

Desarrollo de modulos para el registro de ventas,
contactos y agendamiento de clientes de software
de ventas para la empresa IPSOFT S.A
Proyecto de grado modalidad pasantía

David Ernesto Gómez Gutierrez
Código. 1153681

Universidad del Valle Sede Tuluá
Programa Ingeniería de Sistemas
Octubre 2016

Desarrollo de modulos para el registro de ventas,
contactos y agendamiento de clientes de software
de ventas para la empresa IPSOFT S.A
Proyecto de grado modalidad pasantía

David Ernesto Gómez Gutierrez
Código. 1153681

Trabajo de grado para optar por el titulo de ingeniero de sistemas

Director:
Federico Lopez Gomez, Ing

Universidad del Valle Sede Tuluá
Programa Ingeniería de Sistemas
Octubre 2016

Dedicatoria

A Norberyra, Dagoberto, Gabriela, Karen y Ricardo, a Luz Maria, Judith y a mis abuelos. A mis familiares, los seres que el destino puso en mi camino y a ti. “El recordatorio de una nueva vida para todos ustedes”.

David

Agradecimientos

A los profesores por transmitirnos conocimiento tan valioso, a Federico por su tiempo y apoyo.

A mis familiares por haberme dado todo lo necesario para ingresar a la universidad, por haberme dado todas las herramientas para sobrevivir, por la paciencia y por el cariño

A mis compañeros por las risas, los consejos, los regaños.

A todos, inmensa gratitud.

Índice general

1. Introduccion	1
1.1. Descripcion General	1
1.2. Problema	2
1.2.1. Descripción General del Problema	2
1.3. Objetivos	4
1.3.1. Objetivo General	4
1.3.2. Objetivos Específicos	4
2. Marco Referencial	5
2.1. Marco Conceptual	5
2.1.1. Programación Orientada a Objetos	5
2.1.2. Modelo Vista Controlador	6
2.1.3. Patrón Abstract Factory	7
2.1.4. Frameworks de desarrollo aplicaciones web	7
3. Módulos para la gestión y reserva de beneficios y venta de membresías	9
3.1. Módulo de gestión de beneficios	9
3.1.1. Diagnóstico Inicial	9
3.1.2. Resultado del proceso de rediseño	12
3.2. Módulo Venta de membresías	12
3.2.1. Diagnóstico Inicial	12
3.2.2. Resultado del Proceso de Rediseño	14
3.3. Módulo de Reserva de beneficios	16
3.3.1. Fase de Análisis	16
3.3.2. Fase de Diseño	16
3.3.3. Fase de codificación	16
3.3.4. Pruebas de aceptación	17
3.4. Manuales de usuario para los módulos: venta de membresías, gestión de beneficios y reserva de beneficios	19
4. Aspectos del desarrollo de software	21
4.1. Fase de Análisis	22
4.1.1. Actas	22

4.1.2. Anexos	22
4.1.3. Requerimientos	22
4.1.4. Historias de Usuario	22
4.2. Fase de Diseño	23
4.2.1. Modelo de Datos	23
4.3. Diseño de la arquitectura	23
4.4. Fase de Codificación	24
4.5. Pruebas de Aceptación	25
4.6. Despliegue de la aplicación	25
5. Conclusiones	26

Índice de figuras

2.1. Diagrama MVC	6
3.1. Diagrama de flujo proceso registro beneficios	10
3.2. Diagrama de flujo carga archivos planos	11
3.3. Vista principal modulo archivos planos	12
3.4. Diagrama flujo registro venta	13
3.5. Vista principal modulo ventas	15
3.6. Vista principal modulo reserva	18
3.7. Vista principal WIKI	19
3.8. Detalle WIKI	20

Capítulo 1

Introduccion

1.1. Descripcion General

En los últimos años en la ciencia de la medicina se han logrado notables avances especialmente en el área de la cirugía, esta ha evolucionado a un ritmo vertiginoso apoyandose en el desarrollo tecnológico de otras areas como la anestesiología, la tomografía, la resonancia magnética, entre otras. Permitiendo así que se practiquen procedimientos quirúrgicos menos invasivos y con menor morbilidad ¹ [2]. Por otro lado la transmisión del conocimiento científico y tecnológico es una tarea con una importancia cada vez mayor. La apropiación de este conocimiento por parte de la población en general es una necesidad obvia, pero resulta mucho más obvia y natural para quienes se dedican en forma profesional a la investigación, la enseñanza y la comunicación en las diversas áreas de la ciencia y la tecnología [1], por ejemplo resulta una necesidad obvia y necesaria que los profesionales de la salud se capaciten y aún más importante difundan su conocimiento. Si las anteriores premisas se cumplen la calidad de vida de la población en general aumentaría de forma importante y se extendería la expectativa de vida de pacientes, entre otros logros orientados al bienestar de la humanidad.

El proyecto IMEKT CLINIC (*International Medical Knowledge Transmission*) que se proyecta como una clínica de alta especialidad y centro de capacitación internacional, fundado en 2010 y actualmente en etapa de ejecución, nació como una propuesta enfocada en la difusión del conocimiento médico, específicamente el relacionado con procedimientos quirúrgicos permitiendo a médicos especialistas extranjeros dedicados a la investigación o precursores en algún nuevo procedimiento quirúrgico capacitar a sus homólogos colombianos por medio de congresos, seminarios, procedimientos quirúrgicos didácticos y conferencias. Este proyecto pretende ser un referente en lo que se refiere a la difusión del conocimiento médico.

Para sistematizar todos sus procesos, la clínica ha decidido encomendar la tarea de desarrollar un aplicativo a la empresa IPSOFT-SA. En el presente trabajo de grado se aborda el rediseño y desarrollo de algunos módulos pertenecientes al aplicativo con el fin de proveer a la clínica IMEKT una herramienta infomática apropiada que facilite su

¹cuantía de personas que están enfermas en un sitio y tiempo determinado

funcionamiento.

