



SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE LOS ESTÁNDARES
BÁSICOS DE COMPETENCIAS EN LAS INSTITUCIONES EDUCATIVAS DE
COLOMBIA

DIEGO ALEXANDER PATIÑO GIL
201513003

WALDOR DRADA ARANGO
201556076

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Programa Académico de Ingeniería de Sistemas
Tuluá, Abril 29 de 2016



SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE LOS ESTÁNDARES
BÁSICOS DE COMPETENCIAS EN LAS INSTITUCIONES EDUCATIVAS DE
COLOMBIA

DIEGO ALEXANDER PATIÑO GIL
201513003

WALDOR DRADA ARANGO
201556076

Trabajo de grado para optar por el título de
Ingeniero de sistemas

Director:
Ing. Julián Andrés Rodas Laverde.

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Programa Académico de Ingeniería de Sistemas
Tuluá, Abril 29 de 2016

Nota de aceptación

Presidente del Jurado

Jurado

Jurado

Tuluá, Agosto 03 de 2016

Dedicatoria

A mis padres y hermanos quienes nunca dudaron de mí y me apoyaron en la toma de decisiones en cada etapa de la universidad, sus consejos fueron de mucho ayuda. ya que me dieron las fortalezas para nunca desfallecer en el camino y sobre todo para llegar a la cima de cumplir esta valiosa meta.

Diego Alexander Patiño Gil

A Dios por haberme permitido alcanzar este punto y haberme dado salud para conseguir mis objetivos, además de su infinita bondad y amor .

A mis padres y hermano por los ejemplos de perseverancia que los caracterizan y que me han infundado siempre, por el valor mostrado para salir adelante y por su amor.

A mi princesa por todo el apoyo incondicional y por siempre buscar la manera de mantenerme cuando todo iba mal, gracias por amarme como solo tú lo puedes hacer.

Waldor Drada Arango

Agradecimientos

En primer lugar, quiero agradecer a Dios, por su respaldo y apoyo en cumplir este objetivo.

A mis padres y hermanos que me ayudaron tanto económicamente y emocionalmente.

A mi amigo y compañero de trabajo de grado Waldor Drada Arango por su apoyo y la confianza que deposito en mí en el proceso del desarrollo del proyecto.

Al director de grado, Ingeniero y Magíster Julián Andrés Rodas Laverde, que nos guio y acompaño en este proceso, dándonos apoyo con sus conocimientos, sus experiencias y compromisos.

La Universidad del Valle sede Tuluá, a la escuela de ingeniería de sistemas y computación, y a todo el personal donde difundió conocimiento para lograr tener sólidas bases y lograr ejercerlas en situaciones que se presenten.

Diego Alexander Patiño Gil

Le agradezco a Dios por haberme guiado y acompañado a lo largo de la carrera, por ser esa fortaleza en los momentos difíciles y por brindarme una vida llena de salud, aprendizaje, experiencia, felicidad y sobre todo amor.

A mis padres Paola Andrea Arango Tigreros y Waldor Drada Guzmán por el enorme apoyo en todo momento, por cada uno de los valores que me han inculcado y brindarme la oportunidad de tener una excelente educación a lo largo de mi vida, sobre todo por su ejemplar forma de vivir.

A mi hermano Juan Sebastián Arias Arango por ser una parte importante de mi vida representando esa unión familiar, llenando mi vida de muchas alegrías y amor cuando más lo he necesitado.

A mi novia Stephania González Quintero por darle luz a mis días, gracias por hacerme el hombre más feliz, gracias por entenderme, apoyarme y estar a mi lado en todo momento.

A mi amigo Diego Alexander Patiño Gil por creer y confiar en mí y haber hecho de mi etapa universitaria una trayectoria de experiencias que nunca olvidare.

Al director de grado, Ingeniero y Magíster Julián Andrés Rodas Laverde, por habernos brindando el apoyo, tiempo, amistad y conocimientos que nos transmitió.

La Universidad del Valle sede Tuluá, por haberme permitido ser parte de ella y poder estudiar mi carrera, así como también a los diferentes docentes que brindaron sus conocimientos y su apoyo para seguir adelante día a día

Waldor Drada Arango

Índice General

1. Introducción	1
1.1. Descripción General.....	1
1.2. Planteamiento del Problema.....	2
1.3. Formulación del Problema	3
1.4. Justificación	3
1.5. Objetivos	4
1.5.1. Objetivo General	4
1.5.2. Objetivo Específicos	4
1.6. Estructura del documento.....	5
Marco de Referencia	6
2.1. Marco teórico	6
2.1.1. Arquitectura de la información (Modelo-Vista-Controlador).....	6
2.1.2. Laravel	7
2.1.3. Base de datos.....	9
2.1.4. MYSQL.....	9
2.1.5. Servidor Web	10
2.1.6. HTML	10
2.1.7. CSS (Cascading Style Sheets).....	10
2.1.8. AJAX (Asynchronous JavaScript And XML)	10
2.1.9. PHP	10
2.1.11. ADMINLTE.....	11
2.1.13. SELECT2.....	11
2.1.14. DATATABLES	11
2.1.16. INPUTMASK	12
2.1.17. IOSSLIDER	12
2.1.18. JASNY BOOTSTRAP	12
2.1.19. BOOTSNIPP	12
2.1.20. DATEPICKER.....	12
2.1.21. DATERANGEPICKER	13
2.1.22. CHARTJS	13

2.1.23.	FONTAWESOME	13
2.1.24.	CHECKED.....	13
2.1.25.	Procesos Agiles.....	13
2.1.26.	Descripción de la Metodología Iconix	14
2.1.27.	Estándares básicos de competencia.....	14
2.2.	Antecedentes	17
2.2.1.	Aplicaciones de Escritorio	17
2.2.2.	Aplicaciones Web	18
3.	Desarrollo del Proyecto	21
3.1.	Análisis de Requisitos	24
3.1.1.	Requerimientos funcionales.....	25
3.1.2.	Modelo de Dominio	27
3.1.3.	Prototipo Rápido	29
3.1.4.	Diagrama de Casos de Uso	30
3.2.	Análisis y Diseño Preliminar	34
3.2.1.	Descripción de Casos de Uso.....	34
3.2.2.	Diagrama de Robustez	38
3.3.	Revisión crítica del diseño	40
3.3.1.	Diagramas de navegación	40
3.3.2.	Diagramas de clases	44
3.3.3.	Diagrama de Modelo Relacional.....	44
3.4.	Implementación.....	46
3.4.1.	Herramientas usadas	50
3.4.2.	Descripción de la Interfaz	52
3.4.2.1.	Interfaz calendario año lectivo	53
3.4.2.2.	Interfaz reporte de sedes	56
4.	Pruebas	59
4.1.	Pruebas unitarias	59
4.2.	Pruebas con Tinker.....	62
5.	Conclusiones y trabajos futuros	67
5.1.	Conclusiones	67
5.2.	Trabajos futuros	68
6.	Bibliografía	69

Índice de figuras

Figura 1 Modelo Vista Controlador (MVC).....	6
Figura 2 Interacciones en las capas de laravel.....	7
Figura 3 Fases de la metodología ICONIX [51]	22
Figura 4 Modelo del dominio	28
Figura 5 Prototipo de la pagina de inicio del dashboard	30
Figura 6 Gestión de competencias académicas	31
Figura 7 Gestión de componentes	32
Figura 8 Caso de uso de reportes por áreas	33
Figura 9 Diagrama robustez crear estándar de competencia	39
Figura 10 Diagrama robustez configuración de componentes académicos	39
Figura 11 Diagrama robustez reporte de estudiantes.....	40
Figura 12 Diagrama de navegación de crear estándar de competencia	41
Figura 13 Diagrama de navegación configuración de componentes académicos.....	42
Figura 14 Diagramas de navegación reporte estudiante	43
Figura 15 Modelo entidad relacional.....	45
Figura 16 Diagrama de componentes	46
Figura 17 Ejemplo de migas de pan	50
Figura 18 Ejemplo de herramientas de bootsnipp	51
Figura 19 Interfaz del módulo de gestión de calendario	53
Figura 20 Configuración de fecha del año lectivo.....	53
Figura 21 Configuración de la cantidad de periodos.....	54
Figura 22 Configuración de fecha del año lectivo.....	55
Figura 23 Interfaz de edición del calendario del año lectivo.....	55
Figura 24 Gestión de reporte por sedes	56
Figura 25 Cargar las sedes del sistema.....	56
Figura 26 Configuración de datos para las sedes	57
Figura 27 Grafica de barra para el promedio de competencias	57
Figura 28 Grafica de barra para el promedio de estándares de competencia.....	58

Índice de tablas

Tabla 1 Modelo de estándares de competencias.....	15
Tabla 2 Requerimientos funcionales	25
Tabla 3 Caso de usos crear competencias	34
Tabla 4 Casos de uso configurar componentes académicos	35
Tabla 5 Casos de usos reportes por estudiantes.....	36
Tabla 6 Prueba unitaria crear año lectivo	59
Tabla 7 Prueba unitaria modificar año lectivo	59
Tabla 8 Prueba unitaria eliminar año lectivo.....	59
Tabla 9 Prueba unitaria de crear sedes	60
Tabla 10 Prueba unitaria de crear grupos	60
Tabla 11 Prueba unitaria de reportes por estudiante.....	60
Tabla 12 Prueba unitaria de reportes por grupos	61
Tabla 13 Prueba unitaria de reportes por sedes	61
Tabla 14 Prueba unitaria de reportes por ciclos	61
Tabla 15 Prueba unitaria de reportes de importar usuario	62
Tabla 16 Pruebas de unidad con tinker para la creación de calendario de año lectivo	63
Tabla 17 Pruebas de unidad con tinker para creación de competencias	64
Tabla 18 Pruebas de unidad con tinker para creación de estándares de competencias.....	65

Resumen

El presente proyecto se basó en una problemática que se genera alrededor de la gestión de calificaciones de las instituciones educativas de Colombia, donde no se está evidenciando las falencias que tienen los estudiantes con respecto a los estándares de competencia dadas por el Ministerio de Educación Nacional.

Para dar cumplimiento al problema propuesto se realizó el respectivo análisis, diseño y desarrollo de una aplicación web con modernas tecnologías para los procesos de competencia y estándares para determinar el rendimiento académico de los estudiantes, todo esto relacionado con la programación académica de la Institución. En la solución del proyecto, se basó en la metodología ICONIX y el uso de patrones arquitectónicos que plantea el Framework de Laravel, estableciendo un producto de alta calidad que desempeña las funcionalidades requeridas, ofreciendo los módulos más importantes para el planteamiento del problema como la gestión del año lectivo, la gestión académica, gestión de calificaciones y la más primordial la gestión de reportes donde se va evidenciar el rendimiento de cada estudiante por medio de los estándares de competencia.

Abstract

This project was based on a problem that is generated around the management of qualifications of educational institutions in Colombia, where is not demonstrating the shortcomings which students relative to the standards of competence are given by the Ministry of national education.

To comply with the proposed problem was the respective analysis, design and development of a web application with modern technologies for competitive processes and standards of competence to determine the academic performance of the students, all of this is related to the academic program of the institution. In the solution of the project, was based on the ICONIX methodology and use of architectural patterns arising from the Framework of Laravel, establishing a high quality product that shall perform the required functions, offering the most important modules for the approach of the problem such as the management of the school year, the management, academic qualifications and the most essential management reports where you are going to show the performance of each student through the standards of competition.

Capítulo 1

Introducción

1.1. Descripción General

Uno de las principales tareas del Ministerio de Educación Nacional (MEN) de Colombia es desarrollar políticas y lineamientos para mejorar la calidad de la educación. De esta manera, el MEN ha adelantado propuestas que han sido consignadas en la Ley General de Educación y en varios decretos posteriores. Una de las propuestas del MEN para la calidad de la educación es la reglamentación de los estándares básicos de competencia. Los Estándares son el producto de un trabajo interinstitucional y mancomunado entre el Ministerio de Educación Nacional y las facultades de Educación del país agrupadas en Ascofade (Asociación Colombiana de Facultades de Educación). Los estándares son una descripción de lo que el estudiante debe lograr en una determinada área, grado o nivel. Lo que buscan estos estándares es eliminar las tradiciones de aprendizaje de los alumnos relacionadas a la memorización de contenidos y se enfocan en el análisis y resolución de problemáticas. Los estándares fueron formulados con el propósito de ayudar a las instituciones educativas a definir el plan de área por cada grado buscando el desarrollo de las competencias en el tiempo [1].

A partir de los estándares básicos de competencia y en el marco del proyecto educativo de cada institución, los docentes definen objetivos para cada área específica de los contenidos temáticos y otros requisitos que sean primordiales para desarrollar la competencia respectiva. Se realizó un análisis el cual consistió en evaluar la forma en que se gestiona las calificaciones de los alumnos generadas por los docentes en las instituciones educativas, con este se busca reafirmar los parámetros que establece el MEN y considerar si el sistema de calificaciones se apoyaba por los estándares de competencias.

El presente proyecto fue desarrollar un aplicativo de software que aporte a la solución de los estándares básicos de competencia en la gestión de calificaciones. Por lo cual se planteó un acercamiento tecnológico en la que se establece automatizar en un aplicativo web el ingreso de la información de los estándares básicos de competencia de acuerdo a lo estipulado por el MEN y que también ayude en la generación de informes y reportes que permita analizar las calificaciones de los estudiantes e identificar debilidades y limitaciones.

1.2. Planteamiento del Problema

Actualmente las instituciones educativas en Colombia utilizan un sistema de calificaciones propuestos por el MEN. El sistema está basado en ciertos estándares de nivel de competencia. Un estándar es *“un criterio claro y público que permite juzgar si un estudiante, una institución o el sistema educativo en su conjunto, cumplen con unas expectativas comunes de calidad; expresa una situación deseada en cuanto a lo que se espera que todos los estudiantes aprendan en cada una de las áreas a lo largo de su paso por la Educación Básica y Media, especificando por grupos de grados (1 a 3, 4 a 5, 6 a 7, 8 a 9, y 10 a 11) el nivel de calidad que se aspira alcanzar.”* [2]. Los estándares se han definido para que un estudiante no sólo acumule conocimientos, sino para que aprenda lo que es pertinente para la vida, y de esta manera pueda aplicar estos saberes en su cotidianidad para la solución de problemas nuevos. Los estándares determinan una guía sobre lo que los estudiantes deben saber y saber hacer con lo que aprenden. Se trata de ser competente y no de competir.

En muchos casos, los sistemas de calificaciones de las instituciones educativas no están cumpliendo con los estándares de competencia que el MEN tiene establecidos. Los estándares describen los conocimientos y habilidades que los estudiantes en cada período académico deben lograr y deben ser evaluables por los docentes. A pesar de que se han desarrollado estos estándares para las instituciones educativas, estas no los tienen en cuenta al momento de las calificaciones, ya sea por cuestiones de tiempo o simplemente de disciplina, debido a la gran cantidad de información que contiene los estándares de competencias.

Por lo tanto, las calificaciones registradas por los profesores son muy simples ya que solo determinan una valoración por cada periodo, en la cual se establece un indicador numérico si el alumno pasó o no la materia dada, sin tener en cuenta los temas de calidad que vienen dado por los estándares de competencia. De manera que, el plan de aula que es un instrumento de planificación, es decir, una herramienta a través de la cual, el docente tiene la oportunidad de organizar, programar y evaluar los procesos que se van a desarrollar con los estudiantes a partir del plan de área que es el esquema creado por las instituciones educativas para organizar las áreas principales y fundamentales que forman parte del currículo, no se están apoyando en los estándares básicos de competencia. Debido a que los profesores no están generando calificaciones, en la que se estudie las acciones de mejoramiento o las debilidades de los estudiantes en los diferentes estándares de competencia.

Un análisis de la anterior problemática evidencia dos aspectos fundamentales en los cuales se centran las falencias de las instituciones con respecto a las exigencias del MEN: el plan de área y el registro de calificaciones. En la primera se determina que no se hace un seguimiento válido si el sistema de calificaciones de las instituciones educativas está cumpliendo los esquemas del MEN. En segundo lugar, no hay un reporte evaluativo donde el profesor muestre a los estudiantes que estándares de competencia se está dictando por periodo, para así comparar mediante las calificaciones las posibles falencias o mejoras del rendimiento por partes de los estudiantes. Si examinamos los estándares de competencia en general, que tienen una estructura de factores y subprocesos que el docente debe dictar en el aula de clases [3]. Por consiguiente, en el momento de calificar, el docente a un estudiante ¿Cómo está determinando los conocimientos, habilidades, destrezas y actitudes por parte del estudiante en estos factores y subprocesos? Si solo en el boletín de calificaciones se está generado y visualizando una nota general.

De acuerdo a lo anterior se vio la necesidad de resolver la problemática a través de un software que permita configurar el plan de área apoyados por los estándares básicos de competencia,

permitiendo registrar y almacenar en forma segura las calificaciones asociadas a cada uno de los estándares de competencias que los estudiantes deben evidenciar en el aula, cumpliendo con los estándares de competencias formulados por el MEN. De tal manera que estos procesos puedan ser de fácil y rápido acceso para responder con agilidad, eficiencia y exactitud en el momento en el que esta sea requerida por los miembros de la comunidad educativa.

1.3. Formulación del Problema

¿Cómo determinar las debilidades que se presentan en los estándares de competencias a partir de las calificaciones obtenidas por los estudiantes?

1.4. Justificación

El desarrollo de un sistema de información web que permita configurar el plan de área basados por los estándares básicos de competencias para las instituciones educativas de Colombia tiene como finalidad ser una herramienta de soporte para los entes encargados de la enseñanza y aprendizaje de las instituciones educativas. Esta herramienta permitirá llevar un control detallado de las debilidades de los estándares básicos de competencias encontradas a los estudiantes en el proceso de sus estudios, generando beneficios a los docentes y monitores de aula para hacer un estudio determinado del por qué se generan esas fallas académicas o puedan ofrecer recomendaciones y propuestas que propicien una evaluación formativa eficaz en los procesos de enseñanza y aprendizaje con el propósito de lograr mayores rendimientos en los estudiantes.

En la actualidad en algunas instituciones educativas, se hace necesario que en el boletín académico o en los reportes se visualice el proceso evaluativo de los estudiantes de una manera más detallada, donde las calificaciones estén sujetas a los estándares básicos de competencia y saber exactamente en qué estándar los estudiantes presentan mayor dificultad, así los docentes puedan analizar ese tipo de falencia que presentan sus alumnos y tomar medidas de acompañamiento que apoyen a mejorar el rendimiento académico de ellos. Lo mismo aplica para los padres de familia, estar más informados de los procesos de enseñanza de sus hijos. Dado que en las instituciones educativas generan informes mostrando una nota en particular especificando si el estudiante pasó el área.

El sistema será una herramienta útil para las instituciones educativas ya que facilitará la gestión de procesos de calidad y optimizará el tiempo que los profesores requieren para registrar notas de acuerdo a los estándares de competencia.

El interés del proyecto es ofrecer un apoyo al registro de calificaciones bajo los estándares de competencia reglamentados por el MEN y para llevar a cabo un mejor control de los procesos de evaluaciones hacia los estudiantes y establecer las competencias con sus correspondientes estándares que se les dificulta a los estudiantes aprender, permitiendo que las Instituciones educativas tome decisiones en cuanto a la reestructuración de los planes de aula involucrados.

De esta manera se evidencia la necesidad de transformar estos procesos en un sistema de información de entorno web que generen un manejo eficiente y ágil de la información. El proyecto se realizó en tecnologías de desarrollo web actuales, lo cual potenció la consecución de los alcances establecidos en el presente documento.

1.5. Objetivos

1.5.1. Objetivo General

Diseñar e implementar un aplicativo web para hacer seguimiento al rendimiento académico de los estudiantes en las instituciones educativas de acuerdo a los estándares básicos de competencias dadas por el MEN.

1.5.2. Objetivo Específicos

- Especificar los requisitos solicitados por cada uno de los usuarios involucrados dentro del sistema.
- Diseñar el sistema de acuerdo a las políticas establecidas por el Ministerio de Educación Nacional.
- Implementar un módulo de plan de área donde se permita crear los estándares básicos de competencia para cada una de las áreas dadas por el Ministerio de Educación Nacional.
- Implementar un mecanismo para el registro de calificaciones.
- Implementar un módulo de generación de reportes donde establezca el rendimiento de los estudiantes a partir de los estándares cargados en el sistema.
- Definir y realizar pruebas unitarias con el sistema desarrollado en la Institución Educativa María Antonia Ruiz, verificando que no haya fallas en el producto culminado.

1.6. Estructura del documento

El presente proyecto de trabajo de grado está organizado de la siguiente manera: en el capítulo 1 está la introducción del proyecto el cual consta de la descripción general del problema, el planteamiento del problema, formulación del problema así como los objetivos general y específicos y la estructura del documento, en este capítulo se detalla brevemente el contenido del presente trabajo así como sus objetivos. En el capítulo 2 está el marco referencial que se subdivide en los siguientes marcos: marco teórico, y antecedentes del proyecto, que es la base fundamental en la que se basa el proyecto. En el capítulo 3 está el desarrollo del proyecto, el cual se divide en: análisis de requerimientos que se divide en requerimientos, modelo del dominio, prototipo rápido y diagramas de caso de usos, luego se encuentra el análisis y diseño preliminar que se divide en Descripción de Casos de Uso, diagramas de robustez, Posteriormente se muestra el diseño que se divide en Diagrama de navegación, Diagrama de clases, diagrama de modelo relacional y finalmente la implementación donde se detalla cómo se abordó los tipos de tecnología para la implementación del proyecto.. En el capítulo 4 se describen las pruebas que se realizaron que son: Pruebas unitarias con Tinker y de caja negra. En el capítulo 5 está compuesto por las conclusiones y trabajos futuros, aquí se expresa la experiencia adquirida durante el proyecto y los futuros trabajos que se consideran pertinentes.

Capítulo 2

Marco de Referencia

2.1. Marco teórico

Dado que el objetivo central de este proyecto es desarrollar un aplicativo web que permita determinar las debilidades de estándares de competencias de los estudiantes, en esta sección se presenta el marco teórico del trabajo, la cual enfoca algunos temas: en primer lugar se aborda las bases conceptuales de los modelos y herramientas usados para el desarrollo del proyecto, la metodología Iconix y la importancia de los estándares de competencias que maneja el MEN, que sirvan como ejes conceptuales sobre los cuales apoyar la lectura de este documento

2.1.1. Arquitectura de la información (Modelo-Vista-Controlador)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página y el código que provee los datos dinámicos a la página. El modelo es el sistema de gestión de base de datos y el controlador representa los eventos de la aplicación y la comunicación entre la vista y el modelo [4].

