

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA: INGENIERÍA DE SISTEMAS

**Trabajo de titulación previo a la obtención del título de: INGENIERO DE
SISTEMAS**

TEMA:

**ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN
WEB PARA LA GESTIÓN DE LAS FICHAS DE RESPONSABILIDAD DE LOS
DOCENTES POR CADA TAREA ADMINISTRATIVA ASIGNADA PARA LA
CARRERA DE INGENIERÍA DE SISTEMAS**

AUTOR:

CESAR PATRICIO AYERVE RAMOS

DIRECTOR:

RODRIGO EFRAÍN TUFÍÑO CÁRDENAS

Quito, mayo de 2015

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO
DEL TRABAJO DE TITULACIÓN**

Yo, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaro que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor.

Quito, mayo de 2015

Cesar Patricio Ayerve Ramos

C.C. 171897427-0

DEDICATORIA

Dedico este trabajo de titulación especialmente a la memoria de mi querida madre Mariana de Jesús Ramos Jara, quien ahora goza de la paz eterna de nuestro amado creador, ya que siempre deseó que cumpla todas las metas que me he propuesto a lo largo de mi vida.

A mi padre Cesar Augusto Ayerve Puente, que aparte de ser mi padre es la persona que me ha guiado en este camino y a quien deseo goce de salud para que disfrute con todos quienes le amamos.

A mi segunda madre Dina María Ayerve Puente, quien ha velado por todos en el hogar y que ha sido un ejemplo de fortaleza y humildad.

Cesar Patricio Ayerve Ramos

AGRADECIMIENTO

Un total agradecimiento a la Universidad Politécnica Salesiana, por transmitir el conocimiento necesario que ha sido la herramienta base para el desarrollo del presente trabajo de titulación.

Agradezco a mi tutor Rodrigo Efraín Tufiño Cárdenas, por proveer las pautas necesarias para el correcto desarrollo de la aplicación web, también por ser un gran mentor y una excelente persona.

Cesar Patricio Ayerve Ramos

ÍNDICE

| | |
|--|-----------|
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1 | 2 |
| MARCO REFERENCIAL Y TEÓRICO | 2 |
| 1.1. Planteamiento del problema | 2 |
| 1.2. Diagnóstico de la situación..... | 2 |
| 1.3. Descripción del producto..... | 2 |
| 1.4. Objetivos..... | 3 |
| 1.4.1. Objetivo General..... | 3 |
| 1.4.2. Objetivos Específicos | 4 |
| 1.5. Alcance | 4 |
| 1.6. Marco Teórico | 4 |
| 1.6.1. Fundamentos de la web..... | 4 |
| 1.6.2. HTML5 | 5 |
| 1.6.3. Bootstrap..... | 5 |
| 1.6.4. Codeigniter..... | 5 |
| 1.6.5. Servidor de aplicaciones web Apache | 6 |
| 1.6.6. PostgreSQL..... | 6 |
| 1.6.7. Programación extrema | 7 |
| CAPÍTULO 2 | 10 |
| ANÁLISIS Y DISEÑO | 10 |
| 2.1. Análisis de viabilidad | 10 |
| 2.1.1. Viabilidad técnica | 10 |
| 2.1.2. Viabilidad económica | 11 |
| 2.1.3. Viabilidad operacional..... | 13 |
| 2.2. Análisis de requerimientos | 14 |

| | |
|---|-----------|
| 2.2.1. Alcance | 14 |
| 2.2.2. Identificación de historias de usuario | 14 |
| 2.2.3. Historias de usuario | 16 |
| 2.2.4. Requerimientos del usuario administrador | 17 |
| 2.2.5. Requerimientos del usuario docente | 22 |
| 2.2.6. Requerimientos del usuario dirección de carrera..... | 24 |
| 2.3. Diseño..... | 24 |
| 2.3.1. Diagramas de casos de uso | 25 |
| 2.3.1.1. Diagrama caso de uso administrador | 25 |
| 2.3.1.2. Diagrama caso de uso docente | 26 |
| 2.3.1.3. Diagrama caso de uso dirección de carrera..... | 26 |
| 2.3.2. Diagrama de clases | 27 |
| 2.3.3. Mapas de navegación..... | 29 |
| 2.3.3.1. Mapa de navegación administrador | 29 |
| 2.3.3.2. Mapa de navegación docente | 30 |
| 2.3.3.3. Mapa de navegación dirección de carrera | 30 |
| 2.3.4. Prototipos de interfaces gráficas de usuario | 31 |
| 2.3.4.1. Interfaces para el usuario administrador | 32 |
| 2.3.4.2. Interfaces para el usuario docente | 36 |
| 2.3.4.3. Interfaces para el usuario dirección de carrera..... | 39 |
| 2.5. Diseño conceptual base de datos | 42 |
| 2.6. Atributos del sistema | 44 |
| CAPÍTULO 3..... | 45 |
| CONSTRUCCIÓN Y PRUEBAS..... | 45 |
| 3.1. Herramientas y plataformas..... | 46 |
| 3.1.1 Patrón MVC (modelo, vista, controlador) | 46 |
| 3.1.2. Entorno de desarrollo Netbeans IDE | 47 |

| | |
|---|----|
| 3.1.3. Configuración de Codeigniter 2.2.0..... | 47 |
| 3.1.4. JQuery..... | 50 |
| 3.1.5. Bootstrap 3.3.2..... | 50 |
| 3.1.6. PostgreSQL 9.1..... | 51 |
| 3.1.7. Librerías y helpers de Codeigniter..... | 51 |
| 3.1.8. Librerías externas de Codeigniter..... | 52 |
| 3.1.9. Jmeter..... | 52 |
| 3.1.10. Cliente Filezilla..... | 53 |
| 3.2. Diagrama físico de la base de datos en PostgreSQL..... | 53 |
| 3.3. Diccionario de datos..... | 55 |
| 3.4. Construcción..... | 60 |
| 3.4.1. Arquitectura..... | 60 |
| 3.4.1.1. Cliente web..... | 61 |
| 3.4.1.2. Servidor de aplicaciones..... | 61 |
| 3.4.1.3. Base de datos..... | 62 |
| 3.4.2. Estructura de carpetas y clases..... | 62 |
| 3.4.3. Configuración URL..... | 65 |
| 3.4.4. Clases y métodos..... | 66 |
| 3.4.4.1. Login..... | 66 |
| 3.4.4.2. Index_admin..... | 67 |
| 3.4.4.3. Gestion_tareas..... | 68 |
| 3.4.4.4. Gestion_proyectos..... | 69 |
| 3.4.4.5. Gestion_usuarios..... | 70 |
| 3.4.4.6. Gestion_docentes..... | 71 |
| 3.4.4.7. Gestion_periodos..... | 73 |
| 3.4.4.8. Gestion_tesis..... | 74 |
| 3.4.4.9. Tipos_evidencia..... | 75 |

| | |
|--|------------|
| 3.4.4.10. Index_docente | 76 |
| 3.4.4.11. Tareas_docente..... | 78 |
| 3.4.4.12. Tesis_docente | 78 |
| 3.4.4.13. Index_direccion..... | 79 |
| 3.4.4.14. Cambio_password | 82 |
| 3.4.5. Estructura de las vistas..... | 82 |
| 3.5. Pruebas | 85 |
| 3.5.1. Pruebas de caja negra..... | 85 |
| 3.5.2. Pruebas de rendimiento..... | 90 |
| CAPÍTULO 4..... | 94 |
| IMPLEMENTACIÓN..... | 94 |
| 4.1. Características del servidor..... | 94 |
| 4.2. Configuración de la base de datos | 94 |
| 4.3. Instalación..... | 96 |
| 4.4. Configuración | 97 |
| 4.5. Evidencias de funcionamiento..... | 99 |
| CONCLUSIONES | 101 |
| RECOMENDACIONES | 102 |
| LISTA DE REFERENCIAS | 103 |
| ANEXOS | 104 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Servidor web de aplicaciones..... | 6 |
| Figura 2. Conexión de PostgreSQL..... | 7 |
| Figura 3. Ciclo de vida programación extrema | 8 |
| Figura 4. Diagrama de caso de uso administrador | 25 |
| Figura 5. Diagrama caso de uso docente | 26 |
| Figura 6. Caso de uso dirección de carrera..... | 27 |
| Figura 7. Diagrama de clases..... | 28 |
| Figura 8. Mapa navegacional administrador | 29 |
| Figura 9. Mapa navegacional docentes | 30 |
| Figura 10. Mapa navegacional dirección de carrera..... | 31 |
| Figura 11. Interface de inicio y autenticación | 31 |
| Figura 12. Interface de bienvenida del administrador | 32 |
| Figura 13. Interface opciones menú lateral izquierdo administrador | 33 |
| Figura 14. Interface para agregar y modificar una tarea administrativa..... | 33 |
| Figura 15. Interface para agregar y modificar un proyecto | 34 |
| Figura 16. Interface para agregar y modificar a un usuario | 34 |
| Figura 17. Interface para agregar y modificar un periodo..... | 35 |
| Figura 18. Interface para agregar y modificar un tipo de evidencia..... | 35 |
| Figura 19. Interface para asignar y modificar tareas de los docentes..... | 36 |
| Figura 20. Interface de bienvenida del docente..... | 36 |
| Figura 21. Detalle de tareas y trabajos de titulación del docente | 37 |
| Figura 22. Interface para el detalle de actividades | 37 |
| Figura 23. Interface para subir y descargar evidencias | 38 |
| Figura 24. Interface para agregar actividades..... | 38 |
| Figura 25. Interface de bienvenida de la dirección de carrera..... | 39 |
| Figura 26. Interface de actividades por docente | 40 |
| Figura 27. Interface de actividades por trabajo de titulación | 40 |
| Figura 28. Interface para el perfil de usuario | 41 |
| Figura 29. Interface de cambio de contraseña | 41 |
| Figura 30. Página de créditos | 42 |
| Figura 31. Diseño conceptual base de datos..... | 43 |
| Figura 32. Proceso iterativo del sistema..... | 45 |
| Figura 33. Patrón MVC | 46 |

| | |
|---|-----|
| Figura 34. Localización archivo php.ini | 48 |
| Figura 35. Creación de un proyecto PHP con fuente existente | 48 |
| Figura 36. Instalación Codeigniter en Netbeans..... | 48 |
| Figura 37. Página de bienvenida Codeigniter..... | 49 |
| Figura 38. Asignando dirección del servidor de aplicaciones..... | 49 |
| Figura 39. Estructura de carpetas Bootstrap 3.3.2..... | 51 |
| Figura 40. Diagrama físico de la base de datos | 54 |
| Figura 41. Diagrama de despliegue | 61 |
| Figura 42. Configuraciones archivo config.php | 64 |
| Figura 43. Configuración archivo routes.php..... | 64 |
| Figura 44. Configuración archivo database.php..... | 64 |
| Figura 45. Archivo .htaccess | 65 |
| Figura 46. Configuración archivo httpd.conf | 66 |
| Figura 47. Configuración proxy http | 91 |
| Figura 48. Proxy en Mozilla Firefox | 91 |
| Figura 49. Resultados en árbol (50 usuarios) | 92 |
| Figura 50. Gráfico de resultados (50 usuarios) | 93 |
| Figura 51. Gráfico de..... | 93 |
| Figura 52. Creación de la base de datos | 94 |
| Figura 54. Ubicación archivo de restauración | 95 |
| Figura 53. Restauración base de datos | 95 |
| Figura 55. Base de datos restaurada | 95 |
| Figura 56. Autenticación Filezilla | 96 |
| Figura 57. Depositando archivos en el servidor | 97 |
| Figura 58. Instalación de extensiones para PHP | 97 |
| Figura 60. Página de autenticación..... | 98 |
| Figura 59. Reinicio del servidor | 98 |
| Figura 61. Dashboard administración..... | 99 |
| Figura 62. Gestión de proyectos | 99 |
| Figura 63. Gestión de docentes | 100 |
| Figura 64. Reporte de tareas administrativas asignadas..... | 100 |
| Figura 65. Dashboard docentes | 100 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Requisitos de hardware | 10 |
| Tabla 2. Requisitos de software..... | 11 |
| Tabla 3. Recursos de software..... | 12 |
| Tabla 4. Recursos de hardware..... | 12 |
| Tabla 5. Recursos humanos..... | 13 |
| Tabla 6. Costos de recursos..... | 13 |
| Tabla 7. Historia de usuario control acceso a usuarios | 17 |
| Tabla 8. Historia de usuario cambio de contraseña y perfil | 18 |
| Tabla 9. Historia de usuario registro usuarios al sistema..... | 18 |
| Tabla 10. Historia de usuario creación de tareas administrativas | 19 |
| Tabla 11. Historia de usuario asignación de tareas administrativas a docentes | 19 |
| Tabla 12. Historia de usuario asignación de tareas administrativas a docentes | 20 |
| Tabla 13. Historia de usuario asignación de proyectos a docentes | 20 |
| Tabla 14. Historia de usuario registro de evidencias..... | 21 |
| Tabla 15. Historia de usuario creación de periodos | 21 |
| Tabla 16. Asignación de trabajos de titulación | 22 |
| Tabla 17. Historia de usuario registro de actividades..... | 23 |
| Tabla 18. Historia de usuario evidencias..... | 23 |
| Tabla 19. Historia de usuario reporte de tareas administrativas..... | 24 |
| Tabla 20. Descripción de las tablas de la base de datos | 55 |
| Tabla 21. Diccionario de datos tabla Actividad | 55 |
| Tabla 22. Diccionario de datos tabla Docente..... | 56 |
| Tabla 23. Diccionario de datos tabla Evidencia | 56 |
| Tabla 24. Diccionario de datos tabla Periodo..... | 57 |
| Tabla 25. Diccionario de datos tabla Proyecto..... | 57 |
| Tabla 26. Diccionario de datos tabla Rol | 57 |
| Tabla 27. Diccionario de datos tabla Tarea_Administativa | 58 |
| Tabla 28. Diccionario de datos tabla Tarea_Administrativa_Docente..... | 58 |
| Tabla 29. Diccionario de datos tabla Tesis..... | 59 |
| Tabla 30. Diccionario de datos tabla Tipo_Evidencia..... | 59 |
| Tabla 31. Diccionario de datos tabla Usuario | 59 |
| Tabla 32. Diccionario de datos tabla Usuario_Rol..... | 60 |
| Tabla 33. Modelo login_model | 67 |

| | |
|---|----|
| Tabla 34. Controlador login | 67 |
| Tabla 35. Modelo admin_model..... | 67 |
| Tabla 36. Controlador index_admin..... | 68 |
| Tabla 37. Modelo gestion_tareas..... | 68 |
| Tabla 38. Controlador gestion_tareas..... | 69 |
| Tabla 39. Modelo gproyectos_model..... | 69 |
| Tabla 40. Controlador gestion_proyectos..... | 70 |
| Tabla 41. Modelo gusuarios_model | 71 |
| Tabla 42. Controlador gestion_usuarios..... | 71 |
| Tabla 43. Modelo gdocentes_model..... | 72 |
| Tabla 44. Controlador gestion_docentes | 72 |
| Tabla 45. Modelo gperiodos_model..... | 73 |
| Tabla 46. Controlador gestion_periodos | 74 |
| Tabla 47. Modelo gtesis_model | 74 |
| Tabla 48. Controlador gestion_tesis | 75 |
| Tabla 49. Modelo tevidencia_model | 75 |
| Tabla 50. Controlador tipos_evidencia..... | 76 |
| Tabla 51. Modelo docente_model | 76 |
| Tabla 52. Controlador index_docente | 77 |
| Tabla 53. Controlador tareas_docente | 78 |
| Tabla 54. Controlador tesis_docente | 79 |
| Tabla 55. Modelo direccion_model..... | 80 |
| Tabla 56. Controlador index_direccion..... | 81 |
| Tabla 57. Modelo gpass_model..... | 82 |
| Tabla 58. Controlador cambio_password..... | 82 |
| Tabla 59. Vistas autenticación..... | 83 |
| Tabla 60. Vistas administración | 83 |
| Tabla 61. Vistas docentes | 84 |
| Tabla 62. Vistas dirección de carrera | 84 |
| Tabla 63. Pruebas de caja negra administrador | 85 |
| Tabla 64. Pruebas de caja negra docente..... | 88 |
| Tabla 65. Pruebas de caja negra Dirección de Carrera..... | 89 |

ÍNDICE DE ANEXOS

| | |
|--|-----|
| Anexo 1. Manual administrador | 104 |
| Anexo 2. Manual docente | 104 |
| Anexo 3. Manual dirección de carrera | 104 |

RESUMEN

La aplicación web ha sido desarrollada para la gestión de las fichas de responsabilidad para la Universidad Politécnica Salesiana Sede Quito Campus Sur, esta brinda la posibilidad de administrar tareas administrativas y trabajos de titulación para posteriormente ser asignados a los docentes, estos deberán evidenciar y registrar cada actividad que realicen por medio del sistema y generar su ficha de responsabilidad, además la Dirección de Carrera podrá obtener reportes que evidencien el cumplimiento de las actividades de cada docente y a su vez determinar la carga horaria que tendrá cada uno de ellos, el administrador será el responsable de la correcta gestión y asignación de cada parámetro del sistema, así como su mantenimiento y limpieza periódica.

El capítulo uno presenta el marco referencial y teórico que define los objetivos del trabajo de titulación, los conceptos necesarios y el uso de la metodología.

El capítulo dos presenta el análisis y diseño, se centra en la viabilidad del desarrollo de la aplicación web y en los requerimientos del cliente, cada requerimiento será registrado en historias de usuario que posteriormente se traducirán en el diseño de las interfaces gráficas.

El capítulo tres presenta la construcción y pruebas del desarrollo y las herramientas utilizadas para realizar la construcción de la aplicación web, finalmente se realiza pruebas para comprobar el funcionamiento adecuado del sistema y el cumplimiento de los requerimientos.

El capítulo cuatro presenta la implementación del proyecto finalizado, este es llevado hasta el servidor en el que quedará listo para el uso establecido.

ABSTRACT

The web application has been developed for managing tabs responsibility for Universidad Politécnica Salesiana Sede Quito Campus Sur, this provides the ability to manage administrative tasks and work qualification and then are assigned to teachers, they must demonstrate and record each activity performed by the system and generate their record of responsibility, besides the Career Management can get reports that demonstrate compliance with the activities of each faculty and in turn determine the number of hours they have each of them, the administrator will be the responsible for the proper management and allocation of each parameter of the system and its regular maintenance and cleaning.

Chapter one presents the referential and theoretical framework that defines the aims of this qualification, the necessary concepts and use of the methodology.

Chapter two presents the analysis and design, focuses on the feasibility of developing a web application and customer requirements, each request will be registered user stories that will be translated into the design of graphical interfaces.

Chapter three presents the construction and testing of development and the tools used for building the web application, finally testing is performed to verify proper operation of the system and compliance with the requirements.

Chapter four presents the implementation of the finished project, this is taken to the server where it will be ready for use established.

INTRODUCCIÓN

La Dirección de Carrera de la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Campus Sur Sede Quito, desea obtener reportes de las actividades realizadas por cada docente dentro de un periodo, también requiere de la correcta asignación de tareas administrativas para equilibrar la carga horaria de los docentes.

La aplicación web se desarrollará bajo la metodología de programación extrema (XP), ya que garantiza el control de cambios de los requerimientos del cliente, también permite el desarrollo ágil sin perder mucho tiempo en la documentación ya que se entrega prototipos funcionales al cliente y los cambios se realizan sobre la marcha.

