboradas según horario definido para el día.	Terminado
			14:00	18:00	Problema con actualización de horas al momento de modificar o crear.	Terminado
			19:00	23:00	Ninguno	Terminado
	Realizar validaciones.	1 hora	8:00	10:00	Problemas validando intervalos registrados.	Terminado
	Crear interfaz gráfica.	1 hora	11:00	11:40	Ninguno	Terminado
	Pruebas unitarias.	40 minutos	12:00	13:00	Ninguno	Terminado
	Realizar métodos funcionales.	2 horas	14:00	18:00	Problemas en validación de existencia en inventario según tipo de material	Terminado
			19:00	23:00	Ninguno	Terminado
			8:00	12:00	Problema descontando material al modificar.	Terminado
			14:00	18:00	Ninguno	Terminado
	Realizar validaciones.	1 hora	19:00	20:00	Ninguno	Terminado
	Crear interfaz gráfica.	1 hora	20:00	20:40	Ninguno	Terminado

	Pruebas unitarias.	30 minutos	21:00	21:30	Ninguno	Terminado

Grafica - Tiempo estimado vs real

Los valores están dados en horas.



ANEXO 4



PROYECTO:	Gestión de proyectos			Página 1 de 7
DOCUMENTO:	Estándar de codificación			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Estándar de codificación

Este documento se basa en gran parte por los estándares de codificación del lenguaje Java presentados en Java Language Specification, de Sun Microsystems, Inc.



PROYECTO:	Gestión de proyectos			Página 2 de 7
DOCUMENTO:	Estándar de codificación			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Nombre de paquetes:	3
Nombre de clases:	3
Session beans:	3
Managed beans:	3
Archivos xhtml:	4
Ubicación de librerías:	4
Comentarios de comienzo de clase:	4
Sentencias package e importar:	4
Nombre de métodos:	5
Declaración de variables:	5
Declaracion de constantes:	5
Comentarios	6
Bloque:	6
De línea:	6
De remolque:	6
Indentación	7
Longitud de línea:	7
Romper línea:	7

Nombre de paquetes:

Por defecto todos los paquetes se escribirán en minúscula sin ningún tipo de carácter especial, cada nuevo paquete se definirá dentro de la carpeta **com.myr.<nombre paquete>**

Nombre de clases:

El idioma por defecto en la declaración de cada clase es español, siendo sustantivos y complementando con el nombre del paquete al que pertenece ejemplo:
ListarUsuario.java.

Los nombres de clases iniciaran con mayúscula independientemente de la cantidad de palabras que posea, cada palabra adicional iniciara con mayúscula. No se permiten caracteres como “-” ó “_” en el nombre de la clase.

No se permiten abreviaturas, obviando casos en los que se usan patrones como: DAO, POJO, VO. Estos últimos llevaran mayúscula sostenida.

La llave de apertura “{” aparecerá en la misma línea de declaración separada con un espacio.

Session beans:

El nombre de los session beans 3estarán dados de la siguiente manera:

- Paquete de lógica: El nombre inicia con la entidad a la que se hace referencia, seguido de la palabra “Logica”. Ejemplo **EmpleadoLogica.java**.
Si se trata de una interfaz local el nombre debe terminar con la palabra “Local”. Ejemplo **EmpleadoLogicaLocal.java**. Si se trata de una interfaz remota el nombre debe terminar con la palabra “Remoto”. Ejemplo **EmpleadoLogicaRemoto.java**
- Paquete de Persistencia: El nombre inicia con la entidad a la que se hace referencia, seguido de la palabra “Facade”. Ejemplo **EmpleadoFacade.java**.
Si se trata de una interfaz local el nombre debe terminar con la palabra “Local”. Ejemplo **EmpleadoFacadeLocal**. Si se trata de una interfaz remota el nombre debe terminar con la palabra “Remoto”. Ejemplo **EmpleadoFacadeRemoto.java**

Managed beans:

Los nombres de los ManagedBeans estarán dados según la entidad o funcionalidad a la que se refieran, iniciando con el nombre de la entidad y terminando con la palabra “Vista”. Ejemplo **EmpleadoVista.java**

Archivos xhtml:

Los archivos XHTML se ubicaran dentro de la carpeta **web\jsf** donde se debe crear una carpeta por entidad o funcionalidad, por ejemplo para la entidad empleado se crea de la siguiente manera: **web/jsf/empleado**. Iniciando con minúscula y cada palabra compuesta inicia en mayúscula.

Los archivos XHTML se nombran según la entidad a la que hacen referencia, iniciando un verbo seguido de la entidad a la que hace referencia, el nombre inicia en minúscula y cada palabra compuesta inicia en mayúscula. Ejemplo: **listarEmpleado.xhtml**

Ubicación de librerías:

Las librerías .jar se copiaran dentro del proyecto, en la carpeta: **\web\WEB-INF\lib**. Se deben eliminar las librerías que no sean usadas.

Comentarios de comienzo de clase:

Todos los ficheros deben comenzar con un bloque comentado donde se especificara: funcionalidad, autor, fecha de creación.

```
/*
 * Funcionalidad de clase
 *
 * @author (nombre autor)
 * @creationdate (dia/mes/año)
 * @modificationdate(dia/mes/año)
 *
 */
```

Sentencias package e importar:

Seguido a los comentarios, se encontrará la sentencia **package** seguido de las sentencias **import**, separadas por una línea en blanco.

```
package com.myr.controller;
```

```
import java.io.Serializable;
```

Nombre de métodos:

El nombre iniciara en minúscula y con un verbo en infinitivo describiendo la funcionalidad del método, cada palabra adicional iniciara con mayúscula, evitando abreviaturas sin importar que tan extensa sea la palabra.

Los paréntesis “()” aparecerán seguidos al nombre, separados por un espacio.

La llave de apertura aparecerá en la misma línea de declaración separada con un espacio en blanco.

Declaración de variables:

Se debe declarar una variable por línea, intentando inicializar cada variable donde se declara, a excepción de las variables que dependen de una operación futura.

Las declaraciones se deben ubicar al principio de cada bloque (un bloque es cualquier código encerrado por llaves {}).

```
void myMethod() {  
    int var1 = 0;  
  
    if (condition) {  
        int var2 = 0;  
    }  
}
```

Declaracion de constantes:

Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un subguión (“_”).

Ejemplo:

```
static final int ANCHURA_MINIMA = 4;
```


Comentarios

Bloque:

Son usados para descripciones de ficheros, métodos, estructuras de datos y algoritmos.

Un comentario de bloque debe ir precedido por una línea en blanco que lo separe del resto del código.

```
/*  
* comentario  
*/
```

De línea:

Un comentario de línea puede aparecer a nivel de código, dando una pequeña descripción de la línea que continúa. Si el comentario es extenso debe cumplir el formado de un comentario de bloque.

Debe cumplir con la tabulación del bloque en que esta contenido.

Debe ir precedido de una línea en blanco separándolo del resto del código.

```
    if (condicion) {  
        /* Código de la condicion. */  
    }
```

De remolque:

Pueden aparecer comentarios en la misma línea de código, especificando de manera clara y corta lo que se quiere aclarar, pero deben ser tabulados lo suficientemente lejos del código. Si en el mismo bloque aparecen mas de un comentario, estos deben ser tabulados a la misma profundidad.

```
    if (a == 2) {  
        return TRUE;                /* caso especial */  
    } else {  
        return isPrime(a);          /* caso gerenal */  
    }
```

Identación

Longitud de línea:

En el código fuente se recomienda evitar líneas de más de 80 caracteres, de ser mayor se debe romper la línea.

Romper línea:

Se recomienda romper la línea según los siguientes puntos:

- Romper después de una coma.
- Romper antes de un operador.
- Roturas de alto nivel, tabulando la línea más a la derecha del “padre”.

Ejemplo:

```
unMetodo (expresionLarga1, expresionLarga2, expresionLarga3,  
          expresionLarga4, expresionLarga5);
```

```
operaciones = operacion1 * operacion2 + operacion3  
              +operacion4 + operacion6;
```

ANEXO 5



PROYECTO:	Gestión de proyectos			Página 1 de 5
DOCUMENTO:	Estándar de Base de datos			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Estándar de base de datos

Gestión de proyectos



PROYECTO:	Gestión de proyectos			Página 2 de 5
DOCUMENTO:	Estándar de Base de datos			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Tablas.....	3
Campos	3
Llaves primaria PK:.....	3
Otros campos:.....	4
Llaves foráneas:	4
Consultas:	4
Procedimientos almacenados	5
Triggers o disparadores	5

El siguiente documento describe un estándar que se utilizara durante todo el desarrollo y posteriores mantenimientos sobre el aplicativo de gestión de proyectos en la empresa “Proyectos-mantenimiento industrial M&R”.

Es creado con el propósito de asegurar la legibilidad del modelo de datos por todos los interesados en el proyecto, sin importar que no estén relacionadas con el ámbito de los sistemas. Además, de facilitar la tarea de desarrolladores en futuros mantenimientos del software.

Tablas

Los nombres deben especificarse en singular, iniciando su primera palabra en minúscula, si la palabra es compuesta se debe iniciar con mayúscula cada nueva palabra.

Los nombres de las tablas solo utilizaran caracteres alfabéticos, prohibiendo el uso de caracteres acentuados o especiales.

El nombre deben ser corto y lo más descriptivo posible, evitando abreviaciones.

Las tablas intermedias que representan relaciones N a M. deben nombrarse compuestamente según los nombres de las tablas intervinientes, Ejemplo: inventarioCotizacion.

Campos

Los campos al igual que las tablas, llevaran nombres representativos y lo más cortos posibles, iniciando el nombre en minúscula y para cada palabra que lo compone se debe iniciar con mayúscula.

Llaves primaria PK:

Toda llave primara se debe definir al inicio de cada tabla.

Toda tabla debe contener una llave primaria, la cual será de tipo incremental. Este campo iniciara con la palabra **id** seguido del nombre de la tabla sin abreviaciones. Ejemplo: idEmpleado.

Las tablas intermedias que representan relaciones N a M, llevaran un campo de identificación el cual se nombrara de la siguiente manera: iniciando con **id**, seguido

de las tres primeras letras del nombre de la siguiente tabla, incluyendo todas las tablas que intervienen. Ejemplo idActInvCot.

Otros campos:

Los campos en cada tabla se definen seguido a la llave primaria, los nombres deben representarse en singular e iniciar con minúscula, si es una palabra compuesta se debe continuar con la segunda palabra en mayúscula y así para cada palabra que componga el nombre.

Los campos se nombrarán según el nombre de la tabla, ejemplo:
nombreEmpleado, apellidoEmpleado.

Llaves foráneas:

Las llaves foráneas deben nombrarse de la misma manera que el campo del cual son referenciadas. Ejemplo: tabla cargo -> idCargo

Consultas:

Las consultas SQL se deben organizar según la sentencia ejecutada, un operador en cada línea. Cada operador debe ser escrito en mayúscula sostenida.

Para facilidad de lectura y acortar las sentencias, se debe renombrar las tablas de manera abreviada como se muestra en el ejemplo propuesto.

Ejemplo:

```
SELECT emp.nombre  
FROM empleado emp  
WHERE emp.idEmpleado = 5
```

Procedimientos almacenados

Los procedimientos se deben realizar en el lenguaje **PL/pgSQL**, en el cual todas las sentencias propias del lenguaje deben ser escritas con mayúscula sostenida, conservando el siguiente formato (Todo entre los corchetes [] es opcional):

```
CREATE [ OR REPLACE ] FUNCTION
proc_nombre_funcion([ [ argmodo ] [ argnombre ] argtipo [, ...] ])
RETURNS tipo AS $$

[ DECLARE ]
[ declaraciones de variables ]

BEGIN
    //codigo

END;
$$ LANGUAGE plpgsql
```

El nombre de cada procedimiento debe iniciar con **proc_<nombre función>**, iniciando con la primera letra en minúscula y para cada palabra compuesta iniciar en mayúscula, ejemplo:
Proc_verificarEmpleado().

Si la función será usada en un Trigger, debe retornar un tipo TRIGGER ó NULL.

Triggers o disparadores

Los Triggers o disparadores deben iniciar su nombre con **trg_<nombre trigger>** iniciando el nombre con minúscula y para cada palabra compuesta iniciar en mayúscula, ejemplo:
tgr_ingresarHorasTrabajo().

Los triggers deben hacer uso de una función o procedimiento almacenado.

La definición debe cumplir con la siguiente estructura:

```
CREATE TRIGGER tgr_nombre { BEFORE | AFTER } { INSERT | UPDATE | DELETE [ OR ... ] }
ON tabla [ FOR [ EACH ] { ROW | STATEMENT } ]
EXECUTE PROCEDURE nombre de funcion ( argumentos )
```


ANEXO 6



PROYECTO:	Gestión de proyectos			Página 1 de 4
DOCUMENTO:	Estándar de mensajes			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Estándar de mensajes



PROYECTO:	Gestión de proyectos			Página 2 de 4
DOCUMENTO:	Estándar de mensajes			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Declaración de mensajes de error:.....	3
Mensajes de confirmación:	3
Mensajes de información:	3
Mensajes de error:	4

Declaración de mensajes de error:

Los mensajes estarán definidos en un archivo .properties ubicado en la carpeta src/java, por lo cual deben ser obtenidos y modificados para cada caso. Esto con el fin de garantizar el uso de la misma estructura para cada caso en particular.

Mensajes de confirmación:

Se mostraran mensajes tipo modal con opciones de aceptar y cancelar, en los cuales se pregunte al usuario si realmente desea continuar con una determinada acción. Según la acción seleccionada el mensaje tendrá una estructura diferente, de la siguiente manera:

- **Eliminar**: Si el usuario selecciona eliminar algún registro de la aplicación, el mensaje tendrá la siguiente estructura “¿Está seguro que desea eliminar permanentemente el registro: <ID descriptivo> : <Nombre descriptivo>?”.
- **Salir sin guardar**: Si el usuario selecciona cambiar el flujo de la aplicación mientras crea o modifica un registro, el mensaje tendrá la siguiente estructura: “Los cambios se perderán si continúas con esta acción. ¿Deseas continuar?”.
- **Cambiar el estado de un proyecto**: Si el usuario selecciona modificar un proyecto cambiando su estado original, el mensaje tendrá la siguiente estructura: “¿Está seguro que desea modificar el estado a: <Estado nuevo> para el proyecto: <Nombre de proyecto>?”.

Mensajes de información:

Se mostraran mensajes de información en acciones exitosas realizadas por el usuario. Según la acción realizada el mensaje tendrá una estructura diferente, de la siguiente manera:

- **Crear un registro**: Si el usuario crea un registro en el sistema, el mensaje tendrá la siguiente estructura: “El <tipo de objeto> se registró exitosamente.”
Ejemplo: “El empleado se registró exitosamente.”.
- **Modificar un registro**: Si el usuario modifica un registro en el sistema, el mensaje tendrá la siguiente estructura: “El <tipo de objeto> se <acción> exitosamente.”
Ejemplo: “El empleado se modificó exitosamente.”.

Mensajes de error:

Se mostraran mensajes de error para diferentes problemas encontrados en la ejecución de la aplicación, de la siguiente manera:

- **Campos obligatorios:** Si el usuario no diligencia un campo obligatorio en un formulario, se mostrara un error con la siguiente estructura: “El campo <Nombre de campo> es requerido.”.
- **Tipo de datos:** Si el usuario diligencia un campo con un tipo de dato erróneo, se mostrara un mensaje con la siguiente estructura: “El campo <nombre de campo> espera un: <tipo de dato>”.
- **Formato de datos:** Si el usuario diligencia un campo con un formato erróneo, se mostrara un mensaje con la siguiente estructura: “El campo <nombre de campo> espera un valor con formato: <formato> ”.
- **Error en consultas con base de datos:** Si ocurre un error en una transacción con base de datos, se debe mostrar un error con la siguiente estructura: “Ocurrió un error con la base de datos mientras se <acción> ”.

ANEXO 7



PROYECTO:	Gestión de proyectos			Página 1 de 13
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tareas de ingeniería

Gestión de proyectos

Tabla de contenido

Historia de usuario 4: Crear proveedor.....	3
Tarea de ingeniería 4.1:.....	3
Tarea de ingeniería 4.2:.....	3
Tarea de ingeniería 4.3:.....	4
Tarea de Ingeniería 4.4:.....	4
Historia de usuario 5: Listar proveedores	5
Tarea de ingeniería 5.1:.....	5
Tarea de ingeniería 5.2:.....	5
Tarea de ingeniería 5.3:.....	6
Historia de usuario 6: Consultar/Modificar proveedor	6
Tarea de Ingeniería 6.1:.....	6
Tarea de Ingeniería 6.2:.....	6
Tarea de Ingeniería 6.3:.....	7
Historia de usuario 10: Crear empleado	8
Tarea de ingeniería 10.1:.....	8
Tarea de ingeniería 10.2:.....	8
Tarea de ingeniería 10.3:.....	9
Tarea de ingeniería 10.4:.....	10
Historia de usuario 11: Listar empleados.....	10
Tarea de ingeniería 11.1:.....	10
Tarea de ingeniería 11.2:.....	10
Tarea de ingeniería 11.3:.....	11
Historia de usuario 12: Consultar/Modificar empleado	11
Tarea de ingeniería 12.1:.....	11
Tarea de ingeniería 12.2:.....	12
Tarea de ingeniería 12.3:.....	12

Historia de usuario 4: Crear proveedor

Tarea de ingeniería 4.1:

Tarea de Ingeniería	
Número de tarea: 4.1	Historia de usuario: 4
Nombre tarea: Realizar método para funcionalidad	
Tipo de tarea: Desarrollo	Puntos Estimados: 4 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método que reciba por parámetro un objeto de tipo empleado, con el cual se realizarán validaciones previas como: verificar que los datos obligatorios se encuentren diligenciados, si se encuentra un error se mostrará un mensaje en la parte superior del formulario especificado en la tarea de ingeniería 4.3, detallando que campo contiene error, si por el contrario no encuentra ningún error procede a guardar el objeto en la capa de datos. Si guarda exitosamente redirecciona la aplicación a la interfaz principal con un mensaje de aprobación, si por el contrario ocurre algún error se mostrará un mensaje al usuario notificando el error.	

Tarea de ingeniería 4.2:

Tarea de Ingeniería	
Número de tarea: 4.2	Historia de usuario: 4
Nombre tarea: Realizar validaciones de datos	
Tipo de tarea: Desarrollo	Puntos Estimados: 4 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método el cual, al ser presionada la opción de Guardar verificará que los datos ingresados en el formulario detallados en la tarea de ingeniería 4.3 sean correctos. Teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los tipos de datos son los correctos y tienen el formato correcto. Si se encuentra un dato erróneo se debe mostrar en pantalla el mensaje de error especificando que campo está erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente para crear el proveedor.	

Tarea de ingeniería 4.3:

Tarea de Ingeniería	
Número de tarea: 4.3	Historia de usuario: 4
Nombre tarea: Crear interfaz gráfica	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una interfaz gráfica que contenga los campos necesarios para diligenciar la información básica de un proveedor, de esta manera: Nombre: Campo de texto alfanumérico, no se permite ser nulo Teléfono: Campo de texto alfanumérico, no se permite ser nulo, no se permiten caracteres especiales a excepción de (,)+ que son necesarios para ingresar indicativos. Dirección: Campo de texto alfanumérico. Ciudad: Campo de texto alfanumérico. Correo: Campo de texto alfanumérico con un formato en específico para correos electrónicos. Descripción: Área de texto alfanumérica. Además, en la parte inferior alineados a la derecha del usuario se deben crear dos botones: ○ Volver que al ser presionado debe mostrar un mensaje de confirmación como el siguiente: "Los cambios se perderán si continúas con esta acción. ¿Deseas continuar?", si selecciona Aceptar se redirecciona a la ventana principal, sin guardar ningún cambio. Si presiona cancelar permanece en la interfaz actual. ○ Guardar Su función es guardar en la capa de datos la información diligenciada por el usuario. Al finalizar se redirecciona a la ventana principal en la cual se mostrara un mensaje en la parte superior.	

Tarea de Ingeniería 4.4:

Tarea de Ingeniería	
Número de tarea: 4.4	Historia de usuario: 4
Nombre tarea: Realizar pruebas unitarias	
Tipo de tarea: Pruebas funcionales	Puntos Estimados: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de crear proveedor, para lo cual se deben diligenciar los formularios con datos validos e inválidos verificando que los métodos de validación se encuentran correctamente, además verificar que los métodos guardar exitosamente en la capa de datos, como también que muestran los mensajes de error o aprobación de manera correcta.	

Historia de usuario 5: Listar proveedores

Tarea de ingeniería 5.1:

Tarea de Ingeniería	
Número de tarea: 5.1	Historia de usuario: 5
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben crear métodos funcionales que consulten en la capa de datos la información correspondiente a todos los proveedores que se encuentran en el sistema, retornando una lista de objetos Proveedor que se encontrara de manera ordenada según el nombre de cada proveedor como se especifica en la tarea de ingeniería 5.1 Esta lista estará disponible para ser usada en la interfaz gráfica.	

Tarea de ingeniería 5.2:

Tarea de Ingeniería	
Número de tarea: 5.2	Historia de usuario: 5
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una interfaz la cual, por medio de componentes de primefaces muestre los proveedores existentes en el sistema, se debe mostrar en manera de lista la siguiente información: Nombre, Teléfono, dirección, Ciudad, Correo, Descripción de proveedor, consultar, modificar. Inicialmente la información estará ordenada de manera alfabética según el nombre, pero se podrá reorganizar según el nombre, ciudad o correo, como tambien ser filtrada teniendo en cuenta todos los campos anteriormente mencionados. En la parte inferior alineado a la derecha se mostrará un botón para volver a la ventana principal.	

Tarea de ingeniería 5.3:

Tarea de Ingeniería	
Número de tarea: 5.3	Historia de usuario: 5
Nombre tarea: Realizar pruebas	
Tipo de tarea: Pruebas funcionales	Puntos Estimados: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben de realizar pruebas sobre la funcionalidad de listar proveedor, para lo cual se tendrá en cuenta las especificaciones de la tarea 5.1. Por lo que se debe verificar: • Lista la totalidad de los proveedores. • Muestra los datos correctos para cada proveedor. • Reorganiza según cada columna. • Al buscar una palabra, filtra teniendo en cuenta todos los campos.	

Historia de usuario 6: Consultar/Modificar proveedor

Tarea de Ingeniería 6.1:

Tarea de Ingeniería	
Número de tarea: 6.1	Historia de usuario: 6
Nombre tarea: Crear métodos para funcionalidades	
Tipo de tarea:	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica que reciba como parámetro el registro modificado, verificando que los campos obligatorios sean válidos y posteriormente proceder a guardar los cambios. Si ocurre algún error en el proceso, se debe retornar una excepción la cual será mostrada al usuario, de lo contrario continua el flujo normal de la aplicación.	

Tarea de Ingeniería 6.2:

Tarea de Ingeniería	
Número de tarea: 6.2	Historia de usuario: 6
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una interfaz gráfica que cuente con los campos necesarios para mostrar al usuario	

los datos correspondientes a un proveedor, como: Nombre, teléfono, dirección, ciudad, correo, descripción, los cuales se deben mostrar en campos de texto.

Además, se mostrará una tabla en la parte inferior del formulario inicial donde se especifican los productos que han sido comprados a dicho proveedor.

Si el usuario selecciono **Consultar** en la historia de usuario 5, se mostrará toda la información con los campos de texto deshabilitados.

Si por el contrario, selecciona **Modificar** se mostraran todos los campos de manera editables, además se renderiza un botón en la parte inferior del formulario con la palabra **Guardar cambios**.

En la parte inferior alineado a la derecha se mostrara un botón **Volver** el cual redirecciona a Listar proveedor.

Tarea de Ingeniería 6.3:

Tarea de Ingeniería	
Número de tarea: 6.3	Historia de usuario: 6
Nombre tarea: Realizar pruebas unitarias	
Tipo de tarea: Pruebas funcionales	Puntos Estimados: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas funcionales verificando que se cumple con las especificaciones dadas inicialmente por el usuario. Por lo que se debe tener en cuenta lo siguiente: - Los datos se muestran correctamente en la interfaz de consultar/Modificar siendo los respectivos para cada proveedor seleccionado. - Si selecciona consultar, los campos deben permanecer inactivos. - Si selecciona modificar, los campos estarán activos para modificaciones y se renderiza el botón de <i>Guardar cambios</i>. - Al <i>Guardar cambios</i>, los datos se guardan correctamente. - Se muestran correctamente los mensajes según sea el caso (Exitoso o fallido).	

Historia de usuario 10: Crear empleado

Tarea de ingeniería 10.1:

Tarea de Ingeniería	
Número de tarea: 10.1	Historia de usuario: 10
Nombre tarea: Crear métodos funcionales	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe realizar métodos funcionales que lleven a cabo las funciones para crear empleado. Por lo que se debe crear un método en el paquete de lógica, que reciba como parámetro un objeto de tipo empleado definido en la capa de modelo. Se debe verificar que los datos obligatorios estén diligenciados (Número de identificación, nombre, apellido, cargo y fecha de contratación). Si los datos se encuentran correctos, se debe guardar el empleado en la base de datos, retornando un mensaje a mostrar en la interfaz gráfica en la que informe al usuario si el registro se guardó exitosamente y redireccionar a la ventana principal mostrando un mensaje en la parte superior de la interfaz, o si ocurre un error se debe mostrar un mensaje de error en la parte superior del formulario el que se especifique cual fue el error.	

Tarea de ingeniería 10.2:

Tarea de Ingeniería	
Número de tarea: 10.2	Historia de usuario: 10
Nombre tarea: Realizar validaciones de datos	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Luego de que el usuario diligencie el formulario para crear un empleado, se debe verificar que los datos sean correctos validando los tipos de datos ingresados. Por lo que se debe crear una clase en el paquete de vista con un método que realice dicha función y si encuentra que los datos no cumplen con la estructura debe mostrar un mensaje de error en el que especifique que campo debe ser corregido; en caso contrario, si los datos son correctos ejecuta la función correspondiente a guardar los datos en la capa de lógica (tarea de ingeniería 10.3).	

Tarea de ingeniería 10.3:

Tarea de Ingeniería	
Número de tarea: 10.3	Historia de usuario: 10
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Tiempo estimado: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se diseñara una interfaz compuesta por varios componentes suficientes para diligenciar la información necesaria de un empleado, de la siguiente manera: Número de identificación: Campo de texto, solo acepta datos numéricos, sin puntos ni comas, no puede ser nulo. Nombre: Campo de texto, no puede ser nulo ni contener caracteres numéricos. Apellido: Campo de texto, no puede ser nulo ni contener caracteres numéricos. Teléfono: Campo alfanumérico, permite el ingreso de caracteres como (,)+, si es necesario un indicativo. Cargo: Lista desplegable, mostrará todos los cargos que han sido creados en el sistema. Fecha contratación: Se mostrará un campo de texto que al ser presionado desplegará un calendario con el cual se podrá seleccionar la fecha deseada. El formato será dd-MM-AAAA. Fecha de terminación del contrato: El campo cumplirá con las mismas condiciones que la fecha de contratación, además, al ser diligenciada se mostrara un mensaje de confirmación informando que se le asignara un estado de inactividad al empleado, si confirma se asignara la fecha al campo de texto, de lo contrario conserva el campo vacío. Además, contará con botones de acción, los cuales deben aparecer bajo el formulario, alineados a la derecha del usuario, los botones cumplirán las siguientes condiciones: Cancelar: Al presionar este botón, se mostrará un mensaje de confirmación el cual mostrará el siguiente mensaje: "Los cambios se perderán si continúas con esta acción. ¿Deseas continuar?", si selecciona Aceptar se redirecciona a la ventana principal, sin guardar ningún cambio. Aceptar: Al presionar este botón, se deben bloquear todos los controles de la interfaz, mostrando un mensaje de espera mientras el sistema crea el empleado. Al terminar de guardar, se redirecciona a la ventana principal, mostrando un mensaje de confirmación en la parte superior.	

Tarea de ingeniería 10.4:

Tarea de Ingeniería	
Número de tarea: 10.4	Historia de usuario: 10
Nombre tarea: Realizar pruebas unitarias	
Tipo de tarea: Pruebas funcionales	Puntos Estimados: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar las pruebas funcionales, verificando que se cumplen las especificaciones dadas en la historia de usuario 8. Ingresando diferentes tipos de datos validando que los datos son manipulados de la manera correcta, es decir, validaciones y conversiones necesarias para cumplir dicha función. Si se encuentra un error se debe corregir inmediatamente y documentar que tipo de error se encuentra.	

Historia de usuario 11: Listar empleados

Tarea de ingeniería 11.1:

Tarea de Ingeniería	
Número de tarea: 11.1	Historia de usuario: 11
Nombre tarea: Crear métodos funcionales	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método que consulte todos los empleados que se encuentran activos en el sistema. Asignándolos a un atributo de la clase correspondiente a listar empleado, para posteriormente ser cargados en la interfaz gráfica haciendo uso de componentes de primefaces.	

Tarea de ingeniería 11.2:

Tarea de Ingeniería	
Número de tarea: 11.2	Historia de usuario: 11
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se desarrollará una interfaz gráfica en la cual por medio de una tabla de primefaces, se listaran los empleados que están activos en el sistema. Mostrando información como: Identificación, nombre y apellido, teléfono, cargo, estado y finalmente dos casillas para consultar y modificar respectivamente.	

La tabla podrá filtrarse por todos los campos mencionados anteriormente, o también reacomodar las columnas según como desee el usuario (ascendente o descendente).

Si la no existe ningún empleado en el sistema, se debe mostrar un mensaje sobre la tabla como el siguiente : *“no se encontraron registros para mostrar”*.

Bajo la tabla se tendrá un botón para volver, el cual redirecciona a la ventana principal.

Tarea de ingeniería 11.3:

Tarea de Ingeniería	
Número de tarea: 11.3	Historia de usuario: 11
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas funcionales	Puntos Estimados: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe verificar que los métodos realizados para consultar los empleados en el sistema, funcionen sin ninguna falla, y posterior a esto, verificar que los datos desplegados en la interfaz son los correctos para cada empleado, además verificar que los filtros funcionan correctamente según se especifica en la tarea de ingeniería 9.1.	

Historia de usuario 12: Consultar/Modificar empleado

Tarea de ingeniería 12.1:

Tarea de Ingeniería	
Número de tarea: 12.1	Historia de usuario: 12
Nombre tarea: Crear métodos funcionales para consultar/modificar	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, el cual reciba como parámetro el registro modificado, se debe verificar que los campos obligatorios sean válidos y posteriormente se procede a guardar dicha edición. Si se encuentra algún tipo de error en el proceso debe ser retornado como una excepción, de lo contrario continua el flujo normal de la aplicación.	