1.2. Problema

1.2.1. Descripción General del Problema

La clínica IMEKT contará con diversos espacios físicos entre ellos: doce quirófanos, un centro de convenciones, múltiples auditorios interactivos, oficinas, salas de eventos y salas de prensa para albergar los eventos destinados a la difusión del conocimiento médico (congresos, seminarios, procedimientos quirúrgicos didácticos y conferencias), los servicios que la clínica ofrecerá en estos espacios son denominados beneficios y se dividen en dos categorías, “beneficios de intercambio” que permiten a los terceros tener cupos en los eventos anteriormente mencionados, y “beneficios de mercadeo y servicio” que le dan la posibilidad a los terceros de situar su publicidad en los espacios físicos y usar los anteriores para sus propias actividades o eventos. El modelo de negocio de la clínica consiste en la comercialización de beneficios. Los clientes de la clínica tienen la opción de obtener un grupo de beneficios, comprando una membresía que tiene una duración de cinco años, los terceros que podrán obtener estas membresías serán:

- Universidades: Instituciones de educación superior que deseen usar los quirófanos interactivos para contar una herramienta adicional en el proceso de aprendizaje de sus estudiantes o capacitar a sus docentes ofreciéndoles cupos en los eventos de la clínica.
- Empresas: Compañías que deseen realizar sus eventos o situar su publicidad en los espacios físicos de la clínica y que además quieran obtener descuentos en otros servicios de la clínica.
- Profesionales de la salud: Médicos, especialistas, enfermeros entre otros profesionales que estén interesados en obtener nuevo conocimiento asistiendo a los eventos organizados por la clínica.

Para hacer efectivo el uso de sus beneficios los terceros tendrán la posibilidad de programar su uso, para esto deberán especificar claramente la fecha en la que usarán el servicio y un conjunto de requisitos que dependen del beneficio que desee reservar, por ejemplo al reservar un beneficio para el uso de un espacio publicitario dentro de la clínica el tercero deberá proporcionar la fecha en la que quiere desplegar su publicidad y un archivo con los elementos publicitarios de su marca. Además el tercero tendrá la posibilidad de gestionar sus beneficios, por ejemplo, visualizarlos, consultar la fecha de reserva, mover la fecha y cancelar la reserva de un beneficio.

Adicional a eso la clínica IMEKT reúne sus beneficios en grupos para facilitar su gestión, por ejemplo un grupo llamado “beneficios publicitarios” agrupa los beneficios que permiten pautar una marca en cualquier espacio de la clínica, a cada uno de estos grupos se le asigna un encargado, este encargado o administrador del grupo, tendrá a su cargo las actividades relacionadas con la gestión de beneficios como su reserva y la

recepción de requisitos para su reserva, cancelación de la reserva, movimientos de la reserva, entre otras actividades.

Por todo lo anterior, la clínica IMEKT pacto una alianza con la empresa IPSOFT-SA para proveer un aplicativo que permita sistematizar los procesos de venta de membresías, y reserva y gestión de beneficios, asimismo los procesos propios de la clínica como gestión de usuarios e historia clínica electrónica, este aplicativo aún en etapa de desarrollo tiene como base el “framework” “SIIS (Sistema de Información Integral para la Salud)”, que solo cubre los procesos propios de la clínica, la sistematización de los proceso de venta, gestión y reserva no están al alcance de SIIS. Por lo tanto se hizo necesario iniciar un nuevo proyecto de desarrollo dentro de IPSOFT-SA donde se planeó el desarrollo de tres nuevos módulos que permitieran la sistematización de los siguientes procesos:

1. Ventas de membresías.
2. Gestión de beneficios.
3. Reserva de beneficios.

Al momento de iniciar el presente trabajo se tenían desarrollados los dos primeros módulos: módulo de ventas de membresías y módulo de gestión de beneficios, sin embargo, debido a factores externos los directivos del proyecto IMEKT decidieron hacer cambios en su modelo de negocio. Con base a lo anterior se hace necesario el rediseño de los módulos de ventas de membresías y de gestión de beneficios, además iniciar el desarrollo del módulo de reserva de beneficios, y su integración al aplicativo desarrollado por IPSOFT-SA, asegurando así la correcta sistematización de la totalidad de los procesos de la clínica IMEKT.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollo de módulos para la gestión y reserva de beneficios y venta de membresías como extensión al *framework* SIIS para la clínica IMEKT

1.3.2. Objetivos Específicos

1. Rediseñar módulo de ventas de membresías para IMEKT CLINIC.
2. Rediseñar módulo de gestión de beneficios para IMEKT CLINIC.
3. Desarrollar módulo de reserva de beneficios para IMEKT CLINIC.
4. Construir manuales de usuario para los módulos: venta de membresías, gestión de beneficios y reserva de beneficios.

Capítulo 2

Marco Referencial

En este capítulo se definen los conceptos usados en el presente trabajo de grado

2.1. Marco Conceptual

2.1.1. Programación Orientada a Objetos

La programación orientada a objetos es un paradigma de programación donde los conceptos de clase y objeto son los más importantes, un objeto es cualquier cosa tangible o intangible que se pueda imaginar, que posea atributos [4], por ejemplo, una paciente, un quirófano, un edificio, entre otros. Una clase es una plantilla que permite definir un grupo de objetos con atributos y funciones en común [4], por ejemplo, persona es una clase de objeto caracterizado por tener nombres, apellidos, edad, fecha de nacimiento entre otros. Para diseñar un programa siguiendo estilo o paradigma es necesario identificar cuáles son los objetos de un sistema, las clases en que se pueden agrupar y las relaciones que tienen entre ellos [4]. La Programación orientada a objetos tiene ciertas características fundamentales:

- Modularidad: Proceso de dividir un objeto en piezas más pequeñas, evitando un alto acoplamiento y aumentando la reusabilidad.
- Abstracción: Permite extraer las propiedades más importantes de un objeto, estas propiedades varían dependiendo del contexto en que usará el objeto.
- Encapsulación: Permite empaquetar la funcionalidad de un objeto, de forma que se puede modificar la funcionalidad interna sin afectar la visión externa de un componente del sistema.

- Ocultación de la Información: Consiste en no mostrar al exterior los datos o las funciones que no sean necesarias, un módulo bien encapsulado solo muestra la información necesaria.

En cuanto a sus ventajas se destacan:

- Reusabilidad: Reutilizar componentes de software, al ser modular permite agrupar funcionalidades por clases, para reutilizar estos componentes basta con instanciar un objeto de esta clase.
- Mantenibilidad: Debido a la sencillez para abstraer el problema, los programas orientados a objetos son más sencillos de leer y comprender, pues nos permiten ocultar detalles de implementación dejando visibles sólo aquellos detalles más relevantes.
- Modificabilidad: La facilidad de añadir, suprimir o modificar nuevos objetos nos permite hacer modificaciones de una forma muy sencilla.
- Fiabilidad: Al dividir el problema en partes más pequeñas podemos probarlas de manera independiente y aislar mucho más fácilmente los posibles errores que puedan surgir.