Para ello se analizó, desarrolló e implemento una aplicación web, orientada a la gestión de las fichas de responsabilidad, el administrador será el responsable de la gestión de tareas administrativas, proyectos, usuarios, periodos, trabajos de titulación, tipos de evidencia y asignación de responsabilidades a los docentes, quienes será los que registren y evidencien sus actividades dentro del sistema.

Finalmente la aplicación se desarrollará utilizando la tecnología PHP con el framework Codeigniter, para una rápida implementación el sistema operativo instalado en el entorno de desarrollo será Centos 6.6 y contará con todas las configuraciones necesarias para el correcto funcionamiento del sistema.

CAPÍTULO 1

MARCO REFERENCIAL Y TEÓRICO

1.1. Planteamiento del problema

Los docentes de la Carrera de Ingeniería de Sistemas no poseen un control estandarizado de sus actividades administrativas, que proporcione información medible y cuantificable a la Dirección de Carrera para establecer en cada periodo la carga horaria de los docentes en función a su rendimiento.

1.2. Diagnóstico de la situación

En la actualidad la Carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana se encuentra en un proceso de mejoramiento continuo de excelencia tanto en procesos académicos y administrativos. Por lo tanto la Dirección de Carrera necesita un control de las actividades administrativas que realizan los docentes, ya que es de vital importancia en cada periodo asignar horas administrativas basándose en su rendimiento, se ha establecido un formato para generar la ficha de responsabilidad pero los docentes no lo utilizan de manera adecuada, lo cual impide cuantificar y detallar el número de tareas que han sido cumplidas satisfactoriamente.

Este producto será una herramienta informática para estandarizar el seguimiento de las fichas de responsabilidad, que facilite el análisis del cumplimiento de las tareas administrativas y permita la correcta asignación de las mismas de acuerdo al perfil de cada docente.

1.3. Descripción del producto

El objetivo del sistema es ayudar a la Dirección de Carrera de Ingeniería de Sistemas en la cuantificación y gestión del cumplimiento de las actividades administrativas, registradas por los docentes en su respectiva ficha de responsabilidad. La aplicación web permitirá analizar la distribución de la carga horaria que debe ser asignada a los docentes en cada periodo, así como permitirá la obtención de información útil y detallada.

El sistema será desarrollado en el lenguaje de programación PHP con el framework Codeigniter, el motor de base de datos a utilizar será PostgreSQL.

La aplicación estará estructurada de la siguiente manera:

- Módulo de autenticación

Permitirá el ingreso de los usuarios a la aplicación web, se encargara de verificar el rol correspondiente para cada usuario y contara con la seguridad necesaria para la integridad de la información.

- Módulo de administración

Permitirá la creación de tareas administrativas, proyectos, periodos y tipos de evidencia, también facilitará la gestión de usuarios y docentes, asignado las respectivas tareas administrativas con la distribución de sus horas, finalmente la gestión de trabajos de titulación asignando al docente responsable.

- Módulo de registro

El docente será quien registre sus actividades, la aplicación tendrá un ambiente amigable que proporcionará las herramientas necesarias para ingresar correctamente la información, con sus evidencias que avalen el cumplimiento de su tarea asignada pudiendo ser este un PDF o un documento que lo respalde.

- Módulo de reportes

La finalidad de este módulo es el despliegue de reportes, que serán de utilidad a la Dirección de Carrera para medir el desempeño de cada docente, asignando su correspondiente carga horaria y podrán ser exportados en formato PDF o Excel.

1.4. Objetivos

1.4.1. Objetivo General

Analizar, desarrollar e implementar una aplicación web para la gestión de las fichas de responsabilidad de los docentes por cada tarea administrativa asignada, para la Carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Campus Sur Sede Quito.

1.4.2. Objetivos Específicos

1. Diagnosticar la situación actual de las fichas de responsabilidad y diseñar un proceso que permita registrar las actividades administrativas de los docentes de la Carrera de Ingeniería de Sistemas.
2. Diseñar módulos e interfaces de fácil manejo para los docentes con ayuda de la metodología de programación extrema.
3. Desarrollar la aplicación web utilizando el lenguaje de programación PHP con el framework Codeigniter, basado en el patrón modelo, vista, controlador.
4. Realizar pruebas de rendimiento, rapidez y eficacia del sistema al momento de generar reportes y conectarse con la base de datos.

1.5. Alcance

La aplicación web generará reportes que ayudarán a la Dirección de Carrera en la toma de decisiones, la correcta distribución de la carga horaria de cada docente, evidenciará el desempeño de los mismos en cada periodo y se logrará medir el grado de cumplimiento de las tareas administrativas.

1.6. Marco Teórico

El marco teórico proporciona al lector una idea macro de los elementos, metodología de desarrollo, hardware y software que se utilizará en el desarrollo de la solución tecnológica para la gestión de las fichas de responsabilidad.

1.6.1. Fundamentos de la web

Los conceptos de Internet y web a menudo se confunden y se tiende a creer que son lo mismo, pero en realidad son diferentes. Internet engloba a las tecnologías que hacen posible que los ordenadores ubicados en diferentes lugares del mundo se conecten entre sí y puedan compartir información. Esta conexión es posible mediante cables, módems, líneas telefónicas, routers, protocolos, etc. A todo ese conjunto de tecnologías se le llama Internet. La web es solo uno de los servicios que proporciona Internet.

La web consiste en una forma de organizar la información usando como medio físico de comunicación la red internet y el protocolo HTTP. HTTP (Hypertext Transference

Protocol) es el protocolo de transferencia de hipertexto que los navegadores utilizan para realizar peticiones a los servidores web para recibir las respuestas de ellos. Es el protocolo que se utiliza para ver páginas web (Ramos Martín & Ramos Martín, 2014, pág. 3).

1.6.2. HTML5

HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

HTML5 propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. A partir de ahora, HTML5 provee los elementos estructurales, CSS se encuentra concentrado en cómo volver esa estructura utilizable y atractiva a la vista, y JavaScript tiene todo el poder necesario para proveer dinamismo y construir aplicaciones web completamente funcionales (Gauchat, 2012, pág. Introducción).

1.6.3. Bootstrap

En la actualidad muchos sitios web están elaborando su diseño en HTML5, Bootstrap es el framework de Twitter que permite crear interfaces web con CSS y Javascript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo (Jacob & Otto, 2014, pág. Capítulo 1).

1.6.4. Codeigniter

Codeigniter es un framework PHP para la creación rápida de aplicaciones web, un framework es un programa para desarrollar otros programas, Codeigniter, por tanto; es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web. Es un producto de código libre para cualquier aplicación, cuenta con un amplio apoyo de los desarrolladores y su respectiva documentación.

Como cualquier otro framework, Codeigniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que se debe seguir para obtener provecho de la aplicación.

Codeigniter implementa el proceso de desarrollo modelo, vista, controlador (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales (Foster, 2013, págs. 5-7).

1.6.5. Servidor de aplicaciones web Apache

Apache es el servidor web más utilizado, líder con el mayor número de instalaciones a nivel mundial muy por delante de otras soluciones como el IIS (Internet Information Server) de Microsoft. Apache es un proyecto de código abierto y uso gratuito, multiplataforma, muy robusto y que destaca por su seguridad y rendimiento.

Su misión es crítica, ya que es el encargado de aceptar las peticiones de páginas (o recursos en general) que provienen de los visitantes que acceden a un sitio web y gestionar su entrega o denegación, de acuerdo a las políticas de seguridad establecidas.

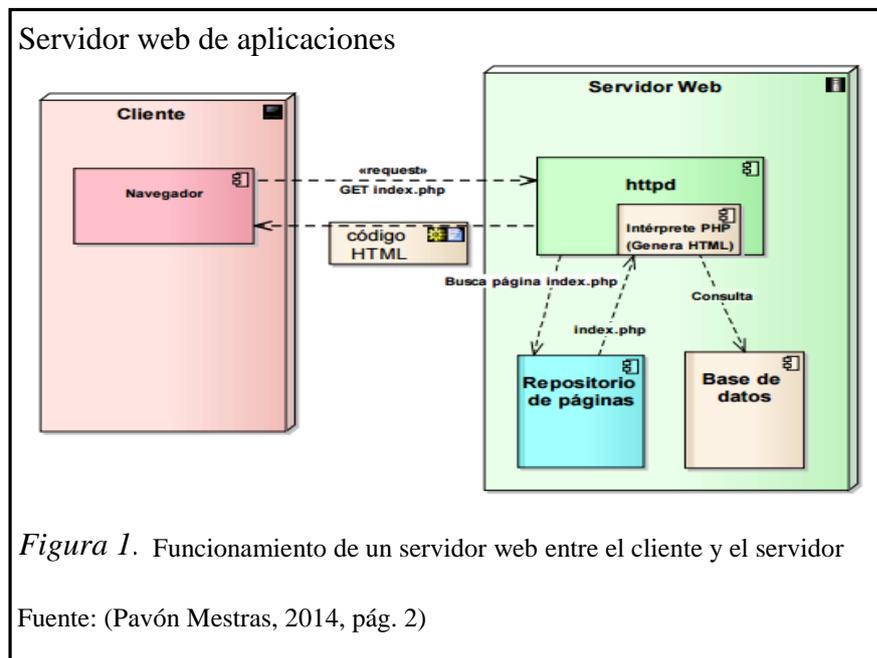


Figura 1. Funcionamiento de un servidor web entre el cliente y el servidor

Fuente: (Pavón Mestras, 2014, pág. 2)

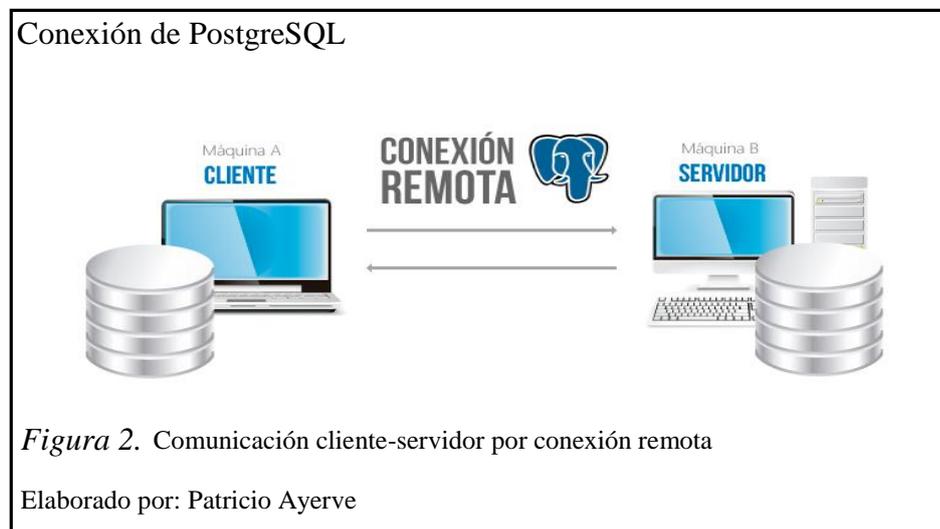
1.6.6. PostgreSQL

PostgreSQL, es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es un sistema de gestión de

bases de datos de código abierto y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en lugar de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Sus características técnicas la hacen una de las bases de datos más potentes por su estabilidad, robustez y facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema (Obe & Hsu, 2012, pág. Preface).



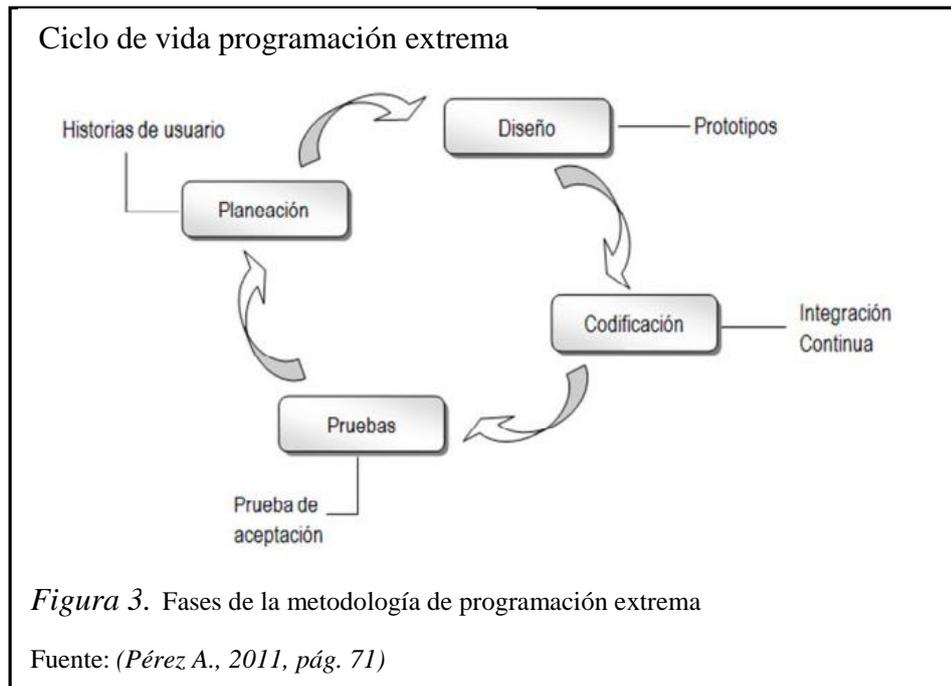
1.6.7. Programación extrema

La programación extrema o extreme programming (XP), es una disciplina de desarrollo de software basada en los métodos ágiles, que evidencia principios tales como el desarrollo incremental, la participación activa del cliente, el interés en las personas y no en los procesos como elemento principal, aceptar el cambio y la simplicidad.

Los principios de XP comprenden diez buenas prácticas que involucran al equipo de trabajo, los procesos y el cliente; los cuales son:

- Planificación incremental: se toman los requerimientos en historias de usuario, las cuales son negociadas progresivamente con el cliente.

- Entregas pequeñas: se desarrolla primero la más mínima parte útil que le proporcione funcionalidad al sistema, y poco a poco se efectúan incrementos que añaden funcionalidad a la primera entrega, cada ciclo termina con una entrega del sistema, el ciclo de vida termina cuando ya no hay más ciclos de entrega y el sistema ha cumplido con el objetivo para el cual fue diseñado, de no ser así, se deberá continuar con el ciclo especificado hasta que la funcionalidad del sistema sea la deseada.



- Diseño sencillo: solo se efectúa el diseño necesario para cumplir con los requerimientos actuales, es decir, no se abordan requerimientos futuros.
- Desarrollo previamente aprobado: una de las características relevantes y propias de XP es que primero se escriben las pruebas y luego se da la codificación, esto con la finalidad de asegurar la satisfacción del requerimiento.
- Limpieza del código o refactorización: consiste en simplificar y optimizar el programa sin perder funcionalidad, es decir, alterar su estructura interna sin afectar su comportamiento externo.
- Programación en parejas: es otra de las características de ésta metodología, que propone que los desarrolladores trabajen en parejas en una terminal, verificando cada uno el trabajo del otro y ayudándose para buscar las mejores soluciones. Se entiende que de esta forma el trabajo será más eficiente y de mayor calidad.

- Propiedad colectiva: el conocimiento y la información deben ser de todos, por lo tanto no se desarrollan islas de conocimiento, todos los programadores poseen todo el código y cualquiera puede sugerir y realizar mejoras.
- Integración continua: al terminar una tarea, ésta se integra al sistema entero y se realizan pruebas de unidad a todo el sistema, ésta práctica permite que la aplicación sea más funcional en cada iteración y garantiza su funcionamiento con los demás módulos del sistema.
- Ritmo sostenible: no es aceptable trabajar durante grandes cantidades de horas ya que se considera que puede reducir la calidad del código y la productividad del equipo a mediano plazo, se sugieren 40 horas semanales.
- Cliente presente: se debe tener un representante (cliente o usuario final) tiempo completo, ya que en XP éste hace parte del equipo de desarrollo y es responsable de formular los requerimientos para el desarrollo del sistema (Pérez A., 2011, págs. 71-73).

CAPÍTULO 2

ANÁLISIS Y DISEÑO

Este capítulo detalla las diferentes fases de análisis y diseño del producto, comenzando por el análisis de viabilidad, los requerimientos necesarios para el desarrollo y diseño del software, también el diagrama entidad-relación de la base de datos y la construcción de la interfaz gráfica en conjunto con la metodología XP.

2.1. Análisis de viabilidad

Tres aspectos son fundamentales para el análisis de viabilidad de un proyecto, estos son: viabilidad técnica, económica y operacional, las cuales se detallan a continuación.

2.1.1. Viabilidad técnica

El proyecto será desarrollado utilizando herramientas de software libre, la plataforma informática sobre la que funcionará son: GNU/Linux, Microsoft Windows y Mac OSX.

El hardware y software para el desarrollo del producto tiene las siguientes características:

Tabla 1.
Requisitos de hardware

| Requisitos de hardware | | | |
|--|----------|-------------------|---|
| Dispositivo | Cantidad | Uso | Descripción |
| Computador portátil Dell Inspiron 14 5447 | 1 | Desarrollo | -Procesador: Intel Core i7 -Memoria: 8 GB -Disco: 1 TB -Arquitectura: 64 bits |
| Router inalámbrico | 1 | Red de desarrollo | -Velocidad: 54 Mbps -protocolo: 802.11g -Puertos: 4 Ethernet -Conectividad: Wireless |

Nota. Requisitos de hardware para el desarrollo de la aplicación web

Elaborado por: Patricio Ayerve

Tabla 2.

Requisitos de software

| Requisitos de software | | | |
|------------------------|----------|-----------------------------------|---|
| Producto | Cantidad | Uso | Descripción |
| CentOS | 1 | Servidor de red, aplicación y BDD | -Licencia: GPL -Versión: CentOS release 6.6 (Final) 64 bits |
| Apache Server | 1 | Servidor de aplicaciones web | -Licencia: GPL -Versión: 2.2.15 |
| PHP | 1 | Lenguaje de programación | -Licencia: GPL -Versión: 5.3.3 |
| Codeigniter | 1 | Framework de desarrollo PHP | -Licencia: GPL -Versión: 2.2.0 |
| PostgreSQL | 1 | Software base de datos | -Licencia: GPL -Versión: 9.1 |
| Netbeans IDE | 1 | IDE de desarrollo | -Licencia: GPL -Versión: 8.0.2 -Módulos: Php, Css3, Html5, JavaScript |
| Google Chrome | 1 | Explorador web | -Licencia: Free -Versión: 41.0.2272.101 m |

Nota. Requisitos de software para el desarrollo de la aplicación web.
Elaborado por: Patricio Ayerve

Tras evaluar diferentes alternativas de lenguajes de programación y plataformas, la aplicación se desarrollará bajo el lenguaje de programación PHP con el framework Codeigniter, dada la sencillez que provee este lenguaje para el trabajo y la gestión de bases de datos, núcleo central de la aplicación.

2.1.2. Viabilidad económica

La inversión requerida para el desarrollo del software está directamente vinculada al pago del salario de un ingeniero de desarrollo. Se estableció un salario mensual de \$900 dólares americanos por el trabajo de ocho horas diarias.

Para el costo real del producto se ha definido diferentes aspectos económicos, los cuales incluyen licenciamiento, costo de personal y gastos varios en general.

Tabla 3.