Tarea de ingeniería 12.2:

Tarea de Ingeniería	
Número de tarea: 12.2	Historia de usuario: 12
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una interfaz gráfica que contenga campos suficientes para cargar la información respectiva al usuario seleccionado como se especifica en la tarea de ingeniería 8.1, además la interfaz debe contar con una tabla en la que se listaran los proyectos en los que se encuentra asignado el empleado y en los que a participado; inicialmente la tabla estará ordenada de manera que los proyectos en los que está asignado actualmente aparezcan en la parte superior. Si el usuario en la tabla de listar empleados (Tarea de ingeniería 9) selecciona la opción “Modificar” se mostrará en una nueva página la interfaz anteriormente nombrada, en los cuales todos los campos serán modificables. Además se debe mostrar un botón en la parte inferior alineado a la derecha del usuario con la frase “Guardar cambios”. Si por el contrario, el usuario selecciona la opción de “Consultar” se mostrará en una nueva página la información anteriormente nombrada, en este caso, los campos no se podrán modificar al igual que el botón para guardar cambios. Bajo el formulario y alineado a la derecha del usuario se debe encontrar un botón para volver a la ventana principal.	

Tarea de ingeniería 12.3:

Tarea de Ingeniería	
Número de tarea: 12.3	Historia de usuario: 12
Nombre tarea: Pruebas unitarias.	
Tipo de tarea: Pruebas funcionales	Puntos Estimados: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de consultar y modificar empleado. Para lo cual se tendrá en cuenta lo siguiente: • Los campos se deshabilitan al seleccionar consultar. • Los campos son editables al seleccionar modificar. • La información desplegada es la correcta para cada empleado (Información personal y	



PROYECTO:	Gestión de proyectos			Página 13 de 13
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

participación en proyectos).

- Valida correctamente los cambios efectuados sobre un empleado.
- Al modificar un empleado se guarda correctamente la información y se redirecciona como se especifica en la tarea de ingeniería 10.3.
- Los mensajes son mostrados de manera correcta.



PROYECTO:	Gestión de proyectos			Página 1 de 56
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tareas de ingeniería

Gestión de proyectos

Tabla de contenido

Historia de usuario 7: Ingresar material a inventario	6
Tarea de ingeniería 7.1:.....	6
Tarea de ingeniería 7.2:.....	6
Tarea de ingeniería 7.3:.....	7
Tarea de ingeniería 7.4:.....	8
Historia de usuario 8: Listar material en inventario.....	9
Tarea de ingeniería 8.1:.....	9
Tarea de ingeniería 8.2:.....	9
Tarea de ingeniería 8.3:.....	10
Historia de usuario 9: Consultar/Modificar inventario	11
Tarea de ingeniería 9.1:.....	11
Tarea de ingeniería 9.2:.....	11
Tarea de ingeniería 9.3:.....	12
Tarea de ingeniería 9.4:.....	13
Historia de usuario 13: Crear proyecto	13
Tarea de ingeniería 13.1:.....	13
Tarea de ingeniería 13.2:.....	14
Tarea de ingeniería 13.3:.....	15
Tarea de ingeniería 13.4:.....	15
Historia de usuario 14: Listar proyectos	16
Tarea de ingeniería 14.1:.....	16
Tarea de ingeniería 14.2:.....	17
Tarea de ingeniería 14.3:.....	18
Historia de usuario 15: Consultar/Modificar proyecto	18
Tarea de ingeniería 15.1:.....	18
Tarea de ingeniería 15.2:.....	19
Tarea de ingeniería 15.3:.....	20
Tarea de ingeniería 15.4:.....	21

Historia de usuario 16: Crear cotización	21
Tarea de ingeniería 16.1:.....	21
Tarea de ingeniería 16.2:.....	22
Tarea de ingeniería 16.3:.....	23
Tarea de ingeniería 16.4:.....	24
Historia de usuario 17: Crear actividades	24
Tarea de ingeniería 17.1:.....	24
Tarea de ingeniería 17.2:.....	25
Tarea de ingeniería 17.3:.....	26
Tarea de ingeniería 17.4:.....	26
Historia de usuario 18: Crear Clientes.....	27
Tarea de ingeniería 18.1:.....	27
Tarea de ingeniería 18.2:.....	28
Tarea de ingeniería 18.3:.....	28
Tarea de ingeniería 18.4:.....	29
Historia de usuario 19: Listar clientes	30
Tarea de ingeniería 19.1:.....	30
Tarea de ingeniería 19.2:.....	30
Tarea de ingeniería 19.3:.....	31
Historia de usuario 20: Consultar/Modificar cliente.....	32
Tarea de ingeniería 20.1:.....	32
Tarea de ingeniería 20.2:.....	33
Tarea de ingeniería 20.3:.....	33
Tarea de ingeniería 20.4:.....	34
Historia de usuario 21: Crear empresa	35
Tarea de ingeniería 21.1:.....	35
Tarea de ingeniería 21.2:.....	35
Tarea de ingeniería 21.3:.....	36
Tarea de ingeniería 21.4:.....	37
Historia de usuario 22: Listar empresas.....	37
Tarea de ingeniería 22.1:.....	37



PROYECTO:	Gestión de proyectos			Página 4 de 56
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tarea de ingeniería 22.2:.....	38
Tarea de ingeniería 22.3:.....	38
Historia de usuario 23: Consultar/Modificar empresa.....	39
Tarea de ingeniería 23.1:.....	39
Tarea de ingeniería 23.2:.....	40
Tarea de ingeniería 23.3:.....	40
Tarea de ingeniería 23.4:.....	41
Historia de usuario 24: Crear riesgo.....	42
Tarea de ingeniería 24.1:.....	42
Tarea de ingeniería 24.2:.....	42
Tarea de ingeniería 24.3:.....	43
Tarea de ingeniería 24.4:.....	44
Historia de usuario 25: Listar riesgo.....	44
Tarea de ingeniería 25.1:.....	44
Tarea de ingeniería 25.2:.....	45
Tarea de ingeniería 25.3:.....	45
Historia de usuario 26: Consultar/Modificar riesgo.....	46
Tarea de ingeniería 26.1:.....	46
Tarea de ingeniería 26.2:.....	47
Tarea de ingeniería 26.3:.....	47
Tarea de ingeniería 26.4:.....	48
Historia de usuario 27: Crear oficio.....	49
Tarea de ingeniería 27.1:.....	49
Tarea de ingeniería 27.2:.....	50
Tarea de ingeniería 27.3:.....	50
Tarea de ingeniería 27.4:.....	51
Historia de usuario 28: Listar oficios	52
Tarea de ingeniería 28.1:.....	52
Tarea de ingeniería 28.2:.....	52
Tarea de ingeniería 28.3:.....	53
Historia de usuario 29: Consultar/Modificar oficios	54



PROYECTO:	Gestión de proyectos			Página 5 de 56
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tarea de ingeniería 29.1:.....	54
Tarea de ingeniería 29.2:.....	54
Tarea de ingeniería 29.3:.....	55
Tarea de ingeniería 29.4:.....	56

Historia de usuario 7: Ingresar material a inventario

Tarea de ingeniería 7.1:

Tarea de Ingeniería	
Número de tarea: 7.1	Historia de usuario: 7
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad la creación de un material en el inventario. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.	

Tarea de ingeniería 7.2:

Tarea de Ingeniería	
Número de tarea: 7.2	Historia de usuario: 7
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear un método en la capa de vista, el cual, al presionar la opción **Guardar** verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente:

- Los datos requeridos no son nulos.
- Los datos tienen el tipo y formato correcto.

Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.

Tarea de ingeniería 7.3:

Tarea de Ingeniería	
Número de tarea: 7.3	Historia de usuario: 7
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de cliente, la cual contenga campos necesarios para diligenciar la información básica de un material, de la siguiente manera: • Nombre: Campo alfanumérico obligatorio. • Cantidad: Campo numérico obligatorio. • Tipo de material: Campo de selección obligatorio. • Precio de compra: Campo numérico obligatorio. • Precio de venta: campo numérico obligatorio. • Fecha de compra: Campo tipo calendario obligatorio. Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera:	

- **Volver** Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continuas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a la página principal, de lo contrario continua en la página actual.
- **Guardar** tiene como funcionalidad guardar en la capa de datos guardar la información diligenciada por el usuario. Si el proceso es exitoso redirecciona a la página principal mostrando un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.

Tarea de ingeniería 7.4:

Tarea de Ingeniería	
Número de tarea: 7.4	Historia de usuario: 7
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de ingresar material al inventario, para lo cual se debe tener en cuenta la especificación de los campos en la tarea de ingeniería 7.3. El proceso de pruebas debe tener las siguientes características: • Verificar validación de datos nulos. • Verificar validación de tipos de datos erróneos. • Verificar que la funcionalidad funcione correctamente. • Verificar que mensajes de error y confirmación se muestren en el momento y manera adecuados.	

Historia de usuario 8: Listar material en inventario

Tarea de ingeniería 8.1:

Tarea de Ingeniería	
Número de tarea: 8.1	Historia de usuario: 8
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad retornar una lista de materiales existentes en el sistema. Solo retorna los materiales que tengan disponible algún tipo de cantidad.	

Tarea de ingeniería 8.2:

Tarea de Ingeniería	
Número de tarea: 8.2	Historia de usuario: 8
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una interfaz gráfica en la capa de cliente, en la cual se muestre mediante una tabla los materiales que están disponibles en el momento. Mostrando la siguiente información: Nombre, Cantidad disponible, Tipo de material, Precio de compra, precio de venta y fecha de	

compra.

Esta información podrá ser filtrada según los siguientes campos: Nombre, Tipo de material.

Además se ofrece la opción para seleccionar un registro en específico para ser modificado o consultado posteriormente.

Se contará con un botón **Volver** en la parte inferior derecha de la tabla, el cual redirecciona a la página principal.

Tarea de ingeniería 8.3:

Tarea de Ingeniería	
Número de tarea: 8.3	Historia de usuario: 8
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de listar inventario, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 8.2. Este proceso debe tener las siguientes características: • Se lista solo el inventario disponible en el sistema. • Se muestra la información descrita en la tarea de ingeniería 8.2. • Se filtra adecuadamente según cada campo descrito en la tarea de ingeniería 8.2 • Botones de acción funcionan correctamente.	

Historia de usuario 9: Consultar/Modificar inventario

Tarea de ingeniería 9.1:

Tarea de Ingeniería	
Número de tarea: 9.1	Historia de usuario: 9
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, el cual reciba como parámetro un registro de tipo inventario, verificando que los campos obligatorios se encuentren debidamente diligenciados y así proceder a guardar los cambios en la capa de datos. Si ocurre algún error en el proceso se debe notificar mediante una excepción la cual será mostrada al usuario, de lo contrario continua el flujo normal de la aplicación.	

Tarea de ingeniería 9.2:

Tarea de Ingeniería	
Número de tarea: 9.2	Historia de usuario: 9
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe realizar un método en la capa de vista, la cual tenga como función verificar que los datos modificados por el usuario estén debidamente diligenciados, de la siguiente manera: Nombre: Campo alfanumérico obligatorio.	

- Cantidad: Campo numérico obligatorio.
- Tipo de material: Campo de selección obligatorio.
- Precio de compra: Campo numérico obligatorio.
- Precio de venta: campo numérico obligatorio.
- Fecha de compra: Campo tipo calendario obligatorio.

Si alguno de los campos no cumplen con el requisito se debe mostrar un mensaje de error al usuario, notificando que campo debe ser corregido. De lo contrario continua con el flujo normal de la aplicación.

Tarea de ingeniería 9.3:

Tarea de Ingeniería	
Número de tarea: 9.3	Historia de usuario: 9
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una interfaz gráfica que cumpla con las mismas características de la tarea de ingeniería 7.3. Si el usuario ha seleccionado consultar, los campos estarán deshabilitados para edición, si por el contrario el usuario selecciono modificar los campos estarán habilitados para ser editados. Además se contará con botones de acción en la parte inferior del formulario, de la siguiente manera: Guardar cambios: Solo será renderizado si el usuario selecciono modificar y tendrá como función guardar el registro con todos los cambios realizados en la capa de datos. Volver: vuelve a la página anterior (Listar inventario).	

Tarea de ingeniería 9.4:

Tarea de Ingeniería	
Número de tarea: 9.4	Historia de usuario: 9
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre las funcionalidades de consultar/modificar inventario, para lo cual se tendrán en cuenta la siguientes características: • Al presionar consultar se muestran los datos correctos al registro seleccionado, los campos están deshabilitados. • Al presionar modificar se muestran los datos correctos al registro seleccionado, los campos están habilitados. • Se valida correctamente la información ingresada. • Se guardan los cambios correctamente. • Los botones de acción funcionan de la manera correcta.	

Historia de usuario 13: Crear proyecto

Tarea de ingeniería 13.1:

Tarea de Ingeniería	
Número de tarea: 13.1	Historia de usuario: 13
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear un método en la capa de lógica, que tenga como funcionalidad la creación de un nuevo proyecto en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados.

Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error.

Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.

Tarea de ingeniería 13.2:

Tarea de Ingeniería	
Número de tarea: 13.2	Historia de usuario: 13
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto. Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a crear el proyecto en el sistema.	

Tarea de ingeniería 13.3:

Tarea de Ingeniería	
Número de tarea: 13.3	Historia de usuario: 13
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual contenga campos necesarios para diligenciar la información del proyecto, de la siguiente manera: • Nombre: Campo alfanumérico obligatorio. • Fecha de inicio, campo de tipo date. • Fecha de entrega: Campo de tipo date. • Usuario responsable: Campo de selección obligatorio. • Cliente: campo de selección obligatorio. Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera: • Volver Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continuas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a la página principal, de lo contrario continua en la página actual. Guardar tiene como funcionalidad guardar en la capa de datos la información diligenciada por el usuario. Si el proceso es exitoso redirecciona a la página de consulta de proyecto, mostrando un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.	

Tarea de ingeniería 13.4:

Tarea de Ingeniería

Número de tarea: 13.4	Historia de usuario: 13
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de crear proyecto, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 13.3. Este proceso debe tener las siguientes características: • Se validan correctamente los datos obligatorios. • Se validan correctamente los tipos de datos. • Crea el proyecto exitosamente. • Botones de acción funcionan correctamente. • Mensaje de confirmación al intentar salir funciona correctamente.	

Historia de usuario 14: Listar proyectos

Tarea de ingeniería 14.1:

Tarea de Ingeniería	
Número de tarea: 14.1	Historia de usuario: 14
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad consultar todos los proyectos existentes en el sistema, ordenando de la siguiente manera:	

- Primero los proyectos que se encuentran en ejecución.
- Segundo los proyectos que se encuentran cotizados.
- Tercero los proyectos que se encuentran terminados.
- Cuarto los proyectos que se encuentran cancelados.
- Dentro de cada categoría se deben ordenar según el nombre.

Tarea de ingeniería 14.2:

Tarea de Ingeniería	
Número de tarea: 14.2	Historia de usuario: 14
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual, por medio de una tabla muestre información básica de los proyectos, de la siguiente manera: • Nombre, puede ser filtrado según este campo. • Responsable, puede ser filtrado según este campo. • Cliente, puede ser filtrado según este campo. • Fecha de inicio, puede ser reordenado según este campo. • Fecha de fin, puede ser reordenado según este campo. • Estado, puede ser filtrado según este campo. • Botón de acción para consultar. • Botón de acción para modificar. Además, se contará con un botón en la parte inferior para volver a la página principal.	

Tarea de ingeniería 14.3:

Tarea de Ingeniería	
Número de tarea: 14.3	Historia de usuario: 14
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de listar proyecto, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 14.2. Este proceso debe tener las siguientes características: • Se listan todos los proyectos en el sistema. • Los proyectos están ordenados de manera correcta. • Se muestra la información descrita en la tarea de ingeniería 14.2. • Se filtra adecuadamente según cada campo descrito en la tarea de ingeniería 14.2 . • Se reordena adecuadamente según cada campo descrito en la tarea de ingeniería 14.2. Botón de volver funciona correctamente.	

Historia de usuario 15: Consultar/Modificar proyecto**Tarea de ingeniería 15.1:**

Tarea de Ingeniería	
Número de tarea: 15.1	Historia de usuario: 15
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	

Descripción:

Se debe crear un método en la capa de lógica, que tenga como funcionalidad modificar un proyecto en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados.

Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error.

Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.

Tarea de ingeniería 15.2:

Tarea de Ingeniería	
Número de tarea: 15.2	Historia de usuario: 15
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar cambios verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto. Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.	

Tarea de ingeniería 15.3:

Tarea de Ingeniería	
Número de tarea: 15.3	Historia de usuario: 15
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, en la cual, por medio de pestañas se debe mostrar: Información general del proyecto, Cotización, Actividades. De este modo se ofrece un formulario en cada pestaña en el que se cargará la información correspondiente al proyecto seleccionado. (Especificación en tareas de ingeniería 15.3, 16.3 y 17.3) Si el usuario selecciona consultar, los campos estarán deshabilitados para edición, si por el contrario selecciona modificar, los campos estarán habilitados para edición. El usuario podrá añadir o eliminar elementos cotizados o actividades. Si el usuario realiza algún tipo de modificación en la cotización se debe crear un nuevo registro con estos cambios. Además se contará con dos botones de acción: • Guardar cambios: Será renderizado solo si el usuario ha seleccionado modificar, tendrá como funcionalidad guardar las modificaciones en la capa de datos. • Volver: Tiene como función redireccionar a la historia de usuario 14 (Listar proyectos).	

Tarea de ingeniería 15.4:

Tarea de Ingeniería	
Número de tarea: 15.4	Historia de usuario: 15
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de Consultar/Modificar proyecto, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 13.3. Este proceso debe tener las siguientes características: • Se valida correctamente que los datos obligatorios no sean nulos. • Se valida correctamente los tipos de datos ingresados. • Se carga correctamente la información del proyecto cargado. • Se guardan correctamente las modificaciones. • Botones de acción funcionan correctamente.	

Historia de usuario 16: Crear cotización

Tarea de ingeniería 16.1:

Tarea de Ingeniería	
Número de tarea: 16.1	Historia de usuario: 16
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear un método en la capa de lógica, que tenga como funcionalidad la creación de una nueva cotización. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados.

Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error.

Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.

Tarea de ingeniería 16.2:

Tarea de Ingeniería	
Número de tarea: 16.2	Historia de usuario: 16
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, mapee una cotización en específico con sus respectivos detalles (Material cotizado, oficios cotizados, gastos adicionales.) al presionar la opción Guardar verifica que los datos ingresados en el formulario son válidos y reconstruya un objeto tipo Cotización para ser asignado al proyecto, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto. Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.	
Observación:	

Tarea de ingeniería 16.3:

Tarea de Ingeniería	
Número de tarea: 16.3	Historia de usuario: 16
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: En la interfaz de crear proyecto (Historia de usuario 13) se debe crear una pestaña tipo Wizard (Secuencia de pasos) en la cual aparecerá como segundo paso la creación de una cotización para el proyecto en creación, la cual contenga campos necesarios para diligenciar la información básica de una cotización, de la siguiente manera: • Se debe ofrecer la manera para agregar la cantidad de horas para un determinado oficio y riesgo asociado. • Se debe ofrecer la manera para agregar gastos adicionales en la cotización. • Se debe ofrecer la manera de agregar materiales. • Finalmente la información debe ser mostrada en manera de tabla, calculando el valor de cada elemento y el valor total para la cotización. • Se debe ofrecer la opción para modificar o eliminar un elemento en la cotización. Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera: • Volver Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continúas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a la página principal, de lo contrario continua en la página actual. • Guardar tiene como funcionalidad guardar en la capa de datos guardar la información diligenciada por el usuario. Si el proceso es exitoso redirecciona a la página principal mostrando un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.	

Tarea de ingeniería 16.4:

Tarea de Ingeniería	
Número de tarea: 16.4	Historia de usuario: 16
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de crear cotización, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 16.3. Este proceso debe tener las siguientes características: • Se agrega adecuadamente la información para riesgos y oficios. • Se agrega adecuadamente la información para materiales a usar. • Se agrega adecuadamente los gastos adicionales. • Se calcula correctamente los valores para cada elemento. • Se calcula correctamente el valor total de la cotización. • Botones de acción funcionan correctamente.	

Historia de usuario 17: Crear actividades**Tarea de ingeniería 17.1:**

Tarea de Ingeniería	
Número de tarea: 17.1	Historia de usuario: 17
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	

Descripción:

Se debe crear un método en la capa de lógica, que tenga como funcionalidad la creación de actividades para un proyecto. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados (Se especifican en la tarea de Ingeniería 17.3).

Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error.

Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.

Tarea de ingeniería 17.2:

Tarea de Ingeniería	
Número de tarea: 17.2	Historia de usuario: 17
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto (Especificación en Tarea de Ingeniería 17.3). Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.	

Tarea de ingeniería 17.3:

Tarea de Ingeniería	
Número de tarea: 17.3	Historia de usuario: 17
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: En la interfaz de Modificar/Consultar proyecto (historia de usuario 15) Se debe crear una nueva pestaña “Actividades” en la cual se ofrecerá una interfaz gráfica con campos necesarios para diligenciar la información básica de una actividad, de la siguiente manera: • Nombre: Campo alfanumérico obligatorio. • Empleados responsables: Campo de selección, se agregan en tabla. • Oficio y riesgo de la actividad: Campo de selección obligatorio, se cargaran según los riesgos cotizados. • Se debe ofrecer la opción para eliminar un empleado asignado a una actividad en específico. • No se permiten valores repetidos en la tabla de responsables.	

Tarea de ingeniería 17.4:

Tarea de Ingeniería	
Número de tarea: 17.4	Historia de usuario: 17
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	

Descripción:

Se deben realizar pruebas sobre la funcionalidad de Crear actividad, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 17.3. Este proceso debe tener las siguientes características:

- Los campos obligatorios son validados correctamente.
- Se agregan correctamente los empleados responsables.
- Se crea correctamente la actividad en el sistema.

Historia de usuario 18: Crear Clientes

Tarea de ingeniería 18.1:

Tarea de Ingeniería	
Número de tarea: 18.1	Historia de usuario: 18
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad la creación de clientes en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.	

Tarea de ingeniería 18.2:

Tarea de Ingeniería	
Número de tarea: 18.2	Historia de usuario: 18
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: - Los datos requeridos no son nulos. - Los datos tienen el tipo y formato correcto (Se especifica en la tarea de ingeniería 18.3). Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.	

Tarea de ingeniería 18.3:

Tarea de Ingeniería	
Número de tarea: 18.3	Historia de usuario: 18
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual contenga campos necesarios para diligenciar la información básica de un cliente, de la siguiente manera:	

- Nombre: Campo alfanumérico obligatorio.
- Dirección: Campo Alfanumérico.
- Ciudad: Campo alfanumérico obligatorio.
- Correo: Campo alfanumérico.
- Teléfono: campo alfanumérico.
- Empresa: Campo de selección obligatorio.

Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera:

- **Volver** Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continuas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a la página principal, de lo contrario continua en la página actual.
- **Guardar** tiene como funcionalidad guardar en la capa de datos guardar la información diligenciada por el usuario. Si el proceso es exitoso limpia el formulario y muestra un mensaje de éxito, de lo contrario muestra un mensaje de error sin limpiar el formulario.

Tarea de ingeniería 18.4:

Tarea de Ingeniería	
Número de tarea: 18.4	Historia de usuario: 18
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de listar inventario, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 18.3. Este proceso debe tener las siguientes características: • Se validan correctamente los tipos de datos.	

- Se validan correctamente los datos obligatorios.
- Se muestran correctamente los mensajes.
- Se crea correctamente el cliente.
- Botones de acción funcionan correctamente.

Historia de usuario 19: Listar clientes

Tarea de ingeniería 19.1:

Tarea de Ingeniería	
Número de tarea: 19.1	Historia de usuario: 19
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad consultar todos clientes en el sistema, los cuales deben estar ordenados de manera alfabética según el nombre del cliente.	

Tarea de ingeniería 19.2:

Tarea de Ingeniería	
Número de tarea: 19.2	Historia de usuario: 19
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:

Programador: Hernán Antonio Ramírez Osorio

Descripción:

Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual muestre en manera de tabla la información de los clientes en el sistema ordenados de manera alfabética según nombre, de la siguiente manera:

- Nombre, se puede filtrar y reordenar por este campo.
- Empresa, se puede filtrar y reordenar por este campo.
- Dirección.
- Ciudad, se puede filtrar y reordenar por este campo.
- Correo.
- Teléfono.
- Botón para consultar cliente.
- Botón para modificar cliente.

Además, se contará con un botón de **Volver** en la parte inferior derecha el cual redirecciona a la página principal.

Tarea de ingeniería 19.3:

Tarea de Ingeniería	
Número de tarea: 19.3	Historia de usuario: 9
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de listar clientes, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 19.2. Este proceso debe tener las siguientes	

características:

- Se listan todos los clientes en el sistema ordenados de manera alfabética según nombre.
- Se filtran los registros según Nombre, empresa o ciudad.
- Se muestra la información correcta para cada cliente.
- Botones de acción funcionan correctamente.

Historia de usuario 20: Consultar/Modificar cliente

Tarea de ingeniería 20.1:

Tarea de Ingeniería	
Número de tarea: 20.1	Historia de usuario: 20
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad modificar un cliente en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.	

Tarea de ingeniería 20.2:

Tarea de Ingeniería	
Número de tarea: 20.2	Historia de usuario: 20
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar Cambios verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto (Especificación en la tarea de ingeniería 18.3). Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.	

Tarea de ingeniería 20.3:

Tarea de Ingeniería	
Número de tarea: 20.3	Historia de usuario: 20.3
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual cargue la información correspondiente a un cliente seleccionado (Información descrita en tarea de ingeniería 19.2).	

Si el usuario selecciona consultar los campos estarán deshabilitados para edición, si por el contrario selecciona modificar, los campos estarán habilitados para edición.

Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera:

- **Volver** Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continuas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a la historia de usuario 19, de lo contrario continua en la página actual.
- **Guardar cambios** tiene como funcionalidad guardar en la capa de datos guardar las modificaciones realizadas por el usuario. Si el proceso es exitoso redirecciona a la página principal mostrando un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.

Tarea de ingeniería 20.4:

Tarea de Ingeniería	
Número de tarea: 20.4	Historia de usuario: 20
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de Consultar/Modificar cliente, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 18.3. Este proceso debe tener las siguientes características: • Se valida correctamente los campos nulos. • Se valida correctamente los tipos de datos. • Se carga correctamente la información respectiva de un cliente. • Se modifica correctamente un cliente. • Botones de acción funcionan correctamente.	

Historia de usuario 21: Crear empresa

Tarea de ingeniería 21.1:

Tarea de Ingeniería	
Número de tarea: 21.1	Historia de usuario: 21
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad la creación de una nueva empresa en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.	

Tarea de ingeniería 21.2:

Tarea de Ingeniería	
Número de tarea: 21.2	Historia de usuario: 21
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar verifica que	

los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente:

- Los datos requeridos no son nulos.
- Los datos tienen el tipo y formato correcto (Especificado en tarea de ingeniería 21.3).

Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.

Tarea de ingeniería 21.3:

Tarea de Ingeniería	
Número de tarea: 21.3	Historia de usuario: 21
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual contenga campos necesarios para diligenciar la información básica de una empresa, de la siguiente manera: • Nombre: Campo alfanumérico obligatorio. • Ciudad: Campo alfanumérico. Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera: • Volver Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continuas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a la página principal, de lo contrario continua en la página actual. • Guardar tiene como funcionalidad guardar en la capa de datos guardar la información diligenciada por el usuario. Si el proceso es exitoso limpia el formulario mostrando un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.	

Tarea de ingeniería 21.4:

Tarea de Ingeniería	
Número de tarea: 21.4	Historia de usuario: 21
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de crear empresa, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 21.3 Este proceso debe tener las siguientes características: • Se validan correctamente los datos obligatorios. • Se registra correctamente la empresa. • Botones de acción funcionan correctamente.	

Historia de usuario 22: Listar empresas

Tarea de ingeniería 22.1:

Tarea de Ingeniería	
Número de tarea: 22.1	Historia de usuario: 22
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear un método en la capa de lógica, que tenga como funcionalidad consultar todas las empresas en el sistema. Los resultados deben estar ordenados de manera alfabética según el nombre de la empresa.

Tarea de ingeniería 22.2:

Tarea de Ingeniería	
Número de tarea: 22.2	Historia de usuario: 22
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, muestre por medio de una tabla información de las empresas, de la siguiente manera: • Nombre, se puede filtrar y reordenar. • Ciudad, se puede filtrar y reordenar. • Consultar, botón de acción. • Modificar, botón de acción. Además, se contará con un botón de Volver que tiene como funcionalidad volver a la página principal.	

Tarea de ingeniería 22.3:

Tarea de Ingeniería	
Número de tarea: 22.3	Historia de usuario: 22
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora

Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de listar empresas, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 22.2 Este proceso debe tener las siguientes características: • Se listan todas las empresas en el sistema. • Se muestra la información correcta de cada empresa. • Se filtra correctamente según los campos especificados. • Botones de acción funcionan correctamente.	

Historia de usuario 23: Consultar/Modificar empresa

Tarea de ingeniería 23.1:

Tarea de Ingeniería	
Número de tarea: 23.1	Historia de usuario: 23
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad modificar una empresa existente. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.	

Tarea de ingeniería 23.2:

Tarea de Ingeniería	
Número de tarea: 23.2	Historia de usuario: 23
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar cambios verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto (Especificado en Tarea de ingeniería 21.3). Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.	

Tarea de ingeniería 23.3:

Tarea de Ingeniería	
Número de tarea: 23.3	Historia de usuario: 23
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de cliente, la cual cumpla con las mismas	

características que la tarea de ingeniería 21.3 ofreciendo campos suficientes para cargar la información de una empresa seleccionada.

Si el usuario selecciona consultar los datos estarán deshabilitados para edición, si por el contrario selección modificar los campos estarán habilitados para edición.

Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera:

- **Volver** Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continuas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a la página principal, de lo contrario continua en la página actual.
- **Guardar** tiene como funcionalidad guardar en la capa de datos guardar la información diligenciada por el usuario. Si el proceso es exitoso redirecciona a la interfaz descrita en la historia de usuario 21 (Listar empresa) un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.

Tarea de ingeniería 23.4:

Tarea de Ingeniería	
Número de tarea: 23.4	Historia de usuario: 23
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de Consultar/modificar empresa, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 21.3. Este proceso debe tener las siguientes características: • Se validan los campos nulos. • Se carga correctamente la información. • Se modifica correctamente la empresa. • Botones de acción funcionan correctamente.	