2.1.2. Modelo Vista Controlador

Es un patrón de arquitectura de software para diseñar aplicaciones, que separa la lógica de la aplicación de la capa de presentación o GUI (Graphic User Interface), esta arquitectura consta de tres capas:

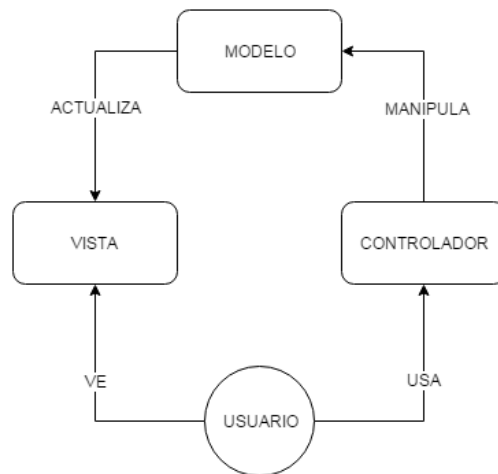


Figura 2.1: Diagrama MVC - Fuente: Elaboracion Propia

1. Modelo: Capa encargada de la lógica del negocio, la persistencia de los datos y la representación del modelo de datos.

2. Vista: Capa responsable de la presentación de los datos en un formato adecuado y amigable para el usuario final [6], además provee elementos gráficos que permiten hacer determinado grupo de acciones sobre los datos.
3. Controlador: Responde a los eventos provocados por el usuario en la vista en el modelo, también se encarga de enviar datos del modelo a la vista [6], por ejemplo el registro de una forma.

Este patrón de arquitectura es bastante ventajoso para aplicaciones a gran escala porque permite mayor desacople entre los componentes de la misma.

2.1.3. Patrón Abstract Factory

El Objetivo del patrón *Abstract Factory* es la creación de objetos agrupados en familias sin tener que conocer las clases concretas destinadas a la creación de estos objetos[3]. La estructura típica de este patrón es la siguiente:

- Cliente: La clase que llamará a la factoría adecuada.
- Abstract Factory: Es la definición de las interfaces de la factoría. Debe de proveer un método para la obtención de cada objeto, por ejemplo: “crearTipoObjeto1()”.
- Factorías Concretas: Estas son las diferentes familias de objetos, proveen la instancia concreta que se creará.
- Clase Abstracta: Definición de las interfaces para la familia de objetos genéricos.
- Clase Concreto: Implementación de las diferentes clases.

2.1.4. Frameworks de desarrollo aplicaciones web

Un “framework” de desarrollo o marco de trabajo es un conjunto grande de código fuente compuesto por clases, funciones, estructuras de datos, organizado bajo cierta arquitectura que puede servir para desarrollar una aplicación web rápidamente. Se puede pensar un framework como una aplicación medio lista que se puede extender y moldear de acuerdo a las necesidades requeridas [5], los “frameworks” de desarrollo por lo regular ofrecen un estilo para codificar bien definido y buenas prácticas que aseguran el buen diseño y rendimiento de una aplicación, sin embargo en su mayoría son lo suficientemente flexibles para permitir hacer cambios profundos en el código fuente e integraciones necesarias dependiendo de las necesidades. Las grandes ventajas de los *frameworks* son la elegancia del código de la aplicación y la reducción en el riesgo de errores. Actualmente algunos de los *frameworks* de desarrollo web escritos en el lenguaje de programación PHP más populares según el portal hongkiat ¹ son:

- Laravel ²

¹www.hongkiat.com/

²<https://laravel.com/>

- [Symfony](https://symfony.com/)³
- [CodeIgniter](https://www.codeigniter.com/) ⁴

³<https://symfony.com/>

⁴<https://www.codeigniter.com/>

Capítulo 3

Módulos para la gestión y reserva de beneficios y venta de membresías

En el presente capítulo se detalla lo aportado a los módulos del aplicativo de la clínica IMEKT abordados en ese trabajo.

3.1. Módulo de gestión de beneficios

3.1.1. Diagnóstico Inicial

Desde este módulo se puede administrar la información más relevante para el funcionamiento de la clínica, este tiene como fin el registro, la visualización y la modificación de los datos de líneas de venta, sublíneas de venta, beneficios, organizaciones y contactos de la clínica, información que será la base de las operaciones de venta y reserva que se realizarán posteriormente. El registro de la información anteriormente mencionada se debe hacer siguiendo un orden claro, detallado en la Figura 3.1.



Figura 3.1: Diagrama proceso registro beneficios - Fuente: Elaboracion propia

El proceso de desarrollo de este módulo estaba totalmente completado al iniciar este trabajo, sin embargo existían algunos errores críticos:

Este módulo permite registrar la información de dos formas: una manual donde es necesario rellenar formularios de registro y una automática (también llamada parametrización) donde es necesario guardar la información en un archivo plano formato CSV¹, donde los registros se separan por saltos de línea y los datos de cada registro por medio de comas u otro carácter. Este es un ejemplo de la estructura un archivo CSV:

CODIGO, NOMBRE, PRECIO, TIPO COD001, BENEFICIO 1, 20000, 2 COD002, BENEFICIO 2, 40000, 1
--

Estos archivos luego son subidos al aplicativo que se encarga de almacenar la información que estos contienen en la base de datos, este último método no estaba implementado en su totalidad, Dado lo anterior los usuarios de la plataforma se veían obligados a registrar la información de aproximadamente 200 registros relacionados con líneas de venta, sublíneas de venta, beneficios, organizaciones y contactos de forma ma-

¹Comma Separated Value

nual. No obstante al registrar y modificar manualmente la información relacionada los inventarios de beneficios y sublineas presentaban incoherencias. Evidenciando errores en la codificación del aplicativo.

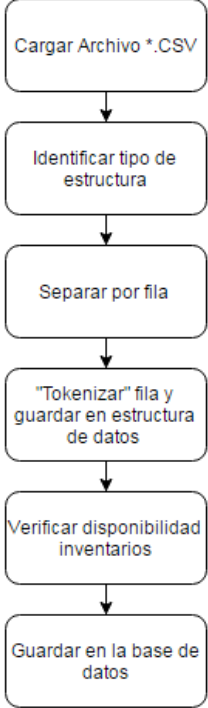


Figura 3.2: Diagrama de flujo carga archivos planos - Fuente: Elaboracion propia

3.1.2. Resultado del proceso de rediseño



Figura 3.3: Vista principal modulo archivos planos - Fuente: Elaboracion propia

Para rediseñar este módulo fue esencial realizar pruebas manuales por cada funcionalidad para detectar los errores, con estos ya identificados y un balance de estado del modulo, lo primordial fue refactorizar el codigo fuente del mismo, para después planear, modelar y codificar el método automático de registro de beneficios. Posterior al proceso de rediseño de este módulo el registro de los datos relacionados con líneas, sublíneas, beneficios, organizaciones y contactos se hace de manera apropiada, sin presentar inexactitudes, Además el proceso de parametrización de los datos quedó completamente funcional añadiendo soporte para las siguientes parametrizaciones solicitadas por el usuario:

- Registrar organizaciones y contactos.
- Registrar espacios físicos.
- Registrar beneficios.
- Relacionar beneficios con sublíneas.
- Relacionar beneficios con grupos.