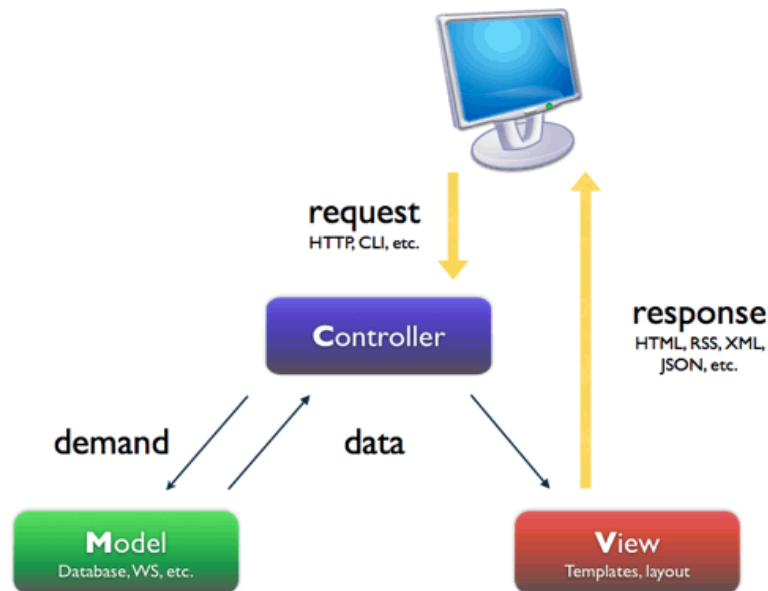


FIGURA 1 MODELO VISTA CONTROLADOR (MVC)

2.1.2. Laravel

Es un Framework de código abierto de PHP, la sintaxis que emplea es simple y elegante y emplea las características de las últimas versiones de PHP. Su modo de trabajo en el desarrollo son sujetas por varios patrones donde el usuario puede emplear el patrón de MVC tradicional o el patrón que maneja Laravel por defecto que es el manejo de Rutas con Clausuras. En su versión 5.3. Además, en la parte de la capa de la vista, Laravel maneja un sistema de plantillas para las interfaces llamado Blade, permitiendo una sintaxis más reducida que facilita el trabajo de las vistas.[5]

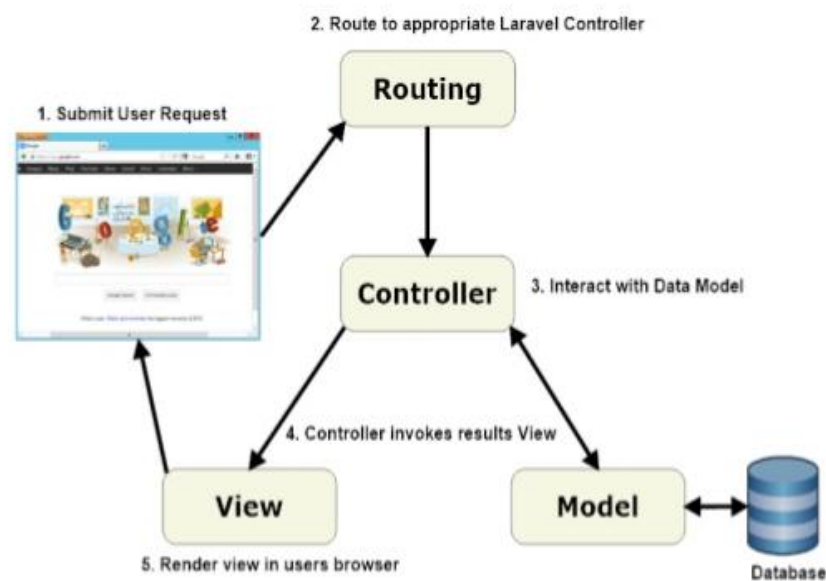


FIGURA 2 INTERACCIONES EN LAS CAPAS DE LARAVEL

Funcionamiento de las capas del framework Laravel

- **Model:** Laravel contiene un sistema que permite el mapeo de datos llamado Eloquent ORM, accediendo y facilitando la creación de modelos; mediante el patrón *active record*, cuyo funcionamiento es sencillo y versátil, no obstante, éste dispone de otros recursos que facilitan interactuar con los datos especialmente la creación de modelos y procesos de consultas o actualizaciones.
- **Controller:** Responde usualmente a los eventos del usuario y genera peticiones al modelo cuando se hace algún tipo de solicitud de información como proceso de actualización según ya sea los privilegios de acceso que se hayan descrito en las especificaciones del aplicativo, Laravel a diferencia de otros contiene la lógica del negocio en los

controladores que permite organizar el código en clases sin tener que escribir las rutas de acceso. Los controladores de Laravel deben de extenderse de la clase BaseController.

- **Routing:** Las rutas de Laravel están definidas en el módulo Routing, los cuales son cargados automáticamente por el Framework y puestas a disposición del Controller para otorgarle características como estados de sesión y protección contra exploit maliciosos transmitidos por usuarios mediante comandos no autorizados.
- **View:** Es la representación del modelo en un formato ideal para interactuar con el usuario, Laravel a diferencia de otros incluye un sistema que permite el procesamiento de plantillas llamado Blade, este favorece ya que permite generar un código mucho más limpio a nivel de interfaz de usuario, incluye sistema de cache con el fin de generar acceso de forma casi inmediata.

En los componentes de Laravel encontramos

- **Artisan:** Es el cliente de consola que permite la ejecución de comandos propios del Framework, es versátil y potente permitiéndonos extenderlo creando nuestras propias tareas para que estén disponibles desde el cliente.
- **Composer:** Permite modificar y agregar los paquetes que se deseen incluso generar paquetes propios con el fin de ser incluidos en el aplicativo a través de un Composer Update que se ejecuta en una consola de comandos
- **Migraciones:** Laravel incorpora la posibilidad de llevar el control de versiones de nuestra base de datos, esta característica integrada con un sistema de Seeding nos permite obtener nuestro aplicativo descargado y corriendo en unos pocos comandos.
- **Resource:** Laravel incorpora el concepto de resource permitiendo programar controladores propios como API Rest, facilitando que a la hora de agregar un elemento el usuario solo deba hacer un POST/resource, para obtener un listado como GET/resource etc. Cabe agregar que esto nos permite consumir y liberar nuestro aplicativo como API si necesidad de realizar ningún tipo de modificación con la posibilidad de integrarla a otros sistemas.
- **ORM:** (Objet-Relational Mapping) permite interactuar con la base de datos como si cada tabla fuera un modelo, lanzando consultas a la base de datos con una sintaxis de PHP, respetando la división de MVC (Model-View-Controller).

Laravel es un Framework con una estructura conceptual y tecnológica de soporte definido, típicamente incluye con herramientas para ayudar al desarrollo y unir diferentes componentes de un proyecto, los más representativo son:

- **Laravel Breadcrumbs:** Esta herramienta permite obtener la página actual dentro de la jerarquía de navegación, indicándole en una línea de texto el recorrido seguido y la forma de regresar con el fin de permitirle al usuario que conozca la ruta de su ubicación en directorios, subdirectorios y navegue a través de ella.
- **LARAVEL COLLECTIVE:** Esta herramienta permiten obtener helpers y funciones extras para las vistas y el uso del framework, como formularios dinámicos, notificaciones

personalizadas y acceso a protocolos SSH permitiendo manejar servidores mediante intérpretes de comandos.

- **LARACASTH/FLAS:** Esta herramienta permite enviar y personalizar mensajes de notificaciones al usuario para que realice alguna acción.
- **LARAVEL EXCEL:** Esta herramienta permite importar los archivos csv, ofreciéndole al desarrollador configurar el modo en el que desea leer el archivo, ya sea por comas, punto y coma o según sea el formato de separación que tenga este, con el fin de importar un sin fin de registro.

2.1.3. Base de datos

Una base de datos es aquella que permite guardar grandes cantidades de información de forma organizada, su función básica es permitir el almacenamiento y la recuperación de información, cuenta con un mecanismo que permite el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

Las principales características de los sistemas de bases de datos son las siguientes:

- Integridad de los datos.
- Independencia lógica y física de los datos.
- Seguridad de acceso y auditoría.
- Acceso a través de lenguajes de programación estándar.

Estas características permiten crear una base de datos con la cual el usuario tiene un fácil acceso, sus datos se encuentran almacenados de forma ordenada y protegida. [6].

2.1.4. MYSQL

Es un sistema gestor de base de datos de código abierto con la capacidad y funcionalidad de almacenar y distribuir cantidades de datos, actualmente en el mercado es uno de los más reconocidos. MySQL permite crear bases de datos donde se puede llevar a cabo procesos como insertar, modificar, eliminar y ordenar datos. Actualmente el propietario es Oracle Corporation.

Las principales características de MySQL, son las siguientes:

- Soporta gran cantidad de datos.
- Gran rapidez y facilidad de uso
- Fácil Instalación.
- Gestión de usuarios y contraseñas, manteniendo un nivel alto de seguridad en los datos.
- Integración de múltiples librerías y herramientas que permiten su uso a través de varios lenguajes de programación.

Con base a estas características se puede establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedidos casi de forma inmediata. [7].

2.1.5. Servidor Web

Es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente.

El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI.

El término también se emplea para referirse al ordenador que ejecuta el programa [8].

2.1.6. HTML

El lenguaje de HTML funciona por medio de “etiquetas” que describen la apariencia o función del texto enmarcado. Este lenguaje puede llegar a incluir un script o código que tenga incidencia en el comportamiento del navegador web de elección.

La funcionalidad del HTML es tan sencilla que puede ser creado y editado en cualquier editor de textos básicos, como el Bloc de Notas típico del sistema operativo Windows. También puede editarse en procesadores de textos, software de diseño web o aplicaciones web directamente, como lo más convencionales programas de administración de contenido [9].

2.1.7. CSS (Cascading Style Sheets)

Es una tecnología desarrollada por la (W3C) con el propósito de separar la estructura de presentación. Su principal objetivo es el control de características graficas que incluye frecuentemente tablas animadas y complicados algoritmos de combinaciones de celdas. CSS trabaja con estándares que permite proporcionar características únicas en elementos referidos a este con el fin de generar plantillas graficas que permitan mayor sencillez al trabajar.

CSS es actualmente una tecnología usada por muchos sitios web para crear páginas visualmente atractivas mediante sintaxis simple y elegante. [10].

2.1.8. AJAX (Asynchronous JavaScript And XML)

Hacen referencia a un conjunto de tecnologías de uso común en la web que, combinadas adecuadamente, permite mejorar enormemente la experiencia del usuario con las aplicaciones de internet. Además, este tipo de tecnología nos permite asociar peticiones al servidor y lo más importante nos permite mandar solicitudes desde el cliente y/o servidor sin refrescar la página, haciendo los aplicativos web más rápidos en tiempo computacional. [11].

2.1.9. PHP

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web. Se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy. La versión de PHP más actual es 7.1.3. Las versiones de PHP han mejorado demasiado ya que permite nuevas funcionalidades donde se ahorra mucho código y hacen más eficiente el tipo de consulta con las nuevas sintaxis. Actualmente existen innumerables Frameworks dirigidos para PHP para facilitar su uso. En el presente proyecto se usó el Framework de Laravel [12].

2.1.11. ADMINLTE

Es un template open source desarrollada con Bootstrap 3, la cual permite vistas reusables y responsive de elementos para la administración de proyectos en la cual se puede visualizar en cualquier equipo con un sistema operativo que tenga un navegador, estaba bajo la licencia MIT que permite modificarla y usarla para fines comerciales. [13]

2.1.12. BOOTSTRAP

Es un Framework que permite simplificar el desarrollo del diseño web, cuya particularidad es adaptar la interfaz del sitio según sea el dispositivo que la visualice, reduce el proceso de maquetación sirviendo de guía para aplicar buenas prácticas de diseño.

Actualmente Bootstrap se instaura con solidez debido a su buen funcionamiento y los resultados positivos que ofrece, también se integra con sistemas de edición y publicación como Drupal o WordPress siendo aún más adaptable y compatible con cualquier terminal. [14]

2.1.13. SELECT2

Con este plugin que es un derivado de JQuery permite obtener un selector de elementos en nuestros formularios, con un cuadro de búsqueda que facilita de manera sencilla encontrar lo que requiere el usuario. Al combinar con consultas al servidor, permite traer elementos sin necesidad de actualizar la página [15].

2.1.14. DATATABLES

Es una herramienta muy destacable cuando se quiere realizar tablas dinámicas donde las peticiones que se hagan al servidor no permita que recargue la página sino que se realice instantemente las consultas, además permite agregar a las tablas HTML controles avanzados como filtros, paginación, manejo automático del ancho de las columnas, entre otros. Facilitando al programador las funcionalidades de esta índole y terminarlo en menos tiempo, además que optimiza el código, para los usuarios son una gran ventaja ya que sus interfaces y sus procesos de mostrar la consultas son muy amigables y neutras para todo tipo de usuarios [16]

.

2.1.15. CSV

Los archivos CSV (del inglés comma-separated values) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, se usa normalmente para almacenar datos en formas de columnas las cuales van separadas por comas y las filas se diferencian mediante saltos de líneas, es un formato sencillo que permite importar o exportar de bases de datos de una aplicación a otra. En el presente proyecto la base de datos donde está todo lo referente

a información del estudiante y los datos de la institución educativa está dada por un archivo con extensión CSV [17]

2.1.16. INPUTMASK

Esta herramienta permite crear una máscara de entrada para garantizar el formato predefinido. Es decir que puedo manipular los campos de entradas de texto para mostrar una acción y validación e indicarle al usuario que debe realizar esa acción de esa manera. [18]

2.1.17. IOSSLIDER

Esta herramienta permite seleccionar un rango de números de inicio y final con el fin de validar la información que se encuentra entre el punto de partida y el punto final. Actualmente esta herramienta es usada en los desarrollos web con el fin de proporcionar fácil uso de los elementos de la vista y a la vez de validar la información obtenida del usuario.

Esta herramienta integra control deslizante de contenido multi navegador personalizable. [19].

2.1.18. JASNY BOOTSTRAP

Esta herramienta proporciona características para construir aplicaciones web altamente interactivas. Donde el programador puede manipular ciertas librerías y adaptarlas a la necesidad del problema, se ahorra demasiado código si fuera hacerlas en Javascript puro. [20]

2.1.19. BOOTSNIPP

Es un proyecto que ha tenido gran crecimiento últimamente gracias a que comparte a toda la comunidad web elementos visuales en forma de "snippets" (trozos de código) basados en bootstrap, éstos van desde una maqueta de un login hasta elementos complejos fusionando CSS3 y JS.

Es ese sitio web que permite la publicación de elementos HTML, con Javascript y CSS, son creados por la comunidad y compartidos allí de manera gratuita [21]

2.1.20. DATEPICKER

Es un componente de la librería jQuery, nos ofrece un calendario totalmente personalizable, en el que podremos realizar selecciones de fechas y asociarlo a elementos HTML, como entradas de formularios. [22]

2.1.21. DATERANGEPICKER

Es un componente selector de rango de fechas para la librería de Bootstrap, donde se crea un menú desplegable desde el cual un usuario puede seleccionar un rango de fecha. Las características incluyen la limitación del rango seleccionable fecha, generando así una matriz de números los cuales podrán ser operados por parte del desarrollador. [23]

2.1.22. CHARTJS

Es un plugin JavaScript simple, flexible y muy completo para incrustar graficas estadísticas, tales como barras. Circulares, lineales, entre otros. Su funcionamiento consiste en obtener una matriz la cual está compuesta por los números ya sea a graficar o procesar mediante estadística.. [24]

2.1.23. FONTAWESOME

Es una librería que une los componentes de JavaScript y Css permitiendo sustituir imágenes de iconos comunes por gráficos vectoriales convertidos en fuentes, con el fin de proporcionar iconos de menor peso a nivel computacional a la hora de cargar en el navegador, permitiendo al desarrollador agregar elementos visuales con excelentes funcionalidades que al momento de visualizarse cargaran casi de forma inmediata. [25]

2.1.24. CHECKED

Es un componente de la librería de JQuery para personalizar los input de formulario de manera gráfica. Estas librerías permiten alterar los componentes ya sea desde su color hasta su funcionalidad, permitiéndole al desarrollador llevar un poco más lejos su creatividad. [26]

2.1.25. Procesos Agiles

Los procesos agiles de desarrollo de software, se enfocan en la capacidad de respuesta a los cambios y en mantener una buena relación con el cliente para llevar al éxito el proyecto. Estos procesos se basan en dos principios, que son: el rastreo de decisiones y la planificación adaptativa, donde la primera permite reducir el número de cambios a realizar en el proyecto, lo cual logra mantener la satisfacción del cliente y por ende el éxito del producto. Y la segunda permite que el grupo de trabajo esté preparado para aceptar cambio y tomar decisiones a lo largo del proyecto.

La metodología sobre la cual se enmarca el desarrollo del aplicativo web es Iconix

2.1.26. Descripción de la Metodología Iconix

Iconix está basada en la rigurosidad de la metodología RUP (Rational Unified Processes) y la practicidad de la metodología XP (Extreme Programing). [13]

‘Fue elaborada por Doug Rosenberg y Kendall Scott a partir de una síntesis del proceso unificado de los “tres amigos” Booch, Rumbaugh y Jacobson y que ha dado soporte y conocimiento a la metodología ICONIX desde 1993.’.

Iconix permite agilizar el proceso de desarrollo de software utilizando herramientas que sirvan para cubrir el ciclo de vida del desarrollo, estas herramientas son una serie de fases en las cuales se especifican los requerimientos y se modela el comportamiento del producto haciendo uso de los diagramas UML (lenguaje de modelamiento de unificado).

Es una metodología pesada-ligera de desarrollo del Software que se halla a medio camino entre un RUP (Rational Unified Process) y un XP (eXtreme Programming). ICONIX deriva directamente del RUP y su fundamento es el hecho de que un 80% de los casos pueden ser resueltos tan solo con un uso del 20% del UML, con lo cual se simplifica muchísimo el proceso sin perder documentación al dejar solo aquello que es necesario. Esto implica un uso dinámico del UML de tal forma que siempre se pueden utilizar otros diagramas además de los ya estipulados si se cree conveniente. ICONIX se guía a través de casos de uso y sigue un ciclo de vida iterativo e incremental. El objetivo es que a partir de los casos de uso se obtenga el sistema final. [27].

2.1.27. Estándares básicos de competencia

Los Estándares de Competencias Básicas son criterios claros y públicos que permiten establecer los niveles básicos de calidad de la educación a los que tienen derecho los niños y las niñas de todas las regiones del país, en todas las áreas que integran el conocimiento escolar en un plan de estudio. Los estándares básicos de competencias se constituyen en una guía para:

- El diseño del currículo, el plan de estudios, los proyectos escolares e incluso el trabajo de enseñanza en el aula.
- La producción de los textos escolares, materiales y demás apoyos educativos, así como la toma de decisión por parte de instituciones y docentes respecto a cuáles utilizar.
- El diseño de las prácticas evaluativas adelantadas dentro de la institución
- La formulación de programas y proyectos, tanto de la formación inicial del profesorado, como de la cualificación de docentes en ejercicio [28].

Para la ejecución de los estándares básicos de competencias, toda institución educativa de Colombia debe plantear un plan de estudio y plan de áreas para determinar que estándares podrán evaluar al estudiante en su proceso académico.

Los estándares son el punto de inicio que permiten especificar y diseñar estrategias de formación y capacitación, a partir de criterios y expectativas compartidas, asegurando que las escuelas ofrezcan educación de alta calidad, permitiendo igualdad de oportunidades educativas para todos los estudiantes.

Los principios de los estándares ofrecen calidad de educación, partiendo de la idea, de que todo niño y niña puede adquirir conocimiento con altos resultados, elevando las expectativas y capacidades de los estudiantes y así poder mejorar su desempeño cotidiano, otro principio se desprende de la necesidad de garantizar equidad en cada alumno, programando y organizando actividades, siguiendo y definiendo planes para lograr el desarrollo en lo que cada estudiante debe y puede aprender, se toma otro paso importante, llamado la democratización de la educación, el cual dirige los esfuerzos y muestra los resultados alcanzados en su proceso de aprendizaje. Por tanto, los estándares de competencia permiten, aseguran y contribuyen a que las diferentes escuelas ofrezcan una educación de alta calidad, garantizando el desarrollo de oportunidades educativas para los alumnos, los cuales se destacan con base a una serie de requisitos para alcanzar los diferentes grados o niveles, obteniendo igualdad en oportunidades para cada uno. Se debe tener una estructura clara, precisa y fácil de entender en las diferentes disciplinas o áreas a estudiar.

Lo que quiere lograr el MEN es que todos los estudiantes planteen y demuestren el “saber y saber hacer, para ser competente. Se han definido para que un estudiante no sólo acumule conocimientos, sino que pueda aplicar estos saberes en su cotidianidad para la solución de problemas nuevos. Se trata de que un niño o joven haga bien lo que le toca hacer. La competencia, muestra que en diversas situaciones de la vida cotidiana el niño, el joven o el adulto, aplican este conocimiento desempeñándose bien. Se trata de ser competente y no de competir.

Además en cumplimiento de la ley 115 de 1994 y considerando que los currículos de las diferentes instituciones educativas dentro de su autonomía institucional deben plantearse al contexto académico teniendo en cuenta aspectos como los desarrollos científicos, y tecnológicos, se han creado los estándares de competencias como bases o guías que sirvan como fundamentos para las pruebas ICFES, entidad que efectúa las evaluaciones tanto a nivel básico y media en todas las instituciones educativas de Colombia.

Con base en estos resultados y teniendo en cuenta los estándares que se proponen en las pruebas saber, cada institución debe preparar un plan para mejorar el rendimiento de los estudiantes de lo que deben saber y saber hacer con lo que hacen según los estándares, y lo que en realidad saben y saben hacer, por tanto, los resultados obtenidos en estas pruebas será la base para promover prácticas pedagógicas que permitan mejorar el aprendizaje de todos los alumnos.

Los estándares básicos de competencias están comprendidos por medios de ciclos de primer a tercero, de cuarto a quinto, de sexto a séptimo, de octavo a noveno y finalmente de decimos a once, donde se hace mención que componentes y que estándares se tienen que ver durante el periodo académico

Componente 1	Componente 2	Componente 3	Componente 4
Estándar 1 del componente 1	Estándar 1 del componente 2	Estándar 1 del componente 3	Estándar 1 del componente 4
Estándar 2 del componente 1	Estándar 2 del componente 2	Estándar 2 del componente 3	Estándar 2 del componente 4
Estándar 3 del componente 1	Estándar 3 del componente 2	Estándar 3 del componente 3	Estándar 3 del componente 4
Estándar n del componente 1	Estándar n del componente 2	Estándar n del componente 3	Estándar n del componente 3

TABLA 1 MODELO DE ESTÁNDARES DE COMPETENCIAS

En cada ciclo se ven unos componentes que dependiendo del área específica el pensamiento que van evaluar y a su vez incluye los estándares que son los parámetros que tendrá en cuenta los docentes en cada institución para evaluar al estudiante y al final determinar qué aspectos son los que tienen más falencias los estudiantes al finalizar cada periodo académico.