Historia de usuario 24: Crear riesgo

Tarea de ingeniería 24.1:

Tarea de Ingeniería	
Número de tarea: 24.1	Historia de usuario: 24
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad la creación de un riesgo nuevo. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.	

Tarea de ingeniería 24.2:

Tarea de Ingeniería	
Número de tarea: 24.2	Historia de usuario: 24
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar verifica que	

los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente:

- Los datos requeridos no son nulos.
- Los datos tienen el tipo y formato correcto (Especificado en la tarea de ingeniería 24.3).

Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.

Tarea de ingeniería 24.3:

Tarea de Ingeniería	
Número de tarea: 24.3	Historia de usuario: 24
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual contenga campos necesarios para diligenciar la información básica de un riesgo, de la siguiente manera: • Nombre: Campo alfanumérico obligatorio. • Porcentaje: Campo numérico obligatorio, con intervalo de [0-100]. Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera: • Volver Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continuas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a la página principal, de lo contrario continua en la página actual. • Guardar tiene como funcionalidad guardar en la capa de datos guardar la información diligenciada por el usuario. Si el proceso es exitoso limpia el formulario y muestra un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.	

Tarea de ingeniería 24.4:

Tarea de Ingeniería	
Número de tarea: 24.4	Historia de usuario: 24
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de crear riesgo, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 24.3. Este proceso debe tener las siguientes características: • Se validan correctamente los datos. • Se crea correctamente los registros. • Botones de acción funcionan correctamente.	

Historia de usuario 25: Listar riesgo

Tarea de ingeniería 25.1:

Tarea de Ingeniería	
Número de tarea: 25.1	Historia de usuario: 25
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear un método en la capa de lógica, que tenga como funcionalidad consultar todos los riesgos que existan en el sistema. Los datos deben estar ordenados de manera alfabética según el nombre, y según el porcentaje de manera descendente.

Tarea de ingeniería 25.2:

Tarea de Ingeniería	
Número de tarea: 25.2	Historia de usuario: 25
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual muestre por medio de una tabla la información correspondiente a los riesgos, de la siguiente manera: • Nombre, se puede filtrar y reordenar. • Porcentaje, se puede reordenar. • Consultar, botón de acción. • Modificar, botón de acción. Además, se contará con un botón de Volver que tiene como funcionalidad volver a la página principal.	

Tarea de ingeniería 25.3:

Tarea de Ingeniería	
Número de tarea: 25.3	Historia de usuario: 25
Nombre tarea: Pruebas unitarias	

Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de listar riesgos, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 25.2. Este proceso debe tener las siguientes características: • Se listan todos los riesgos en el sistema. • La información para cada riesgo es correcta. • Se filtra y reordena correctamente la tabla. • Botones de acción funcionan correctamente.	

Historia de usuario 26: Consultar/Modificar riesgo

Tarea de ingeniería 26.1:

Tarea de Ingeniería	
Número de tarea: 26.1	Historia de usuario: 26
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad la modificación de un riesgo existente. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un	

error en el proceso se debe generar una excepción especificando el error ocurrido.

Tarea de ingeniería 26.2:

Tarea de Ingeniería	
Número de tarea: 26.2	Historia de usuario: 26
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar cambios verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto (Especificado en la tarea de ingeniería 24.3). Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.	

Tarea de ingeniería 26.3:

Tarea de Ingeniería	
Número de tarea: 26.3	Historia de usuario: 26
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:

Programador: Hernán Antonio Ramírez Osorio

Descripción:

Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual cargue la información de un riesgo que ha sido seleccionado.

Si selecciona consultar los campos serán no editables, si por el contrario selecciona modificar, los campos serán editables para modificaciones.

Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera:

- **Volver** Tiene como funcionalidad volver a listar riegos, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continuas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a listar riesgos, de lo contrario continua en la página actual.
- **Guardar Cambios** tiene como funcionalidad guardar las modificaciones en la capa de datos. Si el proceso es exitoso redirecciona a listar riesgos mostrando un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.

Tarea de ingeniería 26.4:

Tarea de Ingeniería	
Número de tarea: 26.4	Historia de usuario: 26
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de consultar/modificar riesgo, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 26.3. Este proceso debe tener las siguientes características: • Se carga correctamente la información para cada registro. • Los campos son editables o no editables según sea el caso. • Se valida correctamente la información.	

- Se guardan las modificaciones correctamente.
- Botones de acción funcionan correctamente.

Historia de usuario 27: Crear oficio

Tarea de ingeniería 27.1:

Tarea de Ingeniería	
Número de tarea: 27.1	Historia de usuario: 27
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad la creación de un oficio en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.	

Tarea de ingeniería 27.2:

Tarea de Ingeniería	
Número de tarea: 27.2	Historia de usuario: 27
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto (Especificación en Tarea de ingeniería 27.3). Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a ingresar material en inventario.	

Tarea de ingeniería 27.3:

Tarea de Ingeniería	
Número de tarea: 27.3	Historia de usuario: 27
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual contenga campos necesarios para diligenciar la información básica de un oficio, de la siguiente manera:	

- Nombre: Campo alfanumérico obligatorio.
- Precio: Campo numérico obligatorio.

Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera:

- **Volver** Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: “Los cambios se perderán si continuas con esta acción. ¿Deseas continuar?” Si selecciona aceptar, redirecciona a la página principal, de lo contrario continua en la página actual.
- **Guardar** tiene como funcionalidad guardar en la capa de datos la información diligenciada por el usuario. Si el proceso es exitoso limpia el formulario mostrando un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.

Tarea de ingeniería 27.4:

Tarea de Ingeniería	
Número de tarea: 27.4	Historia de usuario: 27
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de crear oficio, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 27.3. Este proceso debe tener las siguientes características: • Se validan correctamente los datos. • Se crea correctamente el registro. • Botones de acción funcionan correctamente.	

Historia de usuario 28: Listar oficinas

Tarea de ingeniería 28.1:

Tarea de Ingeniería	
Número de tarea: 28.1	Historia de usuario:28
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como consultar todos los oficinas en el sistema. Los registros deben estar ordenados alfabéticamente según el campo nombre.	

Tarea de ingeniería 28.2:

Tarea de Ingeniería	
Número de tarea: 28.2	Historia de usuario: 28
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual muestre por medio de una tabla la información de los oficinas, de la siguiente manera: • Nombre, se puede filtrar y reordenar. • Precio, se puede reordenar. • Consultar, botón de acción para seleccionar el registro en específico.	

- Modificar, botón de acción para seleccionar el registro en específico.

Además, se contará con un botón de **Volver** que tiene como funcionalidad volver a la página principal.

Tarea de ingeniería 28.3:

Tarea de Ingeniería	
Número de tarea: 28.3	Historia de usuario: 28
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de listar oficios, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 28.2. Este proceso debe tener las siguientes características: • Se listan todos los registros en el sistema. • Se muestra la información descrita en la tarea de ingeniería 28.2. • Se muestra información correcta para cada registro. • Se filtra correctamente según lo descrito en la tarea de ingeniería 28.2. • Botones de acción funcionan correctamente.	

Historia de usuario 29: Consultar/Modificar oficios

Tarea de ingeniería 29.1:

Tarea de Ingeniería	
Número de tarea: 29.1	Historia de usuario: 29
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad modificar un registro existente en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si la ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.	

Tarea de ingeniería 29.2:

Tarea de Ingeniería	
Número de tarea: 29.2	Historia de usuario: 29
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear un método en la capa de vista, el cual, al presionar la opción **Guardar cambios** verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente:

- Los datos requeridos no son nulos.
- Los datos tienen el tipo (Especificación en tarea de ingeniería 27.3).

Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente a modificar el registro.

Tarea de ingeniería 29.3:

Tarea de Ingeniería	
Número de tarea: 29.3	Historia de usuario: 29
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual contenga campos necesarios para cargar la información de un oficio (Especificado en la tarea de ingeniería 27.3). Si el usuario selecciona consultar, los campos estará deshabilitados para edición, si por el contrario selecciona modificar los campos estarán habilitados para edición. Además se contará con dos botones de acción en la parte inferior derecha del formulario, de la siguiente manera: • Guardar cambios: Solo se renderiza si selección modificar, tiene como funcionalidad guardar los cambios del registro modificado. Si el proceso es exitoso redirecciona a la interfaz de listar oficio mostrando un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual • Volver Tiene como funcionalidad volver a la página principal, al ser presionado se debe mostrar un mensaje como el siguiente: "Los cambios se perderán si continúas con esta acción. ¿Deseas continuar?" Si selecciona aceptar, redirecciona a la interfaz de listar	

oficio, de lo contrario continua en la página actual.

Tarea de ingeniería 29.4:

Tarea de Ingeniería	
Número de tarea: 29.4	Historia de usuario: 29
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de consultar/modificar oficio, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 29.3. Este proceso debe tener las siguientes características: • Los campos se habilitan o deshabilitan para edición según la opción seleccionada. • Se valida correctamente la información. • Se carga la información correcta para el registro seleccionado. • Se guardar la información modificada correctamente. • Botones de acción funcionan correctamente.	



PROYECTO:	Gestión de proyectos			Página 1 de 25
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tareas de ingeniería

Gestión de proyectos



PROYECTO:	Gestión de proyectos		Página 2 de 25
DOCUMENTO:	Tareas de ingeniería		
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio

Tabla de contenido

Historia de usuario 30: Registrar cargos	4
Tarea de ingeniería 30.1:.....	4
Tarea de ingeniería 30.2:.....	4
Tarea de ingeniería 30.3:.....	5
Tarea de ingeniería 30.4:.....	6
Historia de usuario 31: Listar cargos	7
Tarea de ingeniería 31.1:.....	7
Tarea de ingeniería 31.2:.....	7
Tarea de ingeniería 31.3:.....	8
Historia de usuario 32: Consultar/Modificar cargo	9
Tarea de ingeniería 32.1:.....	9
Tarea de ingeniería 32.2:.....	9
Tarea de ingeniería 32.3:.....	10
Tarea de ingeniería 32.4:.....	11
Historia de usuario 33: Registrar tiempo trabajado en actividad	12
Tarea de ingeniería 33.1:.....	12
Tarea de ingeniería 33.2:.....	13
Tarea de ingeniería 33.3:.....	13
Tarea de ingeniería 33.4:.....	14
Historia de usuario 34: Consultar tiempo trabajado en actividad	15
Tarea de ingeniería 34.1:.....	15
Tarea de ingeniería 34.2:.....	16
Tarea de ingeniería 34.3:.....	17
Historia de usuario 35: Modificar tiempo trabajado en actividad	17
Tarea de ingeniería 35.1:.....	17
Tarea de ingeniería 35.2:.....	18
Tarea de ingeniería 35.3:.....	19
Tarea de ingeniería 35.4:.....	19



PROYECTO:	Gestión de proyectos			Página 3 de 25
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Historia de usuario 36: Guardar archivo de proyecto	20
Tarea de ingeniería 36.1:.....	20
Tarea de ingeniería 36.2:.....	21
Tarea de ingeniería 36.3:.....	21
Tarea de ingeniería 36.4:.....	22
Historia de usuario 37: Listar archivos de proyecto.....	23
Tarea de ingeniería 37.1:.....	23
Historia de usuario 38: Eliminar archivos de proyecto	23
Tarea de ingeniería 38.1:.....	23
Tarea de ingeniería 38.2:.....	24
Tarea de ingeniería 38.3:.....	24

Historia de usuario 30: Registrar cargos

Tarea de ingeniería 30.1:

Tarea de Ingeniería	
Número de tarea: 30.1	Historia de usuario: 30
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad registrar un cargo en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si ocurre un error en el proceso se debe generar una excepción especificando lo ocurrido.	

Tarea de ingeniería 30.2:

Tarea de Ingeniería	
Número de tarea: 30.2	Historia de usuario: 30
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 40 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear un método en la capa de vista, el cual, al presionar la opción **Guardar** verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente:

- Los datos requeridos no son nulos.
- Los datos tienen el tipo y formato correcto.

Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente en la capa de lógica.

Tarea de ingeniería 30.3:

Tarea de Ingeniería	
Número de tarea: 30.3	Historia de usuario: 30
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de cliente, ubicada en la sección Configuración -> Cargos , inicialmente se mostrará una lista con todos los cargos existentes, y en la parte superior de la lista se ofrece la opción de Agregar nuevo , al seleccionar esta opción se debe cargar en modo de modal un formulario que contenga campos necesarios para diligenciar la información básica de un cargo, de la siguiente manera: • Nombre: Campo alfanumérico obligatorio. • Descripción: Campo alfanumérico. Además, se contará con botones de acción en la parte inferior derecha del formulario, de la siguiente manera: • Cancelar Tiene como funcionalidad cancelar el registro, al ser presionado se debe ocultar el modal y mostrar nuevamente la lista de cargos. • Guardar tiene como funcionalidad guardar en la capa de datos guardar la información	

diligenciada por el usuario. Si el proceso es exitoso se muestra un mensaje de éxito, de lo contrario muestra un mensaje de error y continúa en la interfaz actual.

Tarea de ingeniería 30.4:

Tarea de Ingeniería	
Número de tarea: 30.4	Historia de usuario: 30
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado: 30 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de registrar cargo, para lo cual se debe tener en cuenta la especificación de los campos en la tarea de ingeniería 30.3 . El proceso de pruebas debe tener las siguientes características: • Verificar validación de datos nulos. • Verificar que la funcionalidad funcione correctamente. • Verificar que mensajes se muestren en el momento y manera adecuados.	

Historia de usuario 31: Listar cargos

Tarea de ingeniería 31.1:

Tarea de Ingeniería	
Número de tarea: 31.1	Historia de usuario: 31
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 30 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad retornar la lista de cargos existentes en el sistema. El método debe retornar una lista de objetos tipo Cargo. Si no existe ningún registro en el sistema se debe retornar null.	

Tarea de ingeniería 31.2:

Tarea de Ingeniería	
Número de tarea: 31.2	Historia de usuario: 31
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Tiempo Estimado: 40 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una interfaz gráfica en la capa de cliente, ubicada en Configuración -> Cargos en la cual se muestre mediante una tabla los cargos existentes en el sistema. Mostrando la siguiente información: Nombre, descripción. En la parte superior de la tabla se ubicara un campo de texto para filtrar la información según los	

siguientes campos: Nombre, descripción.

Se ubicara un botón en la parte superior de la tabla para registrar un nuevo cargo como se describe en [la tarea de ingeniería 30.3](#)

Además se tendrán dos botones para Consultar o Modificar un cargo en particular.

Tarea de ingeniería 31.3:

Tarea de Ingeniería	
Número de tarea: 31.3	Historia de usuario: 31
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 20 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de listar Cargos, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 31.2. Este proceso debe tener las siguientes características: • Se listan todos los cargos registrados en el sistema. • Se muestra la información descrita en la tarea de ingeniería 31.2. • Se filtra adecuadamente la información. • Botones de acción funcionan correctamente.	

Historia de usuario 32: Consultar/Modificar cargo

Tarea de ingeniería 32.1:

Tarea de Ingeniería	
Número de tarea: 32.1	Historia de usuario: 32
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 40 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, el cual tenga como funcionalidad guardar cambios sobre un cargo modificado, este método debe recibir como parámetro un registro de tipo Cargo, verificando que los campos obligatorios se encuentren debidamente diligenciados y así proceder a guardar los cambios en la capa de datos. Si ocurre algún error en el proceso se debe notificar mediante una excepción la cual será mostrada al usuario, de lo contrario continua el flujo normal de la aplicación.	

Tarea de ingeniería 32.2:

Tarea de Ingeniería	
Número de tarea: 32.2	Historia de usuario: 32
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 40 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe realizar un método en la capa de vista, el cual tenga como función verificar que los datos modificados por el usuario estén debidamente diligenciados, de la siguiente manera:	

- Nombre: Campo alfanumérico obligatorio.
- Descripción: Campo alfanumérico.

Si alguno de los campos no cumplen con el requisito se debe mostrar un mensaje de error al usuario, notificando que campo debe ser corregido. De lo contrario continua con el flujo normal de la aplicación.

Tarea de ingeniería 32.3:

Tarea de Ingeniería	
Número de tarea: 32.3	Historia de usuario: 32
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Partiendo de la interfaz gráfica descrita en la tarea de ingeniería 31.2 , se debe realizar lo siguiente. La tabla de cargos tendrá la opción de Consultar o Modificar un registro en específico, de esta manera si el usuario selecciona Consultar se mostrará un formulario tipo modal diligenciado con los campos deshabilitados para edición, si por el contrario el usuario selecciona Modificar los campos estarán habilitados para ser editados. Además se contará con botones de acción en la parte inferior del formulario, de la siguiente manera: Guardar cambios: Solo será renderizado si el usuario selecciono Modificar y tendrá como función guardar el registro con todos los cambios realizados en la capa de datos. Cancelar: Cancela la edición y oculta el modal.	

Tarea de ingeniería 32.4:

Tarea de Ingeniería	
Número de tarea: 32.4	Historia de usuario: 32
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 20 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre las funcionalidades de consultar/modificar cargo, para lo cual se tendrán en cuenta la siguientes características: • Al presionar consultar se muestran los datos correctos al registro seleccionado, los campos están deshabilitados. • Al presionar modificar se muestran los datos correctos al registro seleccionado, los campos están habilitados. • Se valida correctamente la información ingresada. • Se guardan los cambios correctamente. • Los botones de acción funcionan de la manera correcta.	

Historia de usuario 33: Registrar tiempo trabajado en actividad

Tarea de ingeniería 33.1:

Tarea de Ingeniería	
Número de tarea: 33.1	Historia de usuario: 33
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 2 Horas
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad registrar un tiempo trabajado en una actividad. El cual debe recibir un objeto diligenciado por el usuario y calcular lo siguiente: • Día en que se trabajó: Se debe verificar que día de la semana se va a registrar para determinar si el tiempo es normal o dominical. • Horas trabajadas: Se debe calcular cuantas horas se trabajó. • Se debe determinar cuántas horas son normales, cuantas extras y cuantas dominicales, teniendo en cuenta la jornada laboral (Lunes a Viernes 7:30-12:00, 13:30-17:30. Sábados 7:30-13:30). • Se debe actualizar automáticamente el tiempo general trabajado de la actividad. Se debe verificar que los campos están diligenciados, si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error. Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.	

Tarea de ingeniería 33.2:

Tarea de Ingeniería	
Número de tarea: 33.2	Historia de usuario: 33
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto. • La hora de inicio no es mayor que la hora de fin. • La hora no es mayor que la fecha actual. • Se seleccionó un empleado. • Se seleccionó una actividad. • Se seleccionó un proyecto. Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos, ejecuta la acción correspondiente en la capa de lógica.	

Tarea de ingeniería 33.3:

Tarea de Ingeniería	
Número de tarea: 33.3	Historia de usuario: 33
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	

Descripción:

Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual contenga campos necesarios para diligenciar la información, de la siguiente manera:

- Proyecto: Campo de selección obligatorio.
- Actividad, campo de selección obligatorio.
- Empleado: Campo de selección obligatorio.
- Fecha a registrar: Campo de selección obligatorio.
- Hora de inicio: campo de selección obligatorio.
- Hora de fin: campo de selección obligatorio.
- Descripción: Campo de texto.
- Reporte de novedad: Campo de selección.

Inicialmente se mostraran los campos para seleccionar proyecto, al seleccionar el proyecto se cargan las actividades relacionadas a este, al seleccionar una actividad se cargan los empleados asignados a está. Al estar cargados los empleados se podrá seleccionar para registrar el tiempo laborado, formulario que se mostrará en un modal.

Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera:

Cancelar: Cierra el formulario de registro y continua con la lista de empleados asignados a la actividad seleccionada.

Guardar tiene como funcionalidad guardar en la capa de datos la información diligenciada por el usuario. Si el proceso es exitoso muestra un mensaje informativo y oculta el modal, de lo contrario muestra un mensaje de error y deja enfocado el formulario.

Tarea de ingeniería 33.4:

Tarea de Ingeniería	
Número de tarea: 33.4	Historia de usuario: 33
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 30 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	
Se deben realizar pruebas sobre la funcionalidad de registrar tiempo trabajado en actividad, para	

lo cual se debe tener en cuenta la especificación de la tarea de ingeniería [33.2](#) y [33.3](#). Este proceso debe tener las siguientes características:

- Se validan correctamente los datos obligatorios.
- Se validan correctamente los tipos de datos.
- Actualiza adecuadamente los elementos de la interfaz.
- Registra exitosamente el tiempo trabajado.
- Calcula los tiempos correctamente (Normales, extras y dominicales).

Historia de usuario 34: Consultar tiempo trabajado en actividad

Tarea de ingeniería 34.1:

Tarea de Ingeniería	
Número de tarea: 34.1	Historia de usuario: 34
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 30 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad consultar en detalle el tiempo trabajado en una actividad registrada en el sistema. El cual debe recibir como parámetro la actividad a ser consultada, y debe retornar el detalle de todos los tiempos registradas en dicha actividad.	

Tarea de ingeniería 34.2:

Tarea de Ingeniería	
Número de tarea: 34.2	Historia de usuario: 34
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual, por medio de una tabla muestre la información detallada del registro de tiempos de una actividad. En la misma interfaz descrita en la tarea de ingeniería 33.3, al ser seleccionada una actividad se debe ofrecer la opción (Tipo botón) para Consultar detalle. Opción que al ser seleccionada mostrará una tabla en una ventana modal con información como: • Fecha. • Empleado. • Hora de inicio. • Hora de fin. • Descripción. • Botón para modificar. En la parte inferior de la tabla se mostrara un total de horas registradas en la actividad. Además, se contará con un botón en la parte inferior para volver.	

Tarea de ingeniería 34.3:

Tarea de Ingeniería	
Número de tarea: 34.3	Historia de usuario: 34
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 30 Minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de consultar tiempo trabajado en actividad, para lo cual se debe tener en cuenta la especificación de las tareas de ingeniería 33.1 y 34.2 Este proceso debe tener las siguientes características: • Se listan todos registros en el sistema. • Se muestran los registros correspondientes a la actividad. • Los tiempos totales trabajados en la actividad corresponden con la suma de los registros. • Botón de volver funciona correctamente.	

Historia de usuario 35: Modificar tiempo trabajado en actividad

Tarea de ingeniería 35.1:

Tarea de Ingeniería	
Número de tarea: 35.1	Historia de usuario: 35
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad modificar un	

registro de tiempo de una actividad en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados.

Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error.

Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.

Tarea de ingeniería 35.2:

Tarea de Ingeniería	
Número de tarea: 35.2	Historia de usuario: 35
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar cambios verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo especificado en la tarea de ingeniería 33.2 . Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos ejecuta la acción correspondiente en la capa de logica.	

Tarea de ingeniería 35.3:

Tarea de Ingeniería	
Número de tarea: 35.3	Historia de usuario: 35
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 Hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Partiendo de la interfaz descrita en la tarea de ingeniera 34.2 , se tiene la opción de modificar un registro de tiempo trabajado en la actividad. Al seleccionar esta opción se debe mostrar un formulario en la parte superior de la tabla donde se carguen los datos del registro respectivo, formulario en el cual se podrán realizar modificaciones. Además se contará con dos botones de acción: Guardar cambios: Tendrá como funcionalidad ejecutar las respectivas funciones en la capa de vista. Cancelar: Oculta el formulario sin modificar el registro.	

Tarea de ingeniería 35.4:

Tarea de Ingeniería	
Número de tarea: 35.4	Historia de usuario: 35
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de modificar tiempo trabajado en actividad.	

Este proceso debe tener las siguientes características:

- Se valida correctamente que los datos obligatorios no sean nulos.
- Se valida correctamente los tipos de datos ingresados.
- Se carga correctamente la información del registro seleccionado.
- Se guardan correctamente las modificaciones.
- Botones de acción funcionan correctamente.

Historia de usuario 36: Guardar archivo de proyecto

Tarea de ingeniería 36.1:

Tarea de Ingeniería	
Número de tarea: 36.1	Historia de usuario: 36
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad guardar un archivo en el servidor. Se debe verificar que no exista un archivo en el proyecto con el mismo nombre, de ser así se debe retornar una excepción. Los archivos se deben guardar en una carpeta con el mismo nombre del proyecto, es decir: <i>Proyectos/[nombreProyecto]/[nombreArchivo]</i>	

Tarea de ingeniería 36.2:

Tarea de Ingeniería	
Número de tarea: 36.2	Historia de usuario: 36
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Se debe crear un método en la capa de vista el cual tenga como funcionalidad validar que los datos necesarios están debidamente diligenciados, de la siguiente manera: • Se ha seleccionado un archivo. Si no se ha seleccionado un archivo, se debe mostrar en pantalla un mensaje de error especificando que el campo es requerido. Si por el contrario, los datos son correctos ejecuta la acción correspondiente en la capa de lógica.	
Observación:	

Tarea de ingeniería 36.3:

Tarea de Ingeniería	
Número de tarea: 36.3	Historia de usuario: 36
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 40 minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: En la interfaz de modificar proyecto se debe crear una pestaña la cual aparecerá como cuarta opción llamada Archivos la cual contenga campos necesarios para diligenciar la información necesaria para cargar un archivo, de la siguiente manera:	

- Componente para seleccionar un archivo en el equipo.
- Lista de archivos cargados en el proyecto actual.
- Botón **Guardar** el cual tiene como funcionalidad ejecutar el método respectivo en la capa de vista.

Tarea de ingeniería 36.4:

Tarea de Ingeniería	
Número de tarea: 36.4	Historia de usuario: 36
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 30 minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de guardar archivo. Este proceso debe tener las siguientes características: • Es posible seleccionar varios archivos. • Los archivos se guardan correctamente. • Se listan los archivos guardados en el proyecto. • Se muestran mensajes de información correctamente.	

Historia de usuario 37: Listar archivos de proyecto

Tarea de ingeniería 37.1:

Tarea de Ingeniería	
Número de tarea: 37.1	Historia de usuario: 37
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado: 20 minutos
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: En la interfaz de Modificar/Consultar proyecto se debe crear una nueva pestaña “ Archivos ” en la cual se ofrecerá una interfaz gráfica en la que se mostrará en forma de tabla los archivos cargados en el sistema que corresponden a dicho proyecto seleccionado. La tabla tendrá las siguientes campos: • Nombre del archivo. • Fecha en que se guardó el archivo. • Botón para eliminar	

Historia de usuario 38: Eliminar archivos de proyecto

Tarea de ingeniería 38.1:

Tarea de Ingeniería	
Número de tarea: 38.1	Historia de usuario: 38
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	

Descripción:

Se debe crear un método en la capa de lógica, que tenga como funcionalidad eliminar un archivo de un proyecto, eliminando el registro en base de datos y el archivo del servidor.

Si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error.

Tarea de ingeniería 38.2:

Tarea de Ingeniería	
Número de tarea: 38.2	Historia de usuario: 38
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado: 1 hora
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Eliminar obtiene el archivo seleccionado y ejecuta la acción respectiva en la capa de lógica. Si ocurre un error se debe mostrar un mensaje informativo especificando que ocurrió. Si por el contrario, el proceso es exitoso, se debe mostrar un mensaje informativo informando que se eliminó correctamente.	

Tarea de ingeniería 38.3:

Tarea de Ingeniería	
Número de tarea: 38.3	Historia de usuario: 38
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Pruebas	Tiempo Estimado: 30 minutos
Fecha de inicio:	Fecha de fin:



PROYECTO:	Gestión de proyectos			Página 25 de 25
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Programador: Hernán Antonio Ramírez Osorio

Descripción:

Se deben realizar pruebas sobre la funcionalidad de eliminar archivo. Este proceso debe tener las siguientes características:

- Al presionar eliminar se debe mostrar un mensaje de confirmación.
- Se elimina el archivo especificado.
- Se muestran correctamente los mensajes.
- Se actualiza correctamente la interfaz.
- Botones de acción funcionan correctamente.



PROYECTO:	Gestión de proyectos			Página 1 de 24
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tareas de ingeniería

Gestión de proyectos

Tabla de contenido

Historia de usuario 39: Crear ODT.....	4
Tarea de ingeniería 39.1:.....	4
Tarea de ingeniería 39.2:.....	4
Historia de usuario 40: Consultar/modificar ODT	5
Tarea de ingeniería 40.1:.....	5
Tarea de ingeniería 40.2:.....	6
Tarea de ingeniería 40.3:.....	6
Tarea de ingeniería 40.4:.....	7
Historia de usuario 41: Exportar ODT	8
Tarea de ingeniería 41.1:.....	8
Tarea de ingeniería 41.2:.....	8
Tarea de ingeniería 41.3:.....	9
Historia de usuario 42: Exportar cotización	10
Tarea de ingeniería 42.1:.....	10
Tarea de ingeniería 42.2:.....	10
Tarea de ingeniería 42.3:.....	11
Historia de usuario 43: Modificar usuarios	12
Tarea de ingeniería 43.1:.....	12
Tarea de ingeniería 43.2:.....	12
Tarea de ingeniería 43.3:.....	13
Historia de usuario 44: Configurar días no laborables	14
Tarea de ingeniería 44.1:.....	14
Tarea de ingeniería 44.2:.....	14
Tarea de ingeniería 44.3:.....	15
Historia de usuario 45: Configurar jornada laboral.....	16
Tarea de ingeniería 45.1:.....	16
Tarea de ingeniería 45.2:.....	17
Tarea de ingeniería 45.3:.....	17



PROYECTO:	Gestión de proyectos			Página 3 de 24
DOCUMENTO:	Tareas de ingeniería			
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Historia de usuario 46: Registrar tiempo trabajado.....	18
Tarea de ingeniería 46.1:.....	18
Tarea de ingeniería 46.2:.....	19
Tarea de ingeniería 46.3:.....	20
Tarea de ingeniería 46.4:.....	21
Historia de usuario 47: Registro de material usado	21
Tarea de ingeniería 47.1:.....	21
Tarea de ingeniería 47.2:.....	22
Tarea de ingeniería 47.3:.....	23
Tarea de ingeniería 47.4:.....	24

Historia de usuario 39: Crear ODT

Tarea de ingeniería 39.1:

Tarea de Ingeniería	
Número de tarea: 39	Historia de usuario: 39
Nombre tarea: Realizar métodos funcionales.	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad registrar una ODT en el sistema. El cual debe recibir por parámetro el objeto diligenciado por el usuario y verificar que los datos obligatorios cumplan con estar diligenciados. Este método será ejecutado automáticamente al momento de crear un proyecto en el sistema, tomando como datos el nombre y la descripción del proyecto, por defecto la descripción de la ODT es “No disponible” en caso de que no se diligencien los datos completos en el proyecto.	

Tarea de ingeniería 39.2:

Tarea de Ingeniería	
Número de tarea: 39.2	Historia de usuario: 39
Nombre tarea: Pruebas unitarias.	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de crear ODT, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 39.1. El proceso de pruebas debe tener las siguientes características:	

- Verificar que los datos registrados son correctos.
- Verificar que se asigna a la descripción un valor por defecto en caso de no diligenciarse por el usuario.
- Se muestra la información de forma clara para el empleado.