3.2. Módulo Venta de membresías

3.2.1. Diagnóstico Inicial

El Módulo de venta de membresías tiene como fin el registro de las ventas realizadas por los vendedores de la clínica. La clínica IMEKT agrupa los beneficios en sublíneas,

una sublínea posee varios beneficios y un beneficio puede ser parte de varias sublíneas. La clínica solo comercializa sublíneas o beneficios agrupados, también llamadas membresías. NO se venden beneficios por separado.

Al registrar la venta de una membresía se requiere vincular datos de la sublínea que se comercializa con los datos del tercero, hacer descuentos en los inventario de los beneficios y de la sublínea, y complementar información del tercero como su identificación, representante legal en caso de ser una compañía, datos de contacto, entre otra información.

Luego de registrada una venta esta debe ser evaluada por el coordinador de ventas de la clínica quien decide si la aprueba o la rechaza, en el caso de ser aprobada el sistema creará un nuevo usuario en la plataforma que contará con alcance para usar el módulo de reserva dándole la posibilidad de reservar y visualizar todos los detalles de los beneficios que adquirió, este modulo sera detallado en la siguiente seccion. En el caso de ser rechazada el sistema revertira los descuentos realizados en los inventarios de la sublinea y los beneficios al momento del registro.

El proceso de desarrollo de este módulo no estaba totalmente completado al iniciar este trabajo, de las anteriores funcionalidades el aplicativo cumplía satisfactoriamente con el registro, aprobación y rechazo de la venta. El formulario de venta no contaba con los campos necesarios para registrar la totalidad de la información requerida del cliente, y al aprobar una venta el sistema no registraba el nuevo usuario.

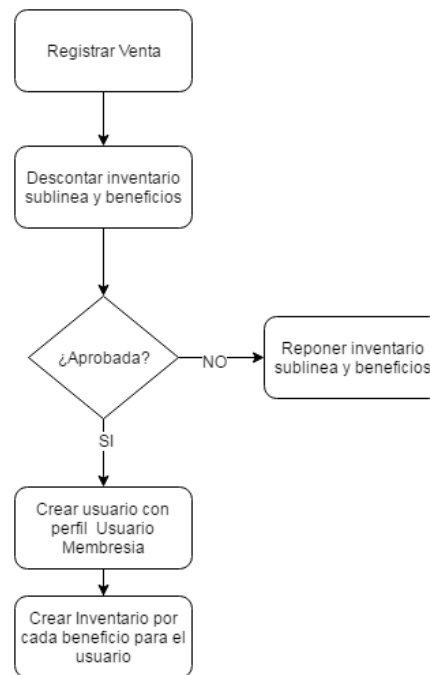


Figura 3.4: Diagrama flujo autorización venta - Fuente: Elaboracion propia

3.2.2. Resultado del Proceso de Rediseño

Para rediseñar este modulo fue necesario refactorizar el codigo fuente del modulo y la codificacion de nuevas funcionaidades Luego del este proceso para el módulo de venta de membresías en el formulario de venta se agregaron los campos faltantes para el registro de la totalidad de la información requerda del cliente. Además se desarrollo la funcionalidad para que luego de aprobada una venta automaticamente se registrara un nuevo usuario y se le asignara: un inventario por cada beneficio adquirido durante 5 años, las credenciales de ingreso (nombre de usuario y contraseña) relacionadas con la informacion de la venta y el perfil “Usuario Memebresia” para que tuviera el alcance de ingresar al aplicativo visualizar y reservar sus beneficios.

Martes, 06 de Diciembre de 2016

SISTEMA DE GESTIÓN DE VENTAS

RAUL ORTIZ JARAMILLO

MODULO DE VENTAS

REGISTRAR VENTA DE SUB LINEA

SELECCIONE LA SUB LINEA A VENDER

SELECCIONAR CLIENTE

INFORMACION DEL CLIENTE

VENDEDOR(A)

IDENTIFICACION CV COZSO016	NOMBRE RAUL ORTIZ JARAMILLO	USUARIO RORTIZ	TELEFONO EXT	TELEFONO CELULAR 316 6323924	EMAIL kreaciongrafica@gmail.com
-------------------------------	--------------------------------	-------------------	-----------------	---------------------------------	------------------------------------

TIPO IDENTIFICACION EMPRESA
ADULTO SIN IDENTIFICACION

No IDENTIFICACION EMPRESA*

DIRECCION DE LA EMPRESA*

TELEFONO DE LA EMPRESA*

INFORMACION REPRESENTANTE LEGAL

NOMBRE REPRESENTANTE LEGAL EMPRESA*

EMAIL REPRESENTANTE LEGAL*

TELEFONO REPRESENTANTE LEGAL EMPRESA*

FECHA DE VENTA DE LA SUB LINEA

[dd-mm-aaaa]

CÓDIGO FIDUCIARIO

ZONA UT

SUR OCCIDENTAL

FECHA DE INICIO DE LA VIGENCIA DE LA MEMBRÍA

[dd-mm-aaaa]

FECHA DE FINALIZACIÓN DE LA VIGENCIA DE LA MEMBRÍA

CALCULAR FIN VIGENCIA

OBSERVACIONES O DETALLES DE LA VENTA

[CONSULTAR](#)
[ADMINISTRACIÓN](#)
[CLIENTES](#)
[REPORTE DE VENTAS](#)
[USUARIOS](#)
[VENTAS](#)
[ARCHIVOS PLANOS](#)
[PARAMETROS INICIALES](#)
 ADMINISTRACION MODULOS
[ADMINISTRACION DE MODULOS](#)
 AGENDAMIENTO
[AGENDAMIENTO](#)

Figura 3.5: Vista principal formulario de ventas - Fuente: Elaboracion propia

3.3. Módulo de Reserva de beneficios

Desde este modulo los terceros pueden visualizar y reservar los beneficios obtenidos en la compra de la membresía, para esto el aplicativo debe proveer a usuario una interfaz que le permita conocer todos los detalles de los beneficios que adquirió y permitir al usuario reservar sus beneficios. Por otra parte el coordinador de la clinica tendra acceso a los detalles de todas las reservas realizadas por los terceros y los diferentes coordinadores acceso a las reservas de cada grupo de beneficios que administran (publicidad, web, etcetera) Para el desarrollo de este módulo se siguieron las etapas de análisis, diseño, desarrollo, pruebas de aceptación y despliegue.