➤ **Plan de estudio**

Documento en el que se relaciona información de las áreas obligatorias y fundamentales al igual que las optativas, es esencial para el buen funcionamiento de las instituciones educativas debido a que sirve como orientador de las decisiones que se tomen en materia administrativa y académica al interior de los establecimientos educativos; además sirve como fuente de información para los usuarios del servicio educativo que buscan intereses particulares a la hora de escoger el tipo de formación que están dispuestos a recibir [29].

➤ **Plan de áreas**

Documentos que detallan aspectos que hacen parte de cada una de las áreas contempladas en el plan de estudio, en este documento se incluyen conceptos, metodologías, sistemas de evaluación, recursos y otros ítems concretos relacionados con la planeación del año escolar en torno a las áreas obligatorias y fundamentales al igual que las optativas.

Sirve de guía para los docentes del área, al igual que los estudiantes y padres de familia en los temas generales y específicos que se abordarán durante el proceso formativo desarrollados al interior de la institución educativa durante el año escolar

Para los directivos docentes sirve como instrumento de verificación y control del cumplimiento de la misión institucional al igual que la adecuada ejecución de los procesos pedagógicos que se adelantan en los planes de aula.

- **Planes de aula:** Es la planeación en detalle de las clases que desarrollan los docentes en las diversas áreas y asignaturas contempladas en el plan de estudio conforme a las especificaciones generales establecidas en los planes de área. Sirven para que los docentes cumplan con la programación establecida en los planes de área y desarrollen un proceso formativo planeado conforme a los estándares de competencias y lineamientos curriculares formulados en el área [30].
- **Planificación curricular:** El término currículo como planificación tienen su propio significado, por ende se procederá a analizar cada uno de ellos para lograr establecer un concepto de Planificación Curricular ¹“El currículo es un proceso educativo integral con carácter de proceso que expresa las relaciones de interdependencia en un contexto histórico social, condición que le permite rediseñarse sistemáticamente en la medida en que se producen cambios sociales, los progresos de la ciencia y las necesidades de los estudiantes, lo que se traduce en la educación de la personalidad del ciudadano que se aspira a formar.”

¹ García y Addine, 2001:3

El diseño curricular ²“es el resultado del trabajo que da respuesta a las exigencias sociales en la formación de profesionales, constituyendo un proyecto educativo, que sirve de guía y condiciona el desarrollo del proceso. El diseño curricular se elabora a partir de las bases mediante una teoría curricular, es el puente entre la teoría curricular y la práctica. “

Una de las funciones principales de los docentes corresponde a la planificación de sus clases. Ellos diseñan, seleccionan y organizan estrategias de enseñanza que otorgan sentido a los contenidos presentados y entregan a sus alumnas y alumnos las mejores herramientas para la adquisición del aprendizaje, esto se denomina planificación curricular [31].

- **Asignación académica:** Es el proceso en el cual se asigna las materias que los docentes dictarán dependiendo del curso al que pertenezca y el periodo académico
- **Periodo Académico:** Conjunto periódico de semana de manera autónoma que planea y especifica la institución educativa, en el que se desarrolla un ciclo completo de los procesos académicos y administrativos. Cada periodo al culminar se evalúa y actualiza el estado académico de los estudiantes.

2.2. Antecedentes

Entre los antecedentes de soluciones sistematizadas que se han hecho en Colombia para gestión de los diferentes procesos académicos en las instituciones educativas se encontraron varios tipos software ya sea de escritorio o en entornos web para hacer una comparación de su funcionamiento. De los programas investigados, citamos los siguientes:

2.2.1. Aplicaciones de Escritorio

Entre los programas de escritorio que tienen como finalidad sistematizar procesos en las instituciones educativas podemos destacar los siguientes:

- **VisualCX** es un software de procesamiento de datos académicos, implementado por la empresa COMPUSERVIX-E LTDA de Cali para satisfacer las necesidades del manejo de la información de estudiantes, docentes, coordinadores y directivas de las Instituciones educativas, es uno de los programas más usados en el Departamento del Valle del Cauca y es compatible solamente con las versiones de Windows (98/millennium/2000/XP/vista/NT/7) [32].
- El software de escritorio denominado **decoded trilogy** desarrollado por la empresa HAWIS Software gráficas de Cartagena es un programa que permite generar las notas estudiantiles, los carnets, los diplomas. Además, para las secretarías Académicas permite generar informes financieros, constancias, libro de promociones y otros documentos; se entrega cd de instalación, licencia de uso, manual digital. Este software es recomendado para instituciones pequeñas ya que al manejar pocos estudiantes la información puede ser procesada por una sola persona, además de su costo reducido” [33].

² Lazo y Castaño, 2001: 6

- **SAS, software académico** es un Software de gestión escolar orientado a instituciones educativas colombianas para el manejo de las inscripciones, matrículas, calificaciones, carnetización y estadísticas de logros evaluativos está implementado con Visual Fox Pro, algunos componentes de flash [34].

2.2.2. Aplicaciones Web

Dentro del entorno web se cita principalmente:

- **Educolombia** es un software web de licencia privativa, el cual permite administrar los procesos de gestión de académica de los colegios a nivel nacional, además tiene módulos adicionales como la gestión de biblioteca e inventarios. En el módulo de proceso de evaluación no permite desplazarse mediante el teclado por las celdas de ingresos de notas, los logros deben crearlo en una vista independiente, llenando así de ventanas muy similar al estilo de software de escritorio y tampoco permite definir la cantidad de logros de la evaluación. Partiendo de este aspecto negativo muchos profesores les resultaría incómodo manejar este tipo de aplicaciones web [35].
- **Quid** Además de tener los módulos principales de una Gestión de procesos académicos, incluso tiene un módulo de mensajería en la cual se envía comunicados, circulares o notificaciones a grupos de estudiantes, profesores o padres de familia por medio mail o SMS es un sistema de licencia privativa que funciona a través de internet. [36].
- **Phidias** es un aplicativo web muy completo. El programa administra los cursos, notas, tareas, exámenes en línea, impresión de boletines y reportes consolidados, además gestiona las preinscripciones, inscripciones, ficha del alumno y núcleo familiar. Además te permite analizar las estadísticas para la toma de decisiones a nivel gerencial. Además, tiene módulos extras como los módulos de cobros y pagos, presenta reportes consolidados, estados financieros, cartera, y permite a los usuarios realizar pagos en línea [37].
- **Web académica 2.0** es una plataforma web desarrollada por la empresa PEGASUS Software de Villavicencio se basa principalmente porque sistematiza la gestión académica tanto para web y aplicaciones móviles, contiene un módulo muy potente que es el de Registro de observaciones de convivencia donde se puede hacer el registro del observador del alumno que incluso el padre lo puede consultar. También tienen un módulo para compartir información con padres y estudiantes donde a través de la opción de correo interno de dicha plataforma web, la institución puede compartir estrategias, mecanismos de prevención, programas y actividades inherentes, manual de convivencia y demás información pertinente al Sistema Nacional de Convivencia [38].
- **Ciudad educativa** es una de las herramientas más usadas en Colombia para las instituciones privadas y públicas por su fácil uso en la parte de interfaz gráfica, es muy factible para los usuarios finales, es desarrollada por la compañía Núcleo Software en Marinilla-Antioquia, además de tener los módulos comunes de la gestión académica contiene un módulo de elegir al personero de la institución correspondiente [39].
- **ServiNotas** es software en entorno web de Barranquilla, muy utilizado en las instituciones educativas de Colombia por su excelente módulo de calificaciones que se adapta a cualquier institución que lo adquiera y cumple con los lineamientos actuales del Ministerio de Educación en Colombia [40].

A nivel internacional se destaca los siguientes software de gestión académica:

- **El proyecto ALBA** es un proyecto de desarrollo de Software para la realización de un sistema Informático Abierto de Gestión Unificada para Unidades Educativas, que brindará una herramienta realizada en software libre, para el trabajo cotidiano en estas unidades. Por su tipo de licenciamiento bajo la GNU/GPL, el sistema permitirá una reutilización y actualización constante, como así también la posibilidad de aprovechamiento de módulos ya realizados por otros emprendedores (siempre que sean compatibles con esta licencia). Es un proyecto web realizado en buenos aires-argentina, el cual puede ser modificado a las necesidades de la institución, busca emprender una comunidad donde se puede satisfacer y desarrollar varias necesidades de los respectivos módulos, gracias al licenciamiento libre, sin embargo, hoy en día no cumple los esquemas de calificación de Colombia [41]
- **Esemtia:** es un software en entorno web desarrollado por la compañía Esemtia Grupo Edebé S.L de España. Además de la gestión académica y la comunicación con las familias, esta plataforma incluye un entorno virtual de aprendizaje y otras utilidades, como gestión de la calidad, facturación, gestión documental, de bibliotecas o de recursos [42].
- **Gescola:** se trata de un sistema de gestión para centros educativos online, flexible y adecuado para cualquier centro. Se integra con la plataforma moodle, con una sincronización automática de cursos, creación de usuarios y matriculación de los alumnos y profesores a sus cursos correspondientes [43].
- **Globaleduca:** Software desarrollado por la empresa LC Ibérica de España, está compuesta por seis módulos, más otros dos opcionales, que pueden integrarse entre sí para la gestión del centro educativo. Uno de ellos es un soporte para realizar las tareas administrativas del centro: gestión de recibos, remesas, nóminas, etc. También ofrece una intranet para gestionar toda la organización interna del centro y un portal la comunicación con las familias [44].
- **iEduca:** es una plataforma de gestión académico-administrativa multiidioma, que tiene como fundamento base la creación de una comunidad educativa entre el centro, personal docente, alumnos y sus familias. Se caracteriza por otorgar a sus usuarios un conjunto de herramientas de gestión, planificación y ejecución para agilizar el desarrollo de las distintas actividades [45].

Por último, se cita la plataforma **Educcare**, cuya plataforma es el resultado de la unión de varias compañías más sobresalientes en España en el ámbito de desarrollo de software como ITforBI, INFORTECNICA Y LIDER IT, dicha unión creó la compañía Integrated Technology Consulting para ser unas de las más potentes compañías de España y plantear en el mercado educativo una solución que satisface las necesidades del país. “La gestión académica, el área financiera, la obtención de resultados analíticos y presupuestarios son las tres áreas de las que dispone esta plataforma.” En concreto, el módulo de Gestión Académica se compone de tres portales web que, siendo independientes, se integran perfectamente en una única base de datos: profesores, padres y secretaría. Ha sido desarrollada con Visual Studio .NET garantizando un software estable y seguro. La robustez y experiencia de Microsoft Dynamics NAV hace de la Solución Financiera de educare un producto único en el mercado es un software muy potente y muy completo en el mercado, no acoge los estándares de calificación de Colombia [46].

Se encontraron aspectos positivos y negativos en las soluciones actuales, se plantea algunos métodos para tener en cuenta y por ende como mejorarlos en la solución que se desea implementar.

En las soluciones de escritorio se encontró que no soportan múltiples usuarios y no son escalables en cuanto al número de usuarios. Además, estos programas dependen del sistema operativo. En las aplicaciones citadas dependen mucho de las versiones de Windows y lenguajes de programación privativos para que los programas se puedan instalar en los computadores.

Mediante un análisis de estos programas en su modo demo, se encuentra que no son parametrizables y modulares. Estos programas requieren cierto tipo de requisitos de hardware para poder ejecutar en el computador. Además, requieren instalación y actualización en cada cliente. Se analizó también que se abren muchas ventanas por cada proceso que el usuario haga.

En las soluciones web se encontró que ningún software en el mercado colombiano tiene implementado un módulo de reportes que siga los estándares de competencias dadas por el MEN.

De los programas citados tienen muy buen soporte y documentación, la mayoría tienen licencia privativa y deben pagar ya sea lo estipulado en el contrato, al igual si desean un nuevo módulo adicional para una función en particular

Se observó que estos aplicativos webs en el módulo de calificaciones, no permite el manejo del teclado.

Algunos programas contienen muchos módulos que a perspectiva sobrarían para una gestión académica para un colegio pequeño.

Las ventajas de implementar un software web es que no se obliga a usar cierto sistema operativo, todo funciona con un navegador web cualquiera. Además, no se necesita instalar nada en el cliente, si se va agregar una nueva terminal solo requiere poner una computadora nueva.

De los programas citados se encontró que el tiempo de respuesta es más lento. En cambio, otros lo han mejorado usando tecnologías como AJAX haciéndolas casi tan rápidas como las de escritorio. Estos aplicativos webs tienen costos adecuados para cualquier institución de pocos recursos.

A partir de los antecedentes descritos se concluyó que en base a las falencias observadas en los elementos existentes en el mercado y las necesidades presentes en las instituciones educativas donde no se están evidenciando de forma clara el rendimiento de cada estudiante a partir de los estándares básicos de competencias que exige el MEN. Por eso en el presente proyecto se tiene un módulo de reportes donde sí se evidencie ese tipo de falencias que no están en los aplicativos descritos anteriormente.

Capítulo 3

Desarrollo del Proyecto

En la fase inicial, para escoger una metodología acorde al proyecto que se desarrolló en este trabajo de grado, se tuvo en cuenta varios aspectos del producto: como el tiempo estimado para su implementación, la complejidad del proyecto y sobre todo la información de cómo funcionan el sistema de los estándares de competencias dadas por el MEN. Para así determinar la cantidad de documentación necesaria, y partiendo del hecho de que un 80 % de los procesos puede ser resueltos tan solo con el 20 % de los diagramas UML, donde las fichas de casos de usos son la guía prioritaria para el desarrollo de los procesos y sigue un ciclo de vida iterativo e incremental, la finalidad es que, a partir de los ficheros y diagramas de casos de usos, se culmine el sistema o aplicativo final.

ICONIX³ es una de las llamadas metodologías ágiles de desarrollo de software más destacadas de los tiempos recientes, se encuentra en un intervalo de complejidad de análisis de la metodología robusta RUP⁴ y la metodología ligera XP⁵, donde se busca entregar la documentación más apropiada para el inicio del desarrollo del aplicativo, sin desviarse del análisis y diseño que fomenta una metodología tradicional, solo que la sintetiza y/o simplifica de acuerdo a la complejidad del problema. En otros términos, plantea lo mejor de la combinación de una metodología robusta y ligera tradicional y lo ofrece al grupo de desarrollo que está ejerciendo la labor.

El actual proyecto tiene una escala de complejidad media por lo cual se usó ICONIX ya que es la más adecuada para llevar el análisis, diseño y desarrollo del proyecto, permitiendo la necesidad de acelerar todos los aspectos de programación, sin perder el enfoque de la gestión de documentación. Esta metodología tiene como características principales:

Iterativo e Incremental: varias iteraciones ocurren entre el desarrollo del modelo del dominio y la identificación de los casos de uso. El ciclo de vida incremental consiste en desarrollar por partes el producto de manera que puedas integrarlas funcionalmente. En cada ciclo de iteración se revisa y mejora el producto.

Trazabilidad: cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos.

Dinámica del UML: Ofrece un uso dinámico del UML porque utiliza algunos diagramas UML, sin exigir la utilización de todos, además simplifica el contorno de los diagramas UML sin tener tanto detalle como en el caso de RUP (Rational Unified Process) cuando el producto es de alta complejidad.

³Metodología pesada-ligera de desarrollo del Software.

⁴Metodología de desarrollo Proceso Unificado de Racional.

⁵Metodología ágil de desarrollo programación extrema.

ICONIX por lo general no se extiende tanto al análisis inicial, porque sería una gran desventaja para el proyecto, debido a que un análisis exhaustivo en la primera fase del proyecto puede convertirse en un obstáculo para iniciar el desarrollo y por ende tener un bajo índice de éxito, pero también se debe resaltar en un contexto contrario que la falta de un análisis complejo con una curvatura establecida en la fase inicial hace que a futuro se realicen cambios del sistema.

La metodología ICONIX ayuda a resolver este tipo de problemática ya que sintetiza los recursos del ciclo de vida del proyecto ya que está en un punto medio de ser una metodología pesada-ligera y además agregando que su naturaleza es de tipo iterativa e incremental para corregir aspectos en a lo largo del proyecto. Por estos aspectos positivos que conlleva esta metodología se pudo abarcar en iteraciones el desarrollo del proyecto, donde en cada iteración se abordaron los contextos más importantes en su análisis y diseño y en las funcionalidades más específicas para la construcción. Dando como objetivo que en cada iteración se lograba un análisis y un desarrollo a los objetivos planteados. Además de acuerdo a la complejidad del mismo, se pueden ejecutar diversas iteraciones dentro de un mismo incremento con la finalidad de optimizar los quehaceres obtenidos y evitar falencias o errores en las iteraciones respectivas. Dentro de cada incremento se encuentran las siguientes fases:

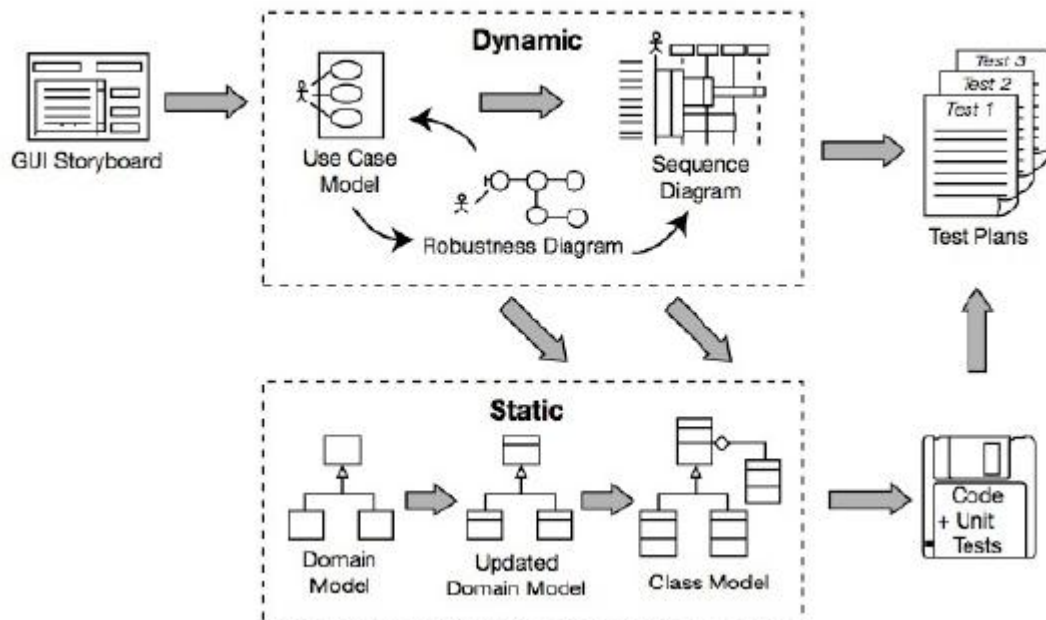


FIGURA 3 FASES DE LA METODOLOGÍA ICONIX [46]

Análisis de requisitos

En esta primera fase se realiza un Modelo de Dominio, que no es más que un Diagrama de Clases extremadamente simplificado. En esta fase se deben analizar todos los requisitos que formarán parte del sistema y con estos construir el, que representa las agrupaciones funcionales que estructurarán el sistema en desarrollo.

En esta primera fase también se realiza un bosquejo o un prototipo de las posibles interfaces graficas que se va diseñar en el sistema y que el cliente quede a gusto de ello, generalmente se toma la interfaz principal y de ella deriva los demás para no perder el esquema y el diseño, generalmente debe ser muy neutra la interfaz

Una vez el prototipo este culminado y se han obtenido todos los requisitos del sistema por parte del cliente, se procede a realizar los casos de uso. Estos diagramas de casos de uso se agrupan en diagramas de paquetes (es decir, utilizan referencias entre diagramas de casos de uso para simplificar su lectura) y se asocia cada requisito a un caso de uso para obtener una trazabilidad.

Análisis y Diseño Preliminar

En esta fase a partir de cada caso de uso se obtendrán una ficha de caso de uso, está formada por un nombre, una identificación, un objetivo, una precondition que debe cumplir antes de iniciarse y una postcondición que debe cumplir si termina correctamente.

Después será necesario realizar lo que se conoce como Diagrama de Robustez, el cual pertenece al proceso ICONIX. Es un híbrido entre un Diagrama de Clases y un Diagrama de Actividades., Facilita el reconocimiento de objetos y hace más sencilla la lectura del sistema.

El objetivo del diagrama de robustez es añadir nuevas relaciones a los diagramas de clase, de forma que ya tendremos un esqueleto aceptable de la arquitectura y del diseño a partir del cual podremos proseguir nuestro proceso. Con esto y las fichas, refinamos el diagrama de clases tanto como sea necesario y obtenemos una nueva versión preparada para la siguiente fase

Revisión crítica del diseño

En esta fase se registran todos los elementos que forman parte del sistema. Se usan diagramas de Secuencia o un tipo de diagrama que determine el flujo. Para el presente proyecto se usó diagramas de navegación en vez de secuencia, ya que el software es totalmente web y por estándar las aplicaciones web se representan mejor con diagrama de navegación.

Los diagramas de navegación es un documento de interfaz y secuencias de acceso inherentes a las aplicaciones Web, tiene gran influencia en las transacciones y funciones de la aplicación, cuando se usa un Framework para el desarrollo, los diagramas deben mostrar sus capas o rutas en la cual se relaciona las funcionalidades.

En caso de que no estemos satisfechos con el resultado, será necesario repasar todo el proceso hasta que éste sea correcto. Es vital que los requisitos se satisfagan correctamente para el éxito del proyecto

Y finalmente se hace un diagrama de clases, que no es necesario ser tan específico, pero en el presente proyecto como es de complejidad media y además como se usó un Framework como Laravel se detalla más explícitamente el comportamiento de sus capas.

Implementación

En esta fase es donde hacemos uso real de la trazabilidad y donde realmente ponemos en práctica esa garantía de calidad. Después de tener un buen diseño, es cuestión de crear un buen software a partir de ese diseño, y mediante los testeos y pruebas adecuados podemos garantizar que el sistema final cumple con los requisitos iniciales y por tanto proceder a su entrega.

3.1. Análisis de Requisitos

En esta fase de análisis de información se comenzó consultando todo lo relevante de los estándares de competencia hechas por el MEN y cómo funciona el sistema de educación en la actualidad, por medio de este estudio, se analizó e identificó las falencias que hay en las instituciones educativas en su sistema de calificaciones al no tener en cuenta los estándares de competencia para determinar el rendimiento de los estudiantes. Se realizó el levantamiento de los requerimientos funcionales y no funcionales alcanzables y medibles teniendo en cuenta el tiempo que establecen dentro de un trabajo de grado. Posteriormente en base a esto, se construyeron los diagramas de casos de usos que indican la interacción de los usuarios y el sistema. Además, en esta fase permitió tener una guía de aprendizaje de las tecnologías que se iban abordar en el proyecto y determinar cuáles eran las más viables para el momento de la etapa de desarrollo. Se hizo un análisis de quienes serían los participantes que estarían involucrados en el sistema, con respecto a las diferentes entrevistas que se les hizo a algunos docentes del colegio María Antonia Ruiz. Posteriormente, se tomó como referencia otros softwares educativos donde se estudiaba las diversas funcionalidades de los participantes del sistema.