Historia de usuario 40: Consultar/modificar ODT

Tarea de ingeniería 40.1:

Tarea de Ingeniería	
Número de tarea: 40.1	Historia de usuario: 40
Nombre tarea: Realizar métodos funcionales.	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, el cual tenga como funcionalidad guardar cambios sobre una ODT en particular, este método debe recibir como parámetro un registro de tipo ODT, verificando que los campos obligatorios se encuentren debidamente diligenciados y así proceder a guardar los cambios en la capa de datos. Los campos editables son: • Nombre, campo alfanumérico obligatorio. • Descripción, campo alfanumérico obligatorio. Si ocurre algún error en el proceso se debe notificar mediante una excepción la cual será mostrada al usuario, de lo contrario continua el flujo de la aplicación.	

Tarea de ingeniería 40.2:

Tarea de Ingeniería	
Número de tarea: 40.2	Historia de usuario: 40
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe realizar un método en la capa de vista, el cual tenga como función verificar que los datos modificados por el usuario estén debidamente diligenciados, de la siguiente manera: • Nombre: Campo alfanumérico obligatorio. • Descripción: Campo alfanumérico. Si alguno de los campos no cumplen con el requisito se debe mostrar un mensaje de error al usuario, notificando que campo debe ser corregido. De lo contrario continua con el flujo normal de la aplicación.	

Tarea de ingeniería 40.3:

Tarea de Ingeniería	
Número de tarea: 40.3	Historia de usuario: 40
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Partiendo de la interfaz descrita en la historia de usuario Consultar/Modificar proyecto , se debe crear una sección exclusiva para la ODT presentada en forma de pestaña. Se debe mostrar la	

información en forma de grupos de la siguiente manera:

- Información de la orden:
 - Nombre o detalle. Campo editable.
 - Nombre de cliente. Campo editable
 - Nombre de responsable.
- Planeación:
 - Fecha de inicio.
 - Fecha de entrega.
- Descripción.
- Materiales cotizados.
- Empleados encargados de la ejecución.

Se debe mostrar un icono para exportar el formato de ODT.

Los datos serán editables solo si el usuario ha seleccionado modificar, de lo contrario solo se mostraran para lectura.

Tarea de ingeniería 40.4:

Tarea de Ingeniería	
Número de tarea: 40.4	Historia de usuario: 40
Nombre tarea: Pruebas unitarias.	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre las funcionalidades de consultar/modificar ODT, para lo cual se tendrán en cuenta las siguientes características: • Se muestran los datos de manera adecuada como se especifica en la tarea de ingeniería 40.3. • Los campos de nombre y descripción de la ODT son modificables. • Al realizar la modificación de los datos se guardan exitosamente los cambios. • El formato de ODT se exporta de manera exitosa, mostrando los datos especificados.	

Historia de usuario 41: Exportar ODT

Tarea de ingeniería 41.1:

Tarea de Ingeniería	
Número de tarea: 41.1	Historia de usuario: 41
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, el cual tenga como funcionalidad generar y exportar un documento PDF con un formato de ODT, este método debe recibir como parámetro el registro ODT con los datos que se mostraran en el reporte. Se debe retornar un flujo de bytes, el cual será recibido en la capa de vista para iniciar la descarga. Si ocurre algún error en el proceso se debe notificar mediante una excepción la cual será mostrada al usuario, de lo contrario continua el flujo normal de la aplicación.	

Tarea de ingeniería 41.2:

Tarea de Ingeniería	
Número de tarea: 41.2	Historia de usuario: 41
Nombre tarea: Realizar reporte	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe realizar un reporte capaz de ser exportado en formato PDF, que contenga la información especificada para la ODT. Para esta tarea se hará uso del software Ireport.	



PROYECTO:	Gestión de proyectos			Página 9 de 24	
DOCUMENTO:	Tareas de ingeniería				
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio		

Se debe ubicar la información en forma igual, como se especifica en [la tarea de ingeniería 40.3](#).
Mostrando además el icono de la empresa en la parte superior del formato.

Los datos a mostrar, se especificaran en forma de “parameters” o “field” según como lo especifica la herramienta.

Tarea de ingeniería 41.3:

Tarea de Ingeniería	
Número de tarea: 41.3	Historia de usuario: 41
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidades de exportar ODT, para lo cual se tendrán en cuenta la siguientes características: • Al presionar el botón de exportar, se genera automáticamente el PDF y se descarga al equipo. • Los datos presentados en el reporte están agrupados de manera correcta. • Los datos mostrados en el reporte son correctos. • El formato presenta de manera clara la información, además de incluir el logo de la empresa.	

Historia de usuario 42: Exportar cotización

Tarea de ingeniería 42.1:

Tarea de Ingeniería	
Número de tarea: 42.1	Historia de usuario: 42
Nombre tarea: Realizar métodos funcionales.	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, el cual tenga como funcionalidad generar y exportar un documento XLSX con la información específica de una cotización en particular, este método debe recibir como parámetro la entidad de la cotización con los datos que se mostraran en el reporte. Se debe retornar un flujo de bytes, el cual será recibido en la capa de vista para iniciar la descarga. Si ocurre algún error en el proceso se debe notificar mediante una excepción la cual será mostrada al usuario, de lo contrario continua el flujo normal de la aplicación.	

Tarea de ingeniería 42.2:

Tarea de Ingeniería	
Número de tarea: 42.2	Historia de usuario: 42
Nombre tarea: Realizar reporte	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe realizar un reporte capaz de ser exportado en formato XLSX, que contenga la información especificada para la cotización seleccionada. Para esta tarea se hará uso de las	



PROYECTO:	Gestión de proyectos		Página 11 de 24
DOCUMENTO:	Tareas de ingeniería		
VERSIÓN:	1.1	ELABORADO POR:	Hernán Antonio Ramírez Osorio

librerías de Apache POI, capaces de generar archivos de XLSX entre otros.

Se debe ubicar la información separada por grupos: Estimación de tiempo, Materiales, gastos adicionales. Cada sección tendrá la suma total de los elementos cotizados.

En la parte inferior se debe ubicar una sección para mostrar el total de la cotización, donde se sumen todos los elementos descritos.

Tarea de ingeniería 42.3:

Tarea de Ingeniería	
Número de tarea: 42.3	Historia de usuario: 42
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidades de exportar Cotización, para lo cual se tendrán en cuenta la siguientes características: • Al presionar el botón de exportar, se genera automáticamente el archivo XLSX y se descarga al equipo. • Los datos presentados en el archivo están agrupados de manera correcta. • Los datos son iguales a los cotizados en la cotización. • Las sumas de cada sección son correctas. • La suma total de la cotización es correcta. El formato presenta de manera clara la información, además de incluir el logo de la empresa.	

Historia de usuario 43: Modificar usuarios

Tarea de ingeniería 43.1:

Tarea de Ingeniería	
Número de tarea: 43.1	Historia de usuario: 43
Nombre tarea: Realizar métodos funcionales.	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, el cual tenga como funcionalidad guardar cambios sobre un usuario modificado, este método debe recibir como parámetro un registro de tipo Empleado, verificando que los campos obligatorios se encuentren debidamente diligenciados y así proceder a guardar los cambios en la capa de datos. Este método solo modifica los datos para: Roles asignados y permiso de acceso al sistema, esto con el fin de limitar el acceso a la aplicación de algunos usuarios. Si ocurre algún error en el proceso se debe notificar mediante una excepción la cual será mostrada al usuario, de lo contrario continua el flujo normal de la aplicación.	

Tarea de ingeniería 43.2:

Tarea de Ingeniería	
Número de tarea: 43.2	Historia de usuario: 43
Nombre tarea: Realizar interfaz grafica	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual, por medio de una tabla muestre todos los empleados registrados en la aplicación.

Se deben mostrar datos básicos como:

- Identificación.
- Nombre.
- Apellido.
- Permiso de acceso, campo de selección con las opciones SI y NO.
- Opción para asignar o quitar roles al usuario seleccionado.

Además, se contará con botones para aceptar o cancelar los cambios sobre un registro en particular.

Tarea de ingeniería 43.3:

Tarea de Ingeniería	
Número de tarea: 43.3	Historia de usuario: 43
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de modificar usuario, para lo cual se debe tener en cuenta lo siguiente: • Se listan todos los usuarios registrados en el sistema, con la información especificada en la tarea de ingeniería 43.2. • Se permite la modificación de un usuario en particular, asignando permisos de usuarios o roles. • Al seleccionar la asignación de roles se muestran los roles asignados con anterioridad para este usuario. • Las opciones de la aplicación de los roles asignados o removidos son mostradas o removidas según sea el caso. • La información de un usuario modificado se guarda de manera exitosa.	

Historia de usuario 44: Configurar días no laborables

Tarea de ingeniería 44.1:

Tarea de Ingeniería	
Número de tarea: 44.1	Historia de usuario: 44
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben crear métodos en la capa de lógica, el cual tenga como funcionalidad para: Crear, modificar o eliminar un registro de la jornada laboral, este método debe recibir como parámetro un registro de tipo noLaborable, verificando que los campos obligatorios se encuentren debidamente diligenciados y así proceder a guardar los cambios en la capa de datos. Si ocurre algún error en el proceso se debe notificar mediante una excepción la cual será mostrada al usuario, de lo contrario continua el flujo normal de la aplicación.	

Tarea de ingeniería 44.2:

Tarea de Ingeniería	
Número de tarea: 44.2	Historia de usuario: 43
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual, por medio de una tabla muestre todos los días no laborales en la empresa.

Se deben mostrar datos básicos como:

- Fecha no laborable.
- Descripción.
- Día festivo, especifica si el día registrado es festivo o no.
- Opción para modificar el día no laborable.
- Opción para eliminar día no laborable.
- Opción para registrar un día no laborable.

Al presionar la opción registrar, se debe mostrar un formulario con los datos necesarios para registrar un día no laboral:

- Fecha, campo de selección obligatorio.
- Descripción, campo alfanumérico.
- Fecha festiva, campo de selección.

Al presionar la opción modificar, se debe mostrar un formulario diligenciado con los datos del registro seleccionado, en el cual se podrán editar todos los campos descritos y guardar los campos.

Al presionar eliminar se debe mostrar un mensaje de confirmación con las opciones para aceptar o cancelar. Si el usuario presiona aceptar el registro se elimina del sistema, de lo contrario se conserva.

Tarea de ingeniería 44.3:

Tarea de Ingeniería	
Número de tarea: 44.3	Historia de usuario: 44
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe probar la funcionalidad para configurar los días no laborables en el sistema, para lo cual se deben tener en cuenta las siguientes características:

- Se validan que el campo fecha no sea nulo.
- Se hace el registro correcto en el sistema.
- Se carga correctamente la información al seleccionar modificar.
- Se modifica correctamente la información.
- Se muestra mensaje de confirmación al seleccionar eliminar un registro.
- Se elimina correctamente los registros.

Historia de usuario 45: Configurar jornada laboral

Tarea de ingeniería 45.1:

Tarea de Ingeniería	
Número de tarea: 45.1	Historia de usuario: 45
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, el cual tenga como funcionalidad guardar cambios, este método debe recibir como parámetro un registro de horario, verificando que los campos obligatorios se encuentren debidamente diligenciados y así proceder a guardar los cambios en la capa de datos. Los datos a validar son: • Una fecha de inicio no es menor a la fecha de salida. • Si se diligencia la hora de entrada se debe diligenciar la hora de salida para esa jornada. • Si se diligencia la hora de salida se debe diligenciar la hora de entrada para esa jornada. Si ocurre algún error en el proceso se debe notificar mediante una excepción la cual será mostrada al usuario, de lo contrario continua el flujo normal de la aplicación.	

Tarea de ingeniería 45.2:

Tarea de Ingeniería	
Número de tarea: 45.2	Historia de usuario: 45
Nombre tarea: Crear interfaz grafica	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual, por medio de una tabla muestre todos los días laborales de la empresa. Se deben mostrar datos básicos como: • Nombre del día. • Entrada en la mañana. • Salida en la mañana. • Entrada en la tarde. • Salida en la tarde. Además, se contará con un botón para seleccionar un día en particular, el cual, al ser seleccionado debe habilitar los campos para edición, permitiendo la edición de los campos de hora de manera individual.	

Tarea de ingeniería 45.3:

Tarea de Ingeniería	
Número de tarea: 45.3	Historia de usuario: 45
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:

Programador: Hernán Antonio Ramírez Osorio

Descripción:

Se debe probar la funcionalidad para configurar la jornada laboral en el sistema, para lo cual se deben tener en cuenta las siguientes características:

- Se muestran los datos de manera correcta para cada día.
- Los días se muestran de manera ordenada según el día especificado.
- Al seleccionar la opción modificar se habilitan los campos para edición.
- Al realizar modificaciones y guardar los cambios se actualiza la información exitosamente.
- Al realizar cambios y seleccionar cancelar se mantiene la información original.
- Las validaciones se hacen de manera correcta.

Historia de usuario 46: Registrar tiempo trabajado

Tarea de ingeniería 46.1:

Tarea de Ingeniería	
Número de tarea: 46.1	Historia de usuario: 46
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de lógica, que tenga como funcionalidad registrar un tiempo trabajado en una actividad. El cual debe recibir un objeto diligenciado por el usuario y calcular lo siguiente: • Día en que se trabajó: Se debe verificar que día de la semana se va a registrar para determinar si el tiempo es normal o dominical. • Horas trabajadas: Se debe calcular cuantas horas se trabajó. • Se debe verificar el horario laboral para el día a registrar, verificando la jornada definida	

para el día. Si el usuario excede alguna de las jornadas se cuenta como hora extra.

- Se debe actualizar automáticamente el tiempo general trabajado de la actividad.

Se debe verificar que los campos están diligenciados, si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error.

Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.

Tarea de ingeniería 46.2:

Tarea de Ingeniería	
Número de tarea: 46.2	Historia de usuario: 46
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto. • La hora de inicio no es mayor que la hora de fin. • Se seleccionó un empleado. • Se seleccionó una ODT. • Se seleccionó una fecha. • Se seleccionó una hora de inicio. • Se seleccionó una hora de fin. • La descripción no es nula. Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos, ejecuta la acción correspondiente en la capa de lógica.	

Tarea de ingeniería 46.3:

Tarea de Ingeniería	
Número de tarea: 46.3	Historia de usuario: 46
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual muestre los registros realizados en forma de tabla, especificando el empleado, la ODT, el intervalo de tiempo. Además, se debe mostrar la opción para realizar un nuevo registro. Al ser seleccionada se muestra un formulario con los siguientes campos: • ODT: Campo de selección obligatorio. • Empleado, campo de selección obligatorio. • Fecha a registrar: Campo de selección obligatorio. • Novedad, campo de selección. • Hora de inicio: campo de selección obligatorio. • Hora de fin: campo de selección obligatorio. • Descripción: Campo de texto. Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera: Cancelar: Cierra el formulario de registro y muestra la tabla con el detalle de los registros realizados. Guardar tiene como funcionalidad guardar en la capa de datos la información diligenciada por el usuario. Si el proceso es exitoso muestra un mensaje informativo y cierra el modal, de lo contrario muestra un mensaje de error y deja enfocado el formulario.	

Tarea de ingeniería 46.4:

Tarea de Ingeniería	
Número de tarea: 46.4	Historia de usuario: 47
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de registrar tiempo trabajado en actividad, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 46.2 y 46.3 . Este proceso debe tener las siguientes características: • Se validan correctamente los datos obligatorios. • Se validan correctamente los tipos de datos. • Actualiza adecuadamente los elementos de la interfaz. • Registra exitosamente el tiempo trabajado. • Calcula los tiempos correctamente (Normales, extras y dominicales).	

Historia de usuario 47: Registro de material usado**Tarea de ingeniería 47.1:**

Tarea de Ingeniería	
Número de tarea: 47.1	Historia de usuario: 47
Nombre tarea: Realizar métodos funcionales	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción:	

Se debe crear un método en la capa de lógica, que tenga como funcionalidad registrar un material usado en un proyecto. El cual debe recibir un objeto diligenciado por el usuario, validando que los datos sean correctos y descontando el material usado del inventario.

Se debe verificar que los campos están diligenciados, si se encuentra un error se debe retornar una excepción con el mensaje correspondiente a dicho error.

Si el objeto cumple con los requisitos, se procede a guardar en la capa de datos, si ocurre un error en el proceso se debe generar una excepción especificando el error ocurrido.

Tarea de ingeniería 47.2:

Tarea de Ingeniería	
Número de tarea: 47.2	Historia de usuario: 47
Nombre tarea: Realizar validaciones	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear un método en la capa de vista, el cual, al presionar la opción Guardar verifica que los datos ingresados en el formulario son válidos, teniendo en cuenta lo siguiente: • Los datos requeridos no son nulos. • Los datos tienen el tipo y formato correcto. • Se seleccionó una ODT. • Se seleccionó un empleado responsable. • Se seleccionó un material. • Se cantidad no es nula. • Se seleccionó una fecha. • Se seleccionó un proyecto. Si alguno de los datos no cumple con la condición, se debe mostrar en pantalla un mensaje de error especificando que campo se encuentra erróneo. Si por el contrario, los datos son correctos, ejecuta la acción correspondiente en la capa de lógica.	

Tarea de ingeniería 47.3:

Tarea de Ingeniería	
Número de tarea: 47.3	Historia de usuario: 47
Nombre tarea: Crear interfaz gráfica.	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se debe crear una nueva interfaz gráfica en la capa de presentación, la cual muestre los registros realizados en forma de tabla, especificando el empleado, la ODT, la fecha y la cantidad usada. Además, se debe mostrar la opción para realizar un nuevo registro. Al ser seleccionada se muestra un formulario con los siguientes campos: • ODT, campo de selección obligatorio. • Empleado, campo de selección obligatorio. • Material, Campo de selección obligatorio. • Cantidad, campo numérico obligatorio. • Largo, campo numérico, opcional según tipo de material seleccionado. • Ancho, campo numérico, opcional según tipo de material seleccionado. • Fecha, campo de selección obligatorio. • Descripción, campo alfanumérico. Además, se contará con botones de acción en la parte inferior derecha, de la siguiente manera: Cancelar: Cierra el formulario de registro y muestra la tabla con el detalle de los registros realizados. Guardar tiene como funcionalidad guardar en la capa de datos la información diligenciada por el usuario. Si el proceso es exitoso muestra un mensaje informativo y cierra el modal, de lo contrario muestra un mensaje de error y deja enfocado el formulario.	

Tarea de ingeniería 47.4:

Tarea de Ingeniería	
Número de tarea: 47.4	Historia de usuario: 47
Nombre tarea: Pruebas unitarias	
Tipo de tarea: Desarrollo	Tiempo Estimado:
Fecha de inicio:	Fecha de fin:
Programador: Hernán Antonio Ramírez Osorio	
Descripción: Se deben realizar pruebas sobre la funcionalidad de registrar material usado, para lo cual se debe tener en cuenta la especificación de la tarea de ingeniería 47.2 y 47.3. Este proceso debe tener las siguientes características: • Se validan correctamente los datos obligatorios. • Se validan correctamente los tipos de datos. • Actualiza adecuadamente los elementos de la interfaz. • Registra exitosamente el material usado.	

ANEXO 8



PROYECTO:	Gestión de proyectos		Página 1 de 22
DOCUMENTO:	Tarjetas CRC		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

Tarjetas CRC

Gestión de proyectos



PROYECTO:	Gestión de proyectos		Página 2 de 22
DOCUMENTO:	Tarjetas CRC		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

Tabla de contenido

Paquete de Lógica	5
Tarjeta CRC 1: EmpleadoLogicaLocal	5
Tarjeta CRC 2: EmpleadoLogica	5
Tarjeta CRC 3: ProveedorLogicaLocal	5
Tarjeta CRC 4: ProveedorLogica	5
Paquete de Modelo	6
Tarjeta CRC 5: Actividad	6
Tarjeta CRC 6: ActividadCotizacionInventario	6
Tarjeta CRC 7: ActividadEmpleado	6
Tarjeta CRC 8: ArchivoProyecto	7
Tarjeta CRC 9: Cargo	7
Tarjeta CRC 10: Cliente	7
Tarjeta CRC 11: Cotizacion	7
Tarjeta CRC 12: CotizacionInventario	8
Tarjeta CRC 13: CotizacionRiesgoOficio	8
Tarjeta CRC 14: DetalleActividadEmpleado	8
Tarjeta CRC 15: Empleado	8
Tarjeta CRC 16: Empresa	9
Tarjeta CRC 17: GastoAdicional	9
Tarjeta CRC 18: Habilidad	9
Tarjeta CRC 19: Inventario	10
Tarjeta CRC 20: Oficio	10
Tarjeta CRC 21: Proveedor	10
Tarjeta CRC 22: Proyecto	11
Tarjeta CRC 23: Riesgo	11
Paquete de persistencia	11
Tarjeta CRC 24: AbstractFacade	11
Tarjeta CRC 25: ActividadCotizacionInventarioFacadeLocal	11



PROYECTO:	Gestión de proyectos			Página 3 de 22
DOCUMENTO:	Tarjetas CRC			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tarjeta CRC 26: ActividadCotizacionInventarioFacade	12
Tarjeta CRC 27: ActividadEmpleadoFacadeLocal	12
Tarjeta CRC 28: ActividadEmpleadoFacade	12
Tarjeta CRC 29: ActividadFacadeLocal	12
Tarjeta CRC 30: ActividadFacade.....	13
Tarjeta CRC 31: ArchivoProyectoFacadeLocal.....	13
Tarjeta CRC 32: ArchivoProyectoFacade	13
Tarjeta CRC 33: CargoFacadeLocal	13
Tarjeta CRC 34: CargoFacade	13
Tarjeta CRC 35: ClienteFacadeLocal	14
Tarjeta CRC 36: ClienteFacade	14
Tarjeta CRC 37: CotizacionFacadeLocal.....	14
Tarjeta CRC 38: CotizacionFacade	14
Tarjeta CRC 39: CotizacionInventarioFacadeLocal	15
Tarjeta CRC 40: CotizacionInventarioFacade	15
Tarjeta CRC 41: CotizacionRiesgoOficioFacadeLocal.....	15
Tarjeta CRC 42: CotizacionRiesgoOficioFacade.....	15
Tarjeta CRC 43: DetalleActividadEmpleadoFacadeLocal.....	16
Tarjeta CRC 44: DetalleActividadEmpleadoFacade.....	16
Tarjeta CRC 45: EmpleadoFacadeLocal	16
Tarjeta CRC 46: EmpleadoFacade	16
Tarjeta CRC 47: EmpresaFacadeLocal	17
Tarjeta CRC 48: EmpresaFacade.....	17
Tarjeta CRC 49: GastoAdicionalFacadeLocal	17
Tarjeta CRC 50: GastoAdicionalFacade	17
Tarjeta CRC 51: HabilidadFacadeLocal	18
Tarjeta CRC 52: HabilidadFacade	18
Tarjeta CRC 53: InventarioFacadeLocal.....	18
Tarjeta CRC 54: InventarioFacade	18



PROYECTO:	Gestión de proyectos			Página 4 de 22
DOCUMENTO:	Tarjetas CRC			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tarjeta CRC 55: OficioFacadeLocal	19
Tarjeta CRC 56: OficioFacade	19
Tarjeta CRC 57: ProveedorFacadeLocal	19
Tarjeta CRC 58: ProveedorFacade.....	19
Tarjeta CRC 59: ProyectoFacadeLocal.....	20
Tarjeta CRC 60: ProyectoFacade	20
Tarjeta CRC 61: RiesgoFacadeLocal.....	20
Tarjeta CRC 62: RiesgoFacade	20
Paquete Vista	21
Tarjeta CRC 63: EmpleadoVista.....	21
Tarjeta CRC 64: AdministrarEmpleadoVista.....	21
Tarjeta CRC 65: ProveedorVista	21
Tarjeta CRC 66: AdministrarProveedorVista	22
Tarjeta CRC 67: UtilVista	22

Paquete de Lógica

Tarjeta CRC 1: EmpleadoLogicaLocal

EmpleadoLogicaLocal	
Funcionalidades	Colaboraciones
- Definir las funcionalidades disponibles para Empleado. - Servir como interfaz para acceder a funcionalidades.	EmpleadoLogica

Tarjeta CRC 2: EmpleadoLogica

EmpleadoLogica	
Funcionalidades	Colaboraciones
- Implementacion de la interfaz EmpleadoLogicaLocal - Define lógica de negocio para cargo. - Registrar nuevo empleado. - Modificar empleado existente. - Eliminar empleado. - Listar empleados existentes. - Consultar Empleado por ID	- EmpleadoLogicaLocal - Empleado

Tarjeta CRC 3: ProveedorLogicaLocal

ProveedorLogicaLocal	
Funcionalidades	Colaboraciones
- Servir como interfaz para acceder a funcionalidades para proveedores. - Define funcionalidades para gestionar proveedores.	ProveedorLogica.

Tarjeta CRC 4: ProveedorLogica

ProveedorLogica	
Funcionalidades	Colaboraciones
Implementación de interfaz proveedorLogicaLocal.	- ProveedorLogicaLocal - Proveedor

- Define lógica de negocio para cargo.
- Registrar nuevo proveedor.
- Modificar proveedor existente.
- Eliminar proveedor.
- Listar todos los proveedores.
- Consultar proveedor por ID.

Paquete de Modelo

Tarjeta CRC 5: Actividad

Actividad	
Funcionalidades	Colaboraciones
• Entidad que representa un registro de Actividad en la base de datos. • Almacenar datos propios de una Actividad.	•

Tarjeta CRC 6: ActividadCotizacionInventario

ActividadCotizacionInventario	
Funcionalidades	Colaboraciones
• Entidad que representa un registro de la relación ActividadCotizacionInventario en la base de datos. • Almacenar datos del inventario usado en una actividad según lo cotizado.	•

Tarjeta CRC 7: ActividadEmpleado

ActividadEmpleado	
Funcionalidades	Colaboraciones
• Entidad que representa un registro de ActividadEmpleado en la base de datos. • Almacenar datos propios de la	•

asignación de una actividad a un empleado específico.

Tarjeta CRC 8: ArchivoProyecto

ArchivoProyecto	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de ArchivoProyecto en la base de datos. - Almacenar datos propios de un archivo que corresponde a un proyecto.	

Tarjeta CRC 9: Cargo

Cargo	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de Cargo en la base de datos. - Almacenar datos propios de un cargo en específico.	- CargoLogicaLocal - CargoLogica - CargoFacadeLocal - CargoFacade

Tarjeta CRC 10: Cliente

Cliente	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de Cliente en la base de datos. - Almacenar datos propios de un cliente es específico.	- ClienteFacadeLocal - ClienteFacade

Tarjeta CRC 11: Cotizacion

Cotizacion	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de Cotizacion en la base de datos. - Almacenar datos propios de una cotización en específico para ser leída, guardada, actualizada o eliminada.	- CotizacionFacadeLocal - CotizacionFacade

Tarjeta CRC 12: CotizacionInventario

CotizacionInventario	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de CotizacionInventario en la base de datos. - Almacenar datos propios del inventario cotizado en un proyecto en específico para leer, guardar, actualizar o eliminar.	- CotizacionInventarioFacadeLocal - CotizacionInventarioFacade

Tarjeta CRC 13: CotizacionRiesgoOficio

CotizacionRiesgoOficio	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de CotizacionRiesgoOficio en la base de datos. - Almacenar datos propios de los riesgos según el oficio para un proyecto en específico para leer, guardar, actualizar o eliminar.	- CotizacionRiesgoOficioFacadeLocal - CotizacionRiesgoOficioFacade

Tarjeta CRC 14: DetalleActividadEmpleado

DetalleActividadEmpleado	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de DetalleActividadEmpleado en la base de datos. - Almacenar datos propios de la actividad realizada por un empleado en específico para ser leída, guardada, actualizada o eliminada.	- DetalleActividadEmpleadoFacadeLocal - DetalleActividadEmpleadoFacade

Tarjeta CRC 15: Empleado

Empleado	
Funcionalidades	Colaboraciones
Entidad que representa un registro de un Empleado en la base de datos.	- EmpleadoFacadeLocal - EmpleadoFacade

- Almacenar datos propios de un empleado en específico para ser leído, guardado, actualizado o eliminado.

Tarjeta CRC 16: Empresa

Empresa	
Funcionalidades	Colaboraciones
• Entidad que representa un registro de Empresa en la base de datos. • Almacenar datos propios de una cotización en específico para ser Consultada, guardada, actualizada o eliminada.	• EmpresaFacadeLocal • EmpresaFacade

Tarjeta CRC 17: GastoAdicional

GastoAdicional	
Funcionalidades	Colaboraciones
• Entidad que representa un registro de GastoAdicional en la base de datos. • Almacenar datos propios de los gastos adicionales en un proyecto en específico para ser Consultados, guardados, actualizados o eliminados.	• GastoAdicionalFacadeLocal • GastoAdicionalFacade

Tarjeta CRC 18: Habilidad

Habilidad	
Funcionalidades	Colaboraciones
• Entidad que representa un registro de Habilidad en la base de datos. • Almacenar datos propios de una habilidad de un empleado en específico para ser Consultada, guardada, actualizada o eliminada.	• HabilidadFacadeLocal • HabilidadFacade

Tarjeta CRC 19: Inventario

Inventario	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de Inventario en la base de datos. - Almacenar datos propios de un material en inventario para ser Consultado, guardado, actualizado o eliminado.	- InventarioFacadeLocal - InventarioFacade

Tarjeta CRC 20: Oficio

Oficio	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de Oficio en la base de datos. - Almacenar datos propios de un oficio en específico para ser Consultado, guardado, actualizado o eliminado.	- OficioFacadeLocal - OficioFacade

Tarjeta CRC 21: Proveedor

Proveedor	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de Proveedor en la base de datos. - Almacenar datos propios de un proveedor en específico para ser Consultado, guardado, actualizado o eliminado.	- ProveedorFacadeLocal - ProveedorFacade

Tarjeta CRC 22: Proyecto

Proyecto	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de Proyecto en la base de datos. - Almacenar datos propios de un proyecto en específico para ser Consultado, guardado, actualizado o eliminado.	- ProyectoFacadeLocal - ProyectoFacade

Tarjeta CRC 23: Riesgo

Riesgo	
Funcionalidades	Colaboraciones
- Entidad que representa un registro de Riesgo en la base de datos. - Almacenar datos propios de un Riesgo en específico para ser Consultado, guardado, actualizado o eliminado.	- RiesgoFacadeLocal - RiesgoFacade

Paquete de persistencia

Tarjeta CRC 24: AbstractFacade

AbstractFacade	
Funcionalidades	Colaboraciones
Clase abstracta que implementa los métodos más comunes para ser usados en la persistencia de la aplicación. Crear, Editar, Eliminar, Consultar, etc.	