3.3.1. Fase de Análisis

En la etapa de análisis de este módulo se cumplieron reuniones presenciales y virtuales con el cliente según la metodología de IPSOFT, de estas reuniones se derivaron varias actas redactadas voluntariamente por el cliente, donde manifestó en un lenguaje informal las funcionalidades requeridas para este nuevo módulo, ademas los perfiles que tendrian acceso al mismo y el alcance de cada uno de los perfiles. Luego se formalizaron estas actas en requerimientos e historias de usuario, esta etapa se desarrollo sin contratiempos arrojando como resultado: 2 actas y 27 requerimientos.

3.3.2. Fase de Diseño

En la etapa de diseño se construyeron los artefactos de diseño esenciales: el diagrama de clases y el diagrama relacional. Con el apoyo del director del equipo de desarrollo, se modelo y se codifico el modulo basándose en la arquitectura M.V.C (Modelo Vista Controlador).

3.3.3. Fase de codificación

En la etapa de codificación del módulo de reserva de beneficios se usó el “framework” SIIS desarrollado por la empresa IPSOFT en el lenguaje de programación PHP. Se implementaron los modelos de datos descritos en el diagrama entidad relación y la lógica del negocio plasmada en los requerimientos, también se diseñaron e implementaron las vistas que permiten al usuario interactuar con el sistema. Como resultado de la fase de codificación se tuvo un módulo de reserva funcional acorde a los requerimientos expresados por el usuario hasta la fecha. Con diferente alcance para cada perfil.

Para el perfil usuario con membresía:

- Visualizar lo detalles de los beneficios adquiridos en la venta
- Reservar los beneficios que requieran la carga de un documento
- Visualizar los documentos subidos y su estado (espera, aprobado, rechazado)

Para el perfil coordinador total:

- Visualizar las reservas hechas por los clientes (Usuarios con membresía).
- Visualizar los documentos subidos por los clientes (Usuarios con membresía).
- Cambiar el estado de los documentos subidos por los usuarios de espera a aprobado, de espera a rechazado.

Para los coordinadores de cada área, la visualización de los beneficios, los documentos y su estado.

- Visualizar los detalles de los beneficios pertenecientes a su grupo adquiridos en la venta.
- Reservar los beneficios pertenecientes a su grupo que requieran la carga de un documento.
- Visualizar los documentos subidos y su estado (espera, aprobado, rechazado).

3.3.4. Pruebas de aceptación

Para aprobar las nuevas funcionalidades integradas en el aplicativo se usaron pruebas de aceptación, las pruebas de aceptación consistieron en la revisión de las funcionalidades por parte del cliente. En este caso el cliente probó cada funcionalidad implementada a medida que se integraba en el aplicativo, y redactó un acta consolidando los resultados de comportamiento del aplicativo. El resultado de este proceso fue la verificación de la totalidad de los requerimientos.

Martes, 06 de Diciembre de 2016

RAUL ORTIZ JARAMILLO

SISTEMA DE GESTIÓN DE VENTAS

imekt clinic
TRANSACCIONES DE VENTAS Y SERVICIOS MÉDICOS

- REPORTE DE VENTAS
- USUARIOS
- VENTAS
- ARCHIVOS PLANOS
- PARAMETROS INICIALES
- ADMINISTRACION MODULOS
 - ADMINISTRACION DE MODULOS
- AGENDAMIENTO
 - AGENDAMIENTO
- CONFIGURACION DEL USUARIO
 - CONFIGURACION USUARIOS
- DICCIONARIO
 - DICCIONARIO

AGENDAMIENTO

IDENTIFICADOR
SELECCION UN USUARIO

Datos Usuario

ID USUARIO: 281

NOMBRE: C00001

SELECCION UN USUARIO

SELECCION CLIENTE

SELECCION UN USUARIO

SELECCION UN USUARIO

SELECCION UN USUARIO

Fecha Reserva

[dd-mm-aaaa] Desde DD/MM/AAAA

6 MESES DE DURACION

Archivo: continuaci_n_.pdf

Acceptar

MES	DICIEMBRE	JAN	FEB	MAR	ABR	MAY	JUN	JUL	AUG	SEPT	OCT	NOV	ANIO	2016	SAB	SUN	DOM										
	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

E EXEC : 0.187

Agendamiento controladorFormasAgendamiento

Figura 3.6: Vista principal modulo reserva beneficios - Fuente: Elaboracion propia

3.4. Manuales de usuario para los módulos: venta de membresías, gestión de beneficios y reserva de beneficios

Para capacitar a los usuarios finales en el uso del aplicativo de la clínica IMEKT se crearon manuales de usuario. En este caso se optó por la utilización de una wiki ², debido a su modularidad, facilidad en su lectura y escritura. Se usó una librería codificada en JavaScript llamada MDWIKI ³, que permite escribir las páginas y secciones de la wiki en un formato MARKDOWN ⁴ esta librería renderiza el contenido de los archivos fuente en el lado del cliente, es decir no tiene ninguna dependencia en el lado del servidor, para su correcto funcionamiento basto con servir el archivo principal en formato HTML y todas las dependencias que son archivos en formato (CSS, JavaScript y MARKDOWN), para servir dichos se uso el servidor web apache.

Se escribieron manuales de usuario para los módulos de gestión de beneficios, venta de membresías y reserva de beneficios, con la firme convicción de acortar la curva de aprendizaje en el funcionamiento del aplicativo por parte de los usuarios.



Figura 3.7: Primer pantallazo WIKI IMEKT- Fuente: Elaboracion Propia

²Sistema de trabajo informático utilizado en los sitios web que permite a los usuarios modificar o crear su contenido de forma rápida y sencilla.

³www.mdwiki.info/

⁴<https://es.wikipedia.org/wiki/Markdown>

Bienvenido | Modulo Archivos Planos | Modulo Administracion | Modulo Ventas | Modulo Agendamiento

[Descripcion](#)

[Menu Lineas de venta](#)

[Menu Sublineas de venta](#)

[Menu Beneficios](#)

[Menu Asignar beneficios a sublineas](#)

[Menu Organizaciones y Contactos](#)

EN ESTE CASO LOS DATOS SON SEPARADOS POR PUNTO Y COMA (,)
EJEMPLO:

```
# CODIGO BENEFICIO; NOMBRE BENEFICIO; DESCRIPCION;
CODBEN001; BENEFICIO UNO; ESTE ES UN BENEFICIO DE PRUEBA
CODBEN002; BENEFICIO DOS; ESTE ES UN BENEFICIO DE PRUEBA2
(...)
(...)
CODIGO-N; BENEFICIO-N; DESCRIPCION-N
```

Al ingresar a cada uno de los menus, tendra el mismo formulario que le permitira adjuntar el archivo que contiene la informacion a parametrizar, y un boton para subir el archivo.