De acuerdo a lo anterior se determinó quienes serían los usuarios finales del sistema asociando el cumplimiento de los objetivos a lo que realmente se busca con el proyecto. A continuación, se describe los usuarios que tendrán importancia en el proyecto:

- **Administrador:** el usuario puede parametrizar los datos institucionales de un determinado colegio, crear el año lectivo, determinar que sedes van a ver en el sistema, importar los usuarios a través de un archivo simat, realizar todo lo que corresponde a la creación de la malla curricular teniendo en cuenta los estándares de competencia y lo más importante determinar las debilidades de competencias que tendrá los estudiantes ya sea individual o grupal ya sea por sede, ciclo académico, o un área académica.
- **Coordinador:** el usuario puede configurar a los docentes la asignación académica donde se deriva la sede, el grado, grupo, jornada, área, asignatura, competencia y estándares que dará el profesor en el año lectivo
- **Profesor:** el usuario elegirá que competencias y estándares desea evaluar en los diferentes periodos académicos del año lectivo.

A continuación, se especifican los requerimientos funcionales del aplicativo web.

3.1.1. Requerimientos funcionales

A continuación, se detalla de forma general los requerimientos funcionales obtenidos en el sistema. Para visualizarlo con más detalle incluyendo los requerimientos no funcionales debe dirigirse al anexo 1.

TABLA 2 REQUERIMIENTOS FUNCIONALES

Código	Descripción
R1	El sistema permitirá crear, editar, consultar y eliminar años lectivos
R2	El sistema permitirá crear, editar, consultar y eliminar Períodos Académicos
R3	El sistema permitirá configurar la fecha de inicio y culminación de cada uno de los periodos.
R4	El sistema permitirá crear, editar consultar, eliminar la institución educativa en la base de datos.
R5	El sistema permitirá crear, editar y consultar las sedes con los siguientes datos: Nombre, NIT, misión, visión y imagen de logo
R6	El sistema permitirá crear y eliminar grupos académicos que pertenecen a dicha sede con los siguientes datos: año lectivo, grado, jornada, grupo
R7	El sistema permitirá cargar un archivo SIMAT donde está toda la información de un año lectivo
R8	El sistema permitirá seleccionar las sedes, donde a su vez debe cargar de manera automática los grupos que corresponde a dicha sede
R9	El sistema permitirá asignarle al docente la carga académica
R10	El sistema permitirá al docente configurar las competencias básicas que depende de cada asignatura
R11	El sistema permitirá configurar al docente los estándares que depende de cada competencia
R12	El sistema permitirá asignarle al docente la cantidad de horas que va dictar de acuerdo al grado y al área de conocimiento en que este
R13	El sistema permitirá asignarle horas semanales al docente
R14	El sistema permitirá validar que las horas asignadas al docente no superen las 30 horas semanales por ley
R15	El sistema permitirá visualizar las horas extras al docente siempre y cuando superen las 22 horas semanales como mínimo
R16	El sistema permitirá crear, editar y consultar un área de conocimiento
R17	El sistema permitirá asignarle las horas semanales al área académica
R18	El sistema permitirá crear, editar y consultar las asignaturas académicas que pertenecen a las áreas de conocimiento
R19	El sistema permitirá asignarle las horas semanales a la asignatura académica
R20	El sistema permitirá validar que la intensidad horaria de las asignaturas no supere la intensidad horaria del área de conocimiento

R21	El sistema permitirá crear, editar y consultar las competencias básicas.
R22	El sistema permitirá, crear, editar y consultar los estándares de competencias
R23	El sistema permitirá, crear, consultar y eliminar los grados académicos
R24	El sistema permitirá, crear, consultar y eliminar ciclos académicos.
R25	El sistema permitirá crear, editar y consultar competencias académicas por cada ciclo.
R26	El sistema permitirá crear, editar y consultar los estándares de competencia por cada ciclo
R27	El sistema permitirá adicionar los componentes de competencias a calificar al docente que pertenece a dicho grupo y año lectivo
R28	El sistema permitirá editar las componentes de competencias de acuerdo al periodo académico
R29	El sistema permitirá eliminar los componentes de competencias de acuerdo a la información
R30	El sistema permitirá cargar de manera dinámica los estudiantes pertenecientes al grupo seleccionado para remitir una calificación en particular.
R31	El sistema permitirá filtrar las calificaciones de acuerdo a la asignatura y periodo
R32	El sistema permitirá ingresar una calificación a los estándares básicos de dicha competencia
R33	El sistema promediara las calificaciones de los componentes académicos obtenidas por los estudiantes pertenecientes al grupo seleccionado
R34	El sistema permitirá configurar las notas académicas de acuerdo a los niveles de desempeño para las calificaciones y reportes
R35	El sistema permitirá generar reportes de gráficos de barras del rendimiento del estudiante por competencias y estándares
R36	El sistema permitirá mostrar gráficamente el rendimiento del grupo estudiantil por competencias y estándares
R37	El sistema permitirá generar reportes de gráficos de barras del rendimiento de la sede institucional por competencias y estándares
R38	El sistema permitirá generar reportes de gráficos de barras del rendimiento de varias sedes institucionales por competencias y estándares
R39	El sistema permitirá generar reportes de gráficos de barras del rendimiento de ciclos académicos por competencias y estándares
R40	El sistema permitirá generar reportes de gráficos de barras del rendimiento de áreas académicas por competencias y estándares
R41	El sistema permitirá la función de iniciar sesión.
R42	El sistema permitirá la función de cerrar sesión.
R43	El sistema permitirá crear, editar y eliminar usuarios.
R44	El sistema permitirá crear, editar y eliminar perfiles.
R45	El sistema permitirá configurar los parámetros de información de la institución educativa: nombre, dirección, teléfono, misión y visión, y el logo de la institución

3.1.2. Modelo de Dominio

Una vez que el trabajo de los requerimientos se culminó satisfactoriamente por varias revisiones, se inicia la construcción de un diagrama de modelo de dominio. El modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema. Esto es así ya que cuando se realiza la programación orientada a objetos, se supone que el funcionamiento interno del software va a imitar en alguna medida a la realidad, por lo que el mapa de conceptos del modelo de domino constituye una primera versión del sistema. Para visualizarlo con más detalle ver anexo 2.

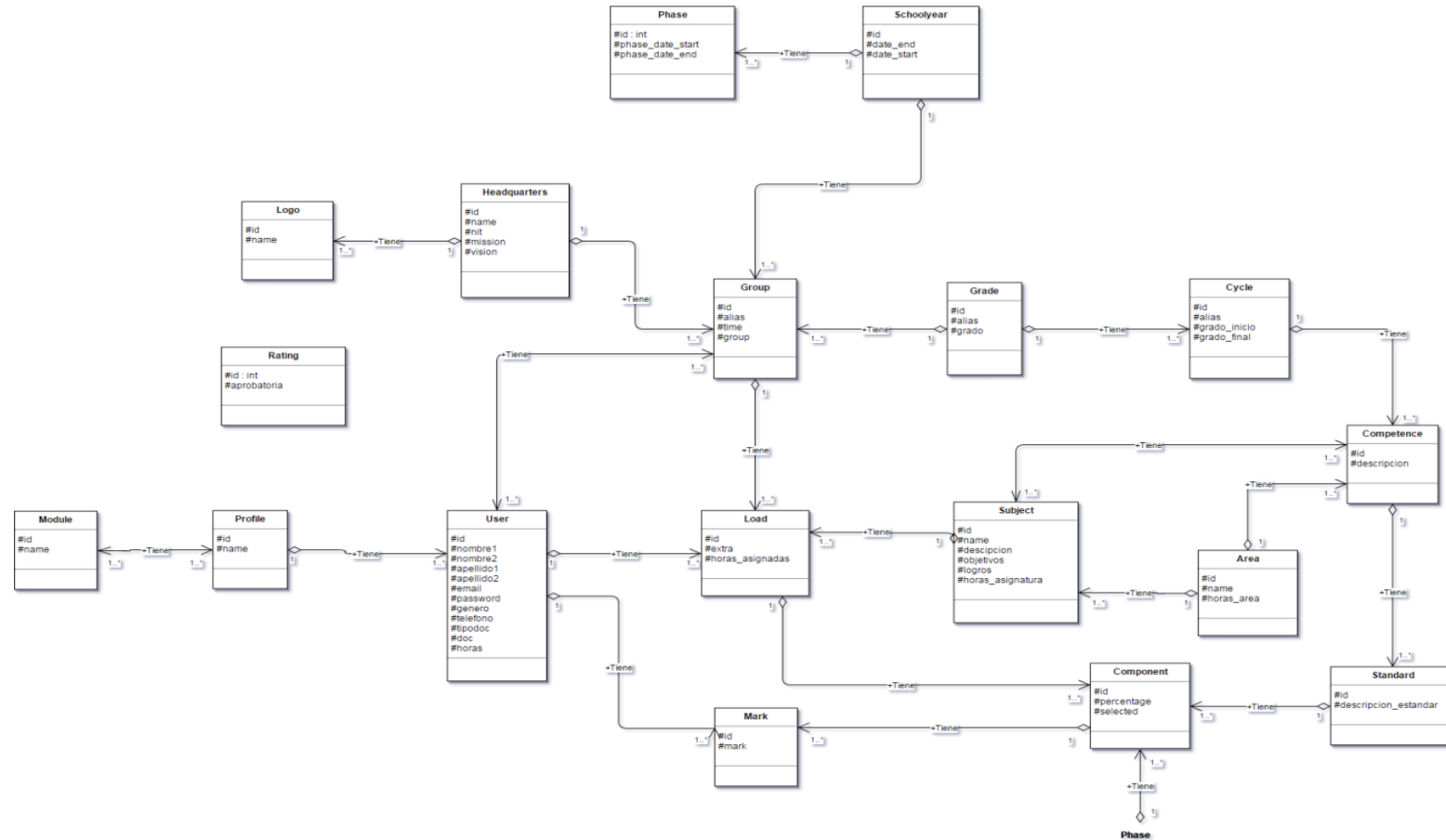


FIGURA 4 MODELO DEL DOMINIO

En este diagrama se especifica de manera simple una perspectiva del flujo que va tener el aplicativo web, independiente del lenguaje de programación y de las tecnologías que se abordaron. Se identifica que objetos del mundo real van a estar vinculados en el sistema. Este diagrama conlleva las abstracciones del sistema y las relaciones de agregación y generalización entre ellos.

3.1.3. Prototipo Rápido

El diseño de prototipos es una técnica muy típica en la ingeniería de software para desarrollar modelos a escala de un producto. Cuando se aplica al desarrollo de sistemas de información el diseño de este tipo de prototipos implica la creación de modelos para el sistema dado, además se centran en el diseño de contenidos e interacción y no en el diseño visual, y es un elemento de gran valor en el análisis, además permite mitigar posibles modificaciones costosas en nuestros contexto en el proceso de diseño web e interacción con el usuario

A continuacion se visualiza el prototipo del sistema inicial, los otros los pueden visualizar en el anexo 3.

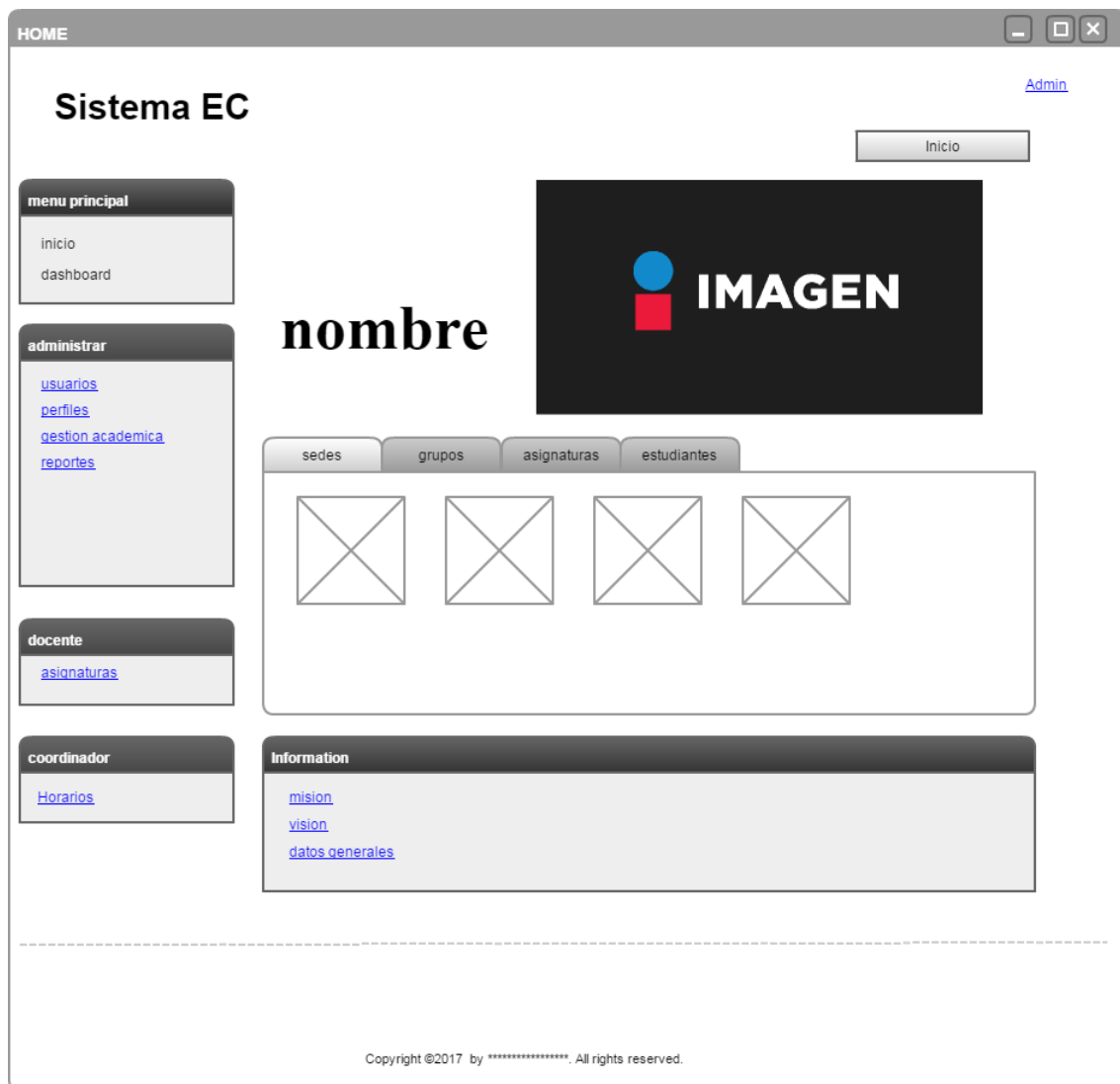


FIGURA 5 PROTOTIPO DE LA PAGINA DE INICIO DEL DASHBOARD

En este prototipo se detalla la vista después del logueo para un usuario con tipo de perfil administrador; donde se encuentran los diferentes menús a los que tiene él acceso.

3.1.4. Diagrama de Casos de Uso

Sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios u otros sistemas, además permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.

A continuación, se muestra algunos diagramas de casos de usos del sistema, los demás pueden ser consultados en el anexo.4.

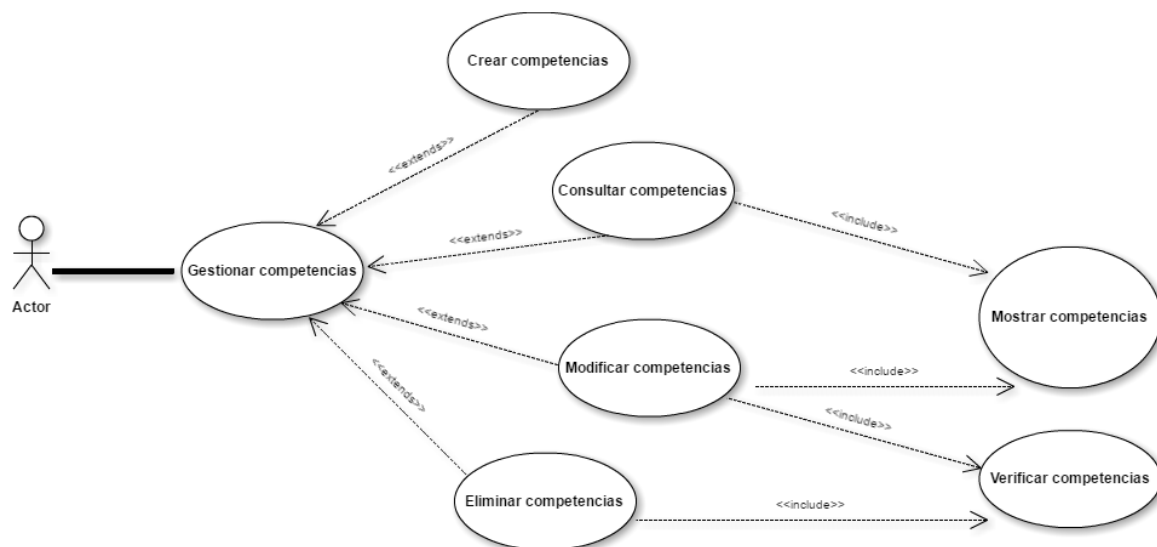


FIGURA 6 GESTIÓN DE COMPETENCIAS ACADÉMICAS

Este diagrama hace referencia al caso de uso de crear, editar eliminar y consultar una competencia, donde se detalla el proceso que debe realizar el usuario con perfil administrador para gestionar los estándares de competencia, siendo esta gestión la base para más adelante realizar los procesos de las configuraciones de componentes, donde se determinara que competencias se evaluarán en cada periodo académico y evidenciar las debilidades de competencias por medio de los reportes.

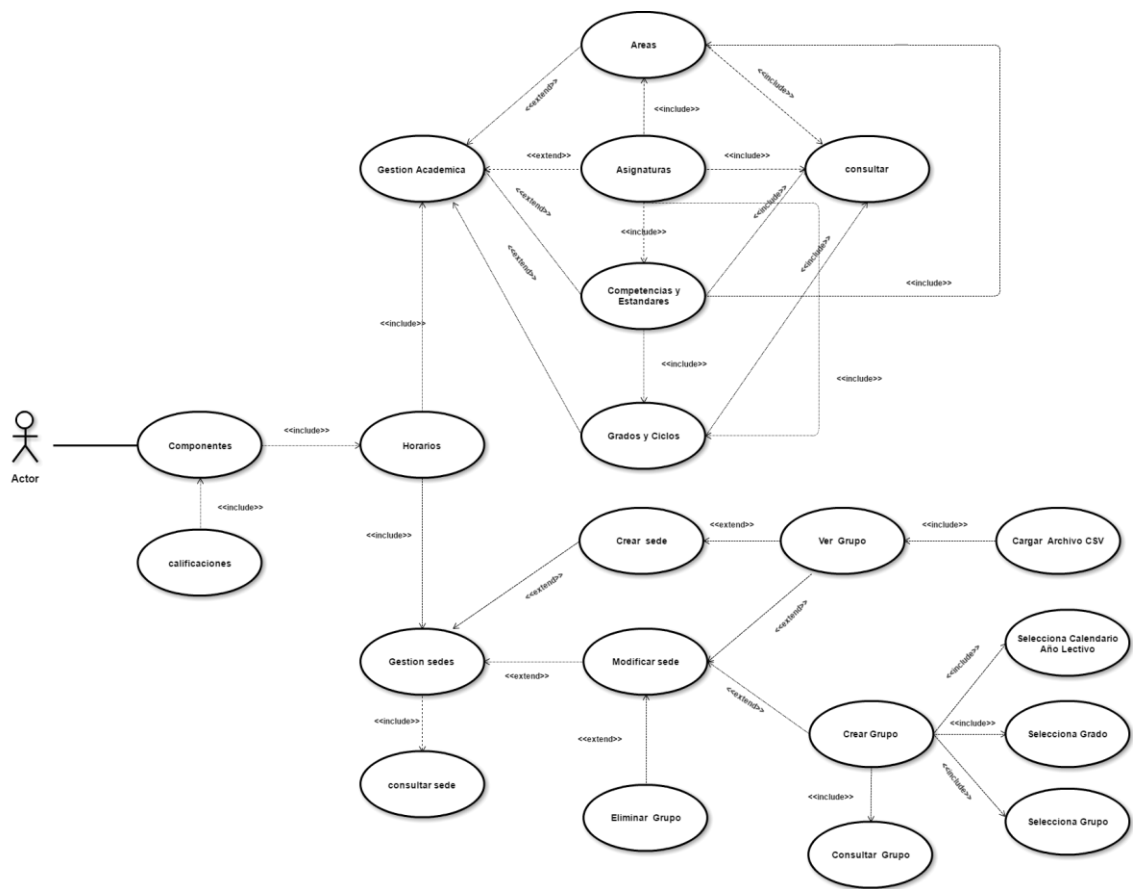


FIGURA 7 GESTIÓN DE COMPONENTES

Este diagrama hace referencia al caso de uso de gestión de componentes, donde muestra los pasos que se debe realizar para las configuraciones de los estándares de competencias que se van evaluar en los periodos académicos. Primero se debe crear las sedes académicas de la institución educativa principal. Luego se debe crear la carga académica (áreas, asignaturas, competencias, estándares, grados). Una vez configurado lo anterior, se procede a realizar el respectivo proceso de cargar los usuarios mediante el archivo SIMAT con sus respectivos datos de la institución educativa, luego se conforman los grupos académicos. Una vez que se haya culminados todos estos procesos, se procede a configurar la carga académica que va tener los docentes (horarios) y finalmente se procede a configurar los componentes donde se determinara que estándares de competencias se van evaluar en determinados periodos académicos.