Tarjeta CRC 25: ActividadCotizacionInventarioFacadeLocal

ActividadCotizacionInventarioFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para ActividadCotizacionInventario. - Servir como fachada para facilidad de	- ActividadCotizacionInventario - ActividadCotizacionInventarioFacade

uso en persistencia.

Tarjeta CRC 26: ActividadCotizacionInventarioFacade

ActividadCotizacionInventarioFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz de ActividadCotizacionInventarioFacadeLocal. - Se define la implementación de las funcionalidades.	ActividadCotizacionInventario

Tarjeta CRC 27: ActividadEmpleadoFacadeLocal

ActividadEmpleadoFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para ActividadEmpleado. - Servir como fachada para facilidad de uso en persistencia.	- ActividadEmpleado - ActividadEmpleadoFacade

Tarjeta CRC 28: ActividadEmpleadoFacade

ActividadEmpleadoFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz de ActividadEmpleadoFacadeLocal. - Se define la implementación de las funcionalidades.	ActividadEmpleado

Tarjeta CRC 29: ActividadFacadeLocal

ActividadFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Actividad. - Servir como fachada para facilidad de uso en persistencia.	- ActividadFacade - Actividad

Tarjeta CRC 30: ActividadFacade

ActividadFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz de ActividadFacadeLocal. - Se define la implementación de las funcionalidades.	Actividad

Tarjeta CRC 31: ArchivoProyectoFacadeLocal

ArchivoProyectoFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para ArchivoProyecto. - Servir como fachada para facilidad de uso en persistencia.	- ArchivoProyectoFacade - ArchivoProyecto

Tarjeta CRC 32: ArchivoProyectoFacade

ArchivoProyectoFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz ArchivoProyectoFacadeLocal - Se define la implementación de las funcionalidades.	ArchivoProyecto

Tarjeta CRC 33: CargoFacadeLocal

CargoFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Cargo. - Servir como fachada para facilidad de uso en persistencia.	- CargoFacade - Cargo

Tarjeta CRC 34: CargoFacade

CargoFacade

Funcionalidades	Colaboraciones
- Implementa la interfaz CargoFacadeLocal. - Se define la implementación de las funcionalidades.	Cargo

Tarjeta CRC 35: ClienteFacadeLocal

ClienteFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Cliente. - Servir como fachada para facilidad de uso en persistencia.	- ClienteFacade - Cliente

Tarjeta CRC 36: ClienteFacade

ClienteFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz ClienteFacadeLocal. - Se define la implementación de las funcionalidades.	Cliente

Tarjeta CRC 37: CotizacionFacadeLocal

CotizacionFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Cotizacion. - Servir como fachada para facilidad de uso en persistencia.	- CotizacionFacade. - Cotizacion.

Tarjeta CRC 38: CotizacionFacade

CotizacionFacade	
Funcionalidades	Colaboraciones
Implementa la interfaz CotizacionFacadeLocal.	Cotizacion.

- Se define la implementación de las funcionalidades.

Tarjeta CRC 39: CotizacionInventarioFacadeLocal

CotizacionInventarioFacadeLocal	
Funcionalidades	Colaboraciones
• Define las funcionalidades que se podrán realizar para CotizacionInventario. • Servir como fachada para facilidad de uso en persistencia.	• CotizacionInventarioFacade. • CotizacionInventario.

Tarjeta CRC 40: CotizacionInventarioFacade

CotizacionInventarioFacade	
Funcionalidades	Colaboraciones
• Implementa la interfaz CotizacionInventarioFacadeLocal. • Se define la implementación de las funcionalidades.	• CotizacionInventario.

Tarjeta CRC 41: CotizacionRiesgoOficioFacadeLocal

CotizacionRiesgoOficioFacadeLocal	
Funcionalidades	Colaboraciones
• Define las funcionalidades que se podrán realizar para CotizacionRiesgooficio. • Servir como fachada para facilidad de uso en persistencia.	• CotizacionRiesgoOficioFacade • CotizacionRiesgoOficio

Tarjeta CRC 42: CotizacionRiesgoOficioFacade

CotizacionRiesgoOficioFacade	
Funcionalidades	Colaboraciones
• Implementa la interfaz	• CotizacionRiesgoOficio.

CotizacionRiesgoOficioFacadeLocal.

- Se define la implementación de las funcionalidades.

Tarjeta CRC 43: DetalleActividadEmpleadoFacadeLocal

DetalleActividadEmpleadoFacadeLocal	
Funcionalidades	Colaboraciones
• Define las funcionalidades que se podrán realizar para DetalleActividadEmpleado. • Servir como fachada para facilidad de uso en persistencia.	• DetalleActividadEmpleadoFacade • DetalleActividadEmpleado

Tarjeta CRC 44: DetalleActividadEmpleadoFacade

DetalleActividadEmpleadoFacade	
Funcionalidades	Colaboraciones
• Implementa la interfaz DetalleActividadEmpleadoFacadeLocal. • Se define la implementación de las funcionalidades.	• DetalleActividadEmpleado.

Tarjeta CRC 45: EmpleadoFacadeLocal

EmpleadoFacadeLocal	
Funcionalidades	Colaboraciones
• Define las funcionalidades que se podrán realizar para Empleado. • Servir como fachada para facilidad de uso en persistencia.	• EmpleadoFacade. • Empleado.

Tarjeta CRC 46: EmpleadoFacade

EmpleadoFacade	
Funcionalidades	Colaboraciones
• Implementa la interfaz EmpleadoFacadeLocal. • Se define la implementación de las	• Empleado.

funcionalidades.	

Tarjeta CRC 47: EmpresaFacadeLocal

EmpresaFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Empresa. - Servir como fachada para facilidad de uso en persistencia.	- EmpresaFacade. - Empresa.

Tarjeta CRC 48: EmpresaFacade

EmpresaFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz EmpresaFacadeLocal. - Se define la implementación de las funcionalidades.	Empresa.

Tarjeta CRC 49: GastoAdicionalFacadeLocal

GastoAdicionalFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para GastoAdicional. - Servir como fachada para facilidad de uso en persistencia.	- GastoAdicionalFacade. - GastoAdicional.

Tarjeta CRC 50: GastoAdicionalFacade

GastoAdicionalFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz de GastoAdicionalFacadeLocal. - Se define la implementación de las funcionalidades.	GastoAdicional.

Tarjeta CRC 51: HabilidadFacadeLocal

HabilidadFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Habilidad. - Servir como fachada para facilidad de uso en persistencia.	- HabilidadFacade. - Habilidad.

Tarjeta CRC 52: HabilidadFacade

HabilidadFacadeLocal	
Funcionalidades	Colaboraciones
- Implementa la interfaz de HabilidadFacadeLocal. - Se define la implementación de las funcionalidades.	Habilidad.

Tarjeta CRC 53: InventarioFacadeLocal

InventarioFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Inventario. - Servir como fachada para facilidad de uso en persistencia.	- InventarioFacade. - Inventario.

Tarjeta CRC 54: InventarioFacade

InventarioFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz de InventarioFacadeLocal. - Se define la implementación de las funcionalidades.	Inventario.

Tarjeta CRC 55: OficioFacadeLocal

OficioFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Oficio. - Servir como fachada para facilidad de uso en persistencia.	- OficioFacade. - Oficio.

Tarjeta CRC 56: OficioFacade

OficioFacadeLocal	
Funcionalidades	Colaboraciones
- Implementa la interfaz de OficioFacadeLocal. - Se define la implementación de las funcionalidades.	Oficio.

Tarjeta CRC 57: ProveedorFacadeLocal

ProveedorFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Proveedor. - Servir como fachada para facilidad de uso en persistencia.	- ProveedorFacade. - Proveedor.

Tarjeta CRC 58: ProveedorFacade

ProveedorFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz de ProveedorFacadeLocal. - Se define la implementación de las funcionalidades.	Proveedor.

Tarjeta CRC 59: ProyectoFacadeLocal

ProyectoFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Proyecto. - Servir como fachada para facilidad de uso en persistencia.	- ProyectoFacade. - Proyecto.

Tarjeta CRC 60: ProyectoFacade

ProyectoFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz de ProyectoFacadeLocal. - Se define la implementación de las funcionalidades.	Proyecto

Tarjeta CRC 61: RiesgoFacadeLocal

RiesgoFacadeLocal	
Funcionalidades	Colaboraciones
- Define las funcionalidades que se podrán realizar para Riesgo. - Servir como fachada para facilidad de uso en persistencia.	- RiesgoFacade. - Riesgo.

Tarjeta CRC 62: RiesgoFacade

RiesgoFacade	
Funcionalidades	Colaboraciones
- Implementa la interfaz de RiesgoFacadeLocal. - Se define la implementación de las funcionalidades.	Riesgo.

Paquete Vista

Tarjeta CRC 63: EmpleadoVista

EmpleadoVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en crear y listar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para crear y listar empleados. - Verificar valides de datos.	- EmpleadoLogicaLocal - CargoLogicaLocal

Tarjeta CRC 64: AdministrarEmpleadoVista

AdministrarEmpleadoVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en consultar o modificar. - Almacenar toda la información diligenciada por el usuario. - Administrar la funcionalidad para modificar empleado. - Verificar valides de datos.	- EmpleadoLogicaLocal - CargoLogicaLocal

Tarjeta CRC 65: ProveedorVista

ProveedorVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de Proveedor en crear y listar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para crear y listar proveedores. - Verificar valides de datos.	ProveedorLogicaLocal

Tarjeta CRC 66: AdministrarProveedorVista

AdministrarProveedorVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de Proveedor en Consultar y modificar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para crear y listar proveedores. - Verificar valides de datos.	ProveedorLogicaLocal

Tarjeta CRC 67: UtilVista

UtilVista	
Funcionalidades	Colaboraciones
- Implementa métodos comunes en la vista. - Subir variables a sesión. - Leer variables en sesión. - Eliminar variables en sesión. - Mostrar mensajes de error. - Validar formato de correos electrónicos. - Verificar si una cadena de texto es nula	



PROYECTO:	Gestión de proyectos			Página 1 de 9
DOCUMENTO:	Tarjetas CRC			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tarjetas CRC

Gestión de proyectos



PROYECTO:	Gestión de proyectos		Página 2 de 9
DOCUMENTO:	Tarjetas CRC		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

Tabla de contenido

Paquete de lógica:	3
Tarjeta CRC 1: ClienteLogicaLocal	3
Tarjeta CRC 2: ClienteLogica.....	3
Tarjeta CRC 3: CotizacionRiesgoOficioLogicaLocal.....	3
Tarjeta CRC 4: CotizacionRiesgoOficioLogica	3
Tarjeta CRC 5: EmpresaLogicaLocal.....	3
Tarjeta CRC 6: EmpresaLogica	4
Tarjeta CRC 7: InventarioLogicaLocal	4
Tarjeta CRC 8: InventarioLogica	4
Tarjeta CRC 9: OficioLogicaLocal	4
Tarjeta CRC 10: OficioLogica	5
Tarjeta CRC 11: ProyectoLogicaLocal	5
Tarjeta CRC 12: ProyectoLogica	5
Tarjeta CRC 13: RiesgoLogicaLocal	5
Tarjeta CRC 14: RiesgoLogica	5
Paquete de vista:	6
Tarjeta CRC 15: ClienteVista.....	6
Tarjeta CRC 16: AdministrarClienteVista.....	6
Tarjeta CRC 17: InventarioVista	6
Tarjeta CRC 18: InventarioModel.....	7
Tarjeta CRC 19: AdministrarInventarioVista	7
Tarjeta CRC 20: EmpresaVista	7
Tarjeta CRC 21: OficioVista.....	7
Tarjeta CRC 22: RiesgoVista	8
Tarjeta CRC 23: ProyectoVista.....	8
Tarjeta CRC 24: ProyectoModel	8

Paquete de lógica:

Tarjeta CRC 1: ClienteLogicaLocal

ClienteLogicaLocal	
Funcionalidades	Colaboraciones
- Definir las funcionalidades disponibles para entidad Cliente. - Servir como interfaz para acceder a funcionalidades.	ClienteLogica

Tarjeta CRC 2: ClienteLogica

ClienteLogica	
Funcionalidades	Colaboraciones
- Implementacion de la interfaz ClienteLogicaLocal - Define lógica de negocio para cargo.	- ClienteLogicaLocal - Cliente

Tarjeta CRC 3: CotizacionRiesgoOficioLogicaLocal

CotizacionRiesgoOficioLogicaLocal	
Funcionalidades	Colaboraciones
- Definir las funcionalidades disponibles para la entidad encargada de controlar la cotización de la mano de obra. - Servir como interfaz para acceder a funcionalidades.	CotizacionRiesgoOficioLogica

Tarjeta CRC 4: CotizacionRiesgoOficioLogica

CotizacionRiesgoOficioLogica	
Funcionalidades	Colaboraciones
- Implementación de la interfaz ClienteLogicaLocal - Define lógica de negocio para registros de mano de obra. - Registra un tiempo de mano de obra. - Modifica un tiempo de mano de obra. - Elimina un tiempo de mano de obra.	- CotizacionRiesgoOficioLogicaLocal - CotizacionRiesgoOficio

Tarjeta CRC 5: EmpresaLogicaLocal

EmpresaLogicaLocal	

Funcionalidades	Colaboraciones
- Definir las funcionalidades disponibles para la entidad empresa. - Servir como interfaz para acceder a funcionalidades.	- EmpresaLogica - Empresa

Tarjeta CRC 6: EmpresaLogica

EmpresaLogica	
Funcionalidades	Colaboraciones
- Implementacion de la interfaz EmpresaLogicaLocal. - Define lógica de negocio para empresa.	- EmpresaLogicaLocal - Empresa

Tarjeta CRC 7: InventarioLogicaLocal

InventarioLogicaLocal	
Funcionalidades	Colaboraciones
- Definir las funcionalidades disponibles para la entidad inventario. - Servir como interfaz para acceder a funcionalidades.	- InventarioLogica - Inventario

Tarjeta CRC 8: InventarioLogica

InventarioLogica	
Funcionalidades	Colaboraciones
- Implementación de la interfaz InventarioLogicaLocal. - Define lógica de negocio para inventario.	- InventarioLogicaLocal. - Inventario.

Tarjeta CRC 9: OficioLogicaLocal

OficioLogicaLocal	
Funcionalidades	Colaboraciones
- Definir las funcionalidades disponibles para entidad Oficio. - Servir como interfaz para acceder a funcionalidades.	- OficioLogica. - Oficio.

Tarjeta CRC 10: OficioLogica

OficioLogica	
Funcionalidades	Colaboraciones
- Implementación de la interfaz OficioLogicaLocal. - Define lógica de negocio para oficio.	- OficioLogicaLocal. - Oficio.

Tarjeta CRC 11: ProyectoLogicaLocal

ProyectoLogicaLocal	
Funcionalidades	Colaboraciones
- Definir las funcionalidades disponibles para la entidad Proyecto. - Servir como interfaz para acceder a funcionalidades.	- ProyectoLogica. - Proyecto.

Tarjeta CRC 12: ProyectoLogica

ProyectoLogica	
Funcionalidades	Colaboraciones
- Implementación de la interfaz ProyectoLogicaLocal. - Define lógica de negocio para proyecto.	- ProyectoLogicaLocal. - Proyecto.

Tarjeta CRC 13: RiesgoLogicaLocal

RiesgoLogicaLocal	
Funcionalidades	Colaboraciones
- Definir las funcionalidades disponibles para la entidad de riesgo. - Servir como interfaz para acceder a funcionalidades.	- RiesgoLogica - Riesgo.

Tarjeta CRC 14: RiesgoLogica

RiesgoLogica	
Funcionalidades	Colaboraciones
- Implementación de la interfaz RiesgoLogicaLocal. - Define lógica de negocio para las	- RiesgoLogicaLocal. - Riesgo.

entidades de riesgo.

Paquete de vista:

Tarjeta CRC 15: ClienteVista

ClienteVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en crear y listar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para crear y listar empleados.	- EmpresaLogicaLocal - ClienteLogicaLocal.

Tarjeta CRC 16: AdministrarClienteVista

AdministrarClienteVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en consultar y modificar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para consultar y modificar empleados.	- EmpresaLogicaLocal - ClienteLogicaLocal.

Tarjeta CRC 17: InventarioVista

InventarioVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en crear y listar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para crear y listar inventario.	- InventarioLogicaLocal. - InventarioModel. - ProveedorLogicaLocal. - CategoriaLogicaLocal.

Tarjeta CRC 18: InventarioModel

InventarioVista	
Funcionalidades	Colaboraciones
Servir como contenedor de variables para la información diligenciada por el usuario.	- Inventario. - Proveedor - Categoría.

Tarjeta CRC 19: AdministrarInventarioVista

AdministrarInventarioVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en los detalles del inventario. - Almacenar toda la información diligenciada por el usuario.	- Inventario. - Proveedor - Categoría.

Tarjeta CRC 20: EmpresaVista

EmpresaVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en crear, listar y modificar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para crear, listar y modificar empresa.	Empresa.

Tarjeta CRC 21: OficioVista

OficioVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en crear, listar y modificar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para crear, listar y modificar oficios.	Oficio

Tarjeta CRC 22: RiesgoVista

RiesgoVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en crear, listar y actualizar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para crear, listar y actualizar riesgos.	Riesgo.

Tarjeta CRC 23: ProyectoVista

ProyectoVista	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de usuario en crear, consultar modificar y listar. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para en crear, consultar modificar y listar empleados. - Crear y modificar cotización de proyecto. - Crear y modificar ODT de proyecto. - Consultar, descargar y subir archivos de proyecto.	- ProyectoLogicaLocal - ClienteLogicaLocal - EmpleadoLogicaLocal - InventarioLogicaLocal - OficioLogicaLocal - RiesgoLogicaLocal - CotizacionLogicaLocal - ArchivoProyectoLogicaLocal - ODTLogicaLocal - CotizacionInventarioLogicaLocal - CotizacionInventario - ProyectoModel

Tarjeta CRC 24: ProyectoModel

AdministrarProyectoVista	
Funcionalidades	Colaboraciones
Servir en almacenamiento de variables relacionadas al proyecto.	- Proyecto - Cliente - Empleado - Inventario - CotizacionInventario - GastoAdicional



PROYECTO:	Gestión de proyectos			Página 9 de 9
DOCUMENTO:	Tarjetas CRC			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

	CotizacionRiesgoOficio



PROYECTO:	Gestión de proyectos			Página 1 de 6
DOCUMENTO:	Tarjetas CRC			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tarjetas CRC

Gestión de proyectos



PROYECTO:	Gestión de proyectos			Página 2 de 6
DOCUMENTO:	Tarjetas CRC			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Paquete de lógica:	3
Tarjeta CRC 1: ActividadLogicaLocal.....	3
Tarjeta CRC 2: ActividadLogica	3
Tarjeta CRC 3: ArchivoProyectoLogicaLocal	3
Tarjeta CRC 4: ArchivoProyectoLogica	3
Tarjeta CRC 5: CargoLogicaLocal	4
Tarjeta CRC 6: CargoLogica.....	4
Tarjeta CRC 7: DetalleActividadEmpleadoLogicaLocal	4
Tarjeta CRC 8: DetalleActividadEmpleadoLogica	4
Paquete de vista:	5
Tarjeta CRC 9: CargoVista.....	5
Tarjeta CRC 10: ProyectoVista.....	5
Tarjeta CRC 11: RegistrarTiemposVista.....	6

Paquete de lógica:

Tarjeta CRC 1: ActividadLogicaLocal

CargoLogicaLocal	
Funcionalidades	Colaboraciones
- Definir funcionalidades disponibles para Actividades. - Servir como interfaz para acceder a funcionalidades.	

Tarjeta CRC 2: ActividadLogica

CargoLogica	
Funcionalidades	Colaboraciones
- Implementación de la interfaz CargoLogicaLocal - Define lógica de negocio para cargo.	

Tarjeta CRC 3: ArchivoProyectoLogicaLocal

ArchivoProyectoLogicaLocal	
Funcionalidades	Colaboraciones
- Definir funcionalidades disponibles para Archivos. - Servir como interfaz para acceder a funcionalidades.	ArchivoProyecto.

Tarjeta CRC 4: ArchivoProyectoLogica

ArchivoProyectoLogica	
Funcionalidades	Colaboraciones
- Implementación de la interfaz CargoLogicaLocal - Define lógica de negocio para los archivos de un proyecto. - Guardar registro de nuevo archivo almacenando nombre y ruta. - Eliminar registro de archivo. - Listar archivos existentes en un proyecto.	- ArchivoProyectoFacadeLocal. - ArchivoProyectoLogicaLocal. - ArchivoProyecto.

Tarjeta CRC 5: CargoLogicaLocal

CargoLogicaLocal	
Funcionalidades	Colaboraciones
- Definir funcionalidades disponibles para Cargo. - Servir como interfaz para acceder a funcionalidades.	Cargo

Tarjeta CRC 6: CargoLogica

CargoLogica	
Funcionalidades	Colaboraciones
- Implementación de la interfaz CargoLogicaLocal - Define lógica de negocio para cargo. - Registrar nuevo cargo - Modificar cargo existente - Eliminar cargo existente - Listar cargos existentes - Consultar cargo por ID	- CargoLogicaLocal - Cargo

Tarjeta CRC 7: DetalleActividadEmpleadoLogicaLocal

DetalleActividadEmpleadoLogicaLocal	
Funcionalidades	Colaboraciones
- Definir funcionalidades disponibles para DetalleActividadEmpleado. - Servir como interfaz para acceder a funcionalidades.	DetalleActividadEmpleado

Tarjeta CRC 8: DetalleActividadEmpleadoLogica

DetalleActividadEmpleadoLogica	
Funcionalidades	Colaboraciones
- Implementación de la interfaz DetalleActividadEmpleadoLogicaLocal. - Define lógica de negocio para funciones de registro de tiempos en una actividad. - Registrar tiempo trabajado en actividad. - Calcula horas trabajadas respecto a	- DetalleActividadEmpleadoLogicaLocal. - DetalleActividadEmpleado. - DetalleActividadEmpleadoFacadeLocal. - ActividadEmpleadoFacadeLocal.



PROYECTO:	Gestión de proyectos		Página 5 de 6
DOCUMENTO:	Tarjetas CRC		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

- hora de inicio y hora de fin.
- Verificar que día se trabajó (Normal, Sábado o Domingo).

Paquete de vista:

Tarjeta CRC 9: CargoVista

CargoVista	
Funcionalidades	Colaboraciones
• Controlar el estado de la interfaz gráfica de cargo en crear, listar, consultar y modificar. • Almacenar toda la información diligenciada por el usuario. • Administrar las funcionalidades para crear, listar, consultar y modificar. • Verificar valides de datos. • Comunicar interfaz con lógica de negocio.	• Cargo. • UtilVista. • CargoLogicaLocal.

Tarjeta CRC 10: ProyectoVista

ProyectoVista	
Funcionalidades	Colaboraciones
• Controlar el estado de la interfaz gráfica de Proyecto. • Administra funcionalidades propias del proyecto en información general, cotización, actividades, archivos. • Copiar archivo seleccionado en el servidor. • Guardar registro de archivo en base de datos. • Descargar archivos del sistema a equipo del usuario. • Eliminar un archivo de proyecto.	• ArchivoProyectoLogicaLocal.

Tarjeta CRC 11: RegistrarTiemposVista

RegistrarTiempos	
Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de registrar tiempos de actividad. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para registrar, modificar y eliminar tiempos de trabajo. - Verificar valides de datos. - Comunicar interfaz con lógica de negocio.	- Proyecto. - Actividad. - ActividadEmpleado. - DetalleActividadEmpleado. - ProyectoLogicaLocal. - ActividadLogicaLocal. - DetalleActividadEmpleadoLogicaLocal.



PROYECTO:	Gestión de proyectos			Página 1 de 4
DOCUMENTO:	Tarjetas CRC			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tarjetas CRC

Gestión de proyectos



PROYECTO:	Gestión de proyectos			Página 2 de 4
DOCUMENTO:	Tarjetas CRC			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Paquete de lógica:	3
GenerarReporteXLS:	3
UsuarioVista:	3
UsuarioModel:	3
SeguridadVista:	3
SeguridadModel:	4
RegistroMaterialVista:	4
RegistroMaterialModelo:	4

Paquete de lógica:

GenerarReporteXLS:

Funcionalidades	Colaboraciones
- Generar reportes en formato XLS para ser descargados posteriormente. - Generar array de bytes para ser descargados posteriormente por el usuario.	Librería Apache POI

UsuarioVista:

Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica de la administración de usuarios para acceso al sistema. - Almacenar toda la información diligenciada por el usuario. - Administrar las funcionalidades para modificar y listar. - Comunicar interfaz con lógica de negocio.	- EmpleadoLogicaLocal - Empleado. - RolLogicaLocal - Rol - UsuarioModel

UsuarioModel:

Funcionalidades	Colaboraciones
- Almacenar la información que será mostrada en la interfaz de administración de usuarios. - Controlar el estado de la interfaz.	- Empleado. - Rol.

SeguridadVista:

Funcionalidades	Colaboraciones
- Controlar el estado de la interfaz gráfica en la creación, modificación de roles. - Almacenar toda la información diligenciada por el usuario.	RolLogicaLocal.

- Administrar las funcionalidades para crear, modificar y listar.
- Comunicar interfaz con lógica de negocio.

SeguridadModel:

Funcionalidades	Colaboraciones
• Almacenar la información que será mostrada en la interfaz de administración de usuarios. • Controlar el estado de la interfaz.	• Rol

RegistroMaterialVista:

Funcionalidades	Colaboraciones
• Controlar el estado de la interfaz gráfica en la creación, modificación y eliminar de registro de materiales. • Almacenar toda la información diligenciada por el usuario. • Administrar las funcionalidades para crear, modificar, eliminar y listar. • Comunicar interfaz con lógica de negocio.	• ODTLogicaLocal • EmpleadoLogicaLocal • CotizacionInventarioLogicaLocal • ODTcotizacionInventarioLogicaLocal • RegistroMaterialModelo

RegistroMaterialModelo:

Funcionalidades	Colaboraciones
• Almacenar la información que será mostrada en la interfaz de administración de usuarios. • Controlar el estado de la interfaz.	• Odt • Empleado • CotizacionInventario • ODTcotizacionInventario

ANEXO 9



PROYECTO:	Gestión de proyectos			Página 1 de 9
DOCUMENTO:	Base de datos			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Base de datos

Gestión de proyectos



PROYECTO:	Gestión de proyectos		Página 2 de 9
DOCUMENTO:	Base de datos		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

Tabla de contenido

ActividadEmpleado	3
ArchivoProyecto:	3
Cargo:	3
Categoría:	3
Cliente:	4
Contrato:	4
Cotizacion:	4
CotizacionInventario:	4
CotizacionRiesgoOficio:	5
DetalleActividadEmpleado:	5
Empleado:	5
Empresa:	5
GastoAdicional:	6
HorarioLaboral:	6
Inventario:	6
ItemAcceso:	6
NoLaborables:	7
Novedad:	7
ODTCotizacionInventario:	7
ODT:	7
Oficio:	8
PermisoAcceso:	8
PermisoAccesoRol:	8
Proveedor:	8
Proyecto:	9
Riesgo:	9
Rol:	9
RolEmpleado:	9



PROYECTO:	Gestión de proyectos		Página 3 de 9
DOCUMENTO:	Base de datos		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

ActividadEmpleado:

ActividadEmpleado
Descripción: Tabla encargada de almacenar el consolidado de horas trabajadas en un proyecto, tiene en cuenta el empleado, la ODT, las horas normales, extras y festivas. Además de indicar si es un empleado encargado de la ejecución del proyecto.

ArchivoProyecto:

ArchivoProyecto
Descripción: Tabla encargada de almacenar la información de los archivos cargados para un proyecto, tiene en cuenta el proyecto, el nombre del archivo y la ruta en el servidor del archivo.

Cargo:

Cargo
Descripción: Tabla encargada de almacenar la información de los cargos en la empresa, los cuales se asignaran a un empleado cuando inicie un contrato. Se tiene en cuenta información como: Nombre del cargo y descripción del cargo.

Categoria:

Categoria
Descripción: Tabla encargada de almacenar la información general de los materiales del inventario, esta información será usada para el inventario, se tiene en cuenta la siguiente información: Nombre de la categoría, unidad de medida de la categoría (und, mm,cm,mtrs), porcentaje adicional para venta del material, porcentaje adicional para costo del material, tipo de material (Inoxidable, hierro, galvanizado, aluminio, entre otros), longitudes del material (1 longitud o dos longitudes).



PROYECTO:	Gestión de proyectos		Página 4 de 9
DOCUMENTO:	Base de datos		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

Cliente:

Cliente

Descripción:

Tabla encargada de almacenar la información relacionada a los clientes de la empresa, los cuales contrataran proyectos. Se tiene en cuenta la siguiente información: Empresa para la cual trabaja el cliente, nombre del cliente, apellido, dirección, ciudad, correo y teléfono.

Contrato:

Contrato

Descripción:

Tabla encargada de almacenar la información relacionada a un contrato de un empleado con la empresa. Se tiene en cuenta la siguiente información: Empleado contratado, cargo asignado, sueldo a pagar, fecha de contratación, fecha fin de contratación, término del contrato, tipo contrato, estado del contrato y una descripción para guardar observaciones relevantes.

Cotizacion:

Cotizacion

Descripción:

Tabla encargada de agrupar los elementos cotizados en un proyecto, como mano de obra, materiales o inventario. Se tiene en cuenta información como: Fecha de creación de la cotización, precio total de la cotización, proyecto al que se realiza la cotización.

CotizacionInventario:

CotizacionInventario

Descripción:

Tabla encargada de almacenar la información de los materiales cotizados para un proyecto, para lo cual se tiene en cuenta lo siguiente: Material cotizado, cantidad cotizada, largo y ancho cotizados, precio del material cotizado, alerta de existencia en caso de que el material no se encuentre en la empresa, alerta de precio antiguo en caso de que se cote sobre precios muy antiguos en una cotización, alerta de existencia incompleta en caso de que la existencia no alcance para lo cotizado. Además de un campo que indica si el material fue incluido en la cotización desde el inicio.



PROYECTO:	Gestión de proyectos		Página 5 de 9
DOCUMENTO:	Base de datos		
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio

CotizacionRiesgoOficio:

CotizacionRiesgoOficio

Descripción:

Tabla encargada de almacenar la información referente a la mano de obra cotizada para un proyecto, se manejan registros individuales según riesgo, oficio y cantidad de horas, además de calcular el precio del elemento cotizado.