Recursos de software

| Recursos de software | | | | |
|---------------------------------|-------------------------|----------|----------------|-------------|
| Descripción del recurso | Descripción de licencia | Cantidad | Valor Unitario | Valor Total |
| Sistema Operativo CentOS 6.5 | Licencia GNU | 1 | \$ 0 | \$ 0 |
| Servidor de Aplicaciones Apache | Licencia GNU | 1 | \$ 0 | \$ 0 |
| Lenguaje de programación | Licencia GNU | 1 | \$ 0 | \$ 0 |
| IDE de desarrollo | Licencia GNU | 1 | \$ 0 | \$ 0 |
| Total | | | | \$ 0 |

Nota. Costos por la adquisición de software para el desarrollo de la aplicación web.

Elaborado por: Patricio Ayerve

Tabla 4.

Recursos de hardware

| Recursos de hardware | | | | |
|----------------------------|--|----------|----------------|-------------|
| Descripción del recurso | Descripción del costo | Cantidad | Valor unitario | Valor total |
| Computador portátil | Precio portátil | 1 | \$ 700 | \$ 700 |
| Infraestructura de red LAN | Switches, routers | 1 | \$ 120 | \$ 120 |
| Servidor web | Se cuenta con un servidor implementado | 1 | \$ 0 | \$ 0 |
| Internet | Costo mensual del servicio (6 meses) | 2 | \$ 35 | \$ 210 |
| Total | | | | \$ 1.030 |

Nota. Costos por la adquisición de hardware para el desarrollo de la aplicación web.

Elaborado por: Patricio Ayerve

Tabla 5.

Recursos humanos

| Recursos de humanos | | | | |
|---------------------|---|----------|----------------|-------------|
| Recurso | Descripción del costo | Cantidad | Valor unitario | Valor total |
| Programador | Salario mensual del programador (6 meses) | 1 | \$ 900 | \$ 5400 |
| Servicios básicos | Precio (6 meses) | 3 | \$ 420 | \$ 420 |
| Gastos operativos | Impresiones, movilización, etc. | 1 | \$ 600 | \$600 |
| Total | | | | \$ 6.420 |

Nota. Costos de recursos humanos para el desarrollo de la aplicación web.

Elaborado por: Patricio Ayerve

Tabla 6.

Costos de recursos

| Referencia | Valor |
|-----------------------------|----------|
| Costos de Software | \$ 0 |
| Costos de Hardware | \$ 1.030 |
| Costos por Recursos humanos | \$ 6.420 |
| Total | \$ 7.450 |

Nota. Costos totales para el desarrollo de la aplicación web.

Elaborado por: Patricio Ayerve

Los costos citados anteriormente serán cubiertos por: Cesar Patricio Ayerve Ramos

2.1.3. Viabilidad operacional

Actualmente la carrera de Ingeniería de Sistemas no tiene un sistema para el control y gestión de las fichas de responsabilidad, el desarrollo será implementado debido a su bajo consumo de recursos económicos, escalabilidad y adaptabilidad.

Los usuarios no técnicos recibirán un manual de usuario que ayudará a comprender como utilizar el producto.

Al realizar este análisis de viabilidad se puede evidenciar lo siguiente:

- Existen los recursos tecnológicos necesarios para desarrollar el producto y la plataforma seleccionada permitirá que el sistema se implante en los sistemas operativos: Microsoft Windows, GNU/Linux y Mac OSX.

- Desde el punto de vista operacional, se pretende que esta aplicación web sea aceptada por los docentes y facilite el control y gestión de sus tareas administrativas.

Por lo tanto, el proyecto resulta viable para su desarrollo e implementación.

2.2. Análisis de requerimientos

El propósito de este subcapítulo es describir de forma detallada los requerimientos de software.

A continuación se mostrará las características que definen el sistema, los perfiles de usuario, sus interfaces y principalmente se especificará en detalle la funcionalidad y operaciones que debe realizar el sistema.

2.2.1. Alcance

Este sistema será creado para gestionar las tareas administrativas llevadas a cabo por los docentes, con esta información se medirá el desempeño de cada uno de ellos de acuerdo a su nivel de cumplimiento.

El sistema será desarrollado para la Carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Campus Sur Sede Quito.

Específicamente, el sistema permitirá a la Dirección de Carrera obtener reportes en cada periodo, lo cual facilitará gestionar la asignación de las tareas administrativas y trabajos de titulación a los docentes y su respectiva carga horaria en base a su desempeño.

También se gestionara la información del administrador, docente y director de carrera, se asignara un rol específico para cada uno de ellos y otros parámetros generales que son indispensables para el funcionamiento del sistema.

2.2.2. Identificación de historias de usuario

Las historias de usuario, como determina la metodología XP, son un conjunto de tarjetas, que describen las necesidades que el cliente desea satisfacer.

Previo a la elaboración de las historias de usuarios se identificará a cada uno de los actores involucrados y su perfil de acceso.

Para el caso de este desarrollo se han identificado tres perfiles los cuales son:

Administrador: será el encargado de crear usuarios, otorgará permisos a los diversos módulos del sistema, gestionara a los docentes con su respectiva asignación de tareas administrativas, su rol realizará lo siguiente:

- Iniciar sesión
- Registro de tareas administrativas
- Registro de proyectos
- Registro de usuarios al sistema
- Registro de docentes
- Asignación de proyectos
- Asignación de tareas administrativas
- Asignación de trabajos de titulación
- Registro de periodos
- Registro de tipos de evidencia
- Cambio de contraseña
- Modificar su perfil
- Descargar manual de usuario
- Imprimir

Docente: será el encargado de registrar sus actividades dentro de las tareas administrativas asignadas por el usuario Administrador, lo mismo sucederá si tiene un trabajo de titulación, el docente subirá al sistema la evidencia correspondiente que avale el cumplimiento de cada actividad, su rol realizara lo siguiente:

- Iniciar sesión
- Registro de actividades
- Registro de evidencias
- Cambio de contraseña
- Modificar su perfil
- Descargar manual de usuario
- Imprimir

Dirección de Carrera: será quien tendrá acceso a reportes del cumplimiento de las tareas administrativas de cada docente dentro de un periodo, su rol realizara lo siguiente:

- Iniciar sesión
- Consulta de reportes de tareas administrativas en cada periodo
- Consulta de reportes de trabajos de titulación
- Descargar manual de usuario
- Imprimir

Las especificaciones sobre los requerimientos iniciales solicitados por el cliente se basaron en las funciones más esenciales.

2.2.3. Historias de usuario

Las historias de usuario son un conjunto de tarjetas escritas por el cliente en un lenguaje natural y permiten obtener los requerimientos del sistema a implementar.

Las tarjetas de usuario contienen los siguientes elementos:

- Número de historia de usuario: es un identificador único que la distingue de otras historias de usuario.
- Usuario: es el cliente que dará uso al requerimiento descrito en la historia de usuario una vez finalizada.
- Nombre de historia: nombre con el que se identifica a la historia de usuario.
- Prioridad: elemento cualitativo que mide la máxima preferencia y permite analizar lo que es de mayor importancia y requiere más atención. Los valores usados en este ítem son: alta, media y baja.
- Riego en desarrollo: elemento cualitativo que indica el grado de dificultad de la historia de usuario.
- Puntos estimados: tiempo en que los desarrolladores estiman realizar la historia de usuario.
- Iteración asignada: identifica a que número de iteración corresponde la historia de usuario, tomando en cuenta que cada iteración está compuesta por los módulos del sistema a los cuales se les asignará una determinada historia de usuario.

- Programador responsable: desarrollador encargado de programar el código necesario para cumplir con los requerimientos inmersos en la historia de usuario asignada.
- Descripción: rápida explicación de lo que trata la historia de usuario, en palabras sencillas que el cliente pueda entender, para que facilite la comunicación con el desarrollador.
- Observaciones: explicaciones extras de funcionalidades del sistema.

Para la especificación de los requerimientos solicitados se planteó una serie de tablas en las cuales se indica cada requerimiento.

A continuación se detalla las historias de usuario correspondientes a todos los usuarios que ingresaran a la aplicación web.

2.2.4. Requerimientos del usuario administrador

El objetivo es reflejar los requerimientos del usuario que será el encargado de la administración del sistema, dichos requerimientos son registrados en las respectivas historias de usuario, acordadas entre el desarrollador y el cliente.

La tabla 7 refleja los requerimientos del administrador para cubrir las necesidades del proceso de inicio de sesión de cada usuario.

Tabla 7.

Historia de usuario control acceso a usuarios

| Control de acceso de usuarios | |
|---|--------------------------------|
| Número: 1 | Usuario: Todos |
| Nombre historia: Control de acceso de usuarios | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: Al ingresar a la aplicación web se debe solicitar una credencial, para que el usuario tenga acceso a la información de acuerdo a su rol y ejecute sus tareas específicas. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para controlar el acceso de los usuarios a la aplicación web.

Elaborado por: Patricio Ayerve

La tabla 8 refleja los requerimientos del administrador para cubrir las necesidades del proceso de cambio de contraseña de cada usuario.

Tabla 8.

Historia de usuario cambio de contraseña y perfil

| Cambio de contraseña y perfil | |
|---|--------------------------------|
| Número: 2 | Usuario: Todos |
| Nombre historia: Cambio de contraseña y perfil | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 3 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: Todos los usuarios podrán cambiar su contraseña de acceso al sistema y modificar su perfil. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para modificar el perfil y contraseña de acceso.

Elaborado por: Patricio Ayerve

La tabla 9 refleja los requerimientos del administrador para cubrir las necesidades del proceso de registro de cada usuario al sistema.

Tabla 9.

Historia de usuario registro usuarios al sistema

| Registro usuarios al sistema | |
|--|--------------------------------|
| Número: 3 | Usuario: Administrador |
| Nombre historia: Cesar Patricio Ayerve Ramos | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: El administrador será capaz de gestionar a los distintos usuarios del sistema y registrarlos dentro del mismo. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para registrar usuarios a la aplicación web.

Elaborado por: Patricio Ayerve

La tabla 10 refleja los requerimientos del administrador para cubrir las necesidades del proceso de creación de tareas administrativas.

Tabla 10.

Historia de usuario creación de tareas administrativas

| Creación de tareas administrativas | |
|---|--------------------------------|
| Número: 4 | Usuario: Administrador |
| Nombre historia: Creación de tareas administrativas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 3 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: El administrador será capaz de crear tareas administrativas las mismas que serán asignadas posteriormente a un docente. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para la creación de tareas administrativas.

Elaborado por: Patricio Ayerve

La tabla 11 refleja los requerimientos del administrador para cubrir las necesidades del proceso de asignación de tareas administrativas a los docentes.

Tabla 11.

Historia de usuario asignación de tareas administrativas a docentes

| Asignación de tareas administrativas a docentes | |
|---|-------------------------------|
| Número: 5 | Usuario: Administrador |
| Nombre historia: Asignación de tareas administrativas a docentes | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 4 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: El administrador será capaz de asignar tareas administrativas a cada docente, según lo determine la Dirección de Carrera. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para asignar tareas administrativas a los docentes.

Elaborado por: Patricio Ayerve

La tabla 12 refleja los requerimientos del administrador para cubrir las necesidades del proceso de creación de proyectos.

Tabla 12.

Historia de usuario asignación de tareas administrativas a docentes

| Creación de proyectos | |
|--|--------------------------------|
| Número: 6 | Usuario: Administrador |
| Nombre historia: Creación de proyectos | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: El administrador será capaz de crear proyectos los mismos que serán asignados posteriormente a un docente. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para la creación de proyectos para los docentes.

Elaborado por: Patricio Ayerve

La tabla 13 refleja los requerimientos del administrador para cubrir las necesidades del proceso de asignación de proyectos a los docentes.

Tabla 13.

Historia de usuario asignación de proyectos a docentes

| Asignación de proyectos a docentes | |
|--|-------------------------------|
| Número: 7 | Usuario: Administrador |
| Nombre historia: Asignación de proyectos a docentes | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: El administrador será capaz de asignar proyectos a cada docente, según lo determine la Dirección de Carrera. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para asignar proyectos a los docentes.

Elaborado por: Patricio Ayerve

La tabla 14 refleja los requerimientos del administrador para cubrir las necesidades del proceso de registro de evidencias.

Tabla 14.
Historia de usuario registro de evidencias

| Registro de tipo de evidencias | |
|---|--------------------------------|
| Número: 8 | Usuario: Administrador |
| Nombre historia: Registro de tipo de evidencias | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: El administrador debe ser capaz de registrar el tipo de evidencia que podrá asignar cada docente. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para registrar los tipos de archivos para las evidencias a subir en el servidor.

Elaborado por: Patricio Ayerve

La tabla 15 refleja los requerimientos del administrador para cubrir las necesidades del proceso de creación de periodos.

Tabla 15.
Historia de usuario creación de periodos

| Creación de periodos | |
|---|-------------------------------|
| Número: 9 | Usuario: Administrador |
| Nombre historia: Creación de periodos | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: El administrador será capaz de crear periodos académicos y a su vez creara una carpeta que contendrá la información de las evidencias de los docentes pero cada uno de ellos. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para la creación de periodos para la asignación de tareas administrativas.

Elaborado por: Patricio Ayerve

La tabla 16 refleja los requerimientos del administrador para cubrir las necesidades del proceso de asignación de trabajos de titulación a los docentes.

Tabla 16.
Asignación de trabajos de titulación

| Asignación de trabajos de titulación | |
|---|--------------------------------|
| Número: 10 | Usuario: Administrador |
| Nombre historia: Asignación de trabajos de titulación | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: El administrador será capaz de crear y asignar trabajos de titulación a los docentes. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para asignar trabajos de titulación a los docentes.

Elaborado por: Patricio Ayerve

2.2.5. Requerimientos del usuario docente

El objetivo es reflejar los requerimientos del usuario docente, dichos requerimientos son registrados en las respectivas historias de usuario, acordadas entre el desarrollador y el cliente.

La Tabla 17 refleja los requerimientos del docente para cubrir las necesidades del proceso de registro de actividades.

Tabla 17.

Historia de usuario registro de actividades

| Registro de actividades | |
|--|-------------------------------|
| Número: 11 | Usuario: Docente |
| Nombre historia: Registro de actividades dentro de tareas administrativas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 3 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: El docente será el responsable de registrar sus actividades dentro de la tarea administrativa o trabajo de titulación asignado por el administrador, también se deberá incluir un documento que evidencie el cumplimiento de cada actividad. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para el registro de actividades de los docentes.

Elaborado por: Patricio Ayerve

La Tabla 18 refleja los requerimientos del docente para cubrir las necesidades del proceso de registro de evidencias.

Tabla 18.

Historia de usuario evidencias

| Evidencias | |
|---|-------------------------------|
| Número: 12 | Usuario: Docente |
| Nombre historia: Evidencias | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: Por cada actividad administrativa el docente sube al sistema una evidencia. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para el registro de evidencias de los docentes.

Elaborado por: Patricio Ayerve

2.2.6. Requerimientos del usuario dirección de carrera

El objetivo es reflejar los requerimientos del usuario que será el encargado de la Dirección de Carrera, dichos requerimientos son registrados en las respectivas historias de usuario, acordadas entre el desarrollador y el cliente.

La Tabla 19 refleja los requerimientos de la Dirección de Carrera para cubrir las necesidades del proceso de reportes de actividades.

Tabla 19.

Historia de usuario reporte de tareas administrativas

| Reporte de tareas administrativas | |
|---|-------------------------------|
| Número: 13 | Usuario: Dirección de Carrera |
| Nombre historia: Reporte de tareas administrativas | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 2 | Iteración asignada: 1 |
| Programador responsable: Cesar Patricio Ayerve Ramos | |
| Descripción: La Dirección de Carrera necesita obtener reportes para medir el cumplimiento de las tareas administrativas asignadas a cada docente de la carrera de Ingeniería de Sistemas cada periodo académico y también se necesita imprimir estos reportes. | |
| Observaciones: Confirmado con el cliente | |

Nota. Historia de usuario para obtener reportes que permitan medir el cumplimiento de las tareas administrativas y trabajos de titulación de los docentes.

Elaborado por: Patricio Ayerve

2.3. Diseño

En los siguientes apartados se mostrará las funcionalidades más importantes que los usuarios podrán realizar dentro del sistema, a través de los diagramas de casos de uso; también se realiza una representación de la navegación con los respectivos mapas de navegación, luego de procederá diseñando las diferentes interfaces gráficas y finalmente se muestra un diagrama conceptual de la base de datos.

2.3.1. Diagramas de casos de uso

Los siguientes diagramas de caso de uso visualizan los principales módulos que conforman el sistema conjuntamente con los actores que intervienen e interactúan en ellos.

2.3.1.1. Diagrama caso de uso administrador

El actor que interviene aquí es el administrador del sistema y las funcionalidades a las que tiene acceso son relacionadas a la gestión de la información que requiere el sistema.

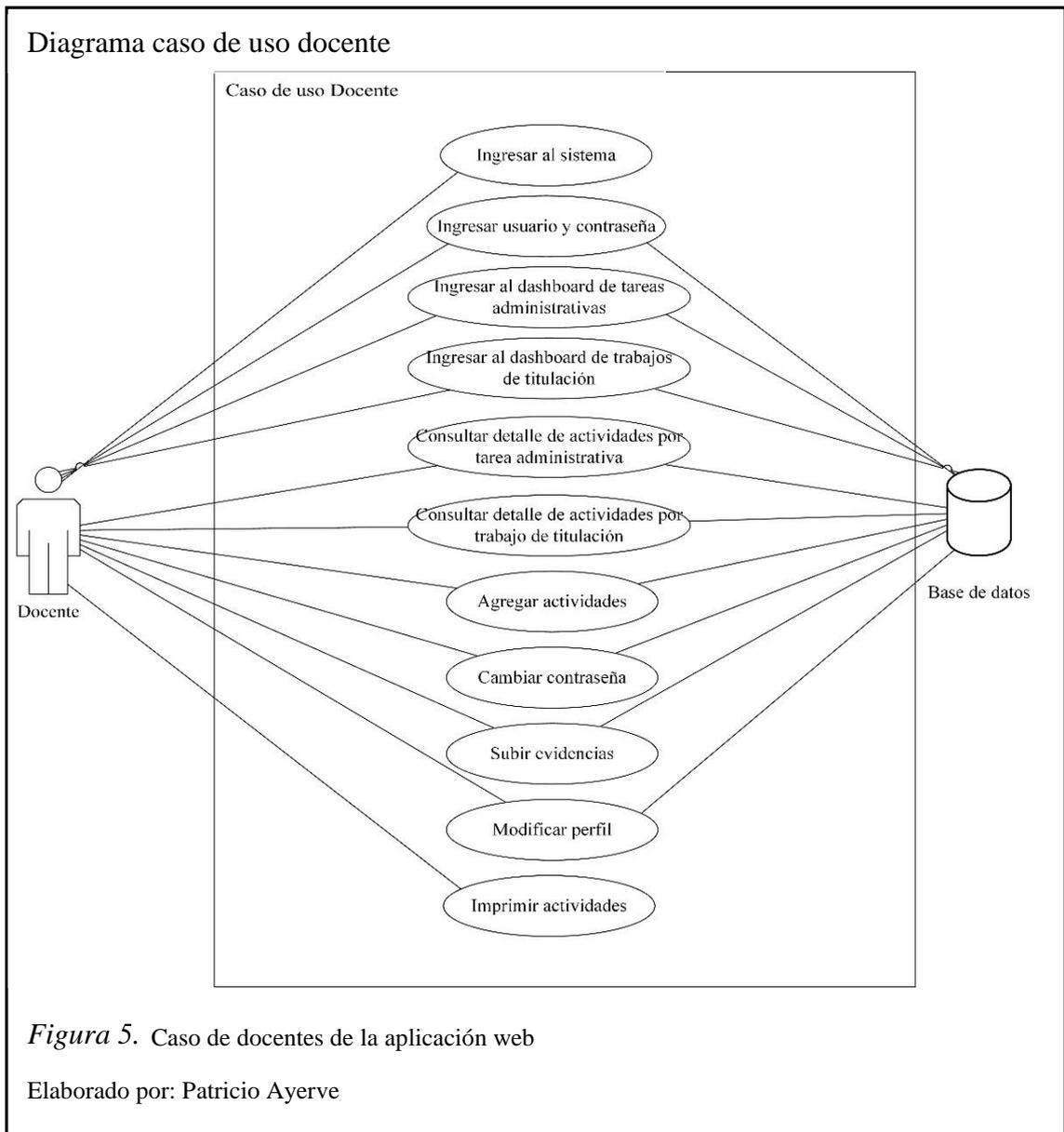


Figura 4. Caso de uso de administración de la aplicación web

Elaborado por: Patricio Ayerve

2.3.1.2. Diagrama caso de uso docente

El actor que interviene aquí es el docente quien será el encargado de las siguientes tareas dentro del sistema a desarrollar.