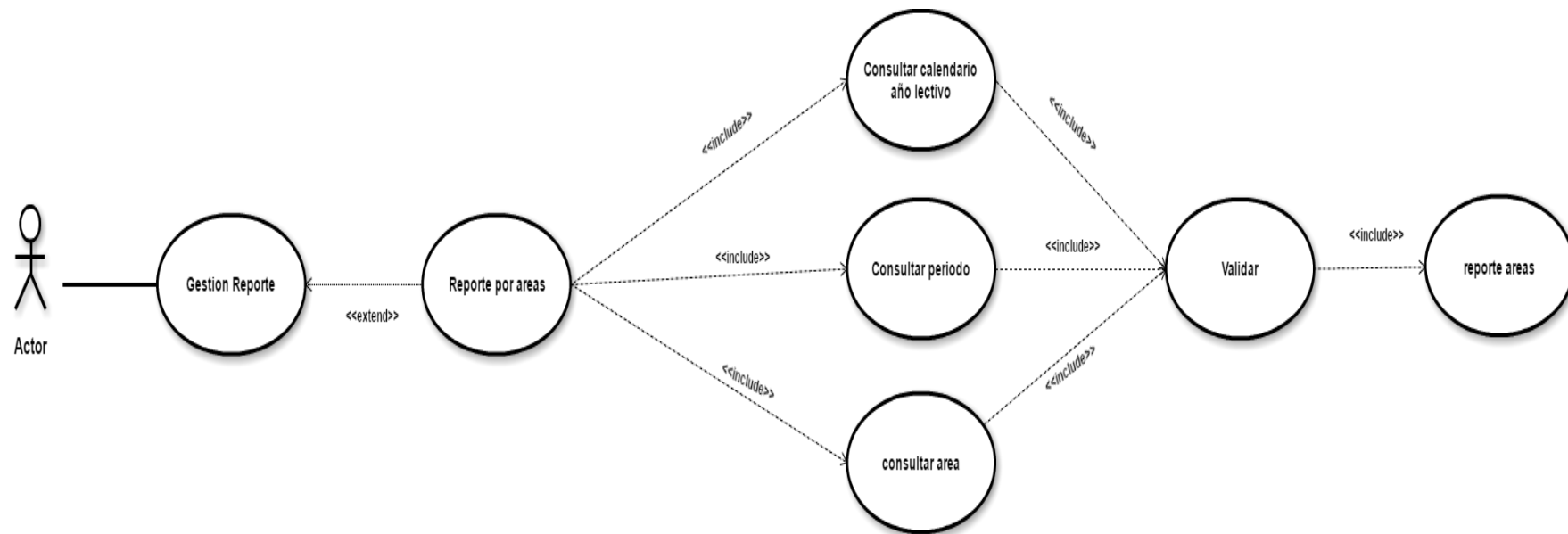


FIGURA 8 CASO DE USO DE REPORTES POR ÁREAS

Este diagrama hace referencia al caso de uso de reportes por áreas académicas, donde se detalla el proceso que debe realizar para determinar las debilidades de los estandartes de competencia que tiene los estudiantes en todas las sedes que conforman la institución educativa.

3.2. Análisis y Diseño Preliminar

3.2.1. Descripción de Casos de Uso

Se siguió con la creación de los ficheros de casos de usos, donde se describen explícitamente el quehacer de los actores dependiendo de su permiso, para brindar un mejor flujo de interacción del actor con el sistema con el objetivo que las funcionalidades sean reflejadas y además los casos de usos tienen cierta base para la construcción de los diagramas UML dependiendo de la metodología y especialmente para los casos de pruebas

A continuación, se describe un resumen de los casos de uso más relevantes, los demás pueden ser consultados en el anexo 5.

TABLA 3 CASO DE USOS CREAR COMPETENCIAS

CASO DE USO		CREAR COMPETENCIAS	
Código:	CU-36	Referencia Requerimiento:	R16-R24- R41-R42-R43
		Referencia Caso de Uso:	CU-18 – CU-20-CU-24-28
Actores	Usuario con este permiso.		
Propósito	Permite crear competencias		
Resumen	Este caso indica la forma como el usuario que tenga un perfil con los permisos correspondientes, debe diligenciar los datos que son indispensables para que crear una nueva competencia.		
Precondición	El usuario debe haberse autenticado y tener los permisos para crear competencias		
Postcondición	Con la competencia creada se permite la asociación de áreas de conocimientos		
CURSO NORMAL DE LOS EVENTOS			
CREAR COMPETENCIAS			
Paso	Acción de los Actores	Respuesta del Sistema	
CN-1	Ingresa al módulo de gestión académica		
CN-2		El sistema despliega la siguiente configuración de la gestión académica: áreas, asignaturas, competencias y grados y ciclos	
CN-3	Ingresa al módulo de competencias y selecciona la opción nueva competencia.		
CN-4		Muestra el formulario que se va a diligenciar teniendo en cuenta los permisos de acuerdo al perfil que tenga asignado el usuario.	
CN-5	Ingresa los datos en los campos obligatorios: Descripción de la competencia, área de conocimiento, ciclo, y estándar de conocimiento, selecciona la opción guardar.		
CN-6		Verifica que los datos ingresados sean válidos de acuerdo con la información almacenada en el sistema.	
CN-7		Se guardan los datos, se informa que la competencia se ha registrado	
CURSO ALTERNO 1			

Datos no validos		
CA1-1	Ingresa los datos en los campos obligatorios de la nueva competencia.	
CA1-2		Verifica que los datos ingresados sean válidos de acuerdo con la información almacenada en el sistema, si existe la competencia mostrara un mensaje como el siguiente “ Ya existe una competencia con este nombre ”
CA1-3		Verifica que los datos ingresados sean válidos de acuerdo con la información almacenada en el sistema, si existe campos vacíos, este mostrara un mensaje como el siguiente “Complete este campo” e indicando el campo respectivo.

TABLA 4 CASOS DE USO CONFIGURAR COMPONENTES ACADÉMICOS

CASO DE USO		CREAR COMPONENTES	
Código:	CU-44	Referencia Requerimiento:	R41-R42-R43-R23-R24-R27-R5-R6-R16-R19
		Referencia Caso de Uso:	CU-18 – CU-20-CU-24-CU-01-CU-04-R40-R28-R32
Actores	Usuario con este permiso.		
Propósito	Permite adicionar componentes a la carga académica		
Resumen	Este caso indica la forma como el usuario que tenga un perfil con los permisos correspondientes, debe diligenciar los datos que son indispensables para crear adicciones de componentes		
Precondición	El usuario debe haberse autenticado y tener los permisos para crear los componentes a su grupo.		
Postcondición	Adiciones de estándares y competencias para el grupo en dicho periodo académico		
CURSO NORMAL DE LOS EVENTOS			
CREAR COMPONENTES			
Paso	Acción de los Actores	Respuesta del Sistema	
CN-1	Ingresa al módulo de asignaturas.		
CN-2		El sistema despliega la información de asignaturas.	
CN-3	Selecciona la opción año lectivo y da click en filtrar.		
CN-4		El sistema despliega la información de edición	
CN-5	Da click en horarios		
CN-6		El sistema despliega la información de edición de componentes.	
CN-7	Selecciona la competencia, el estándar y el periodo al que quiere adicionarle los componentes, da click en el botón adicionar estándar. Ver cuadro 1		
		El sistema guarda la información y muestra en una tabla los registros. Ver cuadro 1	
CURSO ALTERNO 1			
Datos no validos			
CA1-1	Selecciona la competencia, el estándar y el periodo al que quiere		

	adicionarle los componentes, da click en el botón adicionar estándar. Ver tabla18	
CA1-2		
CA1-3		Verifica que los datos ingresados sean válidos de acuerdo con la información almacenada en el sistema, si existe campos vacíos, este mostrara un mensaje como el siguiente “Selecione” e indicando el campo respectivo.
CA1-4		Verifica que los datos ingresados sean válidos de acuerdo con la información almacenada en el sistema, si presenta algún tipo de inconsistencia, este mostrara un mensaje.
CURSO ALTERNO 2		
Datos no validos		
CA1-1	Da click en horarios	
CA1-2		El sistema despliega la configuración de usuarios. Ver tabla13
CA1-3	Realiza los pasos CN-3 al CN-7	
CURSO ALTERNO 3		
Datos no validos		
CA1-1	Da click en horarios	
CA1-2		El sistema despliega la configuración de usuarios. Ver tabla13
CA1-3	Realiza los pasos CN-3 al CN-7	

Cuadro 1. configuración de componentes					
Profesor	Estándares a calificar (xx/xx/xxxx - xx/xx/xxxx)				
Xxxxxxx	Competencia	Estándar	Periodo		
Calificaciones	Adicionar Estándar			Opciones	
	Periodo-Estándar	Periodo	Editar	Guardar	Eliminar

TABLA 5 CASOS DE USOS REPORTES POR ESTUDIANTES

CASO DE USO	GENERAR REPORTE POR ESTUDIANTES		
Código:	CU-13	Referencia Requerimiento:	R01, R02, R03, R04, R05,R06,R07,R08, R21, R22, R23,R24,R25, R26, R27, R28, R29, R30
		Referencia Caso de Uso:	CU-01, CU-05, CU-08, CU-11, CU-12
Actores	Usuario con este permiso		
Propósito	Generar reportes por estudiantes		
Resumen	En este caso de uso el actor puede generar reportes por estudiante sobre su rendimiento en las competencias y estándares de competencia		
Precondición	El usuario debe tener los permisos correspondientes.		
Postcondición	Consultar reportes por estudiantes		
CURSO NORMAL DE LOS EVENTOS			

Paso	<u>Acción de los Actores</u>	<u>Respuesta del Sistema</u>
CN-1	El actor ingresa al módulo de reportes	
CN-2		El sistema despliega la configuración de consultar reportes por estudiantes Además muestra un mensaje de notificación como el siguiente: <i>Selección de sede</i> <i>Seleccione el año lectivo y la sede para filtrar los grados, para realizar el reporte del grupo o estudiante.</i> <i>Si selecciona Reporte de sede(s) desplegara el selector para filtrar las sedes y obtener un reporte de la(s) sede(s).</i>
CN-3	El actor selecciona una fecha del año lectivo y selecciona una sede y da click en siguiente	
CN-4		El sistema muestra la siguiente configuración de la tabla 1
CN-5	El actor selecciona los atributos del cuadro 2 y el da click en siguiente	
CN-6		El sistema muestra el listado de estudiante de acuerdo a la configuración del CN-2 Y CN-4 Además muestra un mensaje de notificación como el siguiente: "Reporte de estudiante o grupo Seleccione el botón de reporte en el estudiante para ir al reporte del estudiante. Puede ver el reporte del grupo"
CN-7	El actor da click en el icono de reportes en uno de los estudiantes del listado respectivo	
CN-8		El sistema muestra la configuración de la tabla 2
CN-9	El actor selecciona los atributos del cuadro 3 y el da click en siguiente	
CN-10		El sistema muestra el reporte de estudiante por medio de gráficos estadísticos(gráficos de barras) para las competencias y estándares de competencias respectivamente
CURSO ALTERNO 1		
CA1-7	El actor da click en reporte del grupo	
CA1-8		El sistema muestra la configuración del cuadro 3
CA1-9	El actor selecciona los atributos del cuadro 3 y el da click en siguiente	
CA1-10		El sistema muestra el reporte del grupo por medio de gráficos estadísticos (gráficos de barras) para las competencias y estándares de competencias respectivamente
CURSO ALTERNO 2		
CA2-3	El actor da click en el boton reporte de sedes	
CA2-3.1		El sistema despliega la configuración de consultar reportes por sedes

CA2-3.3	El actor escoge que sede(s) va cargar para el reporte y da click en siguiente	
CA2-8		El sistema muestra la configuración de la tabla 2
CA2-9	El actor selecciona los atributos de la tabla 2 y el da click en siguiente	
CA2-10		El sistema muestra el reporte de sedes por medio de gráficos estadísticos(gráficos de barras) para las competencias y estándares de competencias respectivamente
CURSO ALTERNO 3		
CA-3	El actor deja vacíos los campos y le da en siguiente	
CA-3.1		El sistema muestra una notificación de mensaje como el siguiente: ¡No es posible consultar los estudiantes debido a que faltan algunos campos!” Y lo devuelve al CN-3

Cuadro 2. Selección de grado, grupo y jornada.	
Descripción	Tipo
Grado	Seleccionable
Grupo	Seleccionable
Jornada	Seleccionable

Cuadro 3. Selección del periodo y área	
Descripción	Tipo
Año lectivo	Seleccionable
Periodo	Seleccionable
Area	Seleccionable

3.2.2. Diagrama de Robustez

Luego de haber finalizado los ficheros de casos de usos, se empieza a proceder a realizar los Diagrama de Robustez, que es una herramienta que nos permite capturar el que hacer y a partir de eso él como hacerlo. Facilita el reconocimiento de objetos y hace más sencilla la lectura del sistema. Ayuda a identificar los objetos que participan en cada caso de uso.

A continuación, se muestra los Diagramas de Robustez más relevantes, los demás pueden ser consultados en el anexo 6.

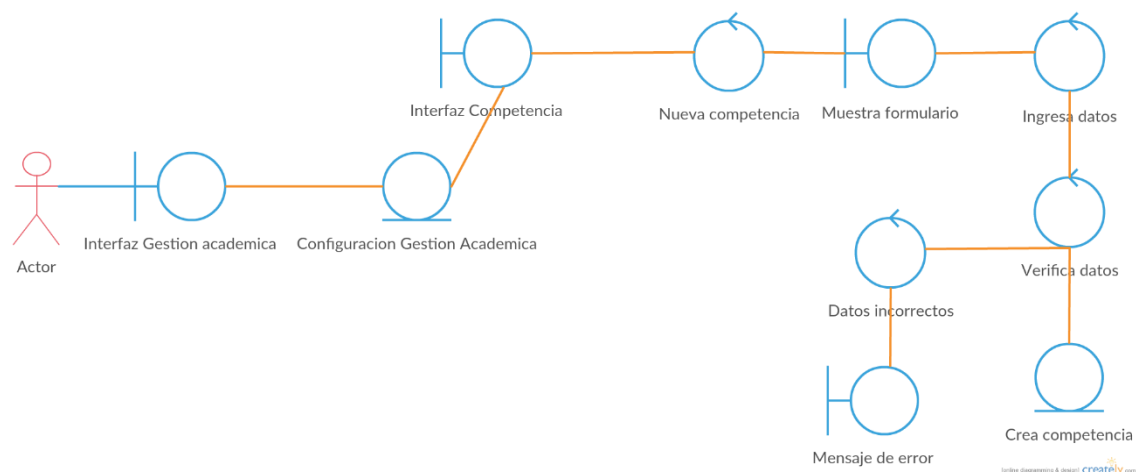


FIGURA 9 DIAGRAMA ROBUSTEZ CREAR ESTÁNDAR DE COMPETENCIA

Este diagrama de robustez representa el caso de uso de crear una competencia académica, mediante el uso de objetos fronterizos, control y entidad, que identifican de forma detallada el proceso que realiza el usuario con perfil administrador para llevar a cabo la gestión de estándares de competencias.

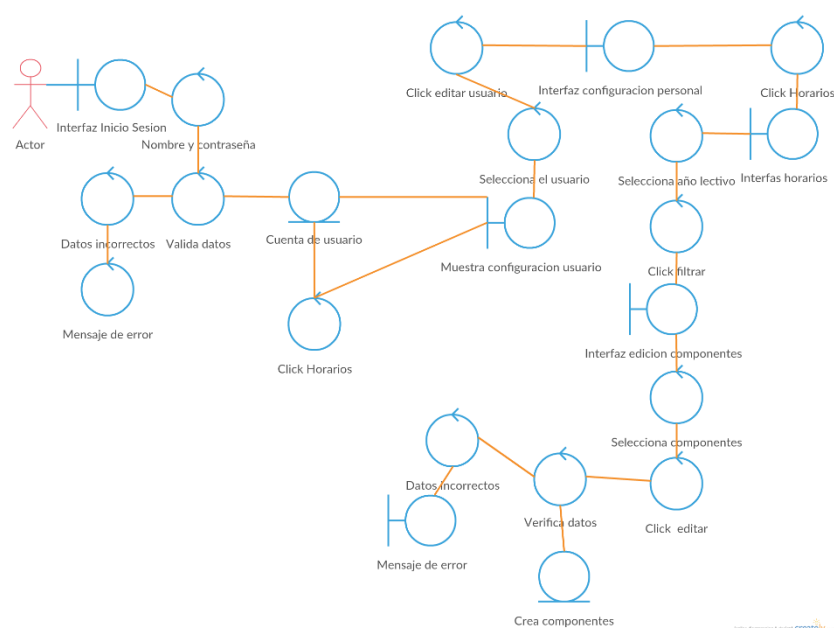


FIGURA 10 DIAGRAMA ROBUSTEZ CONFIGURACIÓN DE COMPONENTES ACADÉMICOS

Este diagrama de robustez hace referencia al caso de uso de crear, editar, eliminar y consultar un componente mediante los perfiles administrador y docente, que permite analizar y observar el modo de interacción entre el sistema y los actores externos a este, determinado por el proceso de selección de competencias, estándares y periodo, si se están realizando de manera razonable y lógica.

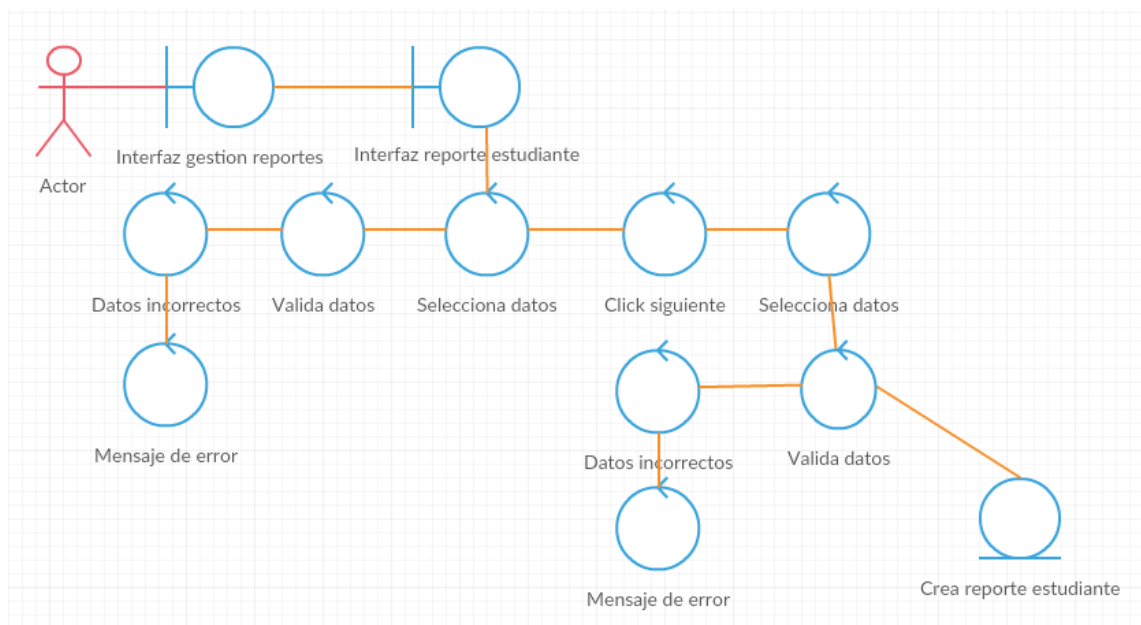


FIGURA 11 DIAGRAMA ROBUSTEZ REPORTE DE ESTUDIANTES

Este diagrama de robustez representa el caso de uso de consultar los reportes por estudiantes, detallando el proceso que realiza un perfil administrador para la gestión de reportes sobre su rendimiento académico en las competencias y estándares de competencias, permitiendo verificar si el flujo es correcto y razonable.

3.3. Revisión crítica del diseño

3.3.1. Diagramas de navegación

Muestran una estructura de los contenidos que van aparecer en el sitio web, además te permite conocer el orden que lleva el aplicativo web, pero no sólo eso, sino que muestra las rutas a nivel de implementación de cómo está funcionando el aplicativo en cuanto a sus patrones de arquitectura de software como MVC ⁶ o cuando se usan Frameworks. En el presente proyecto se describen los modelos de navegación bajo la arquitectura de Laravel.

A continuación, se muestra los Diagramas de Navegación más relevantes, los demás pueden ser Consultados en el anexo 7.

⁶ Modelo-Vista-Controlador

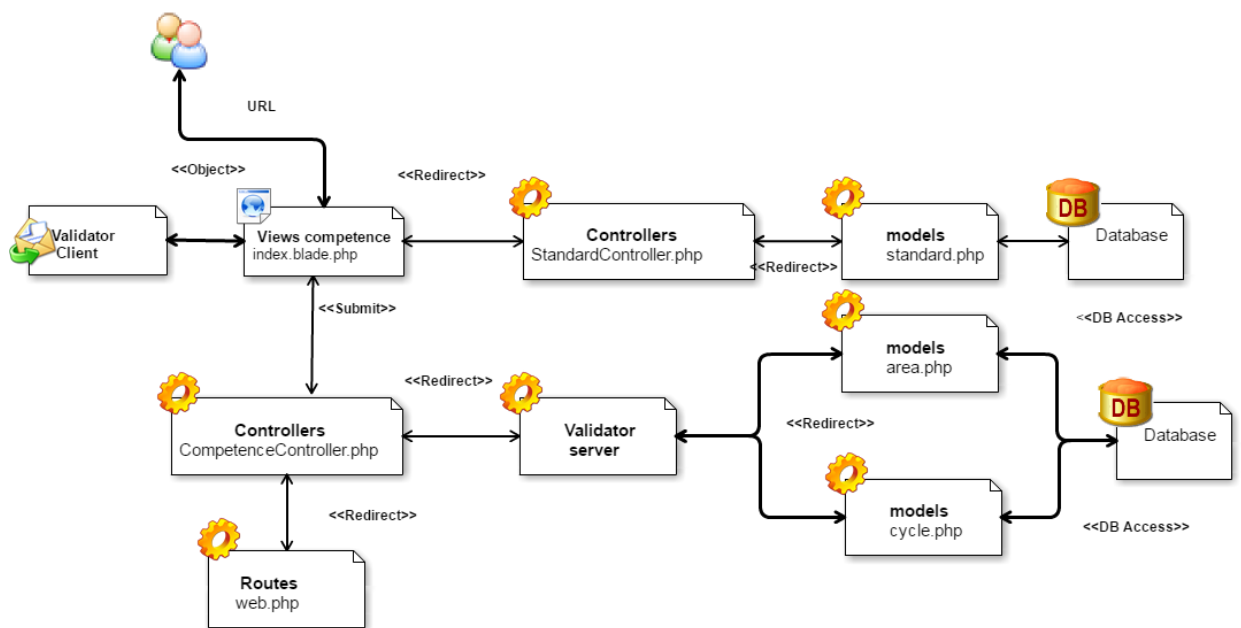


FIGURA 12 DIAGRAMA DE NAVEGACIÓN DE CREAR ESTÁNDAR DE COMPETENCIA

Este diagrama hace referencia al diagrama de navegación de crear un estándar de competencia. Donde se muestra el flujo de las diferentes capas que constituye el framework de Laravel, en cuando se manda una petición al servidor para crear una competencia académica, pasando por las rutas, controladores, modelos y finalmente a la base de datos donde la petición regresara a la capa del cliente.