DetalleActividadEmpleado:

DetalleActividadEmpleado

Descripción:

Tabla encargada de almacenar la información referente a los registros de tiempo trabajado por parte de los empleados, almacenando en detalle cada actividad realizada por un empleado. Se tiene en cuenta la siguiente información: Novedad en caso de que exista, fecha en que se realizó la actividad, hora de inicio y hora de fin en que se realizó la actividad y la descripción del trabajo realizado.

Empleado:

Empleado

Descripción:

Tabla encargada de almacenar la información referente a los empleados de la empresa, se almacena la siguiente información: Número de identificación, nombre y apellido del empleado, teléfono, contraseña para acceder a la aplicación, habilidades del empleado y un atributo que define el empleado tiene acceso al sistema.

Empresa:

Empresa

Descripción:

Tabla encargada de almacenar la información referente a las empresas clientes, la cual será asignada a diferentes clientes. Se tiene en cuenta la siguiente información: nombre de la empresa y ciudad donde se encuentra la empresa.

GastoAdicional:

GastoAdicional

Descripción:

Tabla encargada de almacenar la información referente a un gasto adicional cotizado para un proyecto. Se tiene en cuenta la siguiente información: Nombre del gasto adicional y precio del gasto adicional.

HorarioLaboral:

HorarioLaboral

Descripción:

Tabla encargada de almacenar la información de la jornada laboral en la empresa, no será posible crear ni eliminar elementos, solo podrán ser consultados y modificados según sea necesario. Se tiene en cuenta la siguiente información: Numero del día de la semana (1 para domingo y 7 para sábado), nombre del día, hora de entrada en la mañana, hora de salida en la mañana, hora de entrada en la tarde y hora de salida en la tarde.

Inventario:

Inventario

Descripción:

Tabla encargada de almacenar la información referente a los materiales registrados en el inventario de la empresa, se almacena la siguiente información: Proveedor con quien se adquirió, categoría a la cual pertenece, numero de factura de compra, cantidad comprada, cantidad disponible en inventario, precio de venta por unidad del inventario, fecha de compra del inventario, ancho del material, largo de material disponible.

ItemAcceso:

ItemAcceso

Descripción:

Tabla encargada de almacenar las paginas individuales registradas en la aplicación, las cuales serán usadas para asignarse a roles. Se tiene en cuenta la siguiente información: Ruta url de la página, nombre de la página.

NoLaborables:

NoLaborables

Descripción:

Tabla encargada de almacenar la información referente a los días no laborables en la empresa, los cuales se tendrán en cuenta al momento de registrar un tiempo de actividad desarrollada. Se tiene en cuenta la siguiente información: Fecha, descripción y un indicador que define si la fecha es festiva o normal. Si es festiva se toman todos los tiempos registrados como horas festivas, si por el contrario es normal, significa que es una fecha normal en la cual no se labora, por lo que las horas registradas en esa fecha se tomaran como extras.

Novedad:

Novedad

Descripción:

Tabla encargada de almacenar las novedades tenidas en cuenta en la empresa para realizar análisis posteriores, esta tabla no será administrable desde la aplicación. Se tiene en cuenta la siguiente información:

ODTCotizacionInventario:

ODTCotizacionInventario

Descripción:

Tabla encargada de almacenar la información del material usado en un proyecto registrando el detalle de este. Se tiene en cuenta la siguiente información: Material seleccionado, proyecto donde se usa el material, empleado quien hace el pedido del material, cantidad usada, largo y ancho usado, fecha en que se usó, descripción del uso.

ODT:

ODT

Descripción:

Tabla encargada de almacenar la información referente a la orden de trabajo de un proyecto. Se tiene en cuenta la siguiente información: Proyecto, nombre de la ODT y descripción de la orden de trabajo. Esta tabla será usada para diferentes asignaciones como: Materiales usados y tiempos trabajados.

Oficio:

Oficio

Descripción:

Tabla encargada de almacenar los oficios registrados en la empresa, estos datos serán usados para realizar la cotizaciones de mano de obra. Se tiene información como: Nombre del oficio y precio del oficio.

PermisoAcceso:

PermisoAcceso

Descripción:

Tabla encargada de almacenar las funcionalidades que pueden ser accedidas por un rol, estas funcionalidades tendrán asignadas páginas de acceso (ItemAcceso).
Se tiene en cuenta la siguiente información: Nombre del permiso o funcionalidad y descripción de la funcionalidad.

PermisoAccesoRol:

PermisoAccesoRol

Descripción:

Tabla encargada de relacionar los permisos de acceso con los roles definidos en la aplicación, asignando permisos de acceso de esta manera.
Se tiene en cuenta información como: Rol y permiso de acceso asignado.

Proveedor:

Proveedor

Descripción:

Tabla encargada de almacenar la información referente a los proveedores de la empresa, se almacena información como: Nombre del proveedor, teléfono del proveedor, dirección del proveedor, ciudad del proveedor, ciudad donde se encuentra ubicado el proveedor, correo del proveedor, descripción del proveedor y nit del proveedor.



PROYECTO:	Gestión de proyectos			Página 9 de 9	
DOCUMENTO:	Base de datos				
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio		

Proyecto:

Proyecto
Descripción: Tabla encargada de almacenar la información básica de los proyectos registrados en la empresa, información que será accedida desde diferentes tablas. Se almacena la siguiente información: Cliente a quien se le realiza el proyecto, empleado responsable, nombre del proyecto, fecha de inicio y de fin del proyecto, descripción, estado del proyecto y fecha en que se registra en la aplicación.

Riesgo:

Riesgo
Descripción: Tabla encargada de almacenar los riesgos registrados en la empresa, estos datos serán usados para realizar las cotizaciones de mano de obra. Se tiene información como: Nombre del riesgo y porcentaje adicional que se cobrará por el riesgo.

Rol:

Rol
Descripción: Tabla encargada de registrar la información de los roles creados para el acceso a la aplicación, estos serán asignados a los empleados para que tengan acceso solo a un determinado número de funcionalidades. Se tiene en cuenta información como: Nombre del rol y descripción del rol.

RolEmpleado:

RolEmpleado
Descripción: Tabla encargada de registrar la información referente a las asignaciones de roles que tienen los empleados, relacionando un empleado con los roles deseados. De esta manera un empleado solo tendrá acceso a un determinado número de funcionalidades según lo especifiquen los roles asignados. Se almacena información como: Rol y empleado seleccionado.

ANEXO 10



PROYECTO:	Gestión de proyectos			Página 1 de 8
DOCUMENTO:	Pruebas funcionales			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Pruebas funcionales

Gestión de proyectos



PROYECTO:	Gestión de proyectos			Página 2 de 8
DOCUMENTO:	Pruebas funcionales			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Prueba funcional 1: Crear proveedor.....	3
Prueba funcional 2: Listar proveedores	4
Prueba funcional 3: Consultar/Modificar proveedor.....	5
Prueba funcional 4: Crear empleado	6
Prueba funcional 5: Listar empleados.....	7
Prueba funcional 6: Consultar/Modificar empleado.....	8

Prueba funcional 1: Crear proveedor

Prueba funcional:	
Número de prueba: 1	Historia de usuario: 4
Nombre de la prueba: Crear proveedor	
Descripción: Se probará que se registren exitosamente nuevos proveedores en el sistema, validando correctamente su información.	
Condiciones de ejecución: El proveedor a ser creado no existe en el sistema.	
Entrada/Pasos de ejecución: - Se selecciona la opción proveedor -> crear proveedor. - Se diligencia el formulario según los campos especificados, con datos validos e inválidos según sea el campo. - Nombre – Alfanumérico obligatorio. - Teléfono – Alfanumérico obligatorio. - Correo – Alfanumérico, debe cumplir un formato. - Dirección – Alfanumérico. - Ciudad – Alfanumérico. - Descripción – Alfanumérico. - Se pulsa la opción Guardar.	
Resultado esperado: Registro creado exitosamente, mensaje de información especificando la operación exitosa.	
Evaluación de la prueba: Positiva.	

Prueba funcional 2: Listar proveedores

Prueba funcional:	
Número de prueba: 2	Historia de usuario: 5
Nombre de la prueba: Listar proveedores	
Descripción: Se probará que se listen correctamente los proveedores existentes en el sistema, verificando el correcto funcionamiento de los filtros.	
Condiciones de ejecución: El sistema cuenta con proveedores creados.	
Entrada/Pasos de ejecución: • Se selecciona la opción proveedor -> Listar proveedores. • Se verifican los filtros para: ○ Nombre. ○ Teléfono. ○ Dirección. ○ Ciudad. ○ Correo. ○ Descripción. ○ Filtro general.	
Resultado esperado: Se listan todos los proveedores existentes en el sistema y se filtran correctamente.	
Evaluación de la prueba: Positiva.	

Prueba funcional 3: Consultar/Modificar proveedor

Prueba funcional:	
Número de prueba: 3	Historia de usuario: 6
Nombre de la prueba: Consultar/modificar proveedor	
Descripción: Se probará que al seleccionar Consultar ó Modificar se cargue correctamente la información de un proveedor en el sistema, verificando que los campos cumplen el estado de Habilitados (Modificar) o inhabilitados (Consultar) según sea el caso. Además verificar que permite realizar cambios sobre un proveedor.	
Condiciones de ejecución: El sistema cuenta con proveedores creados.	
Entrada/Pasos de ejecución: - Se selecciona la opción proveedor -> Listar proveedores. - Se selecciona la opción para consultar un proveedor en específico. - Se valida que se muestra la información correcta y que los campos están deshabilitados. - Se selecciona "Volver". - Se selecciona la opción modificar un proveedor en específico. - Se valida que se muestra la información correcta y que los campos están habilitados - Se realizan cambios sobre los valores. - Se pulsa la opción Guardar. - Se valida que los cambios se guardaron correctamente.	
Resultado esperado: - Los campos cambian de estado según la opción que se seleccionó (habilitados o deshabilitados). - Se modificada exitosamente un proveedor, mensaje de información especificando la operación exitosa.	
Evaluación de la prueba: Positiva.	

Prueba funcional 4: Crear empleado

Prueba funcional:	
Número de prueba: 4	Historia de usuario: 10
Nombre de la prueba: Crear empleado	
Descripción: Se probará que se registre exitosamente un nuevo empleado en el sistema. Validando los campos según el tipo de dato requerido.	
Condiciones de ejecución: El empleado a ser creado no existe en el sistema.	
Entrada/Pasos de ejecución: - Se selecciona la opción Empleados -> crear empleado. - Se diligencia el formulario según los campos especificados, con datos validos e inválidos según sea el campo. - Número de identificación – Valor numérico obligatorio. - Nombre – Alfanumérico obligatorio. - Apellido – Alfanumérico obligatorio. - Fecha de contratación – Alfanumérico obligatorio. - Teléfono – Alfanumérico. - Cargo – Lista de selección obligatoria. - Se pulsa la opción Guardar. La prueba se realiza con datos válidos, inválidos y campos vacíos para verificar que se validen correctamente lo datos y se muestren mensajes respectivos al error.	
Resultado esperado: Registro creado exitosamente, mensaje de información especificando la operación exitosa.	
Evaluación de la prueba: Positiva.	

Prueba funcional 5: Listar empleados

Prueba funcional:	
Número de prueba: 5	Historia de usuario: 11
Nombre de la prueba: Listar empleados	
Descripción: Se probará que se listen correctamente los empleados existentes en el sistema, verificando filtros.	
Condiciones de ejecución: El sistema cuenta con empleados creados.	
Entrada/Pasos de ejecución: - Se selecciona la opción Empleados -> Listar Empleados. - Se verifica que se muestra correctamente la lista de proyectos registrados en el sistema con información correspondiente a cada uno de estos. - Se verifican los filtros para: - Identificación. - Nombre. - Apellido. - Filtro general.	
Resultado esperado: Se listan todos los empleados existentes en el sistema y se filtran correctamente.	
Evaluación de la prueba: Positiva.	

Prueba funcional 6: Consultar/Modificar empleado

Prueba funcional:	
Número de prueba: 6	Historia de usuario: 12
Nombre de la prueba: Consultar/modificar empleado	
Descripción: Se probará que se cargue correctamente los datos de un empleado en el sistema, verificando que los campos cumplen el estado de Habilitados o inhabilitados según sea el caso, y se verifica que se guarden correctamente los cambios sobre un empleado.	
Condiciones de ejecución: El sistema cuenta con empleados creados.	
Entrada/Pasos de ejecución: • Se selecciona la opción Empleados -> Listar empleados. • Se selecciona la opción para consultar un empleado en específico. • Se selecciona "Volver". • Se selecciona la opción modificar un empleado en específico. • Se realizan cambios sobre los valores. • Se pulsa la opción Guardar.	
Resultado esperado: Registro modificado exitosamente, mensaje de información especificando la operación exitosa.	
Evaluación de la prueba: Positiva.	



PROYECTO:	Gestión de proyectos			Página 1 de 17
DOCUMENTO:	Pruebas funcionales			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Pruebas funcionales

Gestión de proyectos

Tabla de contenido

Prueba funcional 1: Ingresar material a inventario.	3
Prueba funcional 2: Listar material en inventario.....	3
Prueba funcional 3: Consultar/Modificar inventario.	4
Prueba funcional 4: Crear proyecto.	5
Prueba funcional 5: Listar proyectos.....	6
Prueba funcional 6: Consultar/Modificar proyectos.....	6
Prueba funcional 7: Crear cotización.	7
Prueba funcional 8: Crear actividades.	8
Prueba funcional 9: Crear cliente.....	9
Prueba funcional 10: Listar clientes.	10
Prueba funcional 11: Consultar/Modificar cliente.....	10
Prueba funcional 12: Crear empresa.....	11
Prueba funcional 13: Listar empresas.	12
Prueba funcional 14: Consultar/Modificar empresa.....	12
Prueba funcional 15: Crear riesgo.....	13
Prueba funcional 16: Listar riesgos.	14
Prueba funcional 17: Consultar/Modificar riesgo.....	14
Prueba funcional 18: Crear oficio.....	15
Prueba funcional 19: Listar oficio.....	16
Prueba funcional 20: Consultar/Modificar oficio.....	16

Prueba funcional 1: Ingresar material a inventario.

Prueba funcional	
Número de prueba: 1	Historia de usuario: 7
Nombre de la prueba: Ingresar material a inventario.	
Descripción: Se probará que se registre correctamente nuevos materiales en el sistema, verificando que las validaciones funcionan correctamente.	
Condiciones de ejecución: Ninguna.	
Entrada: - En el menú lateral se selecciona Inventario -> Registrar inventario. - Se diligencia el formulario según los campos especificados: - Nombre: Alfanumérico obligatorio. - Cantidad: Numérico obligatorio. - Tipo material: Lista de selección obligatoria. - Precio de compra: Campo numérico obligatorio. - Precio de venta: Campo numérico obligatorio. - Fecha de compra: Campo seleccionable obligatorio. - Proveedor: Campo seleccionable obligatorio. - Se pulsa la opción guardar. Se debe verificar este proceso para datos válidos, inválidos y campos vacíos.	
Resultado esperado: Registro creado exitosamente, mensaje de información especificando la operación exitosa.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 2: Listar material en inventario.

Prueba funcional	
Número de prueba: 2	Historia de usuario: 8
Nombre de la prueba: Listar materiales en inventario.	
Descripción: Se probará que se listen correctamente los materiales existentes en el sistema, verificando el correcto funcionamiento de filtros y demás componentes.	
Condiciones de ejecución: El sistema cuenta con varios registros de materiales.	
Entrada/Pasos de ejecución: - En el menú lateral se selección la opción <i>Inventario -> listar inventario</i>. - Se verifica que se muestre la siguiente información para cada material: - Nombre. - Cantidad. - Material. - Precio de compra.	

- Precio de venta.
- Fecha de compra.
- Opción para consultar inventario.
- Opción para modificar inventario.
- Filtro general.

Resultado esperado:

Se listan todos los materiales almacenados en el inventario con la información respectiva. El filtro funciona correctamente (Filtra los registros según palabras claves).

Evaluación de la prueba:

POSITIVA.

Prueba funcional 3: Consultar/Modificar inventario.

Prueba funcional

Número de prueba: 3

Historia de usuario: 9

Nombre de la prueba: Consultar/Modificar inventario.

Descripción:

Se probará que al Consultar o Modificar un material en particular se cargue la información correcta, verificando que la edición de los campos es habilitada o deshabilitada según sea el caso. Además, se verifica que los cambios realizados sean almacenados correctamente.

Condiciones de ejecución:

Existen materiales registrados en el sistema.

Entrada/Pasos de ejecución:

- Se selecciona la opción **Inventario -> listar inventario**.
- Se selecciona la opción **Modificar**(Botón) para un material particular.
- Se verifica que los campos cargan la información correcta y están habilitados.
- Se realizan cambios sobre los campos.
- Se selecciona **Guardar cambios**.
- Se selecciona la opción **Inventario -> listar inventario**.
- Se verifica que la información del material se modificó.
- Se selecciona la opción **Consultar**.
- Se verifica que los campos cargan la información correcta y están deshabilitados.
- Se selecciona **Volver**.
- La aplicación redirecciona a **listar inventario**.

Resultado esperado:

- Los campos cambian de estado según la opción que se selecciono (Habilitados o deshabilitados).
- Se modifica exitosamente un material, mostrando mensajes informativos para la operación exitosa.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 4: Crear proyecto.

Prueba funcional	
Número de prueba: 4	Historia de usuario: 13
Nombre de la prueba: Crear proyecto.	
Descripción: Se probará que se registre correctamente un proyecto en el sistema con su respectiva cotización, la cual calculará el precio según los componentes que se adicionen (Oficios, riesgos, materiales, gastos adicionales). Además, se verifica la correcta validación de datos según el tipo requerido.	
Condiciones de ejecución: Para crear el proyecto se debe contar con registros de: • Clientes. • Empleados con cargo Administrador. Para crear la cotización se debe contar con registros de: • Oficios. • Riesgos. • Materiales.	
Entrada/Pasos de ejecución: • Se selecciona la opción Proyectos -> Registrar proyecto. • Se mostrara el primer paso en el registro del proyecto Datos de proyecto, para el cual se diligencia el formulario según los campos especificados: ○ Nombre: campo alfanumérico obligatorio. ○ Cliente: Campo seleccionable obligatorio. ○ Responsable: Campo seleccionable obligatorio. ○ Estado: Campo seleccionable obligatorio. ○ Fecha de inicio: Campo seleccionable obligatorio. ○ Fecha de entrega: Campo seleccionable. • Se selecciona Siguiente. • Se mostrará el segundo paso en el registro del proyecto Crear cotización. La cual no es obligatoria y se detallará en la prueba funcional 7. • Se selecciona la opción Finalizar. Se realizará la prueba con datos válidos, inválidos y campos vacíos para verificar validaciones de datos.	
Resultado esperado: • Se registra el proyecto de manera exitosa, guardando correctamente toda la información diligenciada. • Las validaciones funcionan correctamente, mostrando errores en caso de un dato invalido. • Al registrar el proyecto se mostrará un mensaje de información en que se especifique que el proceso fue exitoso, de lo contrario mostrara un mensaje de error especificando el problema.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 5: Listar proyectos.

Prueba funcional	
Número de prueba: 5	Historia de usuario: 14
Nombre de la prueba: Listar proyectos.	
Descripción: Se probará que se listen de manera correcta los proyectos registrados en el sistema, mostrando información respectiva y general de cada uno de estos. Además se verificara el correcto funcionamiento de filtros y reordenamiento de datos.	
Condiciones de ejecución: Existen proyectos registrados en el sistema.	
Entrada/Pasos de ejecución: - Se selecciona la opción Proyecto -> listar proyectos. - Se verifica que se muestran todos los proyectos registrados en el sistema con su respectiva información. - Se verifica el filtro general el cual busca palabras claves. - Se verifica las opciones de reordenamiento de columnas.	
Resultado esperado: - Se listan todos los proyectos registrados en el sistema con su información respectiva. - Se filtran los datos correctamente según palabras clave.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 6: Consultar/Modificar proyectos.

Prueba funcional	
Número de prueba: 6	Historia de usuario: 15
Nombre de la prueba: Consultar/Modificar proyectos.	
Descripción: Se probará la funcionalidad de Consultar y Modificar un proyecto en particular, verificando que se cargan correctamente los datos respectivos y que las modificaciones se realizan exitosamente. Además, se verifica que las validaciones funcionan correctamente para datos validos e inválidos, mostrando mensajes informativos para cada caso.	
Condiciones de ejecución: Existe por lo menos un proyecto creado en el sistema.	
Entrada/Pasos de ejecución: - En el menú lateral, se selecciona la opción Proyectos -> Listar proyectos. - En la lista de proyectos se selecciona Modificar. - En la pestaña de <i>Información general</i> se verifica que los campos están habilitados a modificaciones y se procede a cambiar valores por datos validos e inválidos según se describe en la prueba funcional 4. - Se presiona la opción Guardar cambios. - Se verifica en la lista de proyectos que los datos fueron modificados exitosamente.	

- Se selecciona la opción **Consultar**.
- En la pestaña de *información general* se verifica que los campos están deshabilitados para edición y que se carga la información correspondiente al proyecto seleccionado.
- Se presiona la opción **Volver**.

Resultado esperado:

- Los campos cambian de estado (Activo o inactivo para edición) según la opción seleccionada.
- Se carga la información correcta a un proyecto en particular.
- Las modificaciones en la información general de los proyectos se realizan exitosamente.
- Los mensajes informativos se muestran oportunamente.
- Las validaciones funcionan correctamente según lo especificado en [la prueba funcional 4](#).

Evaluación de la prueba:

POSITIVA.

Prueba funcional 7: Crear cotización.

Prueba funcional	
Número de prueba: 7	Historia de usuario: 16
Nombre de la prueba: Crear cotización.	
Descripción: Se probará la funcionalidad de crear cotización, verificando que se carga correctamente la información respectiva para la creación (Materiales, oficios, riesgos). Se creará una tabla dinámicamente donde se especifica el detalle cotizado, en la cual se calculará el valor parcial (Valor de materiales, valor de horas cotizadas y valor de gastos adicionales) y el valor total de la cotización. Verificando que los valores son correctos y las validaciones se realizan correctamente.	
Condiciones de ejecución: Deben existir registros en el sistema de: • Materiales. • Riesgos. • Oficios.	
Entrada/Pasos de ejecución: La creación de una cotización se podrá realizar en dos momentos diferentes: 1. En el registro de un nuevo proyecto, en Proyectos -> Registrar proyecto. • Se diligencia el formulario correspondiente a la información general del proyecto, la cual se especifica en la prueba funcional 4. • Se presiona la opción Siguiente. • Se mostrará una tabla en la cual se ofrecen opciones para: Agregar oficio y riesgo, Agregar material, Agregar gasto adicional. • Agregar Oficio y riesgo: ○ Se selecciona un oficio de la lista de selección (Campo obligatorio). ○ Se selecciona un riesgo de la lista de selección (Campo obligatorio). ○ Se diligencia una cantidad de horas a cotizar (Campo numérico obligatorio). ○ Se selecciona aceptar.	

- Agregar Material:
 - Se selecciona un material de la lista de selección (Campo obligatorio).
 - Se diligencia la cantidad de material a cotizar (Campo obligatorio).
 - Se selecciona aceptar.
- Agregar gasto adicional:
 - Se diligencia el nombre del gasto (campo alfanumérico obligatorio).
 - Se diligencia el precio del gasto (campo numérico obligatorio).
 - Se selecciona aceptar.
- Se verifica que la tabla se actualice automáticamente al agregar cada componente en la cotización.
- Se verifica que la suma parcial se calcule correctamente, según los componentes de cada sección (Oficios-Riesgo, Materiales, Riesgos adicionales).
- Se verifica que el total de cotización sea correcto.
- Se presiona **Finalizar**.
- El sistema mostrará un mensaje de éxito en la operación.

Al agregar oficio-riesgo, material, gasto adicional se validara ingresando valores válidos, inválidos y campos vacíos para verificar que se valide correctamente la información.

Resultado esperado:

- Se adicionan correctamente los componentes de la cotización.
- Se calcula correctamente el valor de cada componente según la cantidad.
- Se calcula correctamente el valor parcial de la cotización.
- Se calcula correctamente el total de la cotización.
- La tabla de cotización se actualiza automáticamente al adicionar un componente.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 8: Crear actividades.

Prueba funcional	
Número de prueba: 8	Historia de usuario: 17
Nombre de la prueba: Crear actividades.	
Descripción: Se probará la funcionalidad de crear actividades, validando la correcta asignación de estas a un proyecto, a un grupo de empleados y a una lista de materiales.	
Condiciones de ejecución: Deben existir registros en el sistema de: • Proyectos. • Cotizaciones asignadas a proyectos. • Materiales cotizados en proyectos. • Oficios y riesgos cotizados en proyectos.	
Entrada/ Pasos de ejecución: • En el menú lateral se selecciona la opción de Proyectos -> Listar proyectos. • Se selecciona un proyecto en particular y se presiona en Modificar.	

- Se selecciona la tercera pestaña *Actividades*.
- Se mostrará un formulario en el cual se debe diligenciar lo siguiente:
 - Nombre, campo alfanumérico obligatorio.
 - Oficio-Riesgos, lista de selección obligatorio.
 - Responsables, campo de selección opcional.
 - Materiales, campo de selección opcional.
- Se presiona **Agregar**.

Se verifica que las validaciones sobre los campos sean correctas para el tipo de dato requerido, evaluando datos válidos, inválidos y campos vacíos. Además, se verifica que no se pueda repetir responsables en la lista.

Resultado esperado:

- Validaciones se realizan correctamente según el tipo de dato requerido.
- Se muestran mensajes informativos según la opción realizada.
- Se agrega correctamente la actividad con las asignaciones diligenciadas.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 9: Crear cliente.

Prueba funcional	
Número de prueba: 9	Historia de usuario: 18
Nombre de la prueba: Crear cliente.	
Descripción: Se probará la funcionalidad de crear cliente, para el cual se verifica la correcta validación de datos, mensajes de información al usuario y el registro exitoso del respectivo cliente.	
Condiciones de ejecución: Debe existir por lo menos un registro de empresa en el sistema.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona Clientes -> Registrar cliente. • Se mostrará un formulario en el cual se debe diligenciar la siguiente información: ○ Nombre, campo alfanumérico obligatorio. ○ Apellido, campo alfanumérico obligatorio. ○ Dirección, campo alfanumérico. ○ Ciudad, campo alfanumérico obligatorio. ○ Teléfono, campo alfanumérico. ○ Correo, campo alfanumérico. ○ Empresa, lista seleccionable obligatoria. • Se selecciona Guardar. El registro se realizara con diferentes tipos de datos válidos, inválidos y campos vacíos según se especifica; esto con el fin de verificar las validaciones y los mensajes de información que especifican el error.	
Resultado esperado: • Se valida correctamente la información diligenciada según lo especificado. • Se muestra correctamente los mensajes de información al usuario (Errores y procesos	

exitosos).

- Se registra correctamente el cliente en el sistema.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 10: Listar clientes.

Prueba funcional	
Número de prueba: 10	Historia de usuario: 19
Nombre de la prueba: Listar clientes.	
Descripción: Se probará la funcionalidad de listar clientes, para lo cual se verifica que se listen correctamente los clientes registrados en el sistema, mostrando información correspondiente a cada uno de estos. Además se verifica el funcionamiento del filtro.	
Condiciones de ejecución: Deben existir registros de clientes en el sistema.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona <i>Clientes -> listar clientes</i>. • Se mostrará una lista con los clientes en el sistema. • Se valida que se muestren todos los registros en el sistema. • Se valida que se muestre la información respectiva a cada cliente. • Se valida el funcionamiento del filtro, buscando palabras clave en los clientes.	
Resultado esperado: • Se muestran todos los clientes existentes en el sistema. • Se muestra información respectiva a cada cliente. • Se filtra la información según palabras clave diligenciada.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 11: Consultar/Modificar cliente.

Prueba funcional	
Número de prueba: 11	Historia de usuario: 20
Nombre de la prueba: Consultar/Modificar cliente.	
Descripción: Se probará la funcionalidad de Consultar o Modificar cliente, verificando que se cargue la información respectiva de un cliente, que los campos se activen o inactiven según corresponda y que los cambios se guarden correctamente.	
Condiciones de ejecución: Debe existir un cliente registrado en el sistema.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona <i>Clientes -> Listar clientes</i>.	

- Se selecciona **Modificar** sobre un cliente particular.
 - Se valida que el formulario que se despliega esté diligenciado con la información respectiva al cliente seleccionado.
 - Se valida que los campos estén habilitados para edición.
 - Se realizan modificaciones en los campos cargados.
 - Se selecciona **Guardar cambios**. Lo cual redirecciona a listar clientes nuevamente.
 - En la lista de clientes, se selecciona **Consultar**.
 - Se verifica que la información cargada es la respectiva al cliente seleccionado.
 - Se verifica que los campos estén deshabilitados para edición.
 - Se presiona **Volver** lo cual debe redireccionar a listar clientes.
- Los cambios realizados en modificar cliente se realizaran ingresando valores válidos, inválidos y campos vacíos para verificar el correcto funcionamiento de las validaciones.

Resultado esperado:

- Se carga la información respectiva a un cliente seleccionado.
- Los campos se muestran habilitados o deshabilitados según la opción seleccionada.
- Se muestran mensajes de información al usuario cuando ocurren errores o se realiza una modificación exitosa.
- Las modificaciones se guardan exitosamente.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 12: Crear empresa.

Prueba funcional	
Número de prueba: 12	Historia de usuario: 21
Nombre de la prueba: Crear empresa.	
Descripción: Se probará la funcionalidad de crear empresa, verificando que se realice correctamente la validación de datos.	
Condiciones de ejecución: Ninguno.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona la opción Empresa -> Registrar empresa. • Se muestra un formulario con los siguientes campos: ○ Nombre, campo alfanumérico obligatorio. ○ Ciudad, nombre alfanumérico obligatorio. • Se presiona la opción Guardar. El registro se realizará dejando campos vacíos, con el objetivo de verificar el correcto funcionamiento de las validaciones. En este caso se deben mostrar mensajes de error informando al usuario que el campo es requerido.	
Resultado esperado: • Las validaciones de datos funcionan correctamente. • Al ocurrir un error se muestra un mensaje de error.	

- Se registra exitosamente la empresa en el sistema.

Evaluación de la prueba:
POSITIVA.

Prueba funcional 13: Listar empresas.