2.3.1.3. Diagrama caso de uso dirección de carrera

Con este caso de uso se representa las funcionalidades a las que tiene acceso el usuario con el perfil Dirección de Carrera. Básicamente genera reportes que se han realizado durante un período de tiempo establecido para los docentes, tareas administrativas y trabajos de titulación.

Diagrama caso de uso dirección de carrera

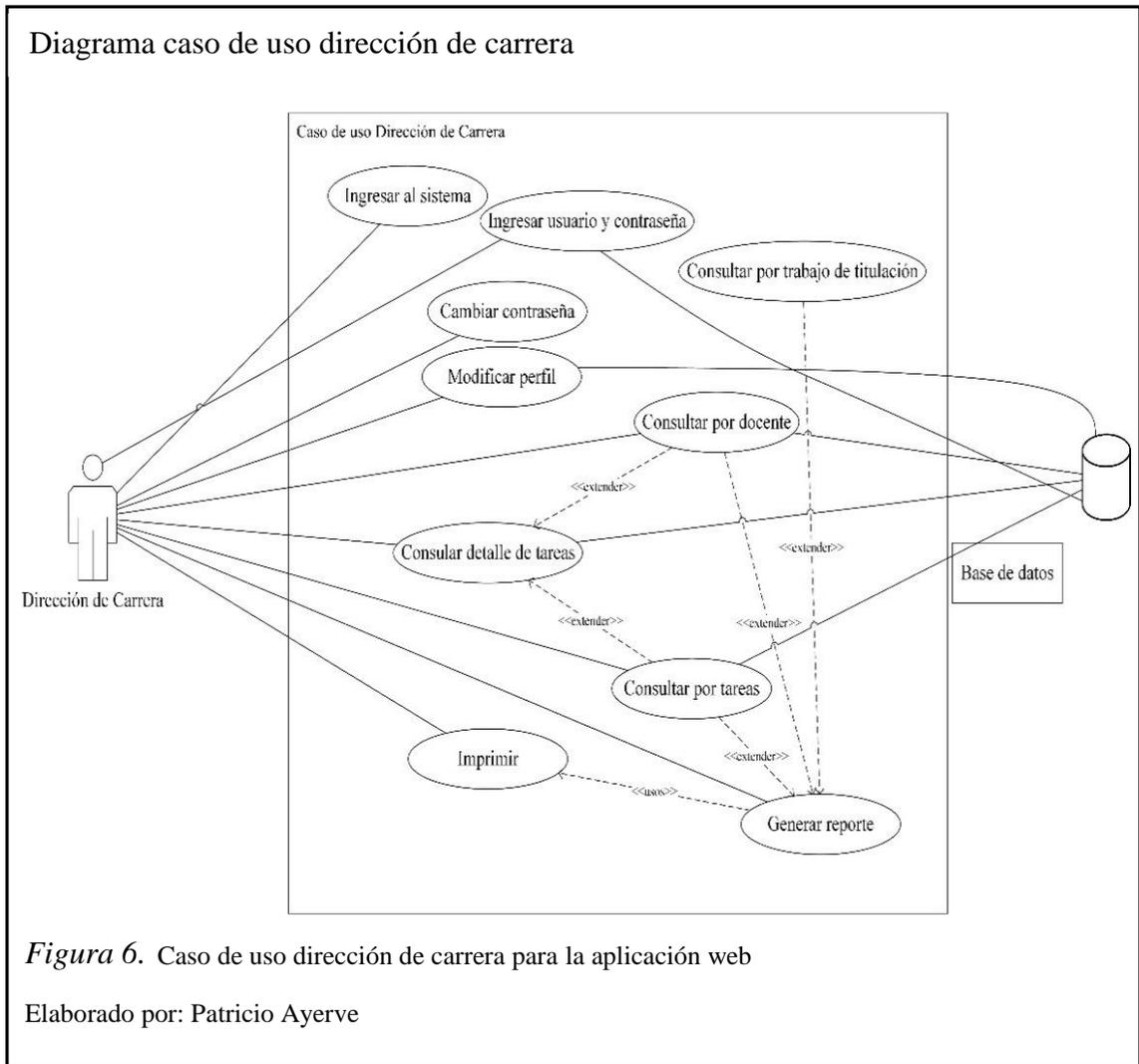


Figura 6. Caso de uso dirección de carrera para la aplicación web

Elaborado por: Patricio Ayerve

2.3.2. Diagrama de clases

A continuación se define el diagrama de clases junto con sus relaciones entre ellas y sus respectivos métodos.

Diagrama de clases

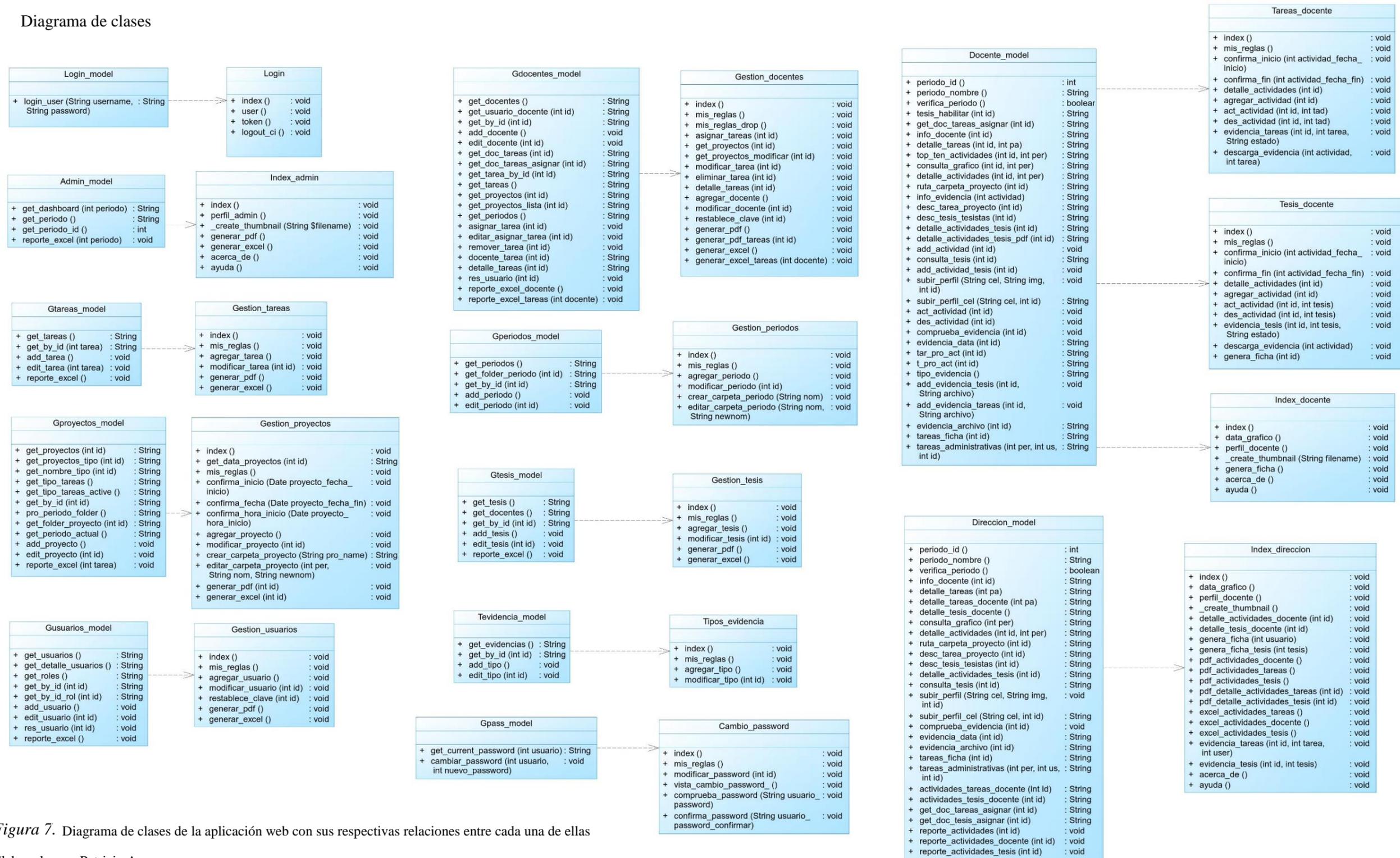


Figura 7. Diagrama de clases de la aplicación web con sus respectivas relaciones entre cada una de ellas

Elaborado por: Patricio Ayerve

2.3.3. Mapas de navegación

En este apartado se describe como el usuario puede desplazarse a través de la aplicación web según el perfil asignado.

2.3.3.1. Mapa de navegación administrador

El usuario con el rol administrador ingresará por la página de autenticación posteriormente se muestra una página de bienvenida. La página de bienvenida de administración muestra los parámetros del sistema y un resumen de las tareas administrativas asignadas a los docentes. Desde aquí, a través del menú de administración, se podrá dirigir a las diferentes páginas para gestionar los datos del sistema.

En la figura 8 se muestra el mapa de navegación para la administración.

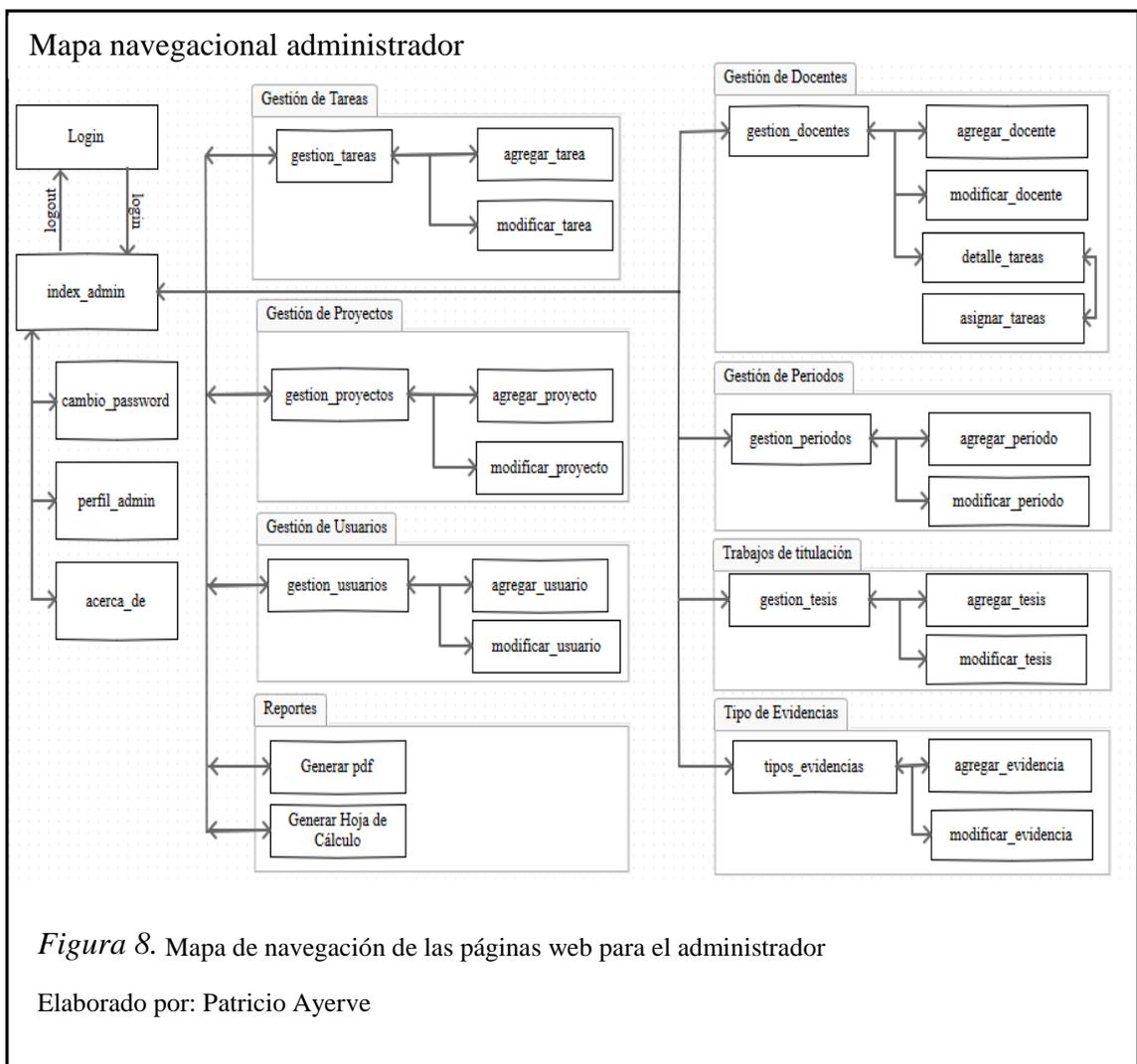


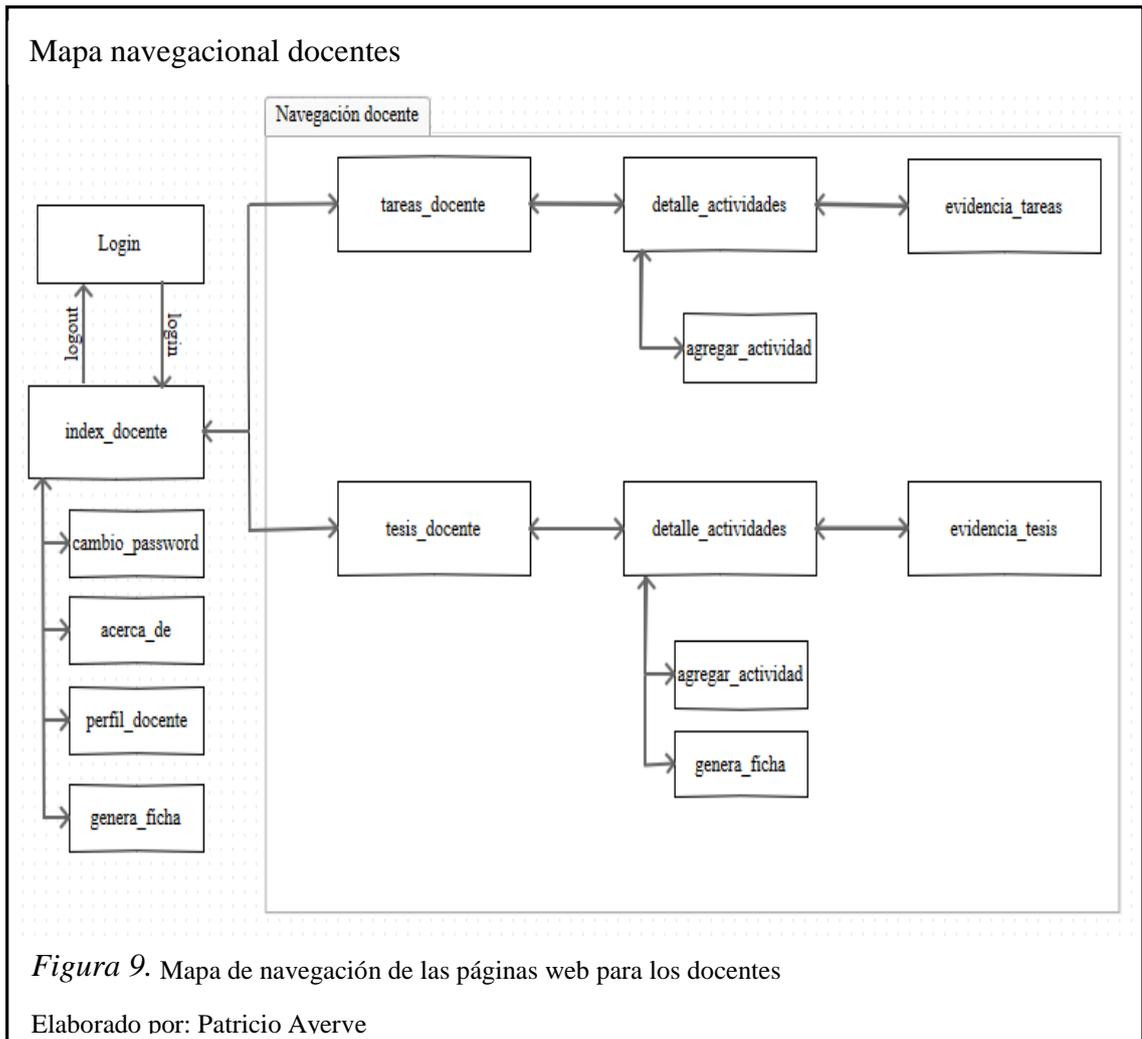
Figura 8. Mapa de navegación de las páginas web para el administrador

Elaborado por: Patricio Ayerve

2.3.3.2. Mapa de navegación docente

El usuario con el rol Docente ingresará por la página de autenticación posteriormente se muestra una página de bienvenida. La página de bienvenida del docente muestra un gráfico de barras del número de actividades por tarea administrativa y un resumen de las mismas en una tabla.