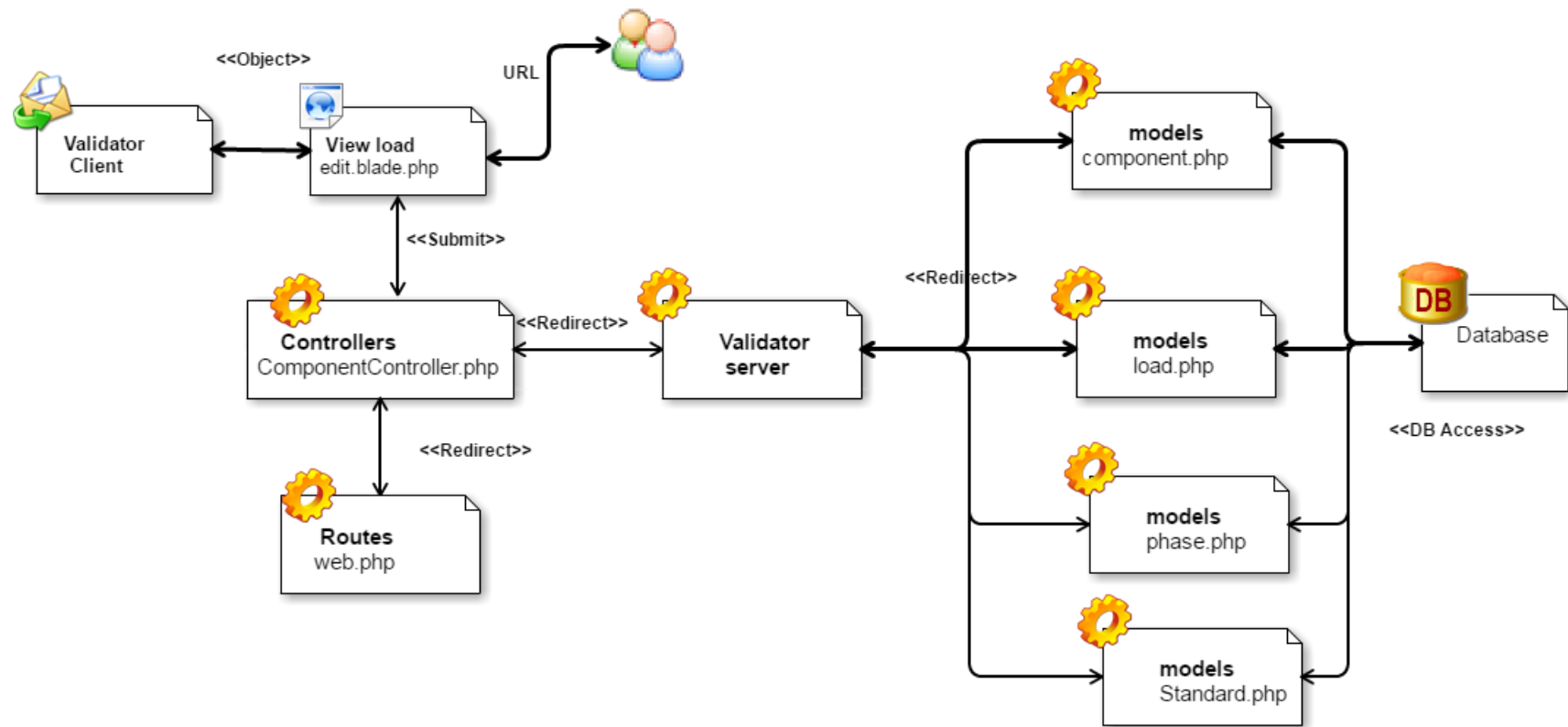


FIGURA 13 DIAGRAMA DE NAVEGACIÓN CONFIGURACIÓN DE COMPONENTES ACADÉMICOS

Este diagrama hace referencia al diagrama de navegación de configuración de componentes académicos. Donde se muestra el flujo de las diferentes capas en el momento que el usuario está configurando los componentes académicos que se evaluará en determinados periodos académicos.

Este diagrama hace referencia al diagrama de navegación de reportes de estudiantes Donde se muestra el flujo de las diferentes capas en el momento que el usuario está filtrando por año lectivo, periodo académico, grupo, grado y área para hacer el debido reporte de las debilidades que hay en dicho alumno.

3.3.2. Diagramas de clases

Describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ello, en la metodología ICONIX se refina el modelo de dominio en un diagrama de clases más elaborado, no tiene que ser tan explícitamente en detalle, pero que si muestre la generalización y herencia del mismo. En el presente proyecto como se usó un Framework se describe las capas y/o rutas que maneja la arquitectura de Laravel. Para mirar el diagrama detalladamente con sus rutas y las relaciones de las clases con sus atributos y métodos puede consultar el anexo 8.

3.3.3. Diagrama de Modelo Relacional

El modelo relacional constituye una alternativa para la organización y representación de la información que se pretende almacenar en una base de datos. Se trata de un modelo teórico o que, además de proporcionarnos los elementos básicos de modelado como las relaciones incluye un conjunto de operadores para su manipulación, sin ambigüedad posible. Detalla las relaciones que hay en cada entidad una con otras para así determinar la manipulación de datos que se tendrá en una consulta cuando se esté desarrollando. Cabe resaltar que una muy buena base de datos optimiza muchos los procesos en cuanto a código, y a testeo y sobre todo optimiza costos computacionales.

En el presente proyecto, no importa que motor de base de datos se use, ya que Laravel nos proporciona crear tablas en la base de datos mediante en una interfaz orientada a objetos, de esta manera se estará creando un sistema compatible con las distintas bases de datos que soporta Laravel por defecto como MySQL, Postgresql, SQLite3, SQL Server. Por otro lado, laravel maneja lo que se denomina las migraciones, que nos permiten tener un registro de las modificaciones que se han hecho o se harán en la base de datos. Esta función es muy útil cuando el desarrollo se hace de más de dos personas.

A continuación, se muestra el diagrama del modelo relacional del presente proyecto de grado. Para verlo con más detalle consultar el anexo 9.

3.4. Implementación

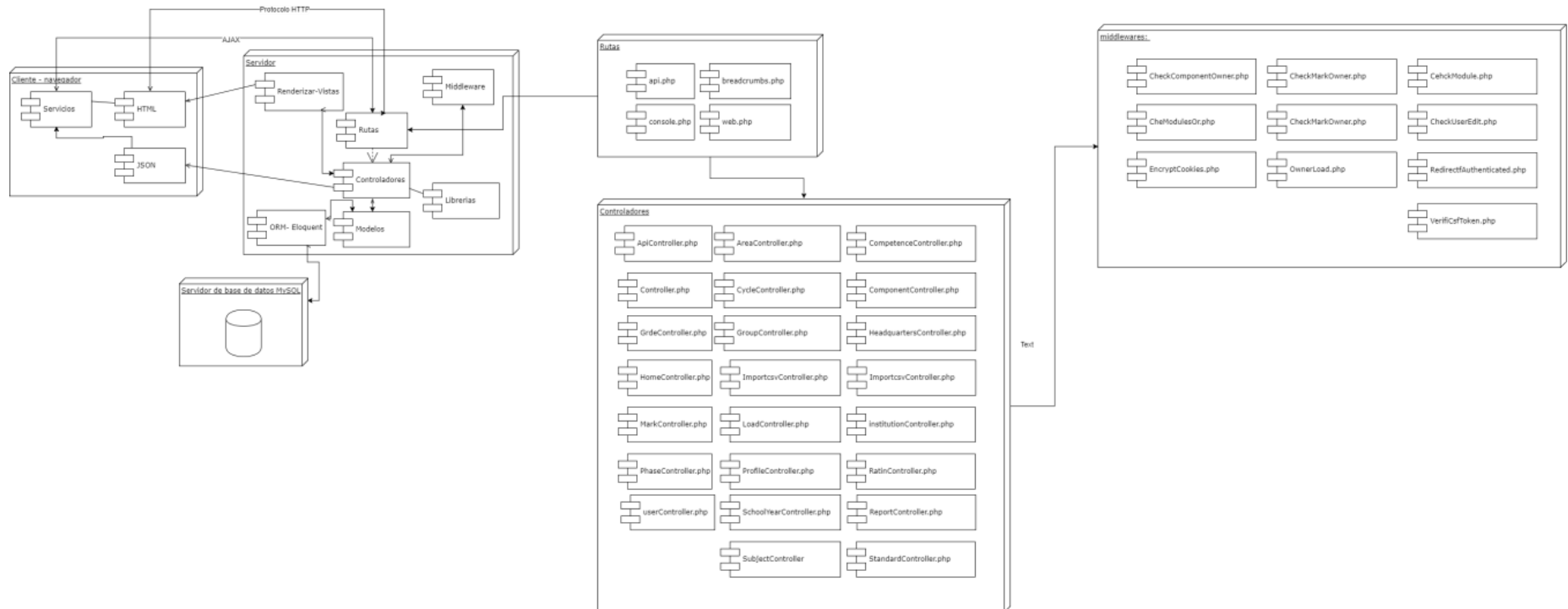


FIGURA 16 DIAGRAMA DE COMPONENTES

Este diagrama hace referencia al diagrama de componentes donde el usuario interactúa con la interfaz enviando peticiones mediante elementos que permitan obtener una respuesta por parte del sistema. Las peticiones realizadas desde el navegador son recibidas en el módulo de rutas mediante las vías de acceso de las diferentes operaciones de rutas acompañado de un método de solicitud, permitiendo definir los puntos finales en los que se pueden realizar solicitudes mediante métodos de acceso como GET y POST.

Una vez obtenida la ruta con la solicitud se procede a enviar al módulo de controladores para responder a las acciones que se solicitan en la aplicación mediante comandos que invocan peticiones al modelo para actualizar su estado al igual que a la vista para cambiar su presentación.

Las solicitudes que se realizan mediante los Middlewares, que tienen acceso a los objetos de solicitud y respuesta permitiendo ejecutar cualquier código, o realizar cambios en las solicitudes y objetos de respuestas, permitiendo invocar y finalizar las funciones y los ciclos de la pila middleware.

Una vez ya recibida la solicitud en el módulo de modelos procede a trabajar con los datos mediante mecanismos para acceder a la información y realizar cambios de su estado.

Este módulo tendrá todas las funciones que le permitirán acceder a las tablas y harán los correspondientes inserts, selects, etc.

Complementos

- **Librerías:**

Son un conjunto de archivos que se usaron con el fin de facilitar la programación y crear grandes resultados en menos tiempo y espacio.

- **Helpers:**

Son un conjunto de funciones ya sean incluidas o creadas por el programador en el framework que se usan de manera interna o globales del sistema las cuales permiten ser utilizadas en nuestras aplicaciones.

En el inicio de la implementación del proyecto se presentaron algunas fallas relacionadas en aplicar nuevas tecnologías y asociarlas con otras para aplicar una funcionalidad. Este tipo de inconvenientes se solucionaron haciendo un proceso de aprendizaje exhaustiva en la parte teórica para tener ciertas bases y de acuerdo al problema solo sería aplicar lo más conveniente. La documentación fue el eje de retroalimentación para adquirir el nivel de aprendizaje necesario, las bases surgieron a través de libros, cursos online, o directamente la documentación de la página principal de la tecnología abordada.

La implementación se desarrolló usando el Framework de Laravel versión 5.3 que funciona bajo el lenguaje de PHP versión 5.4 hasta la actual, entre otras tecnologías que se describirán posteriormente. Laravel es un Framework tan potente que ofrece un patrón de arquitectura de software mucho mejor que el particular MVC (Modelo-Vista-Controlador), donde el controlador es programado como una clase que tiene una variedad de métodos para mandar una petición

HTTTP Request PHP⁷, aunque cabe destacar que Laravel se puede trabajar con la metodología tradicional MVC sin ningún inconveniente.

Sin embargo, Laravel propone trabajar de una manera más rápida y viable, donde se codifica las interacciones HTTP directamente como una función anónima asociada a una Ruta, dando como ventaja minimizar mucho código cuando sólo se requiera mandar una solicitud por una funcionalidad dada.

Las rutas ⁸ en Laravel funcionan similarmente como rutas de seguridad, cuando se consulta la ruta escribes la dirección y el tipo de ruta, en la cual pueden ser de tipo POST, GET, PUT, DELETE, cabe destacar que los navegadores web soporta tipo GET y POST, pero en Laravel se puede enviar una petición de tipo POST emulada, es decir, en una petición de tipo POST donde puedo enviar una ruta DELETE o PUT para eliminación o actualización de algún formulario que quiera enviar al servidor.

Laravel tienen dos carpetas llamadas /web y /api para manejar y escribir nuestras rutas, ya es decisión del usuario como se quiere plantear, esas rutas van a un tipo de controlador con su método asociado o simplemente se crea una función anónima dentro de la ruta para imprimir algo dependiendo de la funcionalidad.

Además Laravel también definen rutas llamadas resource para crear todas las rutas en un CRUD simplificando demasiado código, por ejemplo para editar una funcionalidad, se necesita una ruta para editar y otra para actualizar, el de editar es de tipo GET, el cual genera el formulario general con los datos que tiene el usuario o el modelo propio y para el actualizar (UPDATE) es un tipo POST pero se define como PUT, ya que para el navegador lo recibe como POST, ya que así funciona los navegadores web pero para Laravel como dijimos antes la emula y la define como PUT para enviar la petición.

Ahora se describe el uso de los modelos. Todos los modelos heredan de un modelo padre denominado model, donde abundan demasiadas funciones propias, el modelo como tal hereda muchas funciones que ofrece Eloquent ORM ⁹ para tener persistencia en la base de datos. Un modelo nos ayuda a definir que tabla, atributos que se pueden llenar y que otros se deben mantener ocultos

Eloquent hace uso de los modelos para recibir o enviar la información a la base de datos, Los modelos usan convenciones para que a Laravel se le facilite el trabajo y nos minimice líneas de código, El nombre de los modelos se escribe en singular, en contraste con las tablas de la base de datos que se escriben en plural. Usan notación UpperCamelCase ¹⁰ para sus nombres.

Estas convenciones nos ayudan a detectar automáticamente las tablas, por ejemplo: el modelo User que viene por defecto en Laravel, que se ubica en la carpeta /app. Se encuentra en singular y con notación UpperCamelCase y para Laravel poder definir que tabla es la que está ligada a este modelo le es suficiente con realizar la conversión a notación underscore ¹¹ y plural, dando como resultado la tabla: users. Y esto aplica para cuando queremos crear nuestros modelos, si

⁷ <https://laravel.com/docs/5.3/requests>

⁸ <https://laravel.com/docs/5.3/routing#basic-routing>

⁹ <https://laravel.com/docs/5.3/eloquent>

¹⁰ CamelCase es un estilo de escritura que se aplica a frases o palabras compuestas

¹¹ underscore es un estilo de escritura que se aplica a palabras separas por un guion bajo

tenemos una tabla en la base de datos con la que queremos trabajar que se llama `user_profiles`, vemos que se encuentra con las convenciones para tablas de bases de datos (plural y underscore), entonces el modelo para esta tabla cambiando las convenciones serias: `UserProfile` (singular y UpperCamelCase). Para crear un modelo se usa la consola de elección y se ubica la raíz del proyecto, se procede a ejecutar el comando siguiente: `php artisan make:model User`

Entonces ya ahí entra los controladores donde determinan un papel importante en Laravel, podemos definirlo como quiera, ya que no tienen convenciones como los modelos. Los Controladores puede agrupar las peticiones HTTP relacionada con la manipulación lógica en una clase. Los Controladores normalmente se almacenan en el directorio de aplicación `app/Http/Controllers/`. Además, nos ayudan a agrupar estas peticiones en una clase que se instancia a las rutas, donde las rutas como describimos antes pueden ser tipo resource. Esta ruta nos creara un grupo de rutas de recursos con las peticiones dadas. Por otra parte los controladores nos genera las funciones por defectos que son `index`, `create`, `show`, `edit`, `store`, `update`, `destroy`, que son las operaciones más usadas en una clase y para no tener que crear una ruta para cada método.

Estas operaciones hacen referencia a lo siguiente:

- **Index:** Es el método inicial de las rutas resource, donde se emplea para mostrar una vista en la interfaz principal y donde contiene la información resumida de un modelo dado.
- **Create:** Este método se usa para dirigir la acción a una interfaz donde se agrupa la información para luego almacenarlos en un registro nuevo
- **Show:** esta función permite hacer una consulta de un elemento de la base de datos o de todos los elementos o registros por medio del modelo para realizar una descripción .
- **Edit:** es muy similar al create, pero se usa más en cuestiones de actualizaciones de datos.
- **Store:** Aquí es donde se actualiza un registro en específico que proviene del método create y normalmente redirige al index.
- **Update:** está ligada a la operación edit de edit y en vez de crear un nuevo registro, busca un existente y lo modifica
- **Destroy:** En este método usualmente se destruye o elimina un registro y la petición puede provenir de donde sea siempre y cuando sea llamado con el método DELETE, después puede redirigir al index o a otro sitio dependiendo si logro eliminar o no.

Cabe aclarar que estos métodos no son obligatorios para cada controlador que se cree, se puede usar todas o algunas o crear unas especifica dependiendo de la necesidad. Para crear un modelo ejecutamos el comando siguiente `php artisan make:controller UserController` en nuestra consola de preferencia.

Por último, en la arquitectura que maneja Laravel está la capa de las vistas donde se procede a trabajar con un sistema de plantilla llamado Blade, ¿Qué hace este motor de plantillas? Simplemente nos ayuda a reducir código de nuestras interfaces gráficas, es decir muchas veces las interfaces tienen elementos repetitivos y/o comunes en el sistema en general, Blade lo que hace es mandar a llamar el contenido de una vista en particular llámese pie de páginas, migas de

pan, o menús u otros en una vista aparte donde se quiere que se visualice esos contenidos. Para no estar haciendo varias vistas para una misma acción en particular

Por otro lado, el manejo de las bases de datos difiere mucho a lo convencional. Cuando se programa manualmente sin usar ningún Framework y ORM, toca realizar consultas muy complejas donde particularmente una consulta no sale bien porque la estructura de la base de datos esta mala, toca estructurar el modelo relacionar y actualizar las tablas de la base de datos importando el script o directamente desde el gesto de la BD que estemos generando y no contamos con un control de versiones que nos ayude a simplificar las tareas dadas. En laravel se soluciona este tipo de inconvenientes con el uso de las migraciones.

Las migraciones en Laravel son una herramienta primordial donde podemos hacer un sistema de versiones de nuestros datos sin generar error y maximizando tiempo, podemos definir tablas con POO ¹² en vez de SQL, además y lo más importante de Laravel es compatible con los diferentes motores de base de datos ya que Laravel genera el SQL automáticamente, adaptado al tipo de base de datos con la que estemos trabajando. Ya sea MySQL, Postgres, SQLite, y SQL Server. En el presente proyecto se usó el motor de base de datos MySQL, pero como se describe antes no importa el motor que estemos usando, las modificaciones se hacen directamente en el código

3.4.1. Herramientas usadas

- **Laravel Collective:** Nos ahorra mucha cosa para escribir código en la capa de la vista, usando funciones propias de Laravel que vienen definidas como estilo de ayudas (helpers) en la cual le asocia parámetros de vistas de código de HTML combinado con clase, rutas y a ello se la manda arreglos si es necesario, y así minimiza código y las puede reutilizar para imprimir una etiqueta que necesitemos, en el proyecto se usó para imprimir etiquetas de tipo form e input para cuestiones de formulario.
- **Laravel Breadcrumbs:** En la clase de Breadcrumbs registramos la migas de pan para mandarla en unas rutas y direccionarla para que se muestre en una vista en particular, por ejemplo mandar las migas de pan del registro de inicio para que se vean en la página principal e incluso se puede mandar iconos y tenga mejor navegabilidad. Un ejemplo en el presente proyecto es la sección de calendarios en la cual integramos migas de pan.

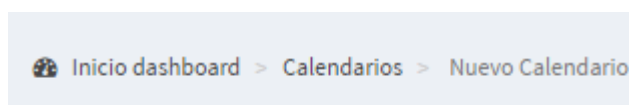


FIGURA 17 EJEMPLO DE MIGAS DE PAN

- **Laracasth/Flash:** se utilizó en el proyecto para generar los mensajes que se muestra en la vista dependiendo de la consulta por ejemplo mensajes de peligro, entrada, riesgo, se maneja directamente desde la capa del controlador.
- **Laravel Excel:** la clase de esta librería se usa para trabajar con archivos de Excel con extensión xls, xlsx y csv, en el contexto del proyecto se usó para extensiones csv, en la

¹² Programación Orientada a Objetos

cual permite recorrer toda la información y sacar lo que necesito mediante técnicas de programación y dar una consulta dependiendo de la necesidad. En el presente proyecto se usó para recorrer los datos necesarios de los estudiantes y sacar información de ello como: nombres y apellidos del estudiante, tipo de documento, institución educativa , sedes, grados, grupos, jornada, sexo

Estas librerías descritas anteriormente vienen incluidas desde el Composer de Laravel

- **Jasny Bootstrap:** Para el funcionamiento de esta librería es bajo el cliente, es simplemente css y javascript, esta librería no se incluye desde el Composer de Laravel, funciona con cualquier página de HTML que incluya script de javascript y Bootstrap. Es una librería que se destaca más por el asunto de ayudas visuales donde se puede pre-visualizar una imagen antes de cargarla, para el presente proyecto se usó por ese motivo para ver la imagen antes de cargarla, su funcionalidad se encuentra en el módulo de sedes para cargar la imagen de institución como tal.
- **Inputmask:** Se utilizó para tener una máscara en los campos de textos de entrada (input) por medio de javascript (js) y css, lo que hace con js en el input es mandarle ciertas propiedades a ese campo para que se visualice guiones bajos para que el usuario pueda determinar que caracteres debe escribir en ese campo y que cantidad por ejemplo quiero escribir números del 1 al 99 número en ese campo, entonces va aparecer los guiones bajos para escribir solo ese intervalo de números. En el proyecto se usó para validar los números en la creación y edición del módulo de calendarios.
- **Ioslider:** Es una librería de JQuery que ayuda a incorporar contenido multi-navegador personalizable, generalmente se usa para aspectos como desplazamiento del banner, o algún efecto deslizante de imágenes y grupos personalizables (Custom Snap Groups) se utilizó esta librería para incorporar Custom Snap Groups en el presente proyecto.
- **Herramientas de Bootsnipp:** se tomó elementos de ese proyecto para organizar de una manera coherente el paso a paso de crear el año lectivo, y del módulo de reportes donde se especifica al usuario que debe realizar primero para seguir con las demás etapas.



FIGURA 18 EJEMPLO DE HERRAMIENTAS DE BOOTSNIPP

- **Datepicker:** En el proyecto se usó necesariamente para las funcionalidades de crear el calendario del año lectivo para determinar las fechas del año lectivo con sus respectivos periodos y darle un mejor aspecto visual.
- **Daterangepicker:** Se trabajó junto con la librería Datepicker, en esta librería se usó particularmente para determinar el límite de rango de fechas que había una con otras

limitando la cantidad de días que el usuario podría seleccionar dependiendo de la fecha inicial escogida.