Prueba funcional	
Número de prueba: 13	Historia de usuario: 22
Nombre de la prueba: Listar empresas.	
Descripción: Se probará la funcionalidad de listar empresas, validando que se muestran todos los registros existentes y que la información es la respectiva para cada registro.	
Condiciones de ejecución: Deben existir registros de empresas en el sistema.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona la opción Empresa -> Listar empresas. • Se muestra una lista con las empresas registradas en el sistema con la información respectiva de cada una. • Se muestra un filtro para buscar palabras clave. • Al diligenciar palabras clave en el filtro se muestran las empresas que estén relacionadas.	
Resultado esperado: • Se muestran todas las empresas registradas en el sistema. • La información mostrada es la respectiva para cada registro. • Se filtra correctamente según palabras clave.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 14: Consultar/Modificar empresa.

Prueba funcional	
Número de prueba: 14	Historia de usuario: 23
Nombre de la prueba: Consultar/Modificar empresa.	
Descripción: Se probará la funcionalidad de Consultar y Modificar empresa, verificando que se cargue correctamente la información de una empresa seleccionada y que se guarden correctamente los cambios realizados, verificando el correcto funcionamiento de las validaciones.	
Condiciones de ejecución: Debe existir un registro de empresa en el sistema.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona la opción Empresa -> Listar empresas. • Se mostrará una lista con todas las empresas registradas en el sistema. • Se selecciona la opción Modificar sobre un registro en particular.	

- Se valida que la información cargada en el formulario es la respectiva a la empresa seleccionada.
- Se valida que los campos estén habilitados para edición.
- Se realizan modificaciones sobre los datos cargados.
- Se presiona la opción **Guardar cambios**.
- Se presiona la opción **Volver**.
- Se muestra la lista de empresas, se verifica que los cambios se guardaron exitosamente.
- Se selecciona **Consultar** sobre una empresa en particular.
- Se verifica que la información cargada en el formulario es la respectiva a la empresa seleccionada.
- Se verifica que los campos están deshabilitados para edición.
- Se presiona **Volver**.

Al realizar modificaciones sobre el registro seleccionado, se verifica la validación con campos vacíos.

Resultado esperado:

- Se carga la información respectiva a una empresa seleccionada.
- Los campos cambian de estado según la opción seleccionada.
- Las validaciones funcionan correctamente.
- Se muestran mensajes de información al usuario cuando ocurre un error o se realizan los cambios exitosamente.
- Se guardan los cambios realizados sobre un registro exitosamente.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 15: Crear riesgo.

Prueba funcional	
Número de prueba: 15	Historia de usuario: 24
Nombre de la prueba: Crear riesgo.	
Descripción: Se probará la funcionalidad de crear riesgo en el sistema, verificando que las validaciones funcionan correctamente.	
Condiciones de ejecución: Ninguna.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona la opción Configuración -> Registrar riesgo. • Se mostrará un formulario en el cual se debe diligenciar la siguiente información: ○ Nombre, campo alfanumérico obligatorio. ○ Porcentaje, campo numérico obligatorio. • Se presiona la opción Guardar. • Se muestra un mensaje de información donde especifica que el proceso fue exitoso. El proceso de registro se realizara diligenciando campos válidos, inválidos y campos vacíos con el fin de validar el correcto funcionamiento de las validaciones.	

Resultado esperado:

- Se valida correctamente los datos diligenciados.
- Se muestran mensajes de información (Error y éxito).
- Se guarda exitosamente el registro en el sistema.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 16: Listar riesgos.

Prueba funcional	
Número de prueba: 16	Historia de usuario: 25
Nombre de la prueba: Listar riesgos.	
Descripción: Se probará la funcionalidad de listar riesgos, verificando que se muestran todos los registros en el sistema y que los datos son los respectivos para cada registro.	
Condiciones de ejecución: Deben existir registros de riesgos en el sistema.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona la opción de Configuración -> Listar riesgos. • Se muestra una lista con todos los registros de riesgos en el sistema y la información respectiva a cada registro. • Se muestra un filtro en el cual se pueden diligenciar palabras clave para filtrar los registros. • Al diligenciar palabras clave se muestran solo los registros relacionados.	
Resultado esperado: • Se muestran todos los registros en el sistema. • Se muestra la información respectiva a cada registro en el sistema. • Se filtra correctamente según las palabras claves diligenciadas.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 17: Consultar/Modificar riesgo.

Prueba funcional	
Número de prueba: 17	Historia de usuario: 26
Nombre de la prueba: Consultar/Modificar riesgo.	
Descripción: Se probará la funcionalidad de Consultar y Modificar riesgos en el sistema, verificando que se cargue correctamente la información del riesgo seleccionado y que se guarden correctamente los cambios realizados, verificando el correcto funcionamiento de las validaciones.	
Condiciones de ejecución: Deben existir registros de riesgos en el sistema.	

Entrada/Pasos de ejecución:

- En el menú lateral se selecciona la opción de **Configuración -> Listar riesgo**.
- Se muestra una lista con todos los registros en el sistema.
- Se selecciona la opción **Modificar** sobre un registro en particular.
- Se muestra un formulario diligenciado con los datos del riesgo.
- Se valida que los datos sean correctos y que los campos estén habilitados para edición.
- Se realizan modificaciones sobre los campos.
- Se presiona **Guardar cambios**.
- Aparece un mensaje de información mostrando que el proceso fue exitoso.
- Se selecciona **Volver**.
- Se carga nuevamente la lista de riesgos.
- Se selecciona la opción **Consultar** sobre un registro en particular.
- Se verifica que los datos cargados en el formulario son correctos y que los campos están deshabilitados para edición.

Las modificaciones se realizan diligenciando datos válidos, inválidos y campos vacíos con el fin de verificar el correcto funcionamiento de los datos.

Resultado esperado:

- Se carga la información correspondiente a cada registro seleccionado.
- Los campos cambian de estado según la opción seleccionada.
- Los cambios se guardan exitosamente.
- Las validaciones se realizan correctamente.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 18: Crear oficio.

Prueba funcional	
Número de prueba: 18	Historia de usuario: 27
Nombre de la prueba: Crear oficio.	
Descripción: Se probará la funcionalidad de crear oficio en el sistema, verificando que las validaciones funcionan correctamente.	
Condiciones de ejecución: Ninguna.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona la opción Configuración -> Registrar oficio. • Se mostrará un formulario en el cual se debe diligenciar la siguiente información: ○ Nombre, campo alfanumérico obligatorio. ○ Precio, campo numérico obligatorio. • Se presiona la opción Guardar. • Se muestra un mensaje de información donde especifica que el proceso fue exitoso. El proceso de registro se realizará diligenciando campos válidos, inválidos y campos vacíos con el fin de validar el correcto funcionamiento de las validaciones.	

Resultado esperado:

- Se valida correctamente lo datos diligenciados.
- Se muestran mensajes de información (Error y éxito).
- Se guarda exitosamente el registro en el sistema.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 19: Listar oficio.

Prueba funcional	
Número de prueba: 19	Historia de usuario: 28
Nombre de la prueba: Listar oficio.	
Descripción: Se probará la funcionalidad de listar oficios, verificando que se muestran todos los registros en el sistema y que los datos son los respectivos para cada registro.	
Condiciones de ejecución: Deben existir registros de oficio en el sistema.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona la opción de Configuración -> Listar oficios. • Se muestra una lista con todos los registros de oficio en el sistema y la información respectiva a cada registro. • Se muestra un filtro en el cual se pueden diligenciar palabras clave para filtrar los registros. Al diligenciar palabras clave se muestran solo los registros relacionados.	
Resultado esperado: • Se muestran todos los registros en el sistema. • Se muestra la información respectiva a cada registro en el sistema. • Se filtra correctamente según las palabras claves diligenciadas.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 20: Consultar/Modificar oficio.

Prueba funcional	
Número de prueba: 20	Historia de usuario: 29
Nombre de la prueba: Consultar/Modificar oficio.	
Descripción: Se probará la funcionalidad de Consultar y Modificar oficios en el sistema, verificando que se cargue correctamente la información del oficio seleccionado y que se guarden correctamente los cambios realizados, verificando el correcto funcionamiento de las validaciones.	
Condiciones de ejecución: Existen registros de oficio en el sistema.	

Entrada/Pasos de ejecución:

- En el menú lateral se selecciona la opción **Configuración -> Listar oficinas**.
- Se mostrará una lista con todos los oficinas registrados en el sistema.
- Se selecciona la opción **Modificar** sobre un registro en particular.
- Se valida que la información cargada en el formulario es la respectiva al oficina seleccionado.
- Se valida que los campos estén habilitados para edición.
- Se realizan modificaciones sobre los datos cargados.
- Se presiona la opción **Guardar cambios**.
- Se presiona la opción **Volver**.
- Se muestra la lista de oficinas, se verifica que los cambios se guardaron exitosamente.
- Se selecciona **Consultar** sobre un oficina en particular.
- Se verifica que la información cargada en el formulario es la respectiva al oficina seleccionado.
- Se verifica que los campos están deshabilitados para edición.
- Se presiona **Volver**.

Al realizar modificaciones sobre el registro seleccionado, se verifica la validación con campos válidos, inválidos y vacíos.

Resultado esperado:

- Se carga la información correspondiente a cada registro seleccionado.
- Los campos cambian de estado según la opción seleccionada.
- Los cambios se guardan exitosamente.
- Las validaciones se realizan correctamente.
- Se muestran mensajes de información (Errores o exitosos).

Evaluación de la prueba:

POSITIVA.



PROYECTO:	Gestión de proyectos			Página 1 de 8
DOCUMENTO:	Pruebas funcionales			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Pruebas funcionales

Gestión de proyectos



PROYECTO:	Gestión de proyectos			Página 2 de 8
DOCUMENTO:	Pruebas funcionales			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Prueba funcional 1: Registrar cargos.	3
Prueba funcional 2: Listar cargos.	3
Prueba funcional 3: Consultar/Modificar cargo.	4
Prueba funcional 4: Registrar tiempo trabajado en actividad.	5
Prueba funcional 5: Consultar/ Modificar tiempo trabajado en actividad.	6
Prueba funcional 6: Guardar archivo de proyecto.	7
Prueba funcional 7: Listar archivos de proyecto.	7
Prueba funcional 8: Eliminar archivos de proyecto.	8

Prueba funcional 1: Registrar cargos.

Prueba funcional	
Número de prueba: 1	Historia de usuario: 30
Nombre de la prueba: Registrar cargos.	
Descripción: Se probará la funcionalidad del registro de cargos verificando que se registre adecuadamente en el sistema, que se valide correctamente la información y que los datos sean los respectivos para cada registro.	
Condiciones de ejecución: Ninguna.	
Entrada: - En el menú lateral se selecciona la opción de Configuración -> Gestionar cargos. - Se muestra una lista con todos los registros de cargos en el sistema y la información respectiva de cada cargo registrado. - Al presionar Registrar se muestra un formulario en el cual se debe diligenciar la siguiente información: - Nombre, campo alfanumérico obligatorio. - Descripción, campo alfanumérico. - Se presiona la opción Registrar. Se verifica que las validaciones sobre los campos sean correctas para el tipo de dato requerido, evaluando datos válidos, inválidos y campos vacíos.	
Resultado esperado: - Validaciones se realizan correctamente según el tipo de dato requerido. - Se muestran mensajes informativos según la opción realizada. - Se muestra la información respectiva al cargo registrado en el sistema. - Se guarda exitosamente los cargos registrados en el sistema.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 2: Listar cargos.

Prueba funcional	
Número de prueba: 2	Historia de usuario: 31
Nombre de la prueba: Listar cargos.	
Descripción: Se probará que se listen de manera correcta los cargos en el sistema, mostrando información respectiva y general de cada uno de estos. Además se verificara el correcto funcionamiento de filtros y reordenamiento de datos.	
Condiciones de ejecución: Existen cargos registrados en el sistema.	
Entrada/Pasos de ejecución:	

- Se selecciona la opción **Configuración -> Gestionar cargos**.
- Se muestra una lista de cargos en el sistema y la información respectiva de cada cargo registrado.
- Se verifica el filtro general el cual busca palabras claves.
- Se verifica las opciones de reordenamiento de columnas.

- Se muestra la lista de cargos registrados en el sistema con su información respectiva.
- Se filtran los datos correctamente según palabras clave.

Evaluación de la prueba:
POSITIVA.

Prueba funcional 3: Consultar/Modificar cargo.

Prueba funcional	
Número de prueba: 3	Historia de usuario: 32
Nombre de la prueba: Consultar/Modificar cargo.	
Descripción: Se probará la funcionalidad de Consultar o Modificar cargo, verificando que se cargue la información respectiva del cargo en el sistema, que los campos se activen y que los cambios se guarden correctamente.	
Condiciones de ejecución: Debe existir un cargo registrado en el sistema.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona la opción configuración-> Gestionar cargos. • Se selecciona Modificar sobre un cargo particular. • Se valida que el formulario que se despliega esté diligenciado con la información respectiva al cargo seleccionado. • Se valida que los campos estén habilitados para edición. • Se realizan modificaciones en los campos cargados. • Se selecciona Guardar cambios. Lo cual redirecciona a listar cargos nuevamente. • En la lista de cargos, se selecciona Consultar. • Se verifica que la información cargada es la respectiva al cargo seleccionado. • Se verifica que los campos estén deshabilitados para edición. • Se presiona Volver lo cual debe redireccionar a listar cargos. Los cambios realizados en modificar cargo se realizaran ingresando valores válidos, inválidos y campos vacíos para verificar el correcto funcionamiento de las validaciones.	
Resultado esperado: • Se carga la información respectiva a un cargo seleccionado. • Los campos se muestran habilitados o deshabilitados según la opción seleccionada.	

- Se muestran mensajes de información al usuario cuando ocurren errores o se realiza una modificación exitosa.

Las modificaciones se guardan exitosamente.

Evaluación de la prueba:
POSITIVA.

Prueba funcional 4: Registrar tiempo trabajado en actividad.

Prueba funcional	
Número de prueba: 4	Historia de usuario: 33
Nombre de la prueba: Registrar tiempo trabajado en actividad.	
Descripción: Se probará la funcionalidad de registrar tiempo trabajado en actividad. Para lo cual se verifica la correcta validación de datos, mensajes de información al usuario y el registro exitoso del respectivo registro para el cual se debe calcular las horas trabajadas según el intervalo de horas diligenciadas.	
Condiciones de ejecución: Debe existir por lo menos un proyecto creado en el sistema.	
Entrada/Pasos de ejecución: • En el menú lateral se selecciona Registros -> Registro de tiempo. • Se muestra la lista de los últimos tiempos registrados en el sistema. • Se selecciona la opción Registrar. • Se despliega una ventana emergente con el formulario a diligenciar, se muestran los siguientes campos: ○ Proyecto: Campo seleccionable obligatorio. ○ Actividad: Campo Seleccionable obligatorio. ○ Fecha: Campo seleccionable obligatorio. ○ Hora de inicio: Campo seleccionable obligatorio. ○ Hora de fin: Campo seleccionable obligatorio. ○ Descripción: Campo alfanumérico obligatorio. • Se selecciona Aceptar. El registro de tiempo se verifica con diferentes tipos de datos válidos, inválidos y campos vacíos según se especifica; esto con el fin de verificar las validaciones y mensajes de información que especifican el error.	
Resultado esperado: • Se valida correctamente la información diligenciada según lo especificado. • Se muestra correctamente los mensajes de información al usuario. • Se registra correctamente el tiempo en el sistema. • Se calcula correctamente el tiempo trabajado según el intervalo ingresado.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 5: Consultar/ Modificar tiempo trabajado en actividad.

Prueba funcional	
Número de prueba: 5	Historia de usuario: 34
Nombre de la prueba: Consultar tiempo trabajado en actividad.	
Descripción: Se probara la funcionalidad de consultar o modificar un registro de tiempo trabajado en actividad, verificando que se cargue la información respectiva de un registro, validando que los campos se activan o inactivan según corresponda.	
Condiciones de ejecución: Deben existir registros de tiempo trabajado en actividad.	
Entrada/Pasos de ejecución: - En el menú lateral se selecciona Registros -> Registro de tiempo. - Se muestra la lista de los últimos tiempos registrados en el sistema. - Se selecciona Modificar sobre un registro en particular. - Se valida que el formulario que se despliega esté diligenciado con la información respectiva al registro seleccionado. - Se valida que los campos estén habilitados para edición. - Se realizan modificaciones de los campos cargados. - Se selecciona Guardar cambios. Se cierra la ventana emergente. - Se verifica que la información se modificó exitosamente. - En la lista de registros, se selecciona consultar. - Se verifica que la información cargada es la respectiva al registro seleccionado. - Se verifica que los campos estén deshabilitados para edición. - Se presiona Cerrar. Los cambios realizados en modificar registro de tiempo se realizaran ingresando valores válidos, inválidos y campos vacíos para verificar el correcto funcionamiento de las validaciones.	
Resultado esperado: - Se carga la información respectiva de un registro seleccionado. - Los campos se muestran habilitados o deshabilitados según la opción seleccionada. - Se muestran mensajes de información al usuario cuando ocurren errores o se realiza una modificación exitosa. - Las modificaciones se guardan exitosamente.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 6: Guardar archivo de proyecto.

Prueba funcional	
Número de prueba: 6	Historia de usuario: 36
Nombre de la prueba: Guardar archivo de proyecto.	
Descripción: Se probará la funcionalidad de guardar archivos en un proyecto. Para lo cual se verifica que es copie el archivo seleccionado al servidor.	
Condiciones de ejecución: Debe existir un proyecto creado en el sistema.	
Entrada/Pasos de ejecución: - En el menú lateral se selecciona la opción de Proyectos -> Listar proyectos. - Se selecciona la opción modificar. - Se muestra un formulario con la información respectiva del proyecto seleccionado. - Se selecciona la opción Archivos - Se muestra la opción para adicionar archivos y consulta de archivos. - Se selecciona la opción Buscar lo cual abre un explorador para buscar un archivo en el equipo. - Se selecciona un archivo y se presiona Aceptar. - El archivo se muestra en la lista de archivos pendientes para guardar. - Se presiona Guardar el sistema copia el archivo en el servidor y lo muestra en la lista de archivos cargados. Se verifica que el archivo se cargue en el servidor.	
Resultado esperado: Se carga el archivo exitosamente, siendo asignado al proyecto seleccionado.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 7: Listar archivos de proyecto.

Prueba funcional	
Número de prueba: 7	Historia de usuario: 37
Nombre de la prueba: Listar archivos de proyecto.	
Descripción: Se probará que se listen de manera correcta los archivos de proyectos registrados en el sistema, mostrando información respectiva y general de cada uno de estos.	
Condiciones de ejecución: Existen archivos de proyectos registrados en el sistema.	

Entrada/ Pasos de ejecución:

- Se selecciona la opción **Proyecto -> listar proyectos.**
- Se selecciona la opción **modificar.**
- Se muestra un formulario con la información respectiva del proyecto seleccionado.
- Se selecciona la opción **Archivos**
- Se muestra la opción para adicionar archivos y consulta de archivos.
- Se verifica que se muestran todos los archivos de proyectos registrados en el sistema con su respectiva información.

Resultado esperado:

- Se listan todos los archivos de proyectos registrados en el sistema con su información respectiva.

Evaluación de la prueba:

POSITIVA.

Prueba funcional 8: Eliminar archivos de proyecto.

Prueba funcional	
Número de prueba: 8	Historia de usuario:38
Nombre de la prueba: Eliminar archivos de proyecto	
Descripción: Se probará la eliminación de archivos cargados con anterioridad en el sistema.	
Condiciones de ejecución: Existen archivos de proyectos registrados en el sistema.	
Entrada/Pasos de ejecución: • Se selecciona la opción Proyecto -> listar proyectos. • Se selecciona la opción modificar. • Se muestra un formulario con la información respectiva del proyecto seleccionado. • Se selecciona la opción Archivos • Se muestra la opción para adicionar archivos y consulta de archivos. • En la lista de consulta de archivos se selecciona la opción eliminar. • El sistema muestra un mensaje de confirmación. Se selecciona Sí. • El sistema elimina el archivo, actualizando la lista de archivos del proyecto.	
Resultado esperado: • Se elimina exitosamente los archivos del sistema. • Se actualiza la lista de archivos, mostrando los archivos existentes.	
Evaluación de la prueba: POSITIVA.	



PROYECTO:	Gestión de proyectos			Página 1 de 8
DOCUMENTO:	Pruebas funcionales			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Pruebas funcionales

Gestión de proyectos



PROYECTO:	Gestión de proyectos			Página 2 de 8
DOCUMENTO:	Pruebas funcionales			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Prueba funcional 1: Crear ODT.....	3
Prueba funcional 2: Consultar/Modificar ODT.....	3
Prueba funcional 3: Calcular costo y precio de venta para inventario.	4
Prueba funcional 4: Listar usuarios. TODAVIA NO	5
Prueba funcional 5: Modificar usuario. TODAVIA NO	5
Prueba funcional 6: Configurar días no laborables.	5
Prueba funcional 7: Modificar días no laborables.	¡Error! Marcador no definido.
Prueba funcional 8: Configurar jornada laboral.....	6
Prueba funcional 9: Registro de tiempo trabajado.	7
Prueba funcional 10: Registrar material usado.....	8

Prueba funcional 1: Crear ODT.

Prueba funcional	
Número de prueba: 1	Historia de usuario: 39
Nombre de la prueba: Crear ODT	
Descripción: Se probará la funcionalidad de crear ODT verificando que se registre automáticamente en el sistema de una manera adecuada, validando que la información creada es la respectiva.	
Condiciones de ejecución: Ninguna.	
Entrada: - En el menú lateral se selecciona la opción Proyecto -> Registrar proyecto. - Se diligencia el formulario de creación de proyecto. - Se selecciona Guardar. - En el menú lateral se selecciona la opción Proyecto -> Listar proyectos. - En la lista se selecciona Consultar sobre el proyecto que se acaba de crear. - Se selecciona la pestaña ODT. - Se verifica que los datos mostrados en la ODT son válidos y se muestran de manera agrupada como se especifica.	
Resultado esperado: - La ODT se crea automáticamente con los datos del proyecto: Nombre, descripción, fecha de inicio, fecha de fin. - Los datos de la ODT se muestran de manera agrupada como se especifica.	
Evaluación de la prueba: POSITIVA	

Prueba funcional 2: Consultar/Modificar ODT.

Prueba funcional	
Número de prueba: 2	Historia de usuario: 40
Nombre de la prueba: Consultar/Modificar ODT	
Descripción: Se proba la funcionalidad de consultar y modificar una ODT, verificando que se muestren los datos correctamente y que se permitan realizar modificaciones sobre el nombre y la descripción.	
Condiciones de ejecución: Debe existir un proyecto con su ODT creadas en el sistema.	
Entrada: - En el menú lateral se selecciona la opción Proyecto -> Listar proyectos. - En la lista se selecciona Consultar sobre un proyecto. - Se selecciona la pestaña ODT. - Se verifica que los datos mostrados en la ODT son válidos y se muestran de manera agrupada como se especifica. - Se selecciona el nombre de la ODT y se modifica.	

- Se presiona en el botón de guardar.
- Se selecciona la descripción de la ODT y se modifica.
- Se presiona en el boto de guardar.
- Se cambia de página.
- En el menú lateral se selecciona la opción **Proyecto -> Listar proyectos**.
- En la lista se selecciona **Consultar** sobre el proyecto que se acaba de modificar.
- Se selecciona la pestaña ODT.
- Se verifica que los cambios realizados si son correctos.

Resultado esperado:

- Verificar que los datos de la ODT se muestren correctamente según lo especificado.
- Verificar que los datos de la ODT son los respectivos al proyecto.
- Verificar que se permita modificar los campos de nombre y descripción y que los cambios se guardan correctamente.

Evaluación de la prueba:

POSITIVA

Prueba funcional 3: Exportar ODT

Prueba funcional	
Número de prueba: 3	Historia de usuario: 41
Nombre de la prueba: Exportar ODT.	
Descripción: Se probara la funcionalidad de exportar una ODT, verificando que se realice la descarga del archivo en formato PDF y que contenga la información respectiva.	
Condiciones de ejecución: Debe existir un proyecto con su ODT creadas en el sistema.	
Entrada: • En el menú lateral se selecciona la opción Proyecto -> Listar proyectos. • En la lista se selecciona Consultar sobre un proyecto. • Se selecciona la pestaña ODT. • Se selecciona la opción Exportar PDF. • El sistema inicia la descarga de un archivo PDF. • Se verifica que el archivo muestre la información respetiva a la ODT seleccionada. • Se verifica que la información se encuentre agrupada como se especifica.	
Resultado esperado: • Se realiza la descarga de un archivo en formato PDF. • El archivo tiene la información respectiva a la ODT seleccionada. • La información esta presentada de forma clara y agrupada.	
Evaluación de la prueba: POSITIVA	

Prueba funcional 4: Listar usuarios. TODAVIA NO

Prueba funcional	
Número de prueba: 4	Historia de usuario: 42
Nombre de la prueba: Listar usuarios	
Descripción:	
Condiciones de ejecución:	
Entrada:	
Resultado esperado:	
Evaluación de la prueba:	

Prueba funcional 5: Modificar usuario. TODAVIA NO

Prueba funcional	
Número de prueba: 5	Historia de usuario: 43
Nombre de la prueba: Modificar usuario	
Descripción:	
Condiciones de ejecución:	
Entrada:	
Resultado esperado:	
Evaluación de la prueba:	

Prueba funcional 6: Configurar días no laborables.

Prueba funcional	
Número de prueba: 6	Historia de usuario: 44
Nombre de la prueba: Configurar días no laborables	
Descripción: Se probará la configuración de días no laborables en el sistema, verificando que se registre adecuadamente en el cada registro, que se valide correctamente la información.	
Condiciones de ejecución: Debe existir un proyecto con su ODT creadas en el sistema.	

Entrada:

- En el menú lateral se selecciona la opción **Configuración -> Días no laborables**.
- Se muestra una tabla con una lista de fechas no laborables.
- Se selecciona **Registrar**.
- Se muestra un formulario con los datos correspondientes para crear un nuevo registro.
- Se diligencia el formulario y se presiona **Registrar**.
- Se verifica que el registro se muestra en la tabla inicial.
- Se selecciona la opción de **Modificar**.
- Se muestra un formulario diligenciado con la información correspondiente al registro seleccionado.
- Se presiona **Modificar**.
- Se verifica que el registro se muestra con los cambios correspondientes.

Resultado esperado:

- Verificar que se listen correctamente los días no laborables configurados en el sistema.
- Verificar que se permita crear un nuevo registro.
- Verificar que las modificaciones surgen efecto en los registros.

Evaluación de la prueba:

POSITIVA

Prueba funcional 7: Configurar jornada laboral.

Prueba funcional	
Número de prueba: 7	Historia de usuario: 45
Nombre de la prueba: Configurar jornada laboral	
Descripción: Se probará la configuración de la jornada laboral en el sistema, verificando que se modifique adecuadamente cada registro y que se valide correctamente la información.	
Condiciones de ejecución: Ninguna.	
Entrada: • En el menú lateral se selecciona la opción Configuración -> Horario laboral. • Se muestra una tabla con una lista de los días de la semana e información respectiva para: Hora entra mañana, hora salida mañana, hora entrada tarde y hora salida tarde. • Se selecciona la opción modificar sobre un día en particular. • El sistema activa los campos para registrar las horas. • Se realizan cambios sobre el horario y se presiona aceptar. • Se muestra la tabla con la información actualizada.	
Resultado esperado: • Verificar que se están listando el horario laboral correctamente.	



PROYECTO:	Gestión de proyectos			Página 7 de 8	
DOCUMENTO:	Pruebas funcionales				
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio		

- Verificar que se realizan modificaciones correctamente.

Evaluación de la prueba:

POSITIVA

Prueba funcional 8: Registro de tiempo trabajado.

Prueba funcional	
Número de prueba: 8	Historia de usuario: 46
Nombre de la prueba: Registro de tiempo trabajado	
Descripción: Se probará la funcionalidad para registrar un tiempo trabajado, verificando que se realizan correctamente las validaciones y el registro se hace exitosamente.	
Condiciones de ejecución: Existe un proyecto en ejecución.	
Entrada: • En el menú lateral se selecciona la opción Registro -> Registro tiempos. • Se muestra una tabla con una lista de registros realizados anteriormente, y una tabla con el consolidado de horas trabajadas en cada proyecto en particular. • Se selecciona la opción Registrar. • Se muestra un formulario con los campos respectivos para realizar el registro. • Se diligencia el formulario. • Se presiona Registrar. • El registro se lista en la tabla con los datos correspondientes. • El consolidado de horas se actualiza según registro realizado. Se verifica que las validaciones sobre los campos del formulario sean correctas para el tipo de dato requerido, evaluando datos válidos, inválidos y campos vacíos.	
Resultado esperado: • Verificar que se muestren los registros realizados con anterioridad. • Verificar que las validaciones se hacen correctamente. • Verificar que se registran nuevos tiempos trabajados. • Verificar que el registro tiene en cuenta la jornada laboral para calcular las horas trabajadas.	
Evaluación de la prueba: POSITIVA.	

Prueba funcional 9: Registrar material usado

Prueba funcional	
Número de prueba: 9	Historia de usuario: 47
Nombre de la prueba: Registrar material usado	
Descripción: Se probará la funcionalidad para registrar un material usado en un proyecto en particular, verificando que se realizan correctamente las validaciones y el registro se hace exitosamente.	
Condiciones de ejecución: Existe un proyecto en ejecución.	
Entrada: • En el menú lateral se selecciona la opción Registro -> Materiales usados. • Se muestra una tabla con una lista de registros realizados anteriormente, donde se especifica el empleado, ODT, material usado y la cantidad usada. • Se selecciona la opción Registrar. • Se muestra un formulario con los campos respectivos para realizar el registro de material usado. • Se diligencia el formulario. • Se presiona Registrar. • El registro se lista en la tabla con los datos correspondientes. • Se hace el descuento de material del inventario. Se verifica que las validaciones sobre los campos del formulario sean correctas para el tipo de dato requerido, evaluando datos válidos, inválidos y campos vacíos.	
Resultado esperado: • Verificar que se muestran los datos registrados con anterioridad. • Verificar que las validaciones se hacen correctamente. • Verificar que se registran nuevos materiales usados y se muestran exitosamente. • Verificar que se realiza el descuento del material en inventario.	
Evaluación de la prueba: POSITIVA	

ANEXO 11



PROYECTO:	Gestión de proyectos			Página 1 de 12
DOCUMENTO:	Manual de instalación			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Manual de instalación



PROYECTO:	Gestión de proyectos			Página 2 de 12
DOCUMENTO:	Manual de instalación			
VERSIÓN:	1.0	ELABORADO POR:	Hernán Antonio Ramírez Osorio	

Tabla de contenido

Instalación de SGBD postgresql.....	3
Instalación de servidor de aplicaciones Glassfish	7
Despliegue de la aplicación	12

Instalación de SGBD postgresql

1. Se ingresa en la URL <https://www.postgresql.org/download/windows/> y se presiona en la opción zip archive.

Windows installers

Interactive installer by EnterpriseDB

[Download the installer](#) certified by EnterpriseDB for all supported PostgreSQL versions.