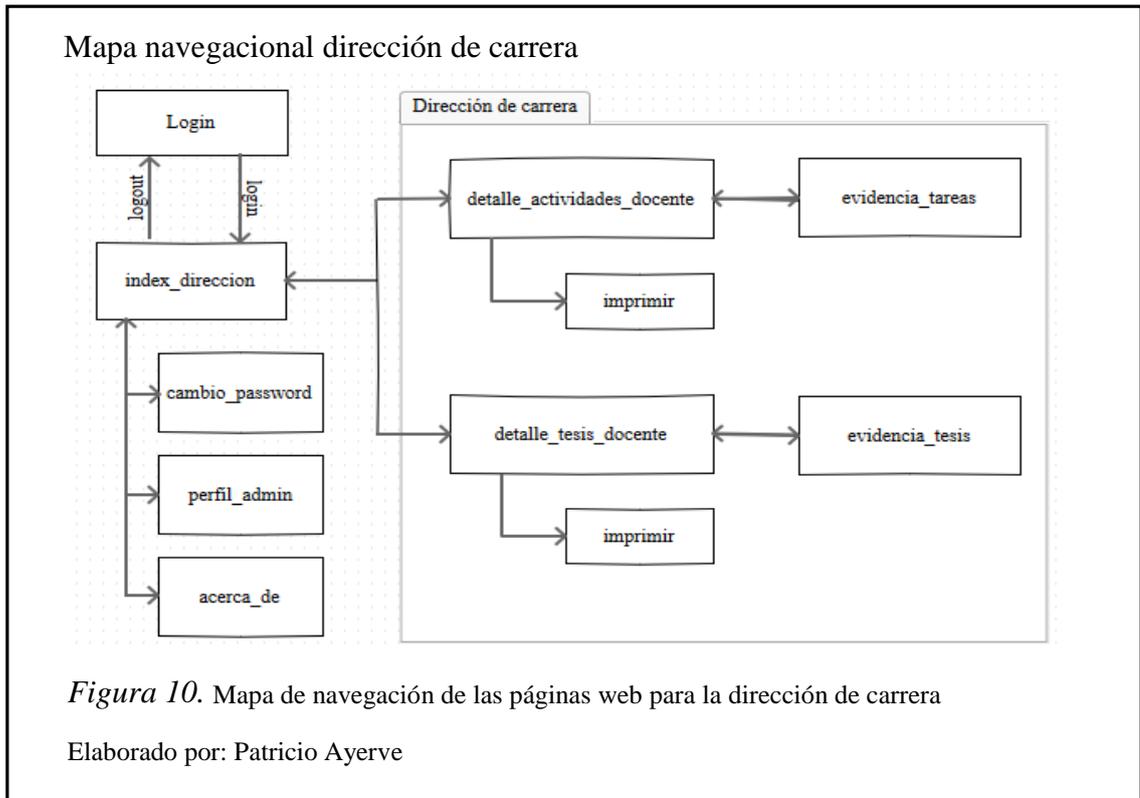
En la figura 9 se muestra el mapa de navegación para el docente



2.3.3.3. Mapa de navegación dirección de carrera

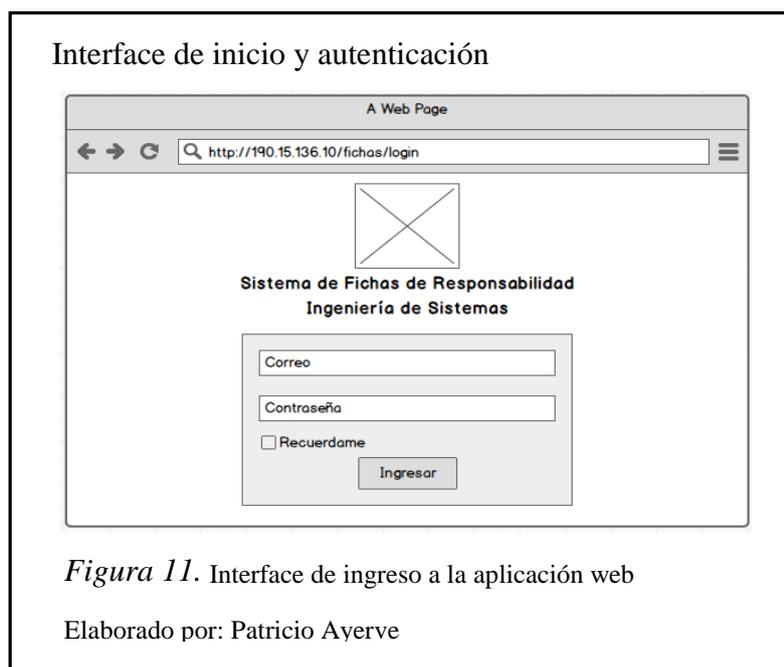
El usuario con el rol de Dirección de Carrera ingresará por la página de autenticación posteriormente se re direccionará a una página de bienvenida, la cual muestra un gráfico de barras con el número de actividades por tarea administrativa con su respectiva tabla de datos, también otra tabla por docentes con información detallada de sus tareas

administrativas y finalmente por trabajo de titulación, cada una ellas permite generar un documento PDF o Excel para ser descargado.



2.3.4. Prototipos de interfaces gráficas de usuario

En este apartado se describen los prototipos de interface de usuario para desplazarse a través de la aplicación web.



- Interface de inicio (Figura 11)

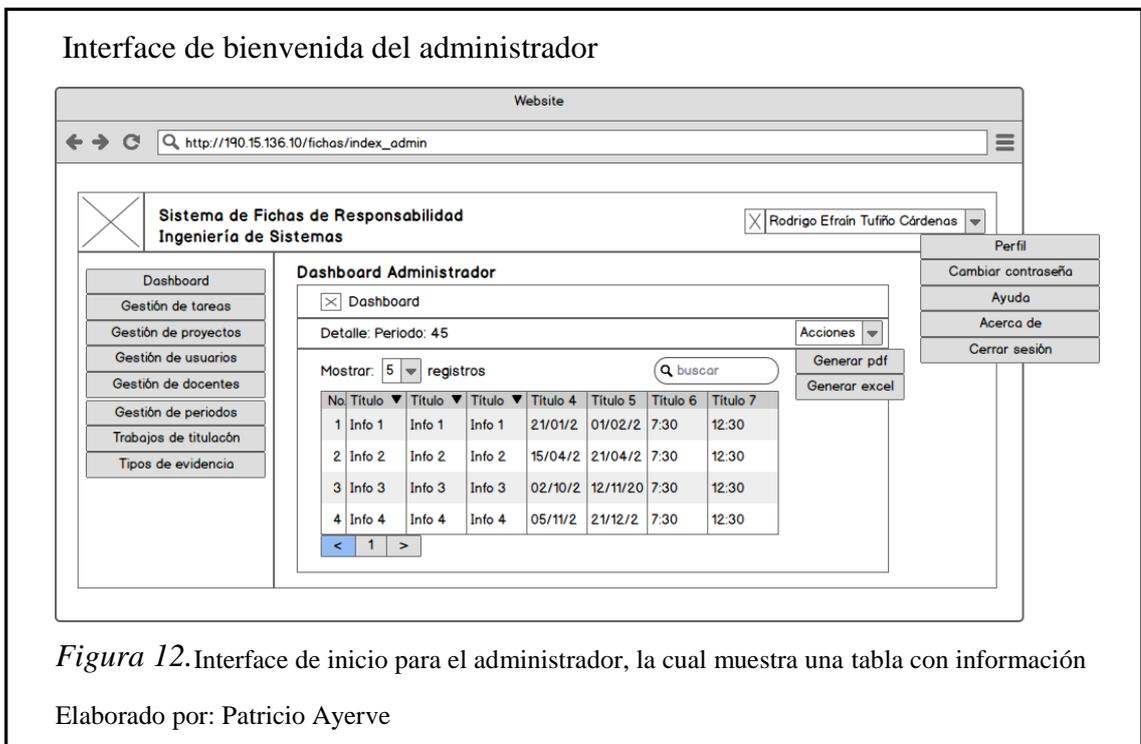
El usuario al acceder a la aplicación web será autenticado por la interface de inicio, la cual solicitará la información de las credenciales propias de cada usuario.

2.3.4.1. Interfaces para el usuario administrador

A continuación se muestran las interfaces que conformaran la administración de la aplicación web.

- Interface de inicio del administrador (Figura 12)

Si el usuario autenticado tiene el rol de administrador, su interface de inicio será la siguiente:



- Interface de opciones del menú lateral izquierdo del administrador (Figura 13)

El diseño de la interface que corresponde al menú lateral izquierdo, excepto para las opciones de Gestión de periodos y Tipos de evidencia las cuales no pueden generar un documento de tipo PDF o Excel, adicionalmente en la opción de Dashboard no se muestra la columna opciones en la tabla ni el botón de agregar/modificar.

Interface opciones menú lateral izquierdo administrador

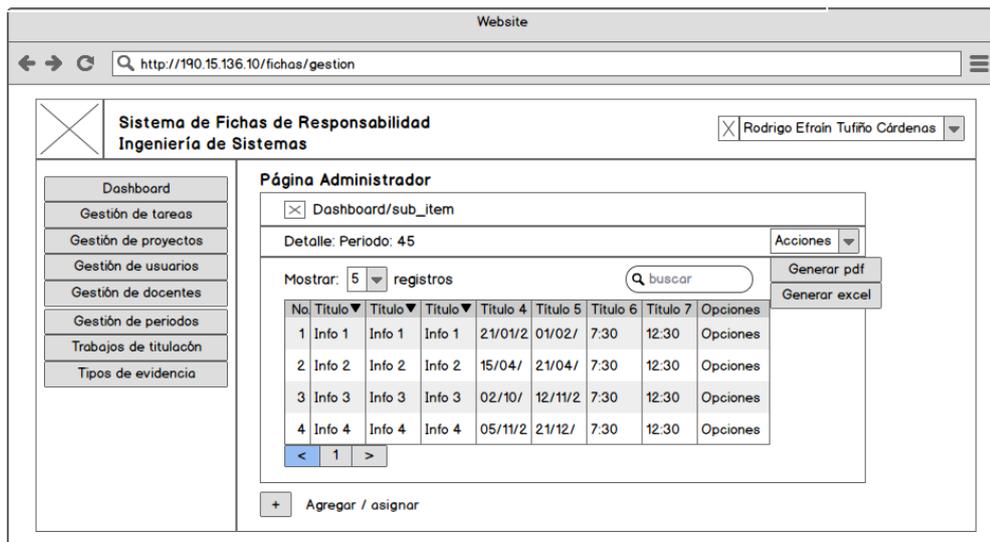


Figura 13. Menú de administración para navegar a través de la aplicación web

Elaborado por: Patricio Ayerve

- Interfaces para agregar o modificar información del Administrador

El diseño de las interfaces para agregar información serán utilizadas para modificar la misma, a continuación se muestra los prototipos para cada una de ellas, con su respectivo mensaje de confirmación:

La primera interface será para agregar y modificar una tarea administrativa mostrada en la Figura 14.

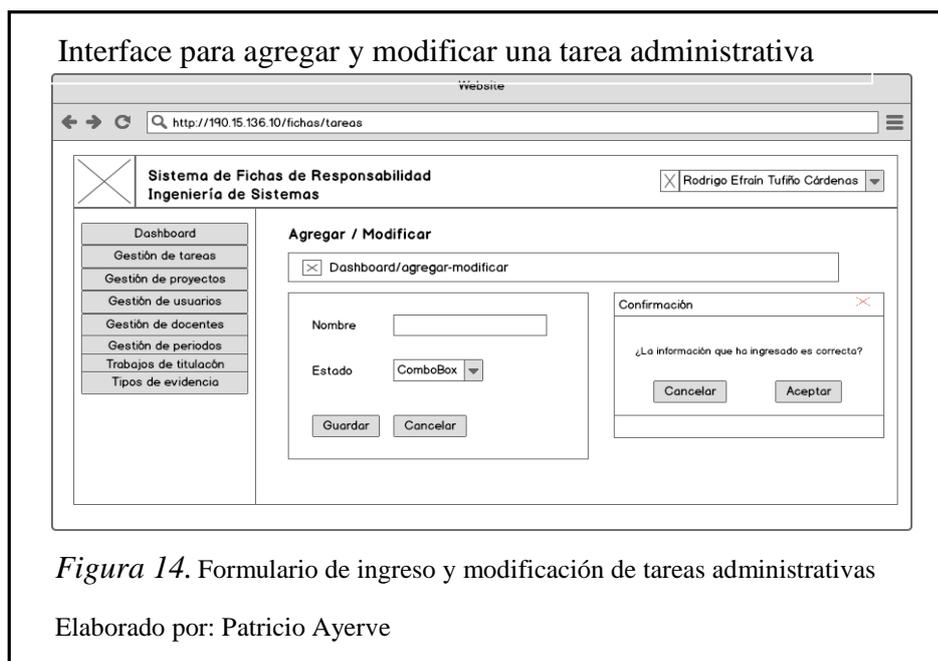
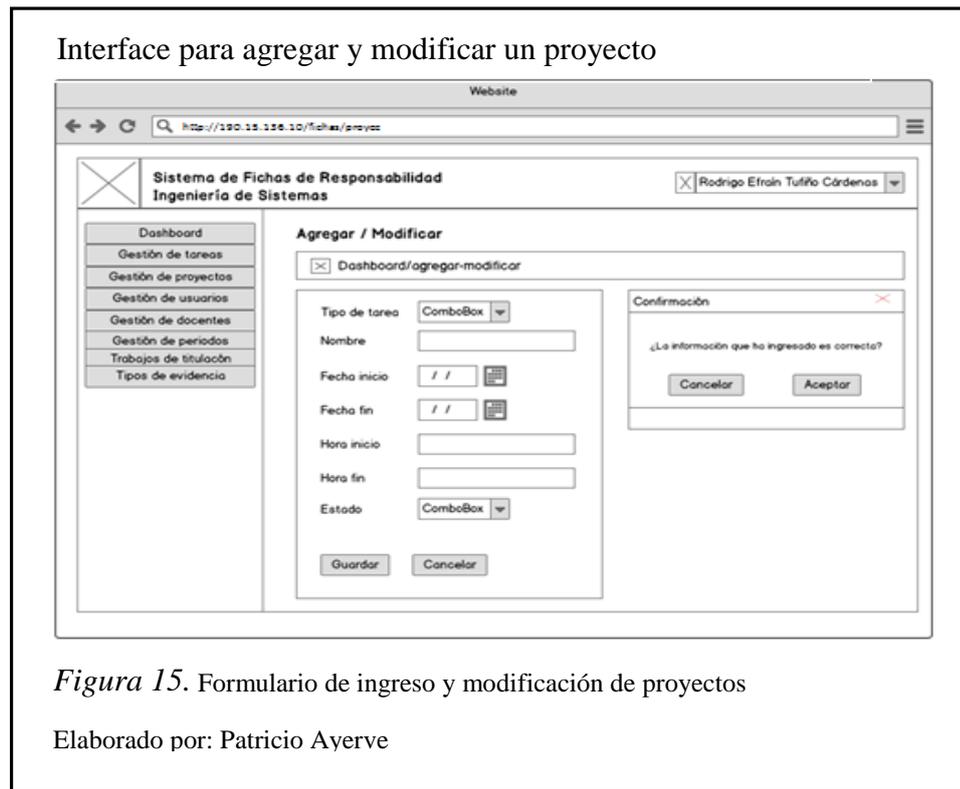


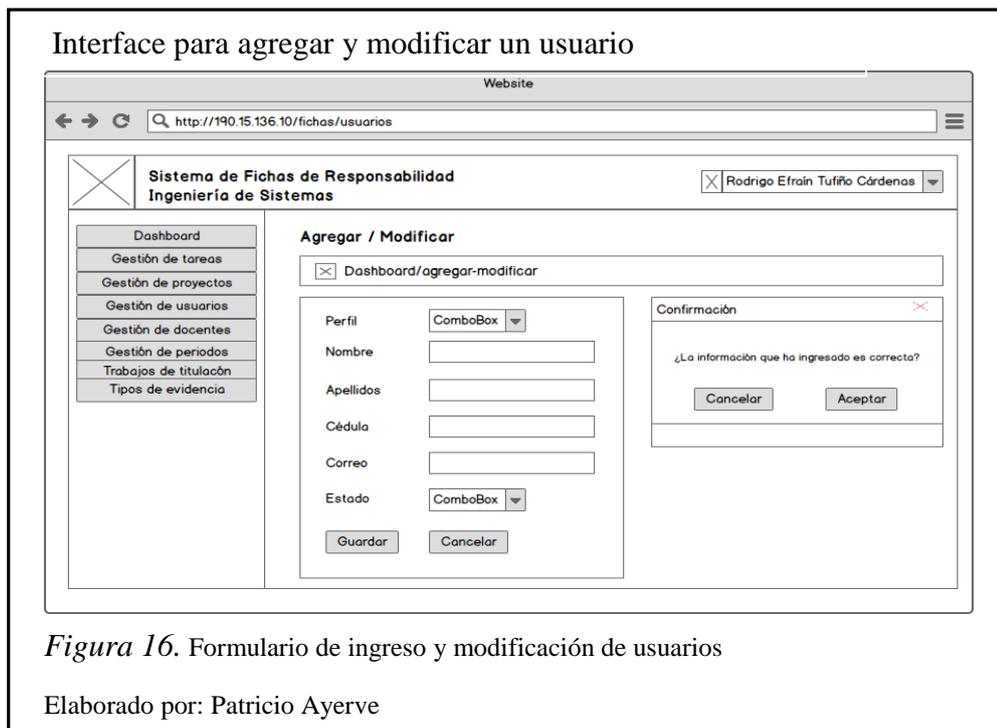
Figura 14. Formulario de ingreso y modificación de tareas administrativas

Elaborado por: Patricio Ayerve

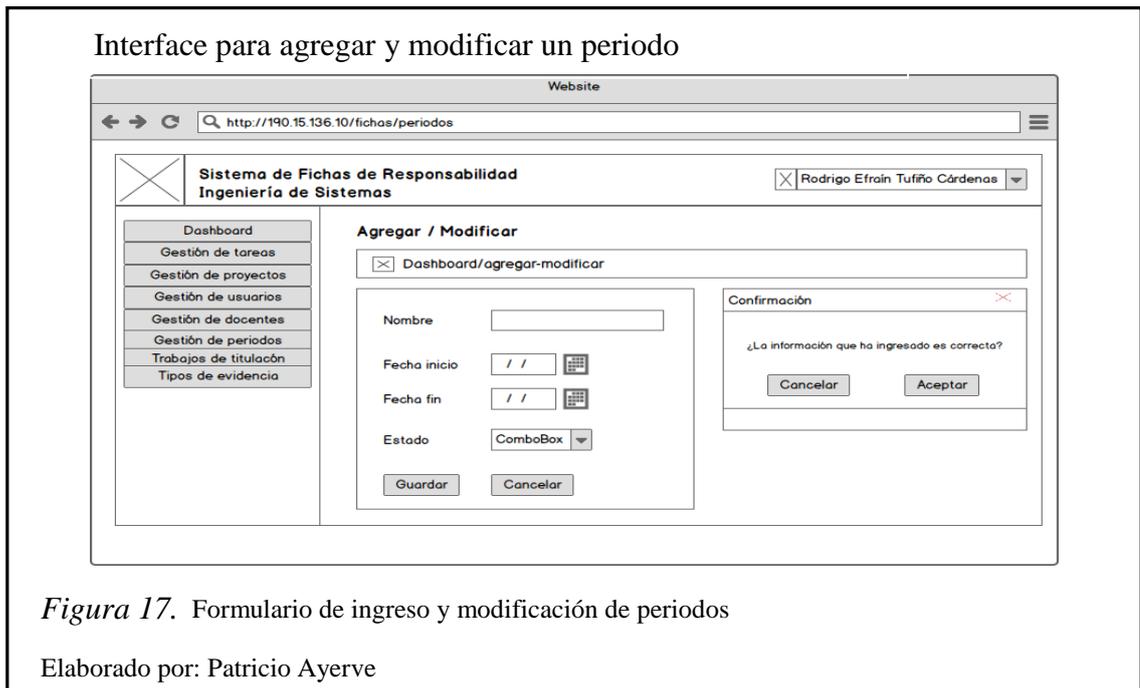
Esta interface será para agregar y modificar un proyecto (Figura 15).