- **ChartJs:** En el proyecto se usó especialmente para el módulo de reportes donde se describió de manera de graficas de barras el rendimiento de los estudiantes de acuerdo a las calificaciones obtenidas por componentes y estándares de competencias.
- **Fontawesome:** Se usó especialmente para colocar iconos en las migas de pan o en los menús para no cargar imágenes y así evitar tanta carga al servidor con tanta diversidad de imágenes que dependiendo del peso de la imagen como tal ocasiona que la carga sea más lenta.
- **Checked:** Esta librería se usó para los elementos de los formularios del presente proyecto, y así personalizarlo y dejar el típico formulario que viene por defectos de las etiquetas de HTML donde no es tan agradable visualmente. Además, esta librería como su nombre lo dice es para checked elementos, con su entorno visual sobresaliente el usuario puede determinar si realmente está validando algún elemento que este llenando.
- **Select2:** Su uso fue muy importante ya que permitió cargar de manera rápida datos de algún formulario en específico, para el proyecto se utilizó mucho para cargar las competencias que dependía del área académica con su grado respectivo, donde el usuario puede agregar o quitar las competencias rápidamente sin necesidad de cargar la pagina nuevamente.
- **AdminLTE:** La plantilla fue la base del diseño y el tipo de navegabilidad para el presente proyecto, se tomó una plantilla de código libre, en la cual se podía realizar los cambios necesarios dependiendo de la necesidad. En el proyecto se usó muchos elementos visuales de JS y CSS que traía esta plantilla, por ejemplo, mostrar de manera gráfica por medio de unas barrar la cantidad de horas que iba dar un docente en con sus respectivas horas extras. La navegabilidad del menú y lo más importante para no saturar las vistas se usó elementos de cajas donde se podía agrupar muchos datos en una sola interfaz sin necesidad que el usuario al dar click en un botón se cargara la página.
- **Datatables:** En el proyecto implementado se usó para filtrar la información de los datos cargados del archivo csv con su respectiva paginación, no sólo eso, sino para filtrar y paginar todo lo relevante de la carga académica

3.4.2. Descripción de la Interfaz

El desarrollo del presente proyecto se realizó bajo plataforma web con tecnologías descritas anteriormente, el sistema está responsive es accesible desde celular, tablets o dispositivos con acceso a internet. El diseño es muy neutro, con las tecnologías por parte del cliente que se usaron fue de gran ayuda para no saturar tanta la página con muchas vistas debido a la magnitud del proyecto.

A continuación, se muestra algunos pantallazos del aplicativo web. Para más información de cómo funciona el sistema, puede consultar el anexo 12 manual de usuarios.

3.4.2.1. Interfaz calendario año lectivo

En la figura No 18 se muestra la primera visualización del módulo del calendario del año lectivo, en la cual se aprecia una vista muy neutra, donde se usó las herramientas de Bootsnipp, para crear el paso a paso de la configuración de dicho modulo, para no saturar tanto la vista y le quedara más amigable la interfaz al usuario. En la figura No 22 que pertenece el segundo paso de la configuración se usa las tecnologías de Datepicker y Daterangepicker para la configuración del rango de fechas. En el tercer paso, figura No 23 se muestra la configuración del rango de fechas para los periodos académicos y finalmente en el paso 4, figura No 24 se da un resumen de la configuración del año lectivo que se va crear.

En la figura No 24, es el apartado de edición del año lectivo donde se podrá editar la fecha del año lectivo y/o las fechas del periodo académico, además se visualiza una gráfica circular de como quedo dividido los periodos en ese año lectivo más los días que están sin asignar.

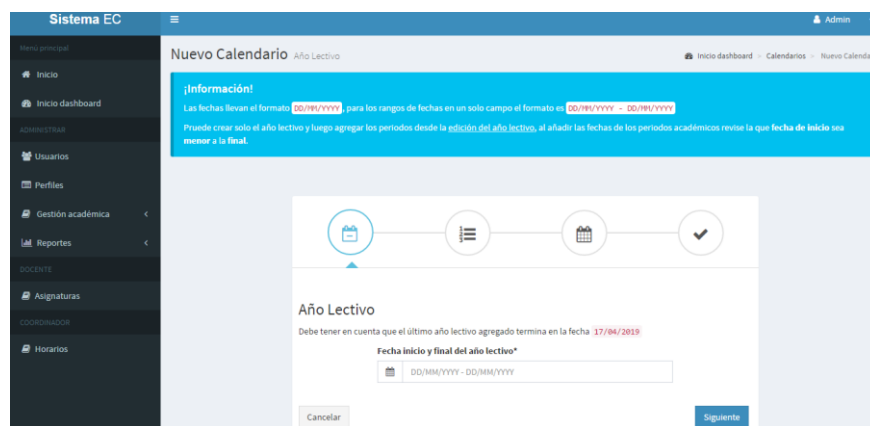


FIGURA 19 INTERFAZ DEL MÓDULO DE GESTIÓN DE CALENDARIO

La interfaz del año lectivo está dada por un filtro de pasos en donde el usuario tendrá que realizar las determinadas configuraciones para crear el año lectivo con sus respectivos periodos.

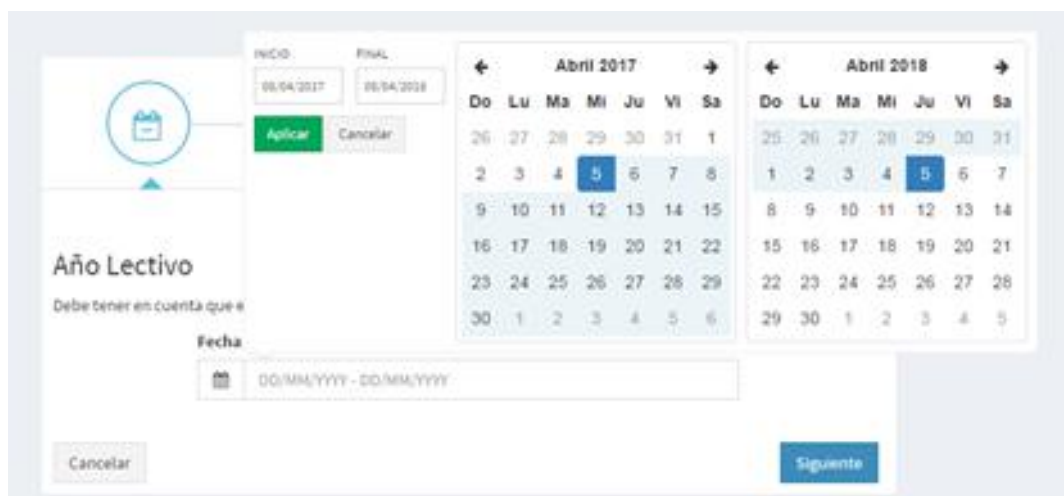


FIGURA 20 CONFIGURACIÓN DE FECHA DEL AÑO LECTIVO

la interfaz se compone de dos calendarios donde el usuario podrá escoger el rango de fecha validando que la fecha final no sea menor que la inicial.



Periodos del año lectivo

Numero de periodos que tendrá el año lectivo, estos podrán ser modificados en la edición del año lectivo.

Numero de Periodos

4

Cancelar Anterior Saltar Continuar

FIGURA 21 CONFIGURACIÓN DE LA CANTIDAD DE PERIODOS

La interfaz está compuesta por un campo de texto donde se determinará cuantos periodos académicos se van crear en el año lectivo correspondiente.

📅 — 1/4/2018 — 📅 — ✓

Periodos del año lectivo

Inicio y final de los periodos, las fechas podan ser modificadas en la e

Periodo	Fechas	
1°	06/04/2017	hasta
2°	24/07/2017	hasta
3°	02/09/2017	hasta
4°	14/01/2018	hasta

« **April 2018** »

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Cancelar
Anterior
Saltar
Continuar

FIGURA 22 CONFIGURACIÓN DE FECHA DEL AÑO LECTIVO

Información 12/04/2018

Fecha de inicio y final*

📅 12/04/2017 - 12/04/2018

ID: 1

Volver
Guardar

Año lectivo periodos

Dias periodo: 13/04/2017 - 13/07/2017
 Dias periodo: 14/08/2017 - 14/11/2017
 Dias periodo: 15/11/2017 - 02/02/2018
 Dias periodo: 03/02/2018 - 10/04/2018
 Dias sin asignar



Periodos academicos

Fecha inicio y final del periodo*

hasta

+ Agregar Periodo

Periodo	Fecha de inicio y final (d/m/Y)			
1°	13/04/2017	hasta 13/07/2017	📅	🗑️
2°	14/08/2017	hasta 14/11/2017	📅	🗑️
3°	15/11/2017	hasta 02/02/2018	📅	🗑️
4°	03/02/2018	hasta 10/04/2018	📅	🗑️

FIGURA 23 INTERFAZ DE EDICIÓN DEL CALENDARIO DEL AÑO LECTIVO.

3.4.2.2. Interfaz reporte de sedes

En la figura No 26 se muestra la interfaz principal del módulo de reportes por sedes, donde debemos escoger que sedes se van cargar el sistema para determinar el rendimiento académico como se aprecia en la figura No 27. Luego se pasa a la configuración de datos donde hay un filtro de datos para hacer el reporte, entre ellos son: el año lectivo, el periodo y el área académica, figura No 28. Finalmente en las figuras No 29 y No 30 se aprecia por graficas estadísticas de barras el rendimiento académico de las sedes teniendo en cuenta sus competencias y los estándares de competencia, además al lado izquierdo de las gráficas se muestra su respectiva nota y la descripción del estándar evaluado

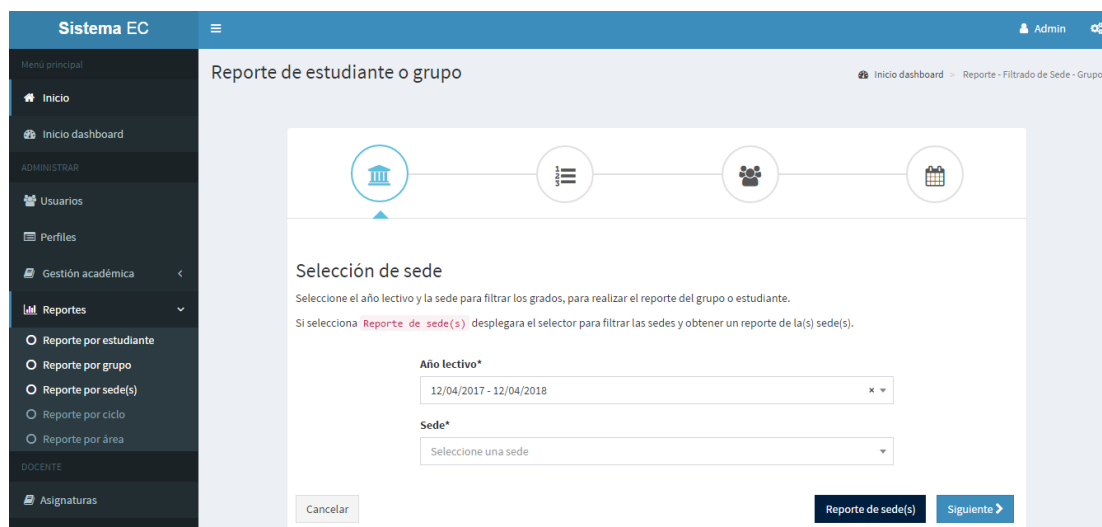


FIGURA 24 GESTIÓN DE REPORTE POR SEDES



FIGURA 25 CARGAR LAS SEDES DEL SISTEMA

Selección del periodo y área

Seleccione un año periodo y un área académica para generar el reporte de la(s) sede(s):

- SEDE PRINCIPAL NO. 7 MARIA ANTONIA RUIZ
- SEDE NO. 1 FRANCISCO JOSE DE CALDAS

Año lectivo*

Año lectivo

Periodo*

Periodo

Seleccionar Área*

Seleccione un área

[Cancelar](#) [Siguiente](#)

FIGURA 26 CONFIGURACIÓN DE DATOS PARA LAS SEDES

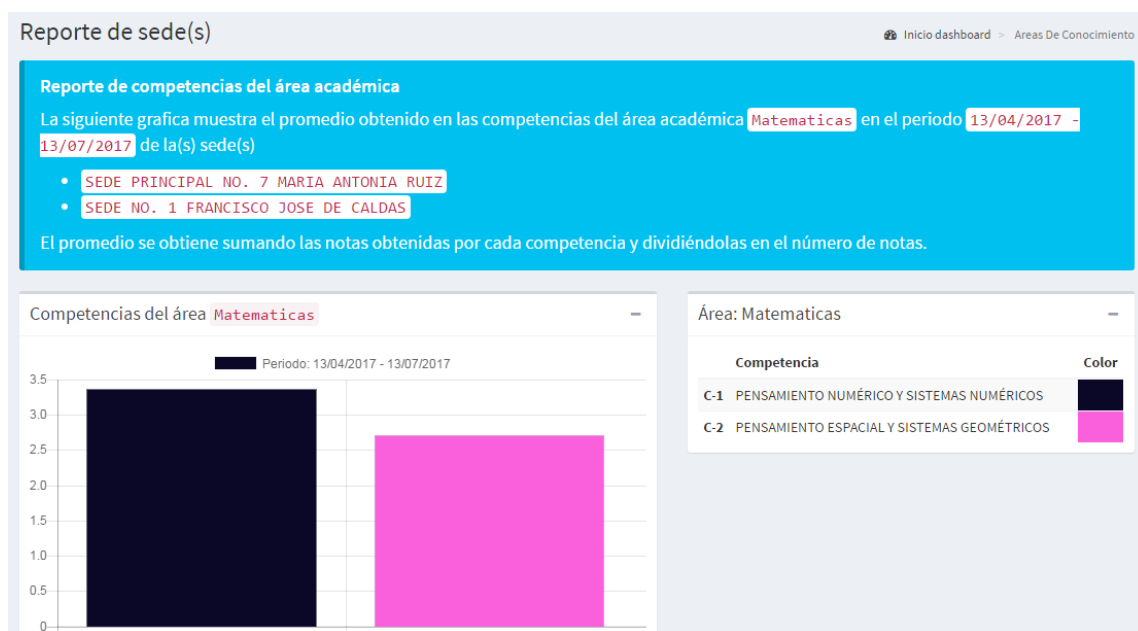


FIGURA 27 GRAFICA DE BARRA PARA EL PROMEDIO DE COMPETENCIAS

La grafica de barras hace referencia a las debilidades que se están presentando los estudiantes en una sede en particularidad, donde se evidencia el promedio de la nota obtenida de las calificaciones de las competencias académicas en determinado periodo y área.



FIGURA 28 GRAFICA DE BARRA PARA EL PROMEDIO DE ESTÁNDARES DE COMPETENCIA

La grafica hace mención a las debilidades que presentan los estudiantes por cada estándar de competencia de acuerdo a su componente. La grafica de barras es calculada por el promedio obtenido por las calificaciones que genero el docente en cada estándar académico en dicho periodo y área.

Capítulo 4

Pruebas

Las pruebas de softwares una de las más costosas del ciclo de vida del proyecto. Debido a que todo producto antes de su fecha de culminación debe pasar por varios procesos valorativos para estimar una alta calidad. Las pruebas fueron realizadas en diferentes momentos del desarrollo del proyecto y según lo que plantea la metodología ICONIX teniendo en cuenta las iteraciones. En el proyecto se usaron pruebas unitarias y pruebas usando la consola de Tinker de Laravel para determinar que los datos que entraban se almacenaran correctamente en la BD.

4.1. Pruebas unitarias

Todos los módulos tuvieron que pasar las pruebas unitarias antes de ser liberados o implementados en el servidor. Una prueba unitaria, o “unit test”, es un método que prueba una unidad estructural de código.

Las demás pruebas pueden ser consultadas en el anexo 10.

TABLA 6 PRUEBA UNITARIA CREAR AÑO LECTIVO

Descripción	Crear año lectivo
Objetivos	Verificar que los datos ingresados sean válidos y almacenados en la base de datos.
Condiciones	Ingresa al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de calendario, selecciona la opción Nuevo, ingresa y selecciona los datos y presiona el botón guardar.
Resultado Esperado	Los datos ingresados se almacenan correctamente.
Resultado Obtenido	La prueba fue aprobatoria, los datos fueron almacenados satisfactoriamente.

TABLA 7 PRUEBA UNITARIA MODIFICAR AÑO LECTIVO

Descripción	Modificar año lectivo
Objetivos	Verificar que los datos ingresados sean válidos y almacenados en la base de datos.
Condiciones	Ingresa al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de calendario, el sistema muestra los años lectivos creados, da click en el icono de editar, determina que acción va realizar y da click en aplicar y guardar .
Resultado Esperado	Los datos ingresados se almacenan correctamente.
Resultado Obtenido	La prueba fue aprobatoria, los datos fueron almacenados satisfactoriamente.

TABLA 8 PRUEBA UNITARIA ELIMINAR AÑO LECTIVO

Descripción	Eliminar año lectivo
Objetivos	Verificar que los datos ingresados sean válidos y eliminados de la base de datos.

Condiciones	Ingresar al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de calendario, se muestra los años lectivos creados y da click en el icono eliminar.
Resultado Esperado	Los datos se eliminan de forma correcta.
Resultado Obtenido	La prueba fue aprobatoria, los datos fueron eliminados satisfactoriamente.

TABLA 9 PRUEBA UNITARIA DE CREAR SEDES

Descripción	Crear sedes
Objetivos	Verificar que los datos ingresados sean válidos y almacenados en la base de datos.
Condiciones	Ingresar al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de sedes, selecciona la opción Nuevo, ingresa los datos y presiona el botón guardar.
Resultado Esperado	Los datos ingresados se almacenan correctamente.
Resultado Obtenido	La prueba fue aprobatoria, los datos fueron almacenados satisfactoriamente.

TABLA 10 PRUEBA UNITARIA DE CREAR GRUPOS

Descripción	Crear grupos
Objetivos	Verificar que los datos ingresados sean válidos y almacenados en la base de datos.
Condiciones	Ingresar al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de sedes, selecciona la opción editar, click en grados y grupos, se seleccionan los datos y presiona el botón guardar.
Resultado Esperado	Los datos ingresados se almacenan correctamente.

TABLA 11 PRUEBA UNITARIA DE REPORTES POR ESTUDIANTE

Descripción	Reporte por estudiantes
Objetivos	Verificar que los datos ingresados sean válidos y almacenados en la base de datos, para generar reportes de estudiantes por medio de gráficos estadísticos para las competencias y estándares de competencias respectivamente
Condiciones	Ingresar al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de reportes, filtra por los parámetros dados, click en el icono reportes del estudiante.
Resultado Esperado	Los datos ingresados se muestran de forma correctamente.

TABLA 12 PRUEBA UNITARIA DE REPORTES POR GRUPOS

Descripción	Crear reporte por grupos
Objetivos	Verificar que los datos ingresados sean válidos y almacenados en la base de datos, para generar reportes de grupos académicos por medio de gráficos estadísticos para las competencias y estándares de competencias respectivamente
Condiciones	Ingresar al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de reportes, selecciona una fecha de año lectivo y sede, y los demás parámetros de filtro y da click en el icono reportes por grupo.
Resultado Esperado	Los datos ingresados se muestran de forma correctamente.
Resultado Obtenido	La prueba fue aprobatoria, los datos fueron almacenados y visualizados satisfactoriamente.

TABLA 13 PRUEBA UNITARIA DE REPORTES POR SEDES

Descripción	Reporte por sedes
Objetivos	Verificar que los datos ingresados sean válidos y almacenados en la base de datos, para generar reportes por sedes por medio de gráficos estadísticos para las competencias y estándares de competencias respectivamente
Condiciones	Ingresar al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de reportes, selecciona una fecha de año lectivo, sede, grado, grupo y jornada, selecciona el estudiante, click en el icono reporte de sedes, selecciona los datos, presiona el botón siguiente.
Resultado Esperado	Los datos ingresados se muestran de forma correctamente.
Resultado Obtenido	La prueba fue aprobatoria, los datos fueron almacenados y visualizados satisfactoriamente.

TABLA 14 PRUEBA UNITARIA DE REPORTES POR CICLOS

Descripción	Crear reporte por ciclos
Objetivos	Verificar que los datos ingresados sean válidos y almacenados en la base de datos, para generar reportes por ciclos académicos por medio de gráficos estadísticos para las competencias y estándares de competencias respectivamente
Condiciones	Ingresar al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de reportes, selecciona un ciclo y los demás parámetros, presiona el botón siguiente.
Resultado Esperado	Los datos ingresados se muestran de forma correctamente.
Resultado Obtenido	La prueba fue aprobatoria, los datos fueron almacenados y visualizados satisfactoriamente.

TABLA 15 PRUEBA UNITARIA DE REPORTES DE IMPORTAR USUARIO

Prueba #	18
Descripción	Importar usuarios
Objetivos	Verificar que los datos ingresados sean válidos y almacenados en la base de datos
Condiciones	Ingresar al sistema con los permisos de acuerdo al perfil que tenga asignado el usuario, ingresa al módulo de usuarios, selecciona la opción importar csv, perfil estudiante, presiona el botón importar.
Resultado Esperado	Los datos ingresados se muestran de forma correctamente.
Resultado Obtenido	La prueba fue aprobatoria, los datos fueron almacenados y visualizados satisfactoriamente.

4.2. Pruebas con Tinker

Es una prueba unitaria que usa una consola de comandos de preferible selección con los paquetes instalados de composer, donde se puede interactuar con las clases y métodos del sistema.

A continuación, se detalla algunas pruebas para visualizar las demás pueden ser consultadas en anexo 11.

TABLA 16 PRUEBAS DE UNIDAD CON TINKER PARA LA CREACIÓN DE CALENDARIO DE AÑO LECTIVO

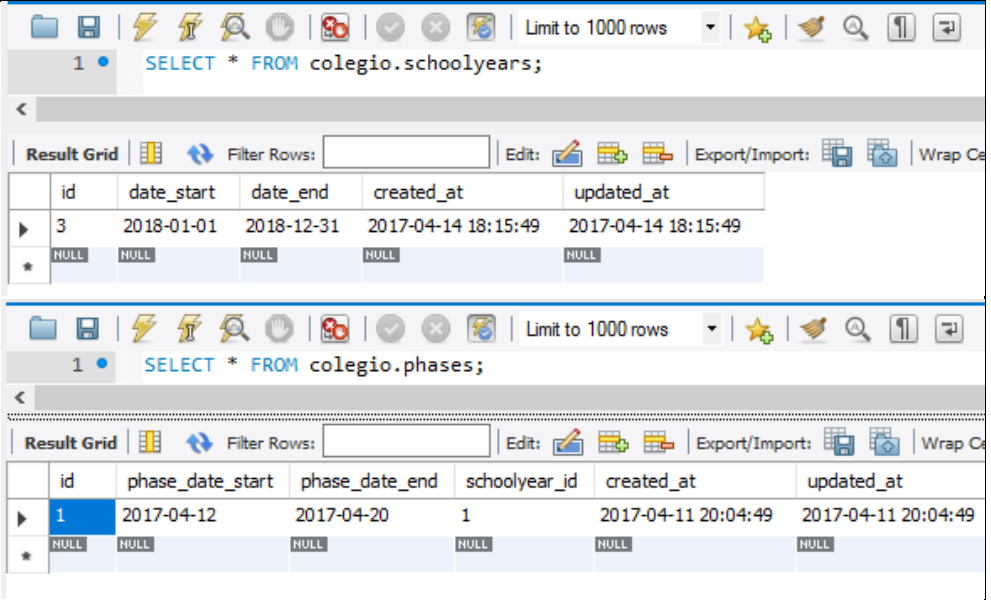
Descripción	Creación año lectivo
Objetivo	Verificar que los datos que se solicitan son almacenados correctamente en la base de datos
Hardware requerido	Un equipo de computo
Software requerido	Consola del sistema operativo, paquete de instalación de Laravel
Datos de prueba	<pre> C:\MAMP\htdocs\colegio [master = +0 ~1 -0 !]> php artisan tinker Psy Shell v0.7.2 (PHP 7.0.9 64-bit) by Justin Hileman >>> \$calendarioLectivo = new App\Schoollyear(['date_start' => '2018-01-01', 'date_end' => '2018-12-31']); => App\Schoollyear {#887 date_start: "2018-01-01 00:00:00", date_end: "2018-12-31 00:00:00", } >>> \$calendarioLectivo->save(); => true >>> \$periodoUno = new App\Phase(['phase_date_start' => '2018-01-01', 'phase_date_end' => '2018-04-10']); => App\Phase {#893 phase_date_start: "2018-01-01 00:00:00", phase_date_end: "2018-04-10 00:00:00", } >>> \$periodoUno->schoollyear()->associate(\$calendarioLectivo); => App\Phase {#893 phase_date_start: "2018-01-01 00:00:00", phase_date_end: "2018-04-10 00:00:00", schoollyear_id: 3, schoollyear: App\Schoollyear {#887 date_start: "2018-01-01 00:00:00", date_end: "2018-12-31 00:00:00", updated_at: "2017-04-14 18:15:49", created_at: "2017-04-14 18:15:49", id: 3, }, } >>> \$periodoUno->save(); => true >>> </pre>
Procedimiento de prueba	se crea un año lectivo con los parámetros necesarios, luego que el sistema confirme el evento, se crea los datos de entrada para crear un periodo académico con sus respectivas relaciones
Resultado esperado	
Resultado obtenido	Prueba exitosa, los datos se almacenan correctamente en la base de datos.

TABLA 17 PRUEBAS DE UNIDAD CON TINKER PARA CREACIÓN DE COMPETENCIAS

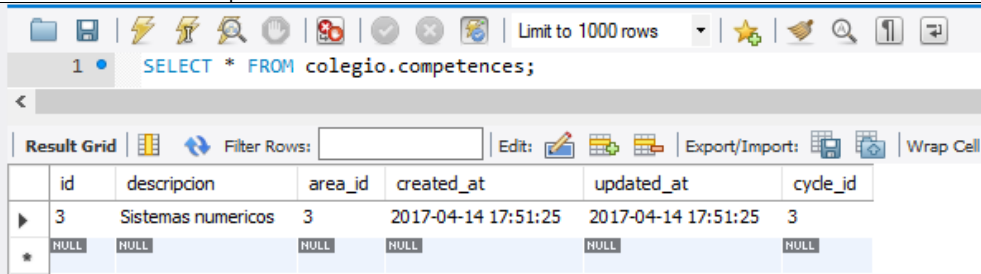
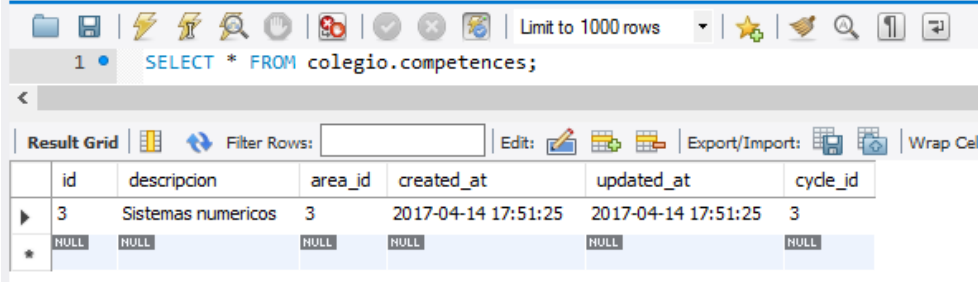
Descripción	Creación de competencias académicas																					
Objetivo	Verificar que los datos que se solicitan son almacenados correctamente en la base de datos																					
Hardware requerido	Un equipo de computo																					
Software requerido	Consola del sistema operativo, paquete de instalación de Laravel																					
Datos de prueba	<pre>posh-git ~ colegio [master] C:\MAMP\htdocs\colegio [master = +0 -1 -0 !] > php artisan tinker Psy Shell v0.7.2 (PHP 7.0.9 0ço cli) by Justin Hileman >>> \$area = new App\Area(['name' => 'matematicas', 'horas_area' => 8]); => App\Area {#887 name: "matematicas", horas_area: 8, } >>> \$area->save(); => true >>> \$ciclo = App\Cycle::find(3); => App\Cycle {#894 id: 3, alias: "5", grado_inicio_id: 10, grado_final_id: 11, created_at: "2017-04-14 17:35:30", updated_at: "2017-04-14 17:35:30", } >>> \$competencia = new App\Competence(['descripcion' => 'Sistemas numericos']); => App\Competence {#892 descripcion: "Sistemas numericos", } >>> \$competencia->cycle()->associate(\$ciclo); => App\Competence {#892 descripcion: "Sistemas numericos", cycle_id: 3, cycle: App\Cycle {#894 id: 3, alias: "5", grado_inicio_id: 10, grado_final_id: 11, created_at: "2017-04-14 17:35:30", updated_at: "2017-04-14 17:35:30", }, } >>> \$competencia->area()->associate(\$area); => App\Competence {#892 descripcion: "Sistemas numericos", cycle_id: 3, area_id: 3, cycle: App\Cycle {#894 id: 3, alias: "5", grado_inicio_id: 10, grado_final_id: 11, created_at: "2017-04-14 17:35:30", updated_at: "2017-04-14 17:35:30", }, area: App\Area {#887 name: "matematicas", horas_area: 8, updated_at: "2017-04-14 17:49:57", created_at: "2017-04-14 17:49:57", id: 3, }, } >>> \$competencia->save(); => true >>></pre>																					
Procedimiento de prueba	Flujo de datos: - Se crea los parámetros necesarios para el área académico y el números horas que va tener dicha área - Se crea los datos de entrada que debe tener un ciclo académico - Se crea la competencia respectiva - Relacionar la competencia con el ciclo académico - Relacionar la competencia con el área académica - Guardar la competencia en la base de datos																					
Resultado esperado	 <pre>SELECT * FROM colegio.competences;</pre> <table><thead><tr><th></th><th>id</th><th>descripcion</th><th>area_id</th><th>created_at</th><th>updated_at</th><th>cycle_id</th></tr></thead><tbody><tr><td>▶</td><td>3</td><td>Sistemas numericos</td><td>3</td><td>2017-04-14 17:51:25</td><td>2017-04-14 17:51:25</td><td>3</td></tr><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></tbody></table>		id	descripcion	area_id	created_at	updated_at	cycle_id	▶	3	Sistemas numericos	3	2017-04-14 17:51:25	2017-04-14 17:51:25	3	*	NULL	NULL	NULL	NULL	NULL	NULL
	id	descripcion	area_id	created_at	updated_at	cycle_id																
▶	3	Sistemas numericos	3	2017-04-14 17:51:25	2017-04-14 17:51:25	3																
*	NULL	NULL	NULL	NULL	NULL	NULL																
Resultado obtenido	Prueba exitosa, los datos se almacenan correctamente en la base de datos.																					

TABLA 18 PRUEBAS DE UNIDAD CON TINKER PARA CREACIÓN DE ESTÁNDARES DE COMPETENCIAS

Descripción	Creación de estándares de competencias académicas
Objetivo	Verificar que los datos que se solicitan son almacenados correctamente en la base de datos
Hardware requerido	Un equipo de computo
Software requerido	Consola del sistema operativo, paquete de instalación de Laravel
Datos de prueba	<pre> A: post-gt-colegio (master) C:\WAMP\htdocs\colegio [master #0 -1 -0 1] > php artisan tinker Psy Shell v0.7.2 (PHP 7.0.9 CLI) by Justin Hileman >>> \$competencia = App\Competence::where('descripcion', '=', 'Sistemas numericos')->first(); => App\Competence {#637 id: 3, descripcion: "Sistemas numericos", area_id: 3, created_at: "2017-04-14 17:51:25", updated_at: "2017-04-14 17:51:25", cycle_id: 3, } >>> \$estandarino = new App\Standard(['descripcion_estandar' => 'Reconozco significados del numero en diferentes contextos (medicion, conteo, comparacion, codificacion, localizacion entre otros).']); => App\Standard {#676 descripcion_estandar: "Reconozco significados del numero en diferentes contextos (medicion, conteo, comparacion, codificacion, localizacion entre otros).", } >>> \$estandarino->competence()->associate(\$competencia); => App\Standard {#676 descripcion_estandar: "Reconozco significados del numero en diferentes contextos (medicion, conteo, comparacion, codificacion, localizacion entre otros).", competence_id: 3, competence: App\Competence {#637 id: 3, descripcion: "Sistemas numericos", area_id: 3, created_at: "2017-04-14 17:51:25", updated_at: "2017-04-14 17:51:25", cycle_id: 3, }, } >>> \$estandarino->save(); => true >>> </pre>
Procedimiento de prueba	Flujo de datos: - Se crea los parámetros necesarios para el área académico y el números horas que va tener dicha área - Se crea los datos de entrada que debe tener un ciclo académico - Se crea la competencia respectiva - Relacionar la competencia con el ciclo académico - Relacionar la competencia con el área académica - Guardar el estándar de competencia en la base de datos
Resultado esperado	
Resultado obtenido	Prueba exitosa, los datos se almacenan correctamente en la base de datos.

Con las pruebas unitarias se mejora la calidad del código, se reducen los tiempos de depuración y la corrección de incidencias y por tanto el cliente estará mucho más contento porque la aplicación hace lo que él quiere que haga. Las pruebas fomentan el cambio y la refactorización.

En estas pruebas se simulan las dependencias lo que permite probar el código sin disponer del resto de módulos.

El objetivo fundamental de las pruebas unitarias es asegurar el correcto funcionamiento de las interfases, o flujo de datos entre componentes de manera tal que a la hora de realizar una unificación de los diferentes componentes que conforman el sistema en general, exista una congruencia que favorezca el desarrollo de la aplicación que se quiere realizar.

De acuerdo a lo anterior, se comprobó el correcto funcionamiento de los módulos del aplicativo mediante objetos mock que imitan el comportamiento de objetos reales de forma controlada, los cuales permitieron asegurar la correcta ejecución por partes individuales, obteniendo las respuestas esperadas del sistema y a su vez tiempos de ejecución, con base a esto, se pudo fomentar el mejoramiento de los algoritmos, adquiriendo mejores tiempos de respuesta en el servidor. Todos los módulos tuvieron que pasar las pruebas unitarias antes de ser liberados o implementados en el servidor.

Por cada una de las pruebas ejecutadas el número de fallos fue con un índice bajo de errores, representando esto como el número de resultados no esperados a partir de los datos de entrada ingresados mediante ejecución de pruebas con la consola de tinker.

Cuando se encuentra un error (“bug”), éste fue corregido inmediatamente, y se tienen precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generaron nuevas pruebas para verificar que el error haya sido resuelto. Este tipo de pruebas no fueron documentadas, dado que fueron realizadas por el desarrollador en el momento de la codificación de tareas de cada iteración.

Al ejecutar de nuevo las pruebas se revelaron un cien por ciento de efectividad; ya que el sistema arrojó lo esperado, es importante aclarar, que éstas no permiten descubrir del todo los errores del código, pero si funcionan como base para el análisis y la optimización del algoritmo.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

En el proceso de entrevista con algunos docentes del colegio María Antonia Ruiz para obtener algunos requerimientos de la aplicación de los usuarios involucrados, es necesario conocer el vocabulario que estos manejan, permitiendo de esa forma entender y evitar ambigüedades en el dialogo que aparten el objetivo al que se quiere lograr.

La recopilación de información sobre los estándares de competencias que maneja el MEN, y la documentación de acuerdo a la metodología que permita tener claro el proceso a seguir en la etapa de desarrollo del sistema, se identificó problemas de cómo abordar toda esta información en un producto intangible donde el factor principal fue la falta de experiencia previa en el desarrollo de estos tipos de sistemas educativos.

El módulo del plan de área fue el eje central del proyecto en cuanto a tiempo invertido en el análisis y desarrollo del mismo. Donde se determinaron aspectos de optimización en cuanto a los quehaceres del sistema. Pensando sobre todo en el usuario docente donde él permitiría realizar la configuración de los estándares básicos de competencias por cada área académica que vaya a dictar por periodo académico. Además, emitiendo una calificación por cada estándar desarrollado.

En el desarrollo del módulo de reportes que permitió mostrar las calificaciones sujetas a los estándares de competencia y de acuerdo a los niveles de desempeño a través del análisis estadístico de gráficos de barras los cuales se generan con los datos obtenidos por cada periodo académico, logro establecer las debilidades que presentan los estudiantes de la institución educativa María Antonia Ruiz , mostrando que en ese colegio hay demasiadas falencias en los procesos de aprendizaje hacia el alumnado.

Al trabajar sobre una problemática real, permitió adquirir un amplio conocimiento tanto en desarrollo de software como en pruebas que es otro campo importante en el área de ingeniería de sistemas, sin embargo, el enfrentarse a la construcción de las aplicaciones bajo parámetros estrictos generó inconvenientes cuando se realizaron pruebas unitarias, se generaron algunos errores que no se había mostrado antes y retardó el desarrollo en múltiples ocasiones. Dando como resultado al final un producto de alta calidad donde se mitigo los errores establecidos.

La estimación del tiempo calculado fue muy distinta al tiempo de ejecución del proyecto, a medida que pasaba de fases para culminar un objetivo se daba cuenta que el tiempo invertido en cumplir ese quehacer no era ni el mínimo del tiempo que se había pronosticado previamente. Esto se debe a la falta de experiencia en el desarrollo de sistemas con gran índice de complejidad, además se debe agregar la falta de experiencia de usar nuevas tecnologías, una cosa es tener la lógica de programación y otra no es tener la experiencia de las aplicabilidades de las herramientas, ya que antes se debe tener un proceso de aprendizaje del uso de ellas.

La utilización de la metodología ICONIX facilitó la adaptabilidad de los procesos en el proyecto, a medida que transcurría el tiempo los requisitos tuvieron cambios, pero gracias a la metodología se pudo adecuar de buena manera el proyecto, permitiendo así entregar un producto más adecuado que cumpliera con la necesidad del trabajo de grado.

El Framework Laravel con otras tecnologías, son herramientas de desarrollo muy primordiales para el servicio de una funcionalidad ya que ayudan a organizar y simplificar de manera rápida las instrucciones y funciones que se generaran al desarrollar en un lenguaje de programación como lo es PHP, porque si se hubiera hecho este proyecto de grado manualmente, es decir sin el uso de un Framework y programar funciones de Javascript puro y maquetear la interfaz con HTML y CSS sin ninguna herramienta de ayuda, el trabajo de grado no se hubiera completado a tiempo, ya que el esfuerzo hubiera sido el doble o el triple ya depende de la experiencia que tenga los desarrolladores.

5.2. Trabajos futuros

Creación de un módulo de notas el cual puede ser accedido por los estudiantes en cualquier momento, estilo campus virtual y no sólo ellos, sino que también puede ser consultado por los padres de familias para que puedan estar más pendiente de las notas de sus hijos a la fecha actual.

Creación de un módulo de boletines donde se especifique la parte cognitiva, social, y personal teniendo en cuenta los criterios de estándares de competencias dadas por el MEN.

Creación de módulo de carnetización estudiantil para que los administradores puedan sacar plantillas de los carnets de los estudiantes e imprimirlo de manera rápida

Creación de un módulo de notificaciones donde los administradores o el perfil que tenga el privilegio puedan crear mensajes con alarmas de un suceso en particular que se va presentar en una fecha establecida en la institución educativa para así estar los docentes estar más informado de ello y no depender tanto de las redes sociales porque se convierte en excusas de que no vieron el mensaje. En cambio, con un módulo que haga este tipo de cosas, es obligación para el docente mirar constantemente que notificaciones hay para ellos.

Capítulo 6

Bibliografía

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

- [1] Ministerio de Educación Nacional (consultado el 29 de abril de 2016). Obtenido de <http://www.mineduacion.gov.co/1621/article-87346.html>
- [2] Ministerio de Educación Nacional (consultado el 21 de abril de 2016). Obtenido de <http://www.mineduacion.gov.co/1759/w3-article-340021.html>
- [3] Ministerio de Educación Nacional (consultado el 21 de abril de 2016). Obtenido de http://www.mineduacion.gov.co/1759/articles-340021_recurso_1.pdf
- [4] W3C “MVC” (consultado el 10 de mayo de 2016). Obtenido de <http://www.w3schools.com/aspnet>
- [5] Laravel(consultado el 16 de febrero de 2016). Obtenido de <https://laravel.com/docs/5.3>
- [6] EcuRed “Base de datos (consultado el 10 de mayo de 2016). Obtenido de http://www.ecured.cu/Bases_de_datos
- [7] MYSQL, “sistemas de Gestión de Base de Datos” (consultado el 29 de abril de 2016). Obtenido de <https://www.mysql.com/>
- [8] Universidad de Barcelona, “Servidor Web.” (consultado el 29 de abril de 2016). Obtenido de <http://tinyurl.com/h4lt0jh>
- [9] MDN Mozilla Developer Network. (consultado 19 de abril de 2016). Obtenido de <https://developer.mozilla.org/es/docs/Web/HTML>
- [10] W3C “Guía breve de css” (10 de mayo de 2016) obtenido de <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- [11] A. J. M. Sierra, *AJAX en J2EE*. alfaomega ra-ma ed., 2008. 9
- [12] PHP (consultado el 10 de mayo de 2016) obtenido de <http://php.net/manual/es/intro-what-is.php>
- [13] AdminLTE(consultado el 16 de febrero de 2016). Obtenido de <https://almsaeedstudio.com/themes/AdminLTE/index2.html>
- [14] Bootstrap (consultado el 16 de febrero de 2016). Obtenido de <http://getbootstrap.com/>
- [15] Select2 (consultado el 16 de febrero de 2016). Obtenido de <https://select2.github.io/>
- [16] DataTables (consultado el 16 de febrero de 2016). Obtenido de <https://datatables.net/>
- [17] IBM “CSV” (consultado el 16 de febrero de 2016). Obtenido de https://www.ibm.com/support/knowledgecenter/es/SS6RBX_11.4.3/com.ibm.sa.help.doc/topics/c_Comm_a_Separated_Value_CSV_Files.html
- [18] inputmask (consultado el 16 de febrero de 2016). Obtenido de <https://github.com/RobinHerbots/Inputmask>

- [19] Iosslider (consultado el 16 de febrero de 2016). Obtenido de <https://iosscripts.com/iosslider>
- [20] Jasny Bootstrap (consultado el 16 de febrero de 2016). Obtenido de <http://www.jasny.net/bootstrap/>
- [21] Bootsnipp(consultado el 16 de febrero de 2016). Obtenido de <http://bootsnipp.com/snippets/featured/form-wizard-using-tabs>
- [22] DatePicker (consultado el 16 de febrero de 2016). Obtenido de <https://jqueryui.com/datepicker/>
- [23] Daterangepicker(consultado el 16 de febrero de 2016). Obtenido de <http://www.daterangepicker.com/>
- [24] ChartJs(consultado el 16 de febrero de 2016). Obtenido de <http://www.chartjs.org/docs/>
- [25] Fontawesome(consultado el 16 de febrero de 2016). Obtenido de <http://fontawesome.io/>
- [26] checked(consultado el 16 de febrero de 2016). Obtenido de <https://api.jquery.com/checked-selector/>
- [27] EcuRed "ICONIX" (consultado el 16 de febrero de 2016). Obtenido de <http://www.ecured.cu/ICONIX>
- [28] Ministerio de Educación Nacional "Estándares básicos de competencias" (consultado el 10 de mayo de 2016) Obtenido de <http://tinyurl.com/jl3ln93>
- [29] Ministerio de Educación Nacional "Plan de estudios" (consultado el 10 de mayo de 2016) Obtenido de <http://www.mineduacion.gov.co/1621/article-79419.html>
- [30] Institución Educativa María de los Ángeles Cano Márquez (consultado el 10 de mayo de 2016) Obtenido de <http://tinyurl.com/h89ry5x>
- [31] Ministerio de Educación (Consultado el 19 de abril de 2016) Obtenido de <http://tinyurl.com/judy7gd>
- [32] COMPUSERVIX. (c19 de 04 de 2016). Obtenido de <http://www.compuservix.com/>
- [33] HAWIS. Tu escuela en línea (consultado el 19 de marzo de 2016). Obtenido de <http://www.tuescuelaenlinea.net/>
- [34] SAS. (Consultado el 19 de abril de 2016). Obtenido de <http://www.rafaelvarela.com/software-academico-sas-notas-boletines.html>
- [35] EDU COLOMBIA. (Consultado el 19 de abril de 2016). Obtenido de sistema integral para la gestión educativa: <http://www.educolombia.org/>
- [36] QUID. (consultado el 19 de abril de 2016). Obtenido de <https://quid.pw/>
- [37] Phidias. (consultado el 19 de abril de 2016). Obtenido de <http://www.phidias.co/>
- [38] WEB ACADEMICA V 2.0. (Consultado el 19 de abril de 2016). Obtenido de <http://web.colpegasus.com/>
- [39] Software, N. Ciudad educativa (consultado el 19 de abril de 2016). Obtenido de <http://www.ciudadeducativa.com/>
- [40] ServiNotas. (consultado el 20 de abril de 2016). Obtenido de <http://www.servinotas.com.co/>
- [41] ALBA. (consultado el 20 de abril de 2016). Obtenido de <http://www.proyectoalba.com.ar/>
- [42] Esemia aplicaciones para la gestión escolar. (consultado el 20 de abril de 2016). Obtenido de <https://esemia.com/es/>
- [43] Gescola. (consultado el 20 de abril de 2016). Obtenido de <https://www.gescola.com/web/es/>

- [44] LCibérica. (consultado el 20 de abril de 2016). Obtenido de <http://www.lciberica.es/>
- [45] IEDUCA. (consultado el 20 de abril de 2016). Obtenido de <http://www.ieduca.com/>
- [46] EDUCCARE gestión académica. (consultado el 20 de abril de 2016). Obtenido de <http://www.educcare.es/>
- [46] Iconixprocessdiagram. (consultado el 20 de abril de 2016). Obtenido de <https://www.simple-talk.com/wp-content/uploads/imported/135-ICONIXProcessDiagram2.jpg>