SUBIR ARCHIVO CSV PARA ASIGNAR BENEFICIOS A GRUPOS

ARCHIVO	No se ha seleccionado ningún archivo.		
SEPARADOR	<input type="radio"/> COMA [,]	<input checked="" type="radio"/> PUNTO Y COMA [.]	<input type="radio"/> TABS []
ARCHIVO CON ENCABEZADO	<input checked="" type="radio"/> SI	<input type="radio"/> NO	
<input type="button" value="Aceptar"/> <input type="button" value="Volver"/>			

Usted debera seguir un orden estricto para subir los archivos, esto garantizara la correcta parametrizacion de la informacion. Este orden es:

- Lineas de venta
- Sublineas de venta
- Espacios Fisicos y Publicitarios
- Beneficios
- Asignar Beneficios a Sublineas
- Organizaciones y Contactos
- Asignar clientes a agentes libres.

Menu Lineas de venta

Figura 3.8: Detalle WIKI IMEKT (Modulo Archivos Planos) - Fuente: Elaboracion propia)

Capítulo 4

Aspectos del desarrollo de software

La metodología de desarrollo usada por la empresa IPSOFT-SA es una adaptación de la metodología de desarrollo de software tradicional llamada cascada. El proceso inicia con una reunión entre los interesados, donde se expone de forma general el proyecto por parte del cliente y el alcance del framework SIIS por parte de los representantes de IPSOFT-SA. Posteriormente se siguen cumpliendo reuniones para especificar los requerimientos de la nueva implementación donde se precisan formalmente las nuevas funcionalidades que tendrá el aplicativo. Luego se diseñan los nuevos módulos y funcionalidades usando artefactos de diseño como diagramas de clase, diagramas entidad-relación y diagrama relacional. Después se codifican los nuevos módulos y funcionalidades basándose en los artefactos de análisis y diseño anteriormente elaborados, la verificación de los requisitos se hace por parte del cliente mediante pruebas de aceptación. Por último se hace el despliegue en el servidor de producción de la nueva implementación, asimismo se parametrizan los datos del cliente en el nuevo aplicativo y los usuarios del sistema son capacitados presencialmente por personal de IPSOFT-SA. Después del proceso de desarrollo es probable que el cliente no haya expresado algunos requisitos o que las funcionalidades existentes presenten fallas, en estos casos el cliente especificará los nuevos requisitos y se llevará a cabo una iteración a pequeña escala del proceso de desarrollo para integrar las nuevas funcionalidades o correcciones al aplicativo.

Para este trabajo se usó fielmente la metodología propuesta por IPSOFT-SA y se siguieron las fases de análisis, diseño, codificación, pruebas, despliegue descritas a detalle en los próximos párrafos. Es importante aclarar que la documentación producida a lo largo del proceso de desarrollo no se puede anexar al presente documento por políticas de la empresa IPSOFT-SA y por decisión de los directivos de la clínica IMEKT, ya que expone de forma explícita su modelo de negocio e información confidencial y de uso exclusivo de estas dos compañías.

4.1. Fase de Análisis

Con el fin de obtener toda la información relacionada con el modelo de negocio de IMEKT CLINIC y funcionamiento detallado del proceso de reserva de los beneficios de la clínica se llevaron a cabo reuniones presenciales y virtuales con el encargado del proyecto IMEKT, de estas reuniones se derivó la siguiente documentación.

4.1.1. Actas

Estos documentos redactados por representantes de la clínica IMEKT contienen de forma muy general y en un lenguaje informal, la descripción y justificación de los nuevos módulos y funcionalidades esperadas en el aplicativo para la clínica.

4.1.2. Anexos

Estos documentos contienen información detallada para la parametrización de datos de la clínica en el aplicativo, por ejemplo la lista de beneficios, los permisos de los usuarios sobre los módulos y beneficios, entre otros.

4.1.3. Requerimientos

Para la especificación de los requisitos del sistema se usaron requerimientos con el propósito de expresar de forma atómica y puntual cada funcionalidad del sistema, se trabajó sobre una plantilla proporcionada por la empresa IPSOFT-SA que contiene lo siguientes campos.

- Identificador: Es un identificador numérico único para cada requerimiento.
- Título: El título del requerimiento o su resumen de la funcionalidad.
- Descripción: El cuerpo del requerimiento, donde se desglosa el mismo siguiendo la siguiente estructura:

[LUGAR, TIEMPO, EVENTO, OBJETO]+ “Debe, deberá, no debe, no deberá” + [ACCIÓN, VERBO, SENTENCIA]+ [A QUIÉN, SUJETO]+ [RESULTADO ESPERADO, DESTINO]

- Usuarios: Perfil(es) que podrán ejecutar esta funcionalidad.

4.1.4. Historias de Usuario

Se usaron como un sustituto para los casos de uso y para dar una idea de cómo los requerimientos interactúan entre sí al ser un artefacto de análisis con un nivel de detalle más amplio, que permite que el proceso de implementación de una funcionalidad del sistema sea más fácil de comprobar, ya que, al contar con criterios de aceptación el desarrollador tiene un objetivo más claro y acotado al momento de codificar el requisito del sistema. Se trabajó con un plantilla que contiene los siguientes campos:

- **Identificador:** Un identificador alfanumérico único para cada Historia de usuario
- **Rol:** Es el rol que está desempeñando el usuario cuando utiliza la funcionalidad que se está describiendo
- **Funcionalidad:** Representa la función que el rol quiere o necesita hacer en el sistema que se está desarrollando.
- **Resultado:** Lo que el rol necesita lograr al ejecutar la acción.
- **Criterio de Aceptación:** Describe el contexto del escenario que define un comportamiento.
- **Comportamiento Esperado:** Dado el contexto y la acción ejecutada por el usuario, la consecuencia es el comportamiento del sistema en esa situación.