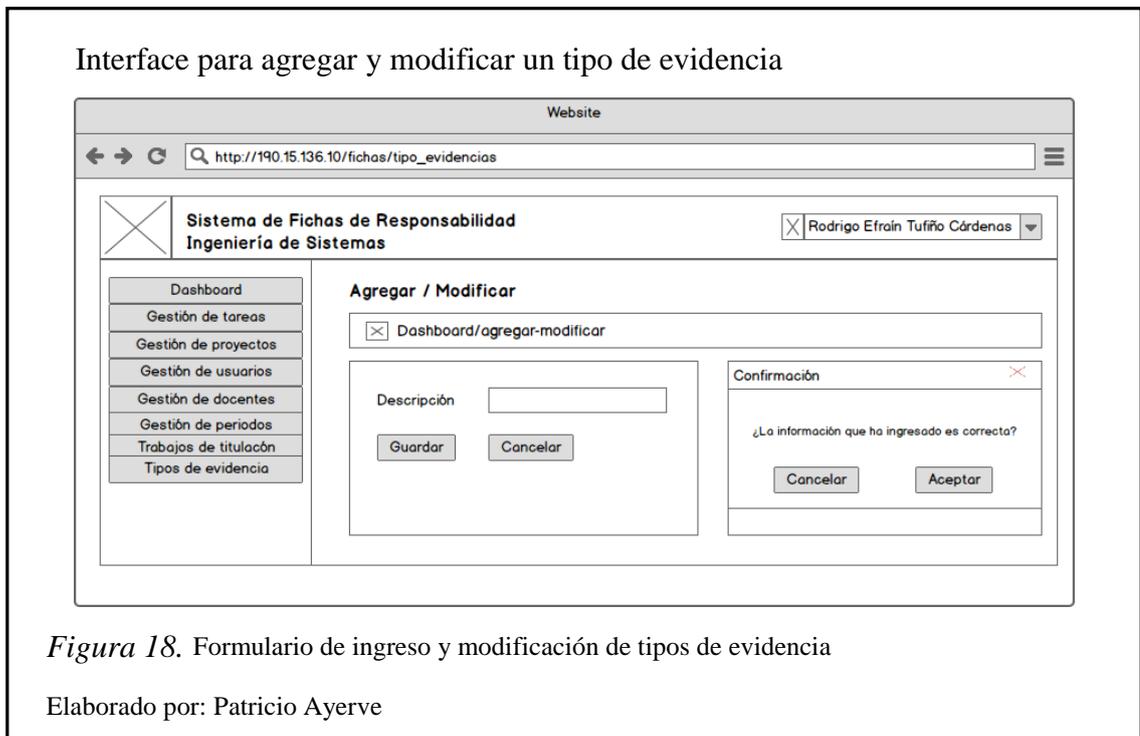
Esta interface será para agregar y modificar un usuario y un docente, pero el docente tiene asignado por defecto su rol así que no dispone del campo perfil (Figura 16).



Esta interface será para agregar y modificar un periodo (Figura 17).



Esta interface será para agregar y modificar los diferentes tipos de evidencia para los archivos que serán subidos al servidor (Figura 18).



Esta interface será para asignar las diferentes tareas administrativas a los docentes con sus respectivas cargas horarias (Figura 19).

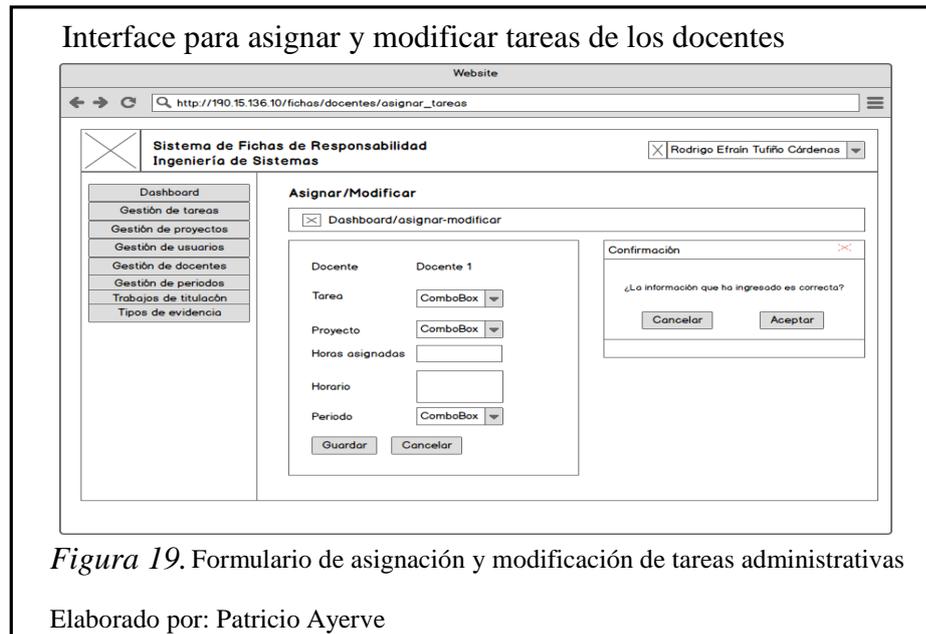


Figura 19. Formulario de asignación y modificación de tareas administrativas

Elaborado por: Patricio Ayerve

2.3.4.2. Interfaces para el usuario docente

A continuación se muestran las interfaces para los docentes en la aplicación web.

- Interface de inicio del docente (Figura 20)

Si el usuario autenticado tiene el rol de Docente, su interface de inicio será la siguiente:

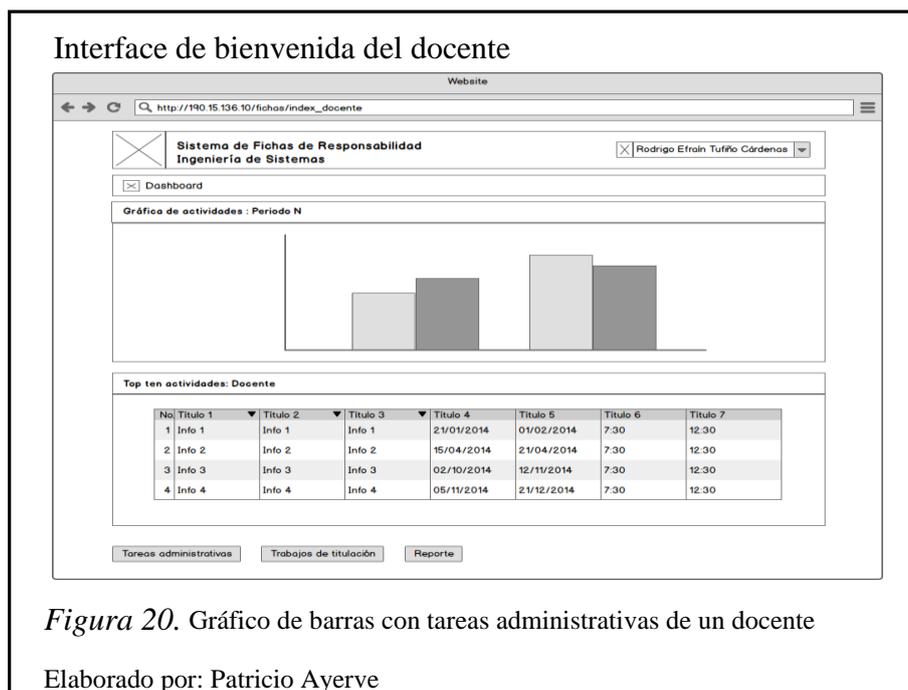
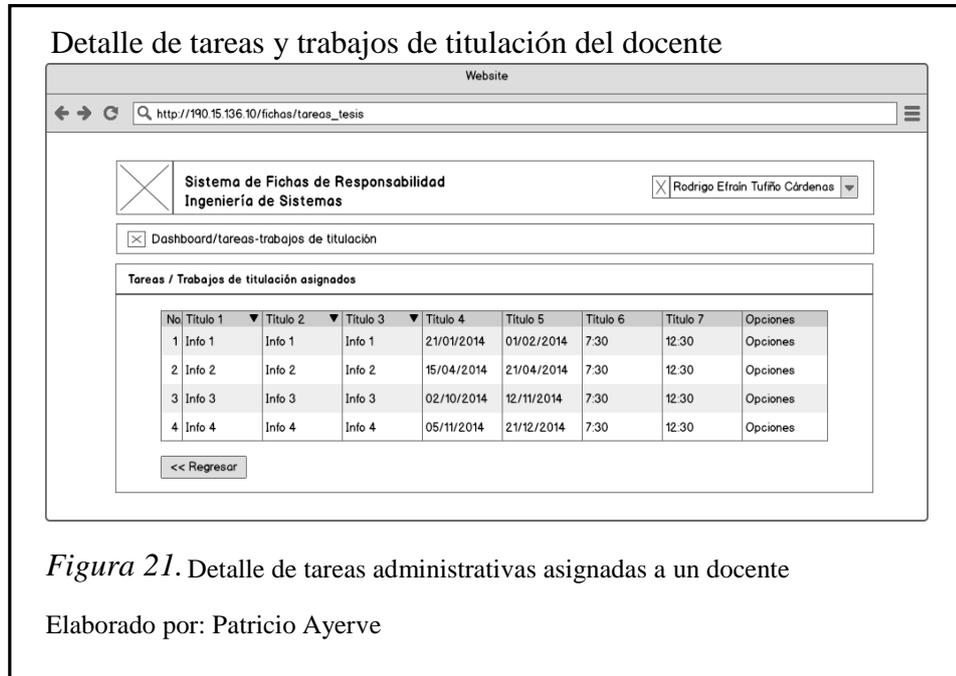


Figura 20. Gráfico de barras con tareas administrativas de un docente

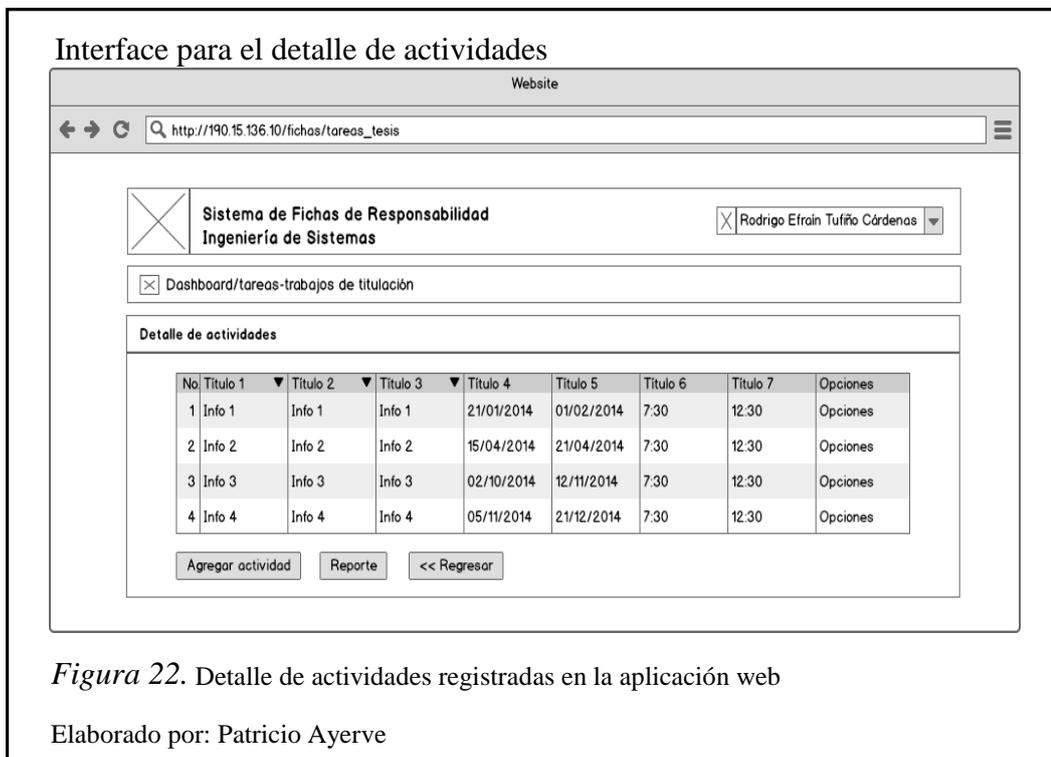
Elaborado por: Patricio Ayerve

- Interfaces de consulta de tareas administrativas y trabajos de titulación

El diseño de las interfaces para consultar las tareas administrativas y los trabajos de titulación asignados mostrados en la Figura 21.



- Interfaces de consulta de actividades para tareas administrativas y trabajos de titulación (Figura 22).



El diseño de la interface para consultar las actividades dentro de las tareas administrativas y trabajos de titulación se muestra a continuación, solo en tareas administrativas no está disponible el botón de reporte.

- Interface para la subida y descarga de evidencias (Figura 23).

La siguiente interface se utilizará para subir y descargar evidencias para el rol de Docente pero el rol de Dirección de Carrera solo tendrá la posibilidad de descarga.

Interface para subir y descargar evidencias

Figura 23. Formulario para subir y descargar evidencias

Elaborado por: Patricio Ayerve

- Interface para agregar actividades para los docentes (Figura 24).

Interface para agregar actividades

Figura 24. Formulario para el registro de actividades de un docente

Elaborado por: Patricio Ayerve

2.3.4.3. Interfaces para el usuario dirección de carrera

A continuación se muestra las interfaces para la Dirección de Carrera.

- Interface de inicio Dirección de carrera (Figura 25).

Si el usuario autenticado tiene el rol de Dirección de Carrera, su interface de inicio será la siguiente:

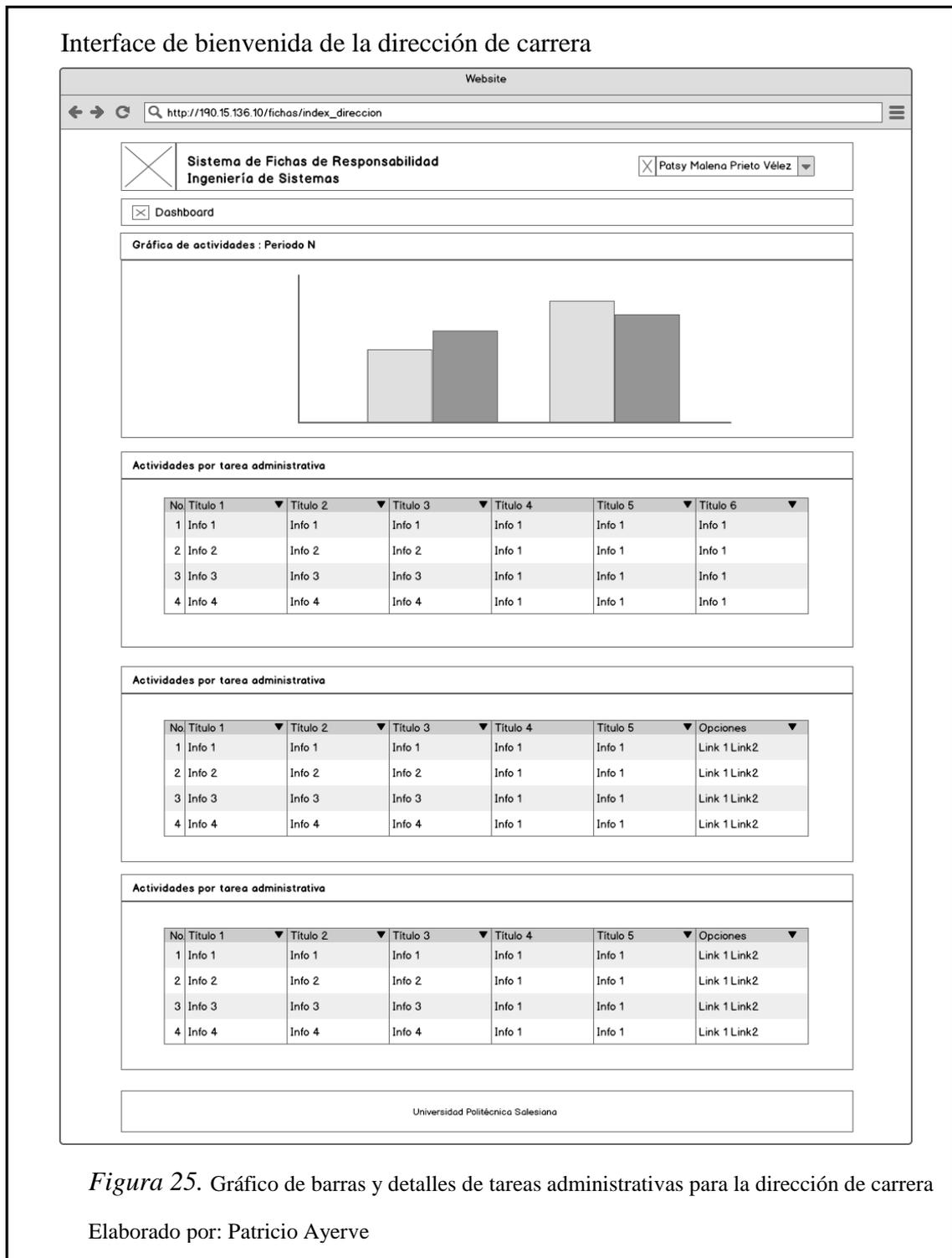


Figura 25. Gráfico de barras y detalles de tareas administrativas para la dirección de carrera

Elaborado por: Patricio Ayerve

- Interface de actividades por docente (Figura 26).

Esta interface mostrará una tabla con todas las actividades de un docente en específico para la Dirección de Carrera, además se podrá imprimir la información generada en una tabla.

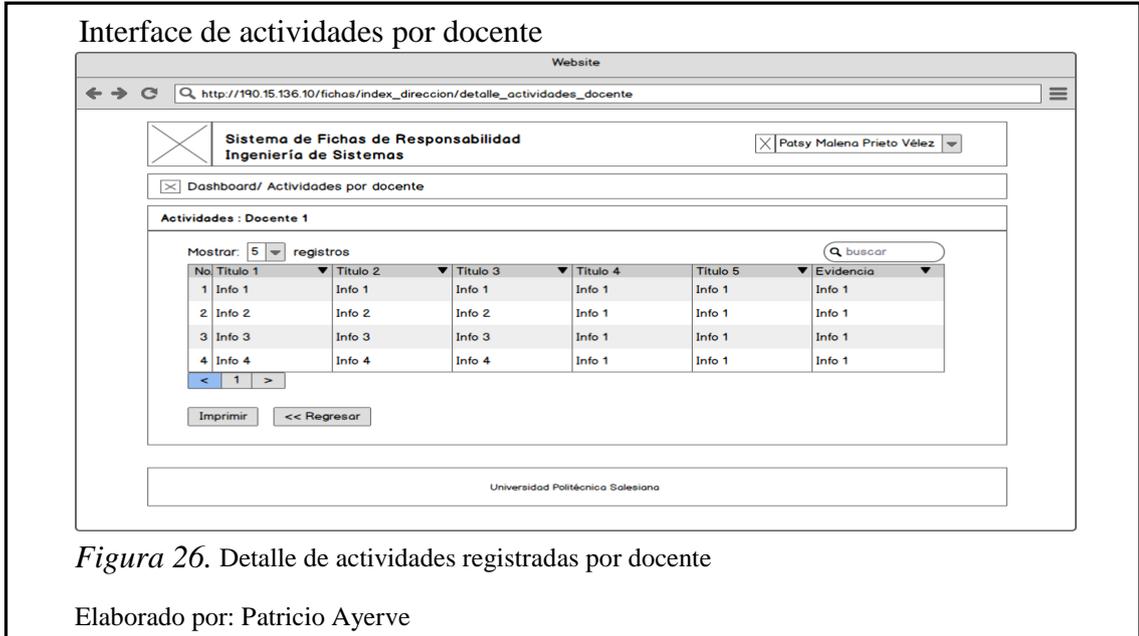


Figura 26. Detalle de actividades registradas por docente

Elaborado por: Patricio Ayerve

- Interface de actividades por trabajo de titulación (Figura 27).