This installer includes the PostgreSQL server, pgAdmin; a graphical tool for managing and developing your databases, and StackBuilder; a package manager that can be used to download and install additional PostgreSQL tools and drivers. Stackbuilder includes management, integration, migration, replication, geospatial, connectors and other tools.

This installer can run in graphical or silent install modes.

The installer is designed to be a straightforward, fast way to get up and running with PostgreSQL on Windows.

Advanced users can also download a [zip archive](#) of the binaries, without the installer. This download is intended for users who wish to include PostgreSQL as part of another application installer.

2. Se busca la sección de postgresql 9.2, y se descarga la versión según el SO. En este caso se descarga una versión de Windows 64bits

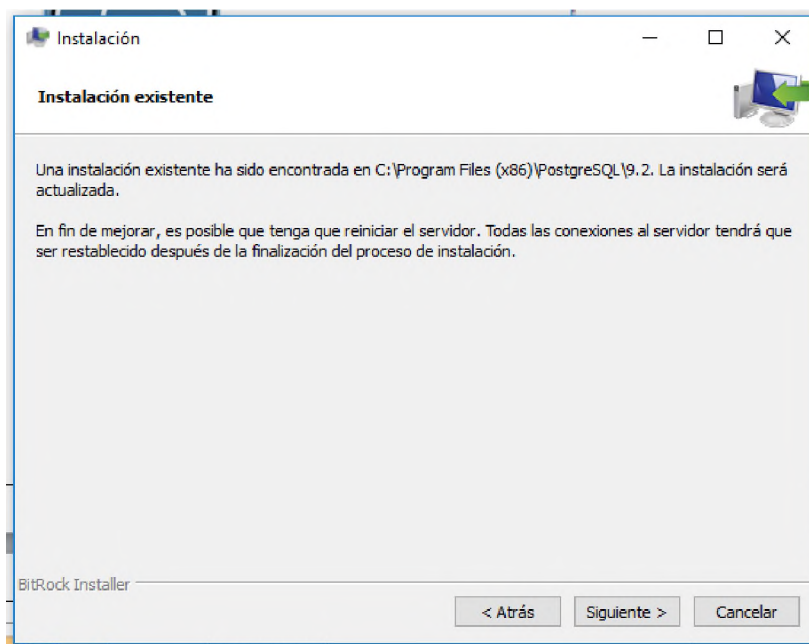
Binaries from installer version 9.2.19



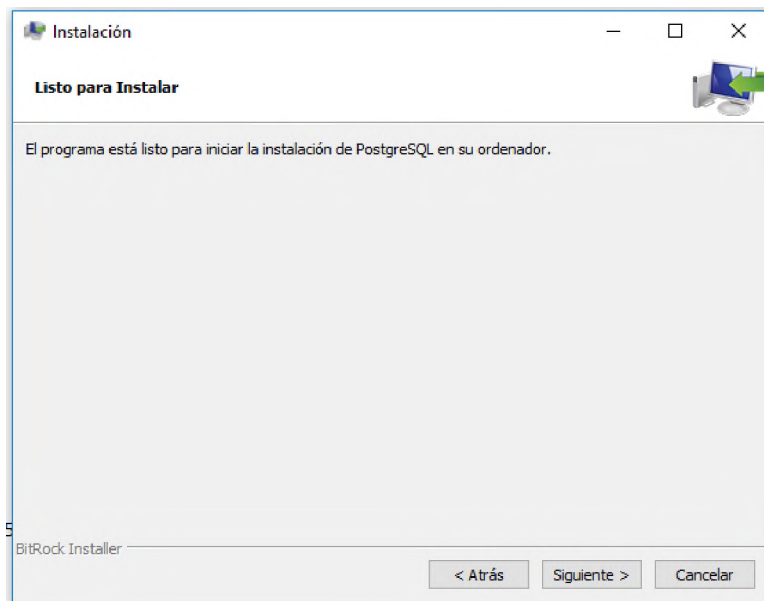
3. Se ejecuta el archivo seleccionado y se siguen los pasos. Se selecciona siguiente.



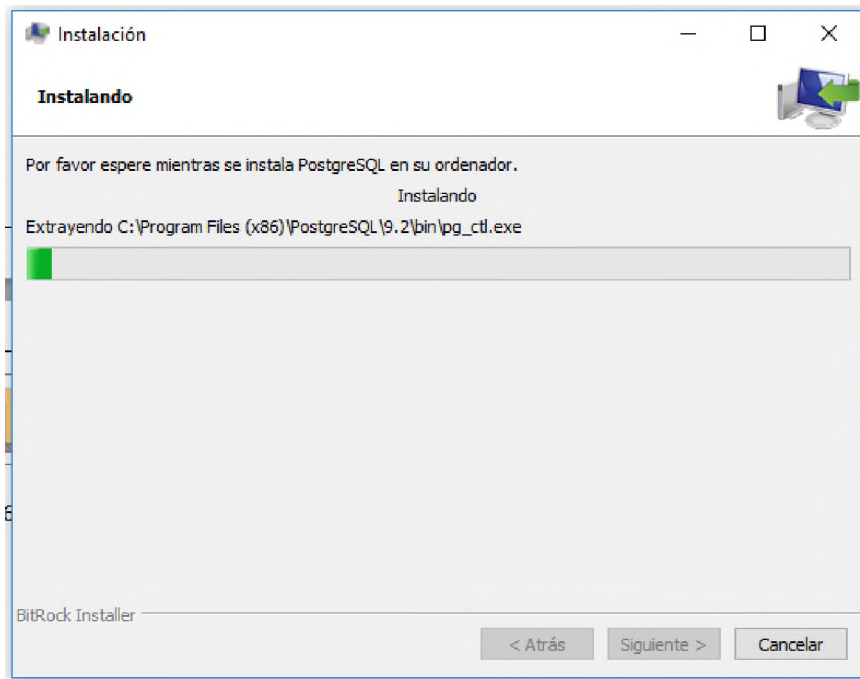
4. Siguiente:



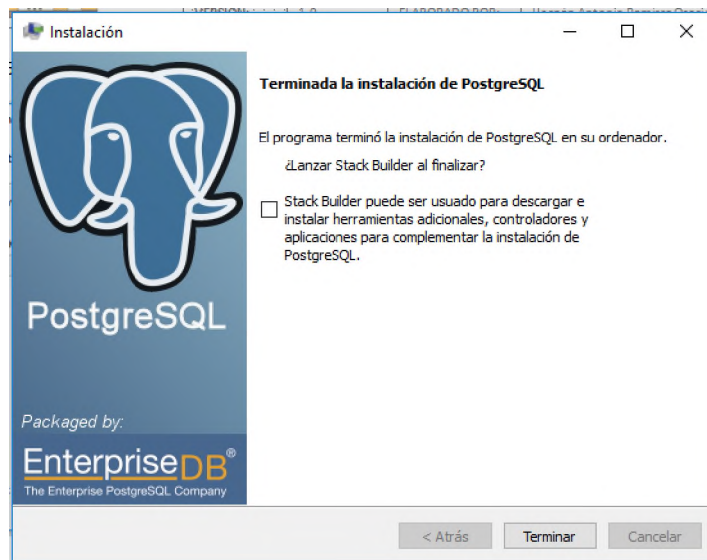
5. Siguiente:



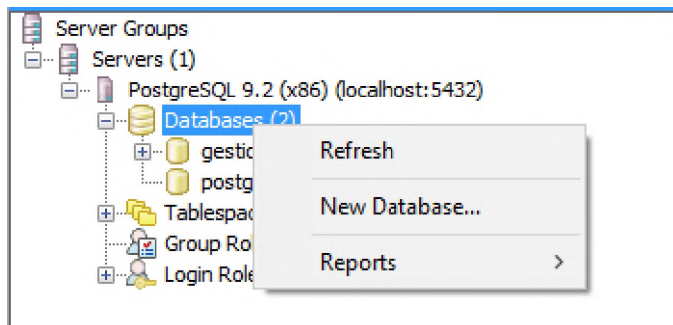
6. Siguiente:



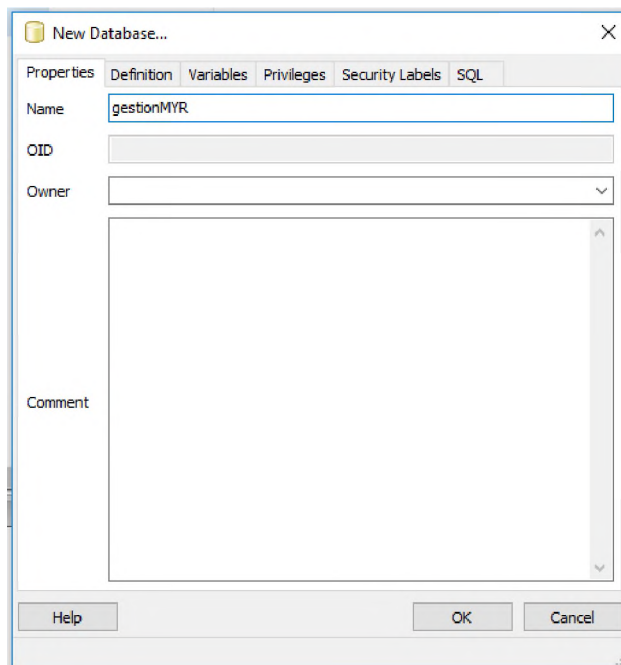
7. Finalizar.



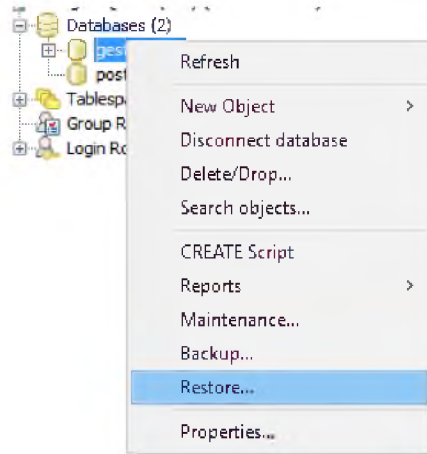
8. Se inicia la aplicación llamada pgAdmin III. Y se da click derecho y se selecciona "New database".



9. Se crea una base de datos con el nombre **gestionMYR**.



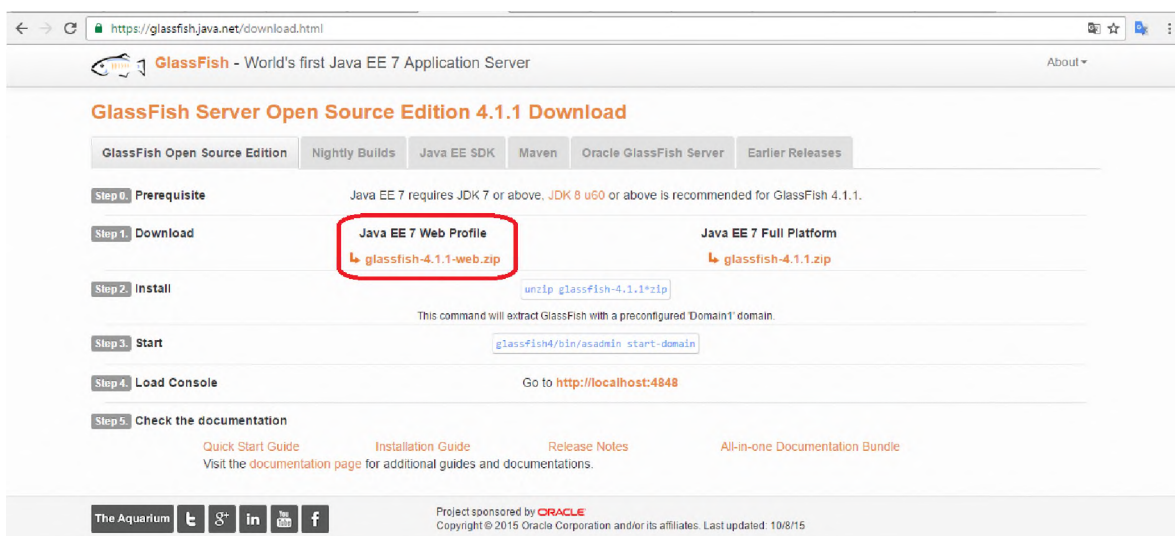
10. Con click derecho sobre la base de datos, se selecciona **Restore**.



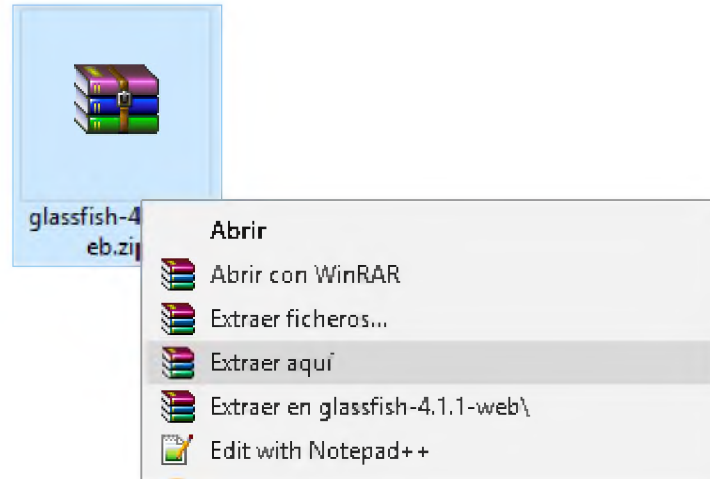
11. Se selecciona el archivo .backup que se encuentra en el proyecto.

Instalación de servidor de aplicaciones Glassfish

1. Se ingresa en la url <https://glassfish.java.net/download.html> y presiona en descargar el servidor de la siguiente manera:



2. Se descomprime el archivo en una carpeta segura, que no se vaya a eliminar.



3. Se ingresa a `\glassfish4\bin`. Se ejecuta el archivo `asadmin.bat`

» Escritorio » Nueva carpeta » glassfish4 » bin

Nombre	Fecha de modifica...	Tipo	Tamaño
asadmin	19/09/2015 11:56 a...	Archivo	3 KB
asadmin.bat	19/09/2015 11:56 a...	Archivo por lotes ...	3 KB
pkg	19/09/2015 11:56 a...	Archivo	9 KB
pkg.bat	19/09/2015 11:56 a...	Archivo por lotes ...	10 KB
updatetool	19/09/2015 11:56 a...	Archivo	9 KB
updatetool.bat	19/09/2015 11:56 a...	Archivo por lotes ...	10 KB

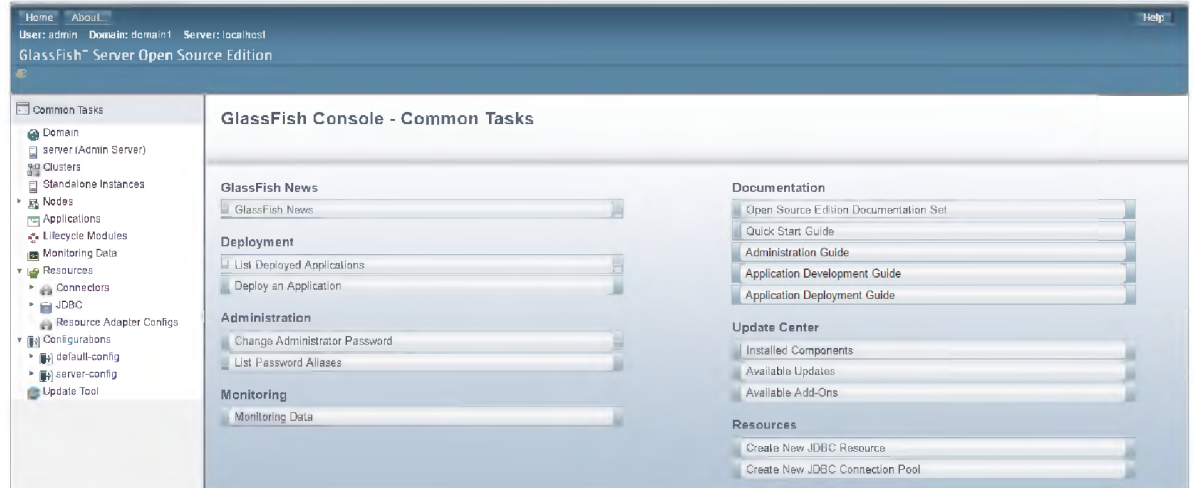
4. Al ejecutar el archivo, se abre una consola, en la cual se debe escribir: **start-domain** para iniciar el servidor.

```

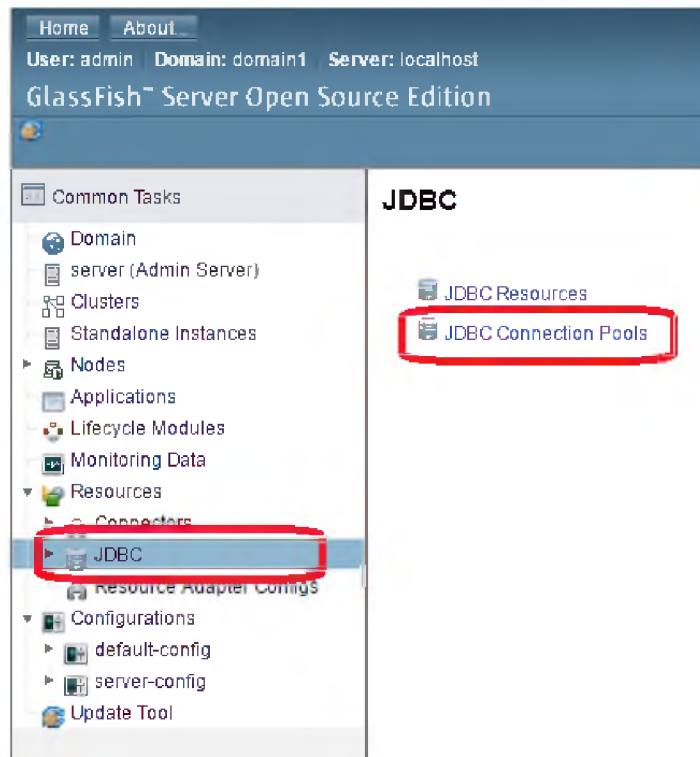
C:\Windows\system32\cmd.exe
Use "exit" to exit and "help" for online help.
asadmin> start-domain
Waiting for domain1 to start .....
Successfully started the domain : domain1
domain Location: C:\Users\RAMIREZ\Desktop\Nueva carpeta\glassfish4\glassfish\domains\domain1
Log File: C:\Users\RAMIREZ\Desktop\Nueva carpeta\glassfish4\glassfish\domains\domain1\logs\server.log
Admin Port: 4848
Command start-domain executed successfully.
asadmin>

```

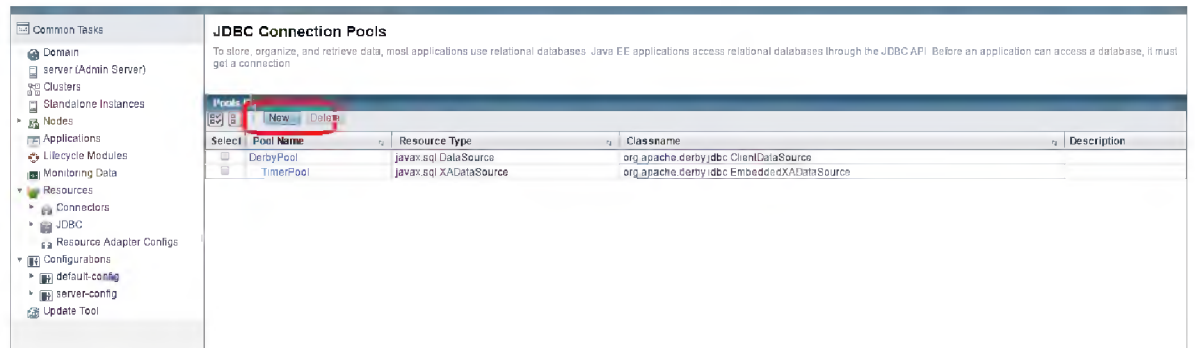
5. Al tener iniciado el servidor, se ingresa por medio de un navegador a la url: **http://localhost:4848/** para ingresar a la consola e iniciar con el proceso de configuración.



6. Se procede a crear el pool de conexiones ingresando en **JDBC** en el menú lateral, y luego seleccionando **JDBC Connection Pools**.



7. Se presiona **new** para iniciar el proceso de creación del pool.



8. En el formulario desplegado se diligencia la siguiente información.

- **Pool name:** post-gre-sql_gestionMYR_postgresPool
- **Resource Type:** javax.sql.DataSource
- **Database Driver Vendor:** Postgresql

Se presiona siguiente

New JDBC Connection Pool (Step 1 of 2)

Identify the general settings for the connection pool.

General Settings

Pool Name: *

Resource Type:

Database Driver Vendor:

Introspect: ☐ Enabled

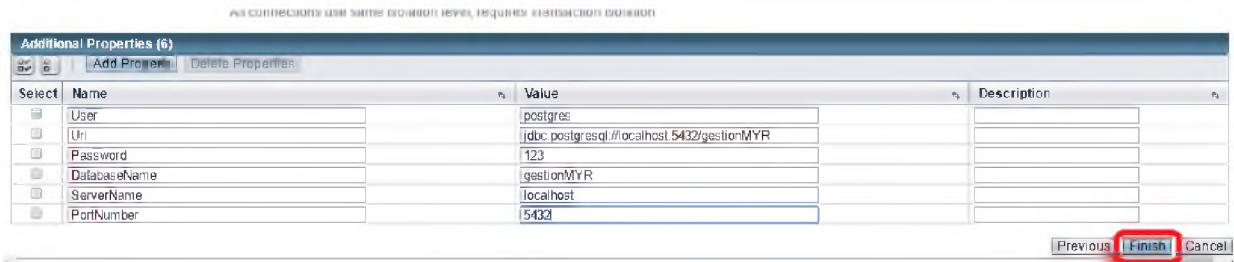
* Indicates required field

9. En el formulario que se muestra, se ubica en la parte inferior en la opción de “Additional Properties”, donde se debe diligenciar los siguientes datos:

- **serverName:** localhost
- **portNumber:** 5432
- **databaseName:** gestionMYR

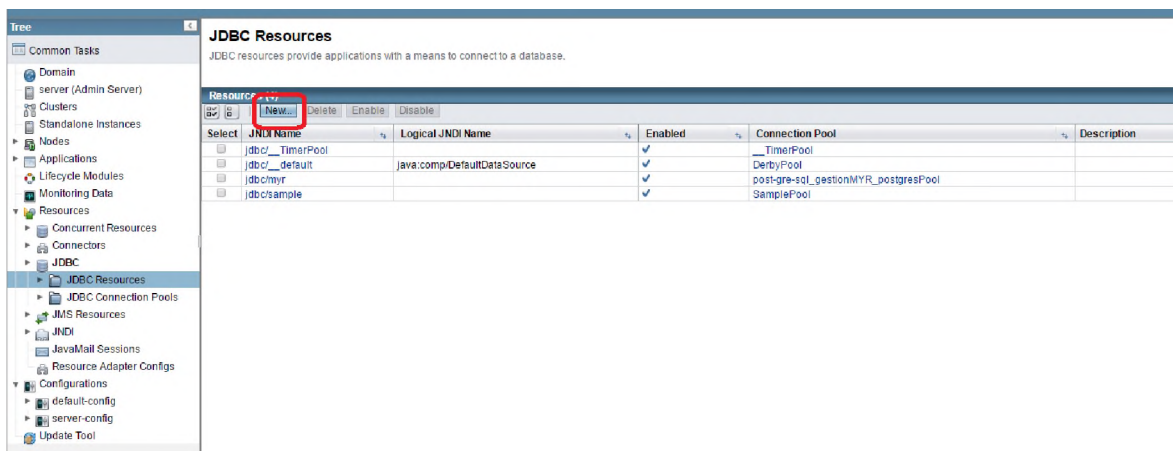
- **User:** postgres
- **Password:** 123
- **URL:** jdbc:postgresql://localhost:5432/gestionMYR
- **driverClass:** org.postgresql.Driver

Y se presiona en **Finish**.



Select	Name	Value	Description
☐	User	postgres	
☐	Uri	jdbc:postgresql://localhost:5432/gestionMYR	
☐	Password	123	
☐	DatabaseName	gestionMYR	
☐	ServerName	localhost	
☐	PortNumber	5432	

10. Se debe realizar la creación del **JDBC Resources**, seleccionando en el menú lateral **JDBC -> JDBC Resources**. Y se presiona New.



Select	JNDI Name	Logical JNDI Name	Enabled	Connection Pool	Description
☐	jdbc/_TimerPool		☒	_TimerPool	
☐	jdbc/_default	java.comp/DefaultDataSource	☒	DerbyPool	
☐	jdbc/myr		☒	post-gre-sql_gestionMYR_postgresPool	
☐	jdbc/sample		☒	SamplePool	

11. En el formulario que se muestra, se diligencia la siguiente información:

- **JNDI Name:** jdbc/myr
- **Pool Name:** post-gre-sql_gestionMYR_postgresPool

ANEXO 13



GESTIÓN Y CONTROL DE PROYECTOS - PROYECTOS MANTENIMIENTO INDUSTRIAL M&R
PRUEBA DE ACEPTACIÓN - PRIMER ENTREGA

Fecha:

N°	Historia de usuario	Funcionalidad	Presentacion de datos	Interfaz grafica	Observación
4	Registrar proveedor	CUMPLE	BUENA	BUENA	Se debe agregar campo para diligenciar NIT. Permitir la opcion de agregar proveedor desde listar proveedores.
5	Consultar/Modificar proveedor	CUMPLE	BUENA	BUENA	Falta NIT del proveedor.
10	Registrar empleado	CUMPLE	BUENA	BUENA	Agregar fecha de terminación de contrato.
11	Listar empleados	CUMPLE	BUENA	BUENA	Permitir registrar un empleado desde esta ubicación.
12	Consultar/Modificar empleado	CUMPLE	BUENA	BUENA	Ninguna.



GESTIÓN Y CONTROL DE PROYECTOS - PROYECTOS MANTENIMIENTO INDUSTRIAL M&R
PRUEBA DE ACEPTACIÓN - SEGUNDA ENTREGA

Fecha:

N°	Historia de usuario	Funcionalidad	Presentacion de datos	Interfaz grafica	Observación
7	Registrar material en inventario	CUMPLE	BUENA	BUENA	Ninguna.
8	Listar material en inventario	CUMPLE	BUENA	BUENA	Ordenar en orden alfabetico según nombre.
9	Consultar/Modificar inventario	CUMPLE	BUENA	BUENA	Ninguna.
13	Crear proyecto	CUMPLE	BUENA	BUENA	Ninguna.
14	Listar proyectos	CUMPLE	BUENA	BUENA	Ninguna.
15	Consultar/Modificar proyecto	CUMPLE	BUENA	BUENA	Permitir generar ODT's.
16	Crear cotizacion	CUMPLE	BUENA	BUENA	Ninguna.
17	Crear actividades	NO CUMPLE	REGULAR	BUENA	Se debe generar un formato unico ODT (Orden de Trabajo).
18	Registrar clientes	CUMPLE	BUENA	BUENA	Ninguna.
19	Listar clientes	CUMPLE	BUENA	BUENA	Permitir el registro de un cliente desde esta ubicación.
20	Consultar/Modificar cliente	CUMPLE	BUENA	BUENA	Mostrar que proyectos se han desarrollado para un cliente.
21	Registrar empresa	CUMPLE	BUENA	BUENA	Ninguna.
22	Listar empresas	CUMPLE	BUENA	BUENA	Permitir el registro de una empresa desde esta ubicación.
23	Consultar/Modificar empresa	CUMPLE	BUENA	BUENA	Ninguna.
24	Registrar riesgo	CUMPLE	BUENA	BUENA	Ninguna.
25	Listar Riesgos	CUMPLE	BUENA	BUENA	Permitir el registro de un riesgo desde esta ubicación.
26	Consultar/Modificar riesgo	CUMPLE	BUENA	BUENA	Ninguna.
27	Registrar oficio	CUMPLE	BUENA	BUENA	Ninguna.
28	Listar oficios	CUMPLE	BUENA	BUENA	Permitir el registro de un oficio desde esta ubicación.
29	Consultar/Modificar oficio	CUMPLE	BUENA	BUENA	Ninguna.



GESTIÓN Y CONTROL DE PROYECTOS - PROYECTOS MANTENIMIENTO INDUSTRIAL M&R
PRUEBA DE ACEPTACIÓN - TERCER ENTREGA

Fecha:

N°	Historia de usuario	Funcionalidad	Presentación de datos	Interfaz grafica	Observación
30	Registrar cargos	CUMPLE	BUENA	BUENA	Ninguna.
31	Listar Cargos	CUMPLE	BUENA	BUENA	Ordenar en orden alfabetico según nombre.
32	Consultar/Modificar cargo	CUMPLE	BUENA	BUENA	Ninguna.
33	Registrar tiempo trabajado en actividad	NO CUMPLE			Se debe registrar en una ODT unica para cada proyecto.
34	Consultar tiempo trabajado en actividad	NO CUMPLE			Se debe manejar una ODT unica para cada proyecto.
35	Modificar tiempo trabajado en actividad	NO CUMPLE			Se debe manejar una ODT unica para cada proyecto.
36	Guardar archivo de proyecto	CUMPLE	BUENA	BUENA	Ninguna.
37	Listar archivos de proyecto	CUMPLE	BUENA	BUENA	Ninguna.
38	Eliminar archivos de proyecto	CUMPLE	BUENA	BUENA	Ninguna.



GESTIÓN Y CONTROL DE PROYECTOS - PROYECTOS MANTENIMIENTO INDUSTRIAL M&R
PRUEBA DE ACEPTACIÓN - CUARTA ENTREGA

Fecha:

N°	Historia de usuario	Funcionalidad	Presentación de datos	Interfaz grafica	Observación
39	Crear ODT.	CUMPLE	BUENA	BUENA	
40	Consultar/Modificar ODT.	CUMPLE	BUENA	BUENA	
41	Exportar ODT	CUMPLE	BUENA	BUENA	Se deben eliminar la lista de materiales cotizados del reporte generado.
42	Exportar cotización	CUMPLE	BUENA	BUENA	
43	Modificar usuarios.	CUMPLE	BUENA	BUENA	
44	Configurar días no laborables.	CUMPLE	BUENA	BUENA	
45	Configurar jornada laboral.	CUMPLE	BUENA	BUENA	
46	Registro de tiempo trabajado	CUMPLE	BUENA	BUENA	
47	Registro de materiales usados	CUMPLE	BUENA	BUENA	