4.2. Fase de Diseño

4.2.1. Modelo de Datos

Luego de tener una especificación clara y formal del comportamiento esperado del sistema, vino la creación y adaptación de modelos y estructuras de datos con el propósito de abstraer de la mejor forma posible todos los datos involucrados en el modelo de negocio de la clínica IMEKT, se usó el artefacto de diseño diagrama entidad-relación y diagramas de clases para lograr este fin.

4.3. Diseño de la arquitectura

Teniendo en cuenta que el proceso de desarrollo está relacionado con la extensión del “framework” SIIS, la arquitectura del aplicativo ya está definida, el aplicativo se basa en el patron de arquitectura M.V.C (Modelo-Vista-Controlador). El “framework ” SIIS está diseñado basado en el patrón de arquitectura de tres capas conocido como M.V.C (Modelo-Vista-Controlador) , donde la capa del modelo es la que se encarga de la interacción con la base de datos, en esta capa se declaran las sentencias SQL necesarias para registrar, leer, editar y eliminar los datos y modelos del aplicativo. La vista se encarga de mostrar la información de forma legible y amigable al usuario del aplicativo, también de mostrar los componentes gráficos necesarios para que el usuario interactúe con el aplicativo. Finalmente el controlador se encarga de responder a eventos, generalmente acciones del usuario, e invoca funciones del modelo, de igual forma envía información a la vista para su presentación al usuario, por tanto se puede decir que el controlador actúa como un intermediario entre el modelo y la vista.

4.4. Fase de Codificación

Previo a la etapa de codificación, por parte de la empresa IPSOFT-SA se me brindó una capacitación, donde se me transmitió el conocimiento necesario acerca del funcionamiento del *framework* SIIS esta capacitación se hizo mediante el acompañamiento personal por el director del área de desarrollo y poniendo a disposición una wiki que contiene la documentación del framework, esto es, información detallada de las clases, métodos, módulos y funcionalidades que lo componen.

En la etapa de codificación se respetó el estilo de programación de la empresa IPSOFT-SA, no se añadieron dependencias adicionales a las existentes, y se procuró la utilización de buenas prácticas de desarrollo. Durante esta etapa como primera medida, se centraron esfuerzos en la implementación de las entidades y sus relaciones en la base datos al igual que las consultas en lenguaje SQL (Structured Query Language) para obtener, registrar, actualizar y eliminar información de los mismos, para ese fin se usó el motor de base de datos POSTGRESQL ¹ versión 9. Luego se definieron los modelos que permiten obtener y modificar los datos, esto usando una capa intermedia denominada “Database Access Abstraction Layer” (DAAL) o capa de abstracción de acceso a la base de datos, que facilita la comunicación con la base de datos y el aplicativo, exponiendo una API consistente que permite desde determinado lenguaje de programación acceder a las funciones del motor de base de datos, en este caso se usó ADOdb una librería que abstrae la comunicación entre PHP y algunos motores de base de datos. Además se definieron los controladores que capturan la información desde la vista mediante formas, validan dicha información y lanzan funciones en el modelo, de forma análoga obtiene información del modelo y la envía a la vista, para la definición de los modelos y los controladores se usó el lenguaje de programación PHP.

Para realizar comunicaciones asíncronas entre el servidor y el cliente (AJAX), se usó la librería Xajax ² escrita en PHP, esta librería permite realizar llamadas asíncronas sin escribir métodos o funciones en el lado del cliente, es decir la librería se encarga de traducir una función en el lenguaje PHP a el lenguaje JavaScript e inyectarla en tiempo de ejecución en el navegador. Por último se implementaron las interfaces de usuario, que son artefactos gráficos que permiten al usuario interactuar con el aplicativo, para ese propósito se usó el lenguaje de hipertexto HTML, el lenguaje de estilos CSS y el lenguaje de programación JavaScript apoyado en la librería JQUERY ³.

Para facilitar la escritura y edición del código fuente del aplicativo se usó el editor de texto ATOM ⁴ desarrollado por la compañía Github⁵ que cuenta con licencia MIT que permite sea usado de forma gratuita, ATOM es muy potente y cuenta con componentes gráficos muy amigables, además cuenta con una gran comunidad que brinda soporte y diseña temas y paquetes para personalizar el editor en distintos aspectos y para llevar a cabo tareas específicas, en este caso se integraron paquetes gratuitos para la revisión

¹www.postgresql.org.es/

²<http://www.xajax-project.org//>

³<https://jquery.com/>

⁴<https://atom.io/>

⁵github.com

de la sintaxis de los lenguajes: PHP, JavaScript, HTML, CSS. Y un visualizador de archivos CSV, estos fueron los paquetes que se integraron al editor:

- language-php ⁶.
- language-css ⁷.
- tablr ⁸.
- linter ⁹.

4.5. Pruebas de Aceptación

La empresa IPSOFT no cuenta con una metodología para realizar pruebas unitarias o funcionales, dado esto para la verificación de la correcta implementación de las funcionalidades se realizaron pruebas de aceptación por parte del cliente, se hicieron de forma manual, simulando el funcionamiento de la plataforma.

4.6. Despliegue de la aplicación

Para el proceso de despliegue de la aplicación se usó el cliente FTP WINS CP ¹⁰ para copiar el código fuente en el servidor de despliegue, las dependencias que debe cumplir el servidor de despliegue son las siguientes:

- Sistema Operativo LINUX.
- Motor de base de datos POSTGRES QL ⁹.
- Servidor Web Apache ¹¹.
- Lenguaje de programación PHP versión 5 ¹².

⁶<https://atom.io/packages/language-php>

⁷<https://atom.io/packages/language-css>

⁸<https://atom.io/packages/tablr>

⁹<https://atom.io/packages/linter>

¹⁰<https://winscp.net/>

¹¹<https://httpd.apache.org/>

¹²www.php.net/

Capítulo 5

Conclusiones

Al finalizar la pasantía se integraron dos módulos rediseñados y un módulo desarrollado al aplicativo de IMEKT CLINIC de acuerdo a las necesidades del cliente, dicho aplicativo permite la gestión de beneficios, la venta de membresías y la reserva de beneficios. El aplicativo se codificó usando el “framework” SIIS desarrollado por los ingenieros de la empresa IPSOFT.

En el presente trabajo de grado modalidad pasantía se propusieron los siguientes objetivos específicos, de los cuales se concluye:

- **Rediseñar módulo de gestión de beneficios para IMEKT CLINIC y Rediseñar venta de membresías para IMEKT CLINIC**

Se propuso el rediseño de dos módulos existentes en el aplicativo de la clínica IMEKT, después de terminado este proceso se corrigieron procesos importantes e indispensables para su funcionamiento, con las nuevas funcionalidades integradas se mejoró la experiencia de usuario, y se agilizo el proceso de registro de beneficios.