Esta interface mostrará una tabla con todas las actividades de un docente en específico por trabajo de titulación para la Dirección de Carrera, además se podrá imprimir la información generada en una tabla.

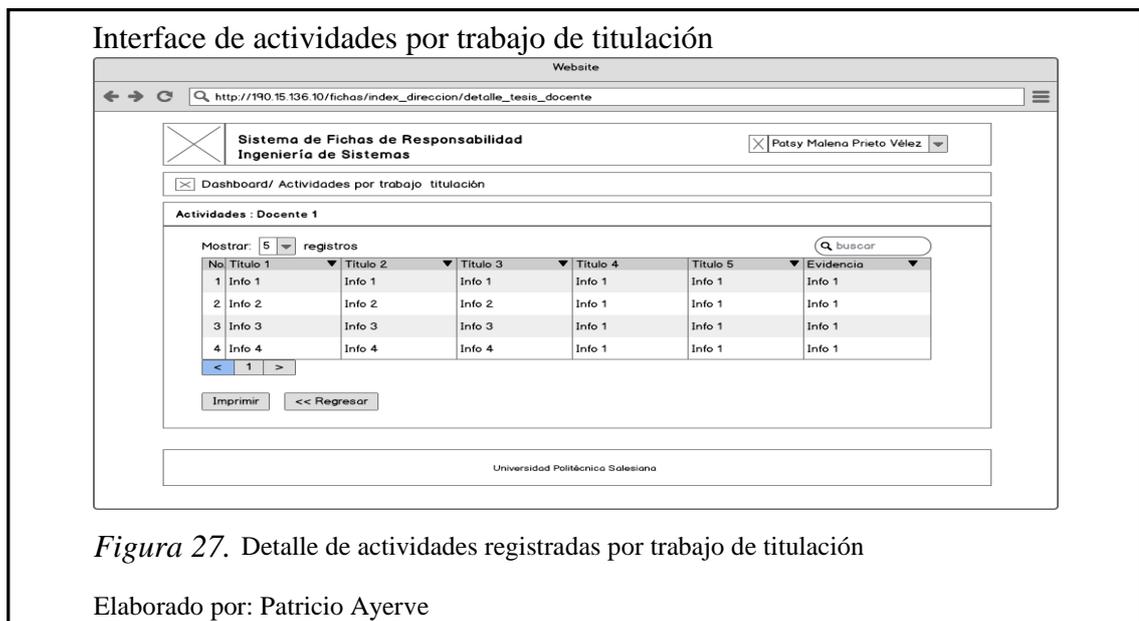
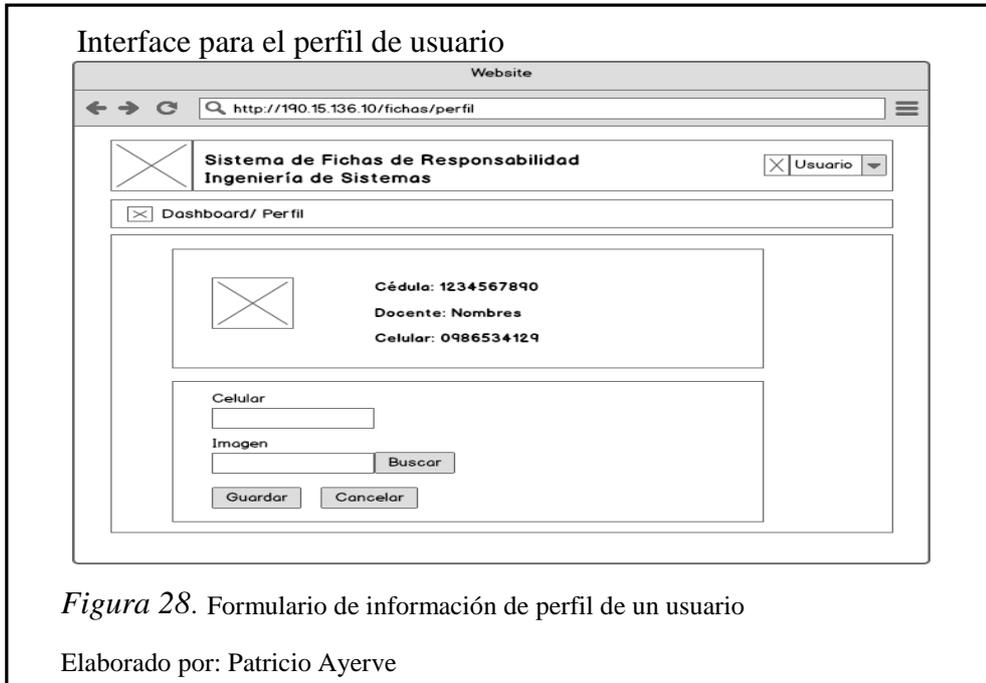


Figura 27. Detalle de actividades registradas por trabajo de titulación

Elaborado por: Patricio Ayerve

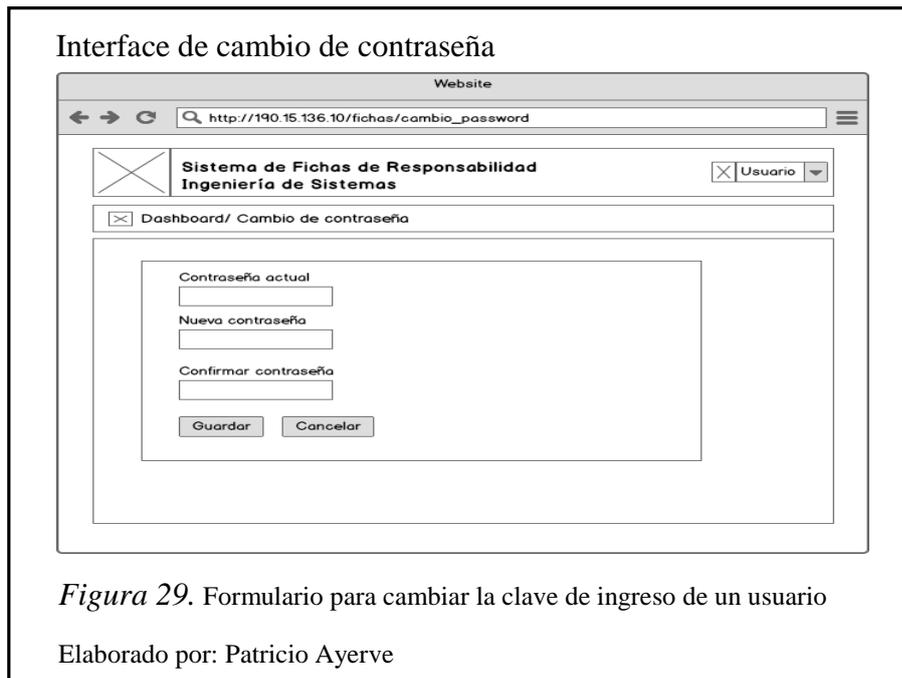
- Interface de perfil de usuario (Figura 28).

Esta interface estará presente en todos los usuarios del sistema, los cuales podrán subir una imagen para su perfil y un número de celular.



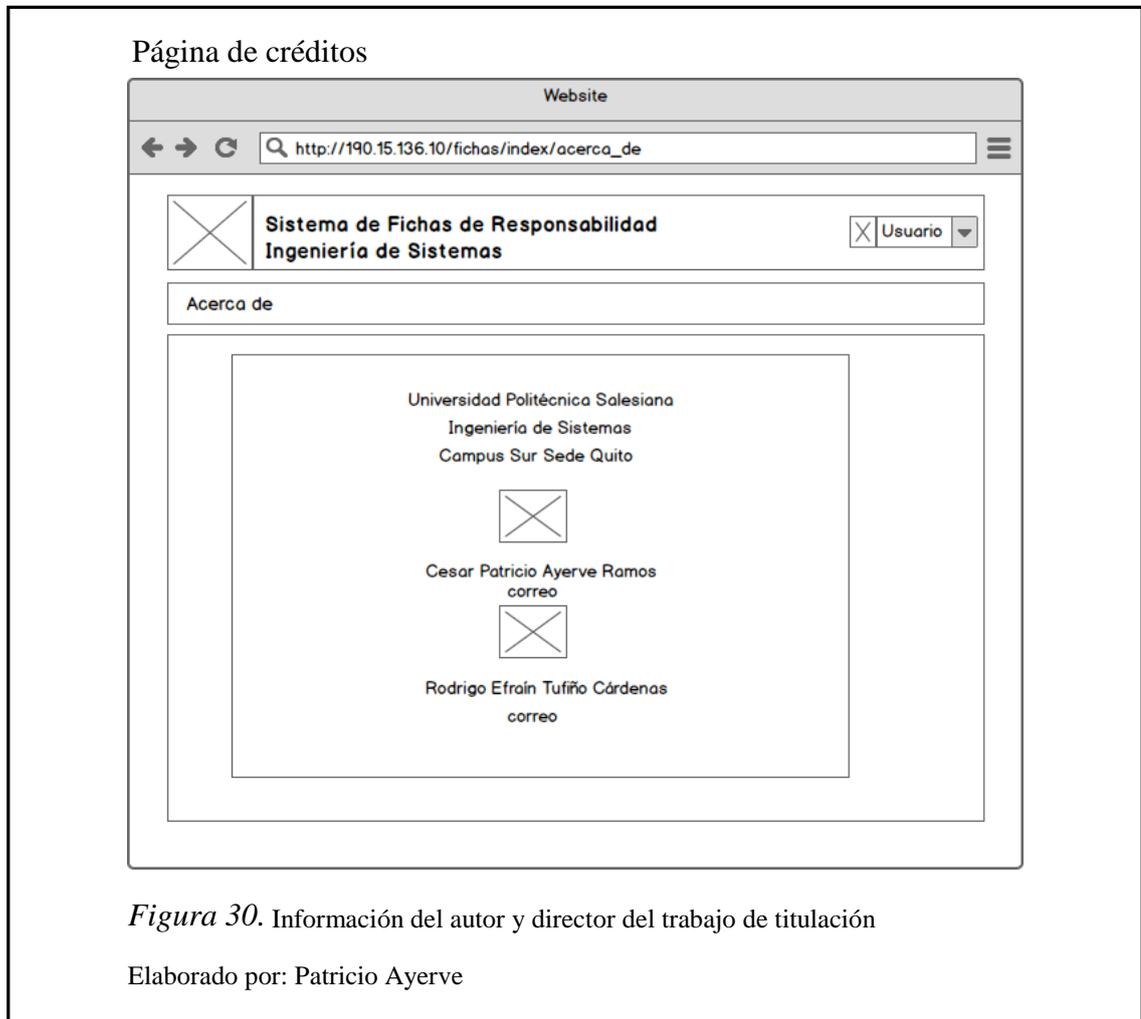
- Interface de cambio de contraseña (Figura 29).

Esta interface estará presente en todos los usuarios del sistema, los cuales podrán cambiar su contraseña.



- Interface de créditos (Figura 30).

Esta interface muestra la información relacionada con el autor del presente trabajo de titulación y el docente tutor de la misma.



2.5. Diseño conceptual base de datos

La base de datos ha sido diseñada en función de los requerimientos planteados por la Dirección de Carrera. Se ha desarrollado el siguiente diagrama conceptual que muestra las tablas necesarias para almacenar toda la información con sus relaciones entre cada una de ellas.

Diseño conceptual base de datos

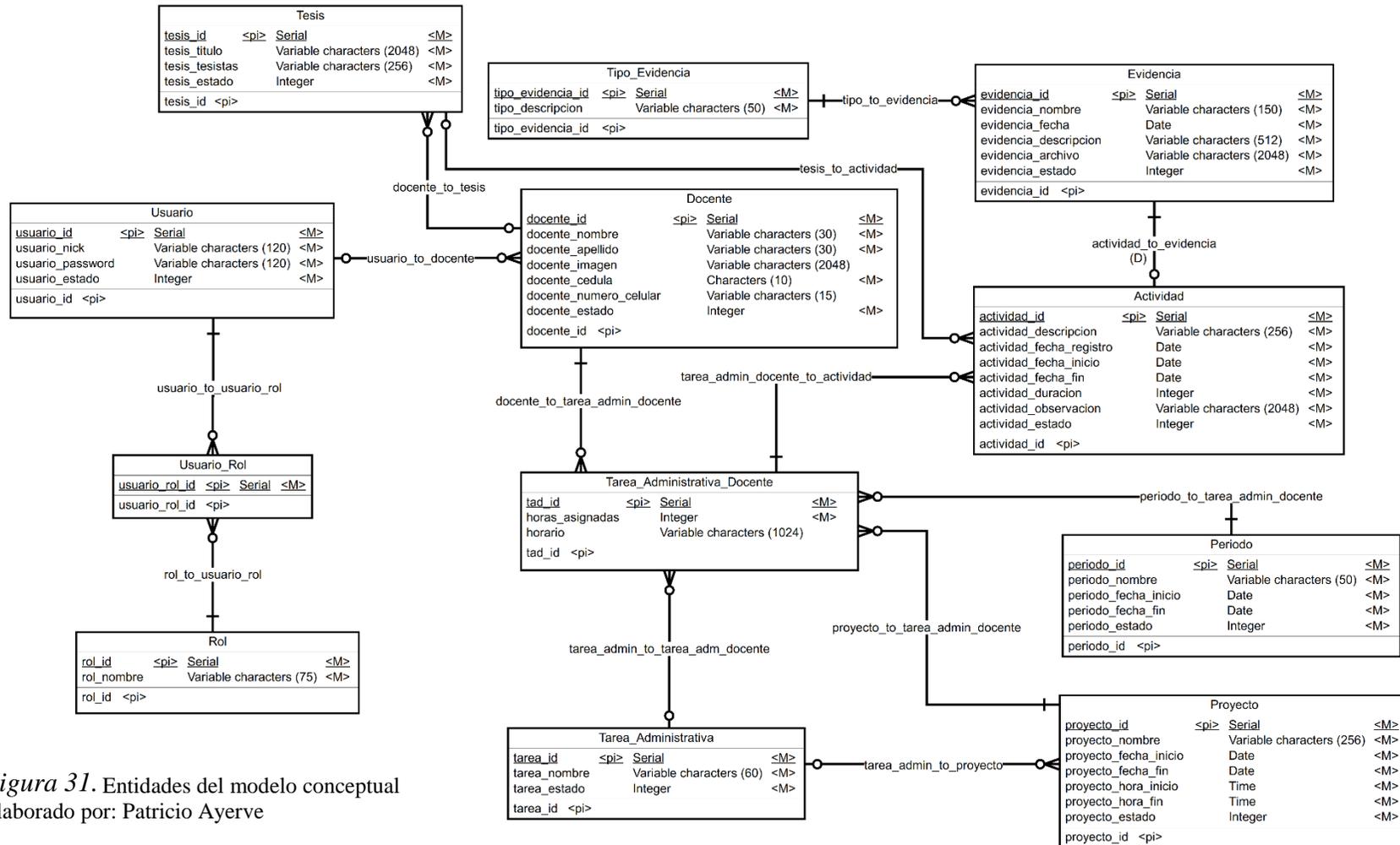


Figura 31. Entidades del modelo conceptual
Elaborado por: Patricio Ayerve

2.6. Atributos del sistema

- El sistema deberá estar disponible 24 horas al día y 7 días a la semana (24/7).
- Deberá soportar al menos 20 usuarios concurrentes.
- Las herramientas y plataformas que utilice para el desarrollo del sistema deberán ser Software Libre.
- El sistema debe proveer mecanismos de seguridad para que solo usuarios autorizados ingresen, y definir un tiempo límite de inactividad para la sesión de los usuarios.
- Se desea que las contraseñas de los usuarios se almacenen encriptadas.
- Todos los usuarios del sistema deben ingresar a través de un proceso de autenticación definido por un nombre de usuario y una contraseña.
- Cada usuario podrá cambiar su contraseña sin la necesidad de solicitar ayuda al administrador.

CAPÍTULO 3

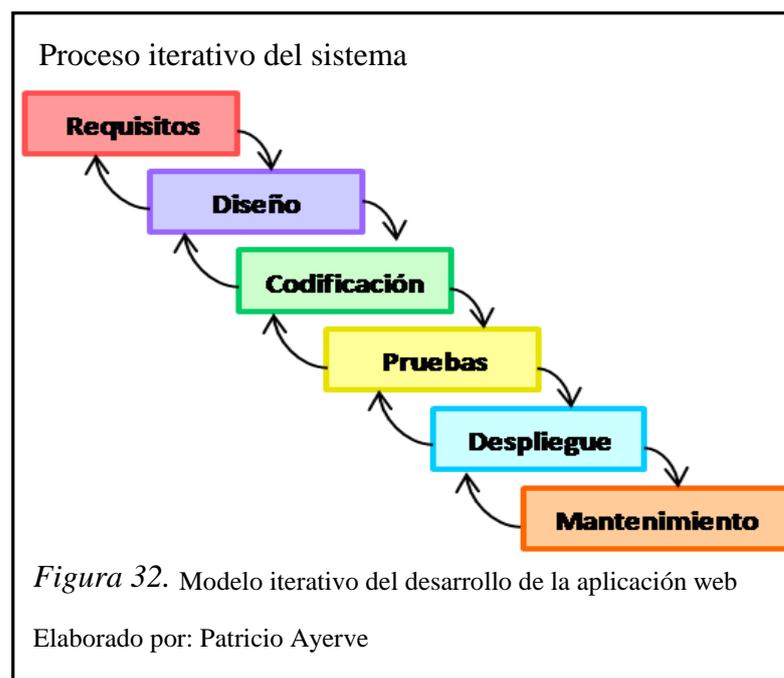
CONSTRUCCIÓN Y PRUEBAS

En este capítulo se detallará el proceso de construcción, las herramientas tecnológicas a utilizar y finalmente las pruebas realizadas sobre la aplicación web para que sea funcional y cumpla con las expectativas del usuario final.

Es preciso mencionar que para la programación del sistema se aplicó la metodología XP, para lo cual se establecía requerimientos de un módulo en específico, se desarrollaba los prototipos, se los probaba y una vez que el módulo era funcional, se procedía a establecer nuevos requerimientos. Es así que el desarrollo del sistema se lo realizó en el siguiente orden:

1. Autenticación de usuarios
2. Módulo de administración
3. Módulo de docentes
4. Módulo de dirección de carrera
5. Reportes de administración
6. Reportes de docentes
7. Reportes de dirección de carrera

En los siguientes apartados se muestra la programación y el resultado final del sistema. Por motivos de tiempo y espacio no se ha colocado el desarrollo iterativo que siguió la programación del sistema.



3.1. Herramientas y plataformas

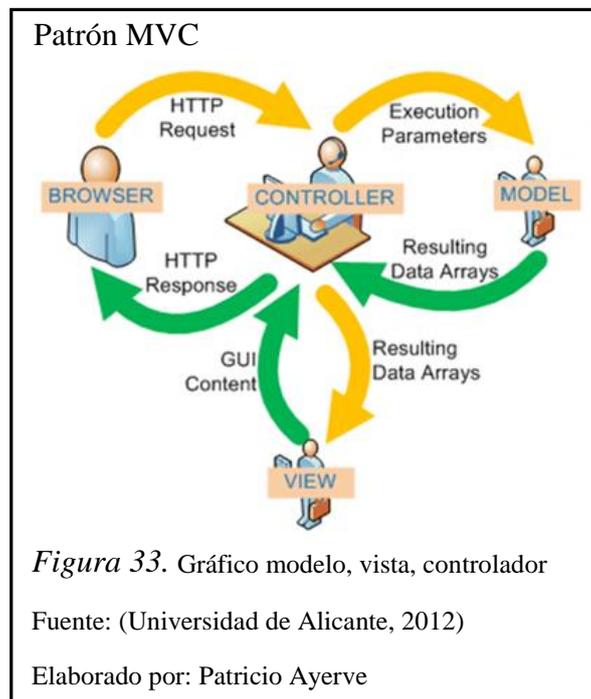
La Dirección de Carrera tiene un servidor de aplicaciones implementado en el cual se realizará todas las configuraciones necesarias, los cambios de los scripts y el depósito de la aplicación web se realizara de forma remota utilizando el cliente ftp Filezilla.

A continuación se muestra la descripción de las herramientas tecnológicas que se utilizarán para el desarrollo de la aplicación web.

3.1.1 Patrón MVC (modelo, vista, controlador)

Modelo-Vista-Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.



- El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La **Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

- El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno (Universidad de Alicante, 2012).

3.1.2. Entorno de desarrollo Netbeans IDE

Netbeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación.

Netbeans IDE permite rápida y fácilmente desarrollar sistemas de escritorio Java, móviles y aplicaciones web, así como aplicaciones HTML5, Javascript y CSS. El IDE también proporciona un gran conjunto de herramientas para desarrolladores de PHP, Java y C / C ++. Es gratuito y de código abierto y tiene una gran comunidad de usuarios y desarrolladores de todo el mundo.

Para el desarrollo de la aplicación se ha optado por la versión del IDE más actual al momento es la 8.0.2.

3.1.3. Configuración de Codeigniter 2.2.0

La configuración de Codeigniter dentro de Netbeans 8.0.2 se realiza de la siguiente manera:

- Descargar la versión comprimida estable de Codeigniter de la siguiente página web: <http://www.codeigniter.com/download>, actualmente la versión estable es la 2.2.0.
- Activar las características de desarrollo para PHP dentro del IDE, posteriormente buscar el archivo php.ini el cual será el intérprete del lenguaje.
- Una vez concretado el paso anterior proceder con la creación de un nuevo proyecto PHP con fuente existente, buscar la carpeta descomprimida de Codeigniter y establecer los parámetros mostrados en la figura 34, luego definir la dirección del servidor de aplicaciones y finalmente al ejecutar el proyecto, se desplegara en el navegador la página de bienvenida del framework, cabe mencionar que el servidor de aplicaciones debe estar iniciado para que todo funcione.

Localización archivo php.ini

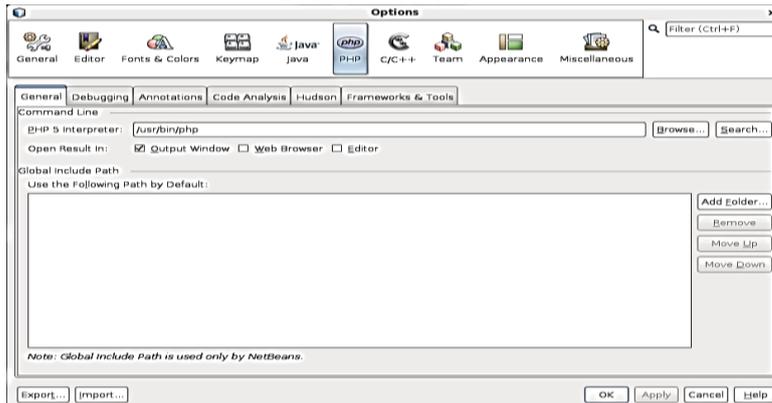


Figura 34. Localizar el archivo php.ini para que Netbeans lo interprete
Elaborado por: Patricio Ayerve

Creación de un proyecto PHP con fuente existente

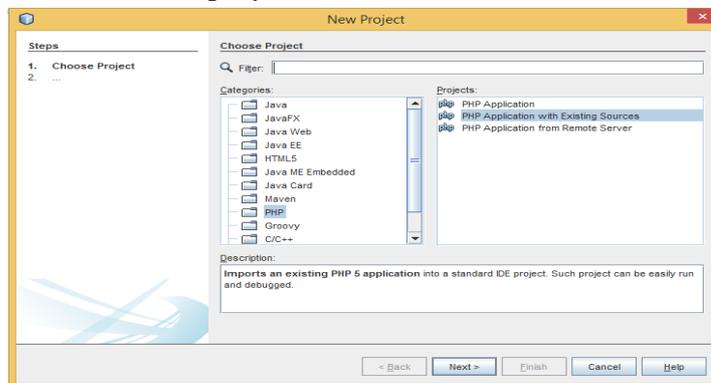


Figura 35. Al crear el proyecto se crean las carpetas del framework
Elaborado por: Patricio Ayerve

Instalación Codeigniter en Netbeans

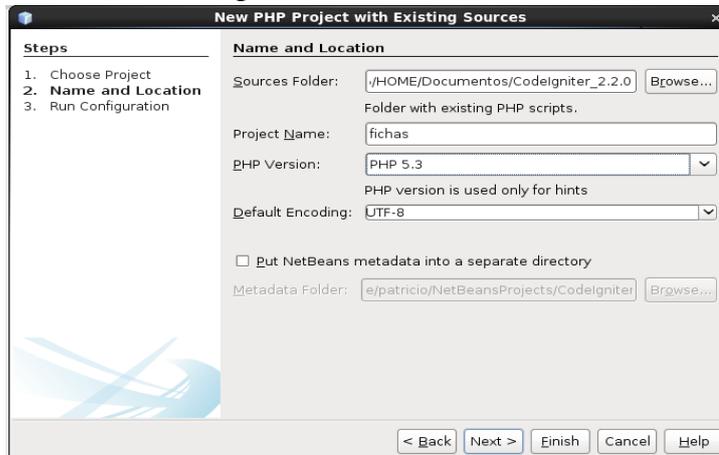


Figura 36. Al crear el proyecto se debe escoger la versión de PHP 5.3
Elaborado por: Patricio Ayerve

Asignando dirección del servidor de aplicaciones

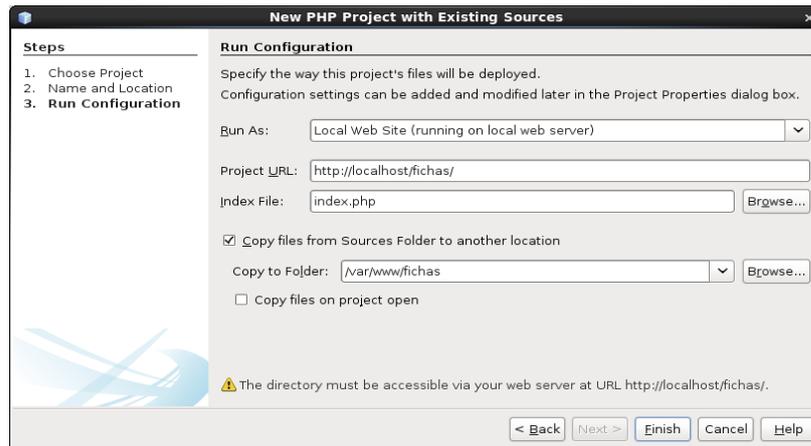


Figura 38. Al crear el proyecto asignar la dirección del servidor de desarrollo
Elaborado por: Patricio Ayerve

Página de bienvenida Codeigniter

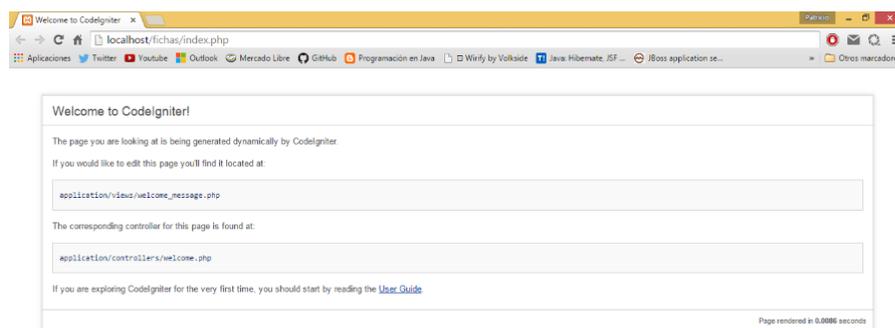


Figura 37. Página de bienvenida se muestra si se ha configurado correctamente
Elaborado por: Patricio Ayerve

Para el autocompletado de código en Netbeans solo basta con copiar la carpeta autocomplete en nbproject dentro de la carpeta con el nombre del proyecto, a continuación un enlace para descargar la carpeta autocomplete.

<https://mega.co.nz/#!OwwmRbqD!gFETd2rMJgCZu9C9y71U0QDu7Ot6RucEvQG8I2PAo9g>

3.1.4. JQuery

Jquery se encuentra escrito en JavaScript, un lenguaje de programación muy rico y expresivo. JQuery es un framework de JavaScript para facilitar, entre otros, el acceso a los elementos del DOM, los efectos, interactuar con los documentos HTML, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

Jquery podría ser otro framework más como script.aculo.us, MooTools, YUI pero en cambio disponemos una gran potencia con una facilidad mucho mayor que sus competidores.

Jquery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX al servidor (D'Onofrio, 2013).

Para el desarrollo de la aplicación se utilizará JQuery con la versión 1.11.2.

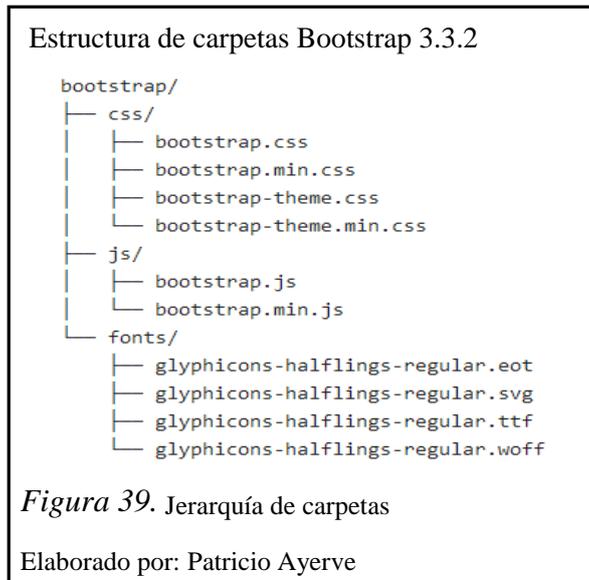
3.1.5. Bootstrap 3.3.2

Para la implementación de las vistas en Codeigniter se ha optado por la utilización de Bootstrap 3.3.2, ya que provee de un gran número de herramientas para crear diseños adaptativos a la pantalla de un dispositivo y también ahorra el tiempo de creación de estilos.

Bootstrap se puede descargar de dos maneras, compilado o mediante el código fuente original. Dependiendo de la forma que se haya elegido, se verá una estructura de directorios u otra en la siguiente dirección:

<http://getbootstrap.com/getting-started/#download>

Todos los plugins JavaScript de Bootstrap requieren la librería JQuery para funcionar, por lo que deberá ser incluirlo en las plantillas. Después de descomprimir el archivo descargado con la versión compilada de Bootstrap, se verá la siguiente estructura de archivos y directorios:



Estos archivos son la forma más sencilla de utilizar Bootstrap en cualquier proyecto web. Para cada archivo se ofrecen dos variantes: los archivos compilados (cuyo nombre es bootstrap.*) y los archivos compilados más los comprimidos (cuyo nombre es bootstrap.min.*). También se incluyen las fuentes de los iconos del proyecto Glyphicons y el tema opcional de Bootstrap (Jacob & Otto, 2014).

3.1.6. PostgreSQL 9.1

El gestor que se utilizará será PostgreSQL 9.1, ya que es la versión que está implementada en el servidor de desarrollo facilitado por la Dirección de Carrera, se procederá a cargar el script de la base de datos, así como sus respectivos procedimientos almacenados que permitirán registrar información y las vistas para la consultar información.

El servidor para la base de datos cuenta con un usuario y contraseña de acceso, estas serán configuradas dentro de Codeigniter para poder establecer una conexión vía TCP/IP para realizar las operaciones correspondientes solicitadas por la aplicación web.

3.1.7. Librerías y helpers de Codeigniter

Codeigniter como framework de desarrollo de aplicaciones provee de librerías y helpers que facilitan el desarrollo de las aplicaciones web, ya que estas son comúnmente usadas por muchos sistemas informáticos, a continuación se detallan que librerías y helpers serán utilizados:

- Librería Session: la clase permite mantener el "estado" de un usuario y el seguimiento de su actividad mientras navegan el sitio web.
- Librería File Uploading: permite subir archivos, se puede configurar varias preferencias, restringir el tipo y tamaño de los archivos.
- Librería Form Validation: el framework provee validación de formularios que ayuda a minimizar la cantidad de código a escribir.
- Librería Security: la clase de seguridad contiene métodos que ayudan a crear una aplicación segura, el procesamiento de los datos de entrada serán seguros.
- Librería Image Manipulation: la clase permite realizar las siguientes acciones: cambio de tamaño, miniaturas, recorte de imágenes, rotación de imagen, imagen con marca de agua. Las tres principales bibliotecas de imágenes son compatibles con: GD / GD2, NetPBM e ImageMagick
- Helper Download: ayudante que permite descargar datos al escritorio.
- Helper Form: contiene funciones que ayudan en el trabajo con formularios.
- Helper URL: contiene funciones que ayudan en el trabajo con las direcciones URL.

3.1.8. Librerías externas de Codeigniter

Codeigniter permite que se vinculen librerías propias de PHP para permitir en este caso la generación de reportes, a continuación se detallaran brevemente las librerías PHPExcel y TCPDF.

- PHPExcel: es una librería creada en PHP que permite exportar y leer diferentes formatos de Excel, como .xls (Excel 2007), .xlsx (OfficeOpenXML), CSV, PDF, HTML, entre otros. Para utilizar la librería es necesario tener instalado PHP 5.2 o superior, esta licenciado bajo LGPL.
- TCPDF: es una biblioteca de gran alcance para generar PDF con PHP. Es compatible con UTF-8, y es capaz de convertir fragmentos de HTML a PDF. Para utilizar la librería es necesario tener instalado PHP 5.2 o superior.

3.1.9. Jmeter

Apache JMeter es un software de código abierto de Java puro, que fue desarrollado por primera vez por Stefano Mazzocchi de Apache Software Foundation, diseñado para

cargar pruebas de funcionalidad y medir el rendimiento. Puede utilizar JMeter para analizar y medir el rendimiento de una aplicación web o varios servicios. Las pruebas de rendimiento sirven para probar una aplicación web, medir la carga múltiple y el tráfico de usuarios concurrentes. JMeter originalmente se utiliza para la aplicación de pruebas en aplicaciones web o aplicaciones que se conecten a un FTP (Guru99, 2015).

3.1.10. Cliente Filezilla

Filezilla es un cliente de ftp gratuito desarrollado para las plataformas Linux, Windows y Mac OSX válido para todas las versiones de este último, que se caracteriza por su sencillez de uso. Filezilla cubre un área de trabajo para transferir archivos de las páginas web al servidor, procedimiento en el que se utiliza el protocolo FTP (File Transfer Protocol) (Lazaro, 2013).

La versión de cliente a utilizar será la 3.10.2 que al momento es la más actual, y provee una fácil conexión al servidor de aplicaciones para depositar todos los archivos necesarios para el funcionamiento de la aplicación web.

3.2. Diagrama físico de la base de datos en PostgreSQL

Se ha desarrollado el siguiente diagrama físico de la base de datos que muestra las tablas necesarias para almacenar toda la información con sus relaciones entre cada una de ellas.

Diagrama físico de la base de datos

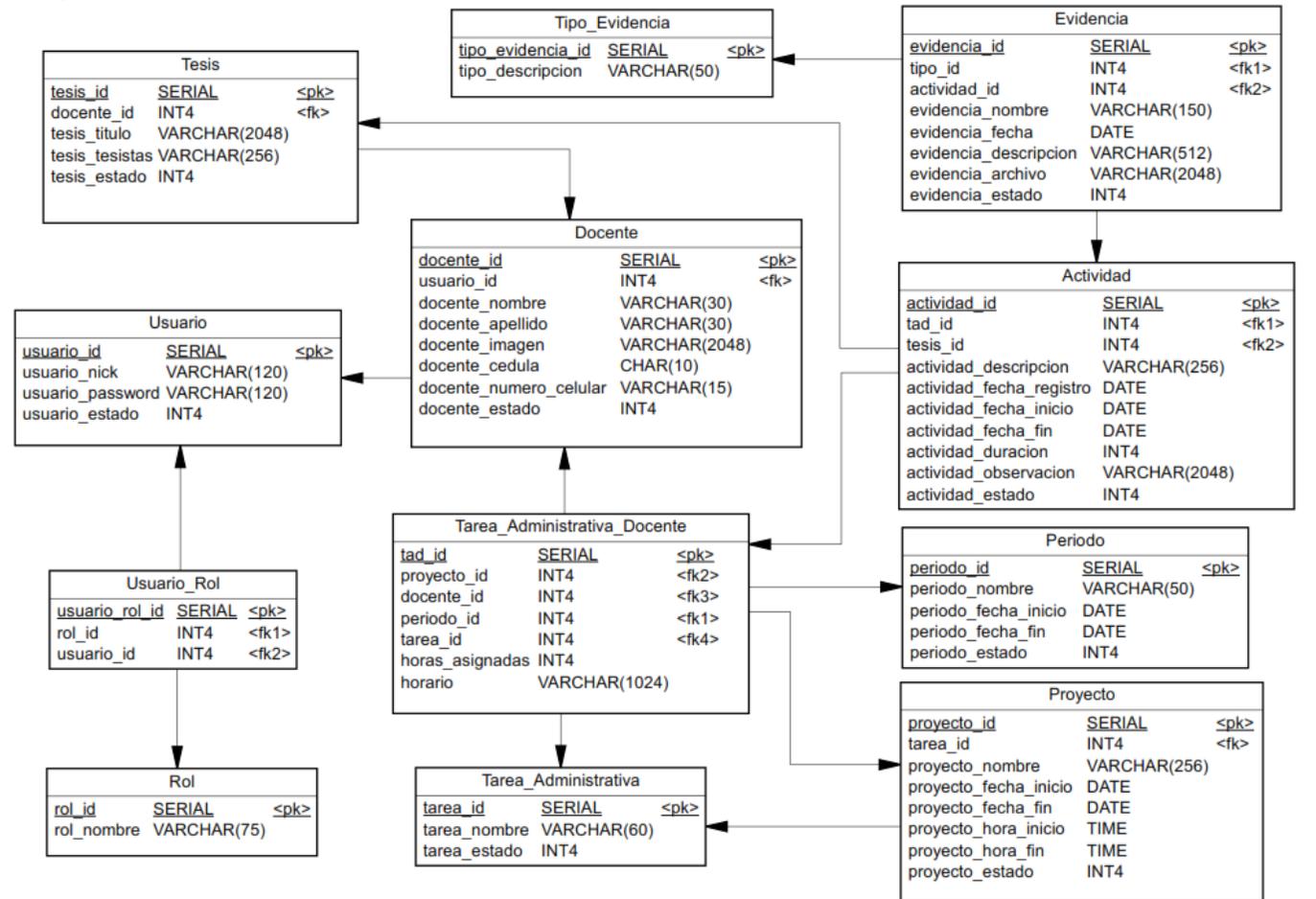


Figura 40. Diagrama físico que contiene las tablas y contendrán la información de la aplicación web

Elaborado por: Patricio Ayerve

La siguiente tabla muestra la descripción de cada una de las tablas involucradas