- **Desarrollar módulo de reserva de beneficios para IMEKT CLINIC**

Se propuso el desarrollo de un módulo para la reserva de beneficios de la clínica, el módulo de agendamiento cuenta con las funcionalidades necesarias que permiten al usuario, luego de autorizada su venta, ingresar a la plataforma para visualizar sus beneficios. El usuario puede adjuntar archivos a sus beneficios en el caso de necesitarlo, y los administradores de la clínica podrán visualizar, rechazar o aprobar dichos archivos.

- **Construir manuales de usuario para los módulos: venta de membresías, gestión de beneficios y reserva de beneficios.**

Los usuarios finales podrán visualizar los manuales de uso de la plataforma en un formato más amigable mediante una WIKI.

Respecto las metodologías y tecnologías usadas por la empresa IPSOFT-SA:

Referente a la metodología de desarrollo, si en lugar de una metodología tradicional se implementara una metodología de desarrollo ágil como SCRUM ¹ se mejoraría notablemente la comunicación entre los integrantes de los equipos de desarrollo y soporte, además haría el proceso de desarrollo software más expedito mejorando la productividad y haciendo el proceso de software mucho más ameno principalmente para el cliente.

Referente a las tecnologías, el proyecto IMEKT y los demás proyectos desarrollados por la empresa IPSOFT están contruidos con base en el *framework* SIIS desarrollado por los ingenieros de la misma empresa, este *framework* sigue una arquitectura M.V.C (Modelo Vista Controlador). En ocasiones para el desarrollador implementar funcionalidades relativamente sencillas se convierte en una tarea compleja debido a la errónea delegación en las responsabilidades en las tecnologías, se hace una sobrecarga de responsabilidades sobre la capa del servidor, haciéndolo responsable, por ejemplo, de actualizar el D.O.M (Document Object Model) de forma asíncrona mediante la librería Xajax.

Una correcta delegación de responsabilidades entre las tecnologías podría ser solucionada usando una arquitectura RESTFUL ², donde la aplicación contaría con dos capas, la primera capa o “BACKEND” que por medio de servicios web expondría los datos en un formato JSON mediante los metodos HTTP : GET, POST, DELETE, PUT. Y en la segunda capa o “FRONTEND” la interfaz entre el usuario y el sistema que consistiría en una aplicación corriendo en el navegador preferiblemente desarrollada en el lenguaje de programación JavaScript, con los métodos necesarios para consumir los servicios de la primera capa mediante llamadas asíncronas (AJAX), dado el reciente avance de las herramientas de desarrollo web, también se podría utilizar un “framework” de desarrollo web del lado del cliente, M.V.C como AngularJS ³ o EmberJS ⁴, que dada la experiencia además de agilizar el desarrollo de la aplicación, garantizaría su robustez y velocidad. Otra opción que se puede considerar es usar algún “framework” de desarrollo que esté basado en el patrón de arquitectura M.V.C (Modelo Vista Controlador), la ventaja de estos “frameworks” es su gran soporte, reflejado en sus grandes comunidades y documentación, además que la mayoría exponen una filosofía para desarrollar aplicaciones, garantizando buenas prácticas de programación y buen rendimiento en la aplicación.

El gran inconveniente con las dos últimas opciones es que implicaría reescribir los aplicativos de los proyectos desarrollados por la empresa IPSOFT exigiendo gran cantidad de tiempo, capital y recurso humano. La opción más factible es mejorar el diseño y arquitectura del framework SIIS; esto se puede lograr añadiendo una interfaz ORM (Object Relational Mapping) a la capa del modelo, existen bastantes librerías que gozan de buen soporte y popularidad como Doctrine ⁵ o RedBeanPHP ⁶ que pueden ayudar a cumplir este propósito, otra mejora factible es la incorporación de un motor de plantillas en la aplicación esto facilita notablemente el desarrollo si se compara con el método utilizado actualmente mediante la concatenación de cadenas, también existen bastantes

¹<https://www.scrum.org/>

²<http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>

³<https://angularjs.org/>

⁴emberjs.com/

⁵www.doctrine-project.org/

⁶www.redbeanphp.com/

librerías para cumplir este propósito como Mustache ⁷, Smarty ⁸.

En cuanto a la interfaz gráfica el “framework”, SIIS usa hojas de estilo desactualizadas, no soportadas por dispositivos móviles y no optimizadas lo que además de producir una experiencia negativa en el usuario, alarga sustancialmente el tiempo de carga de la aplicación. Sería pertinente el uso de un “framework” HTML/CSS/JavaScript como Bootstrap⁹ o Foundation¹⁰, con soporte para dispositivos móviles, de la mano de “task runners” como GULPJS¹¹, para el diseño y optimización de las vistas de los aplicativos.

Mi experiencia al hacer parte de un proyecto real de desarrollo de software fue muy excitante y enriquecedora en todos los aspectos, desde el aspecto personal hasta el profesional, conocí la realidad de un ambiente laboral, y dada la relación directa con el cliente adquirí experiencia en el levantamiento de requerimientos, relaciones interpersonales y comunicación asertiva. Además al pertenecer a un equipo de desarrollo aprendí a trabajar en equipo.

Mejore notablemente mis habilidades en el diseño de los modelos de datos y su implementación, en la resolución de problemas, y en las tecnologías PHP, POSTGRESQL, HTML, CSS, JAVASCRIPT.

⁷<https://mustache.github.io/>

⁸www.smarty.net/

⁹getbootstrap.com/

¹⁰foundation.zurb.com/

¹¹gulpjs.com

Bibliografía

- [1] La difusión del conocimiento. *Interciencia*, 33:786 – 786, 11 2008.
- [2] Eduardo Barboza Besada. Avances de la cirugía en los últimos 30 años. *DIAGNOSTICO*, 47:122 –128, 9 2008.
- [3] L. DEBRAUWER and Y. EVAIN. *Patrones de diseño en PHP: Los 23 modelos de diseño: descripciones y soluciones ilustradas en UML2 y PHP*. Expert IT. Ediciones ENI, 2015.
- [4] F. Durán, F. Gutiérrez, and E. Pimentel. *Programación orientada a objetos con Java*. Ediciones Paraninfo. S.A., 2007.
- [5] B. Porebski, K. Przystalski, and L. Nowak. *Building PHP Applications with Symfony, CakePHP, and Zend Framework*. IT Pro. Wiley, 2011.
- [6] S.A.C.T.D.A.V.Z. Sonia Jaramillo Valbuena. *Programación Avanzada en Java*. Elizcom S.a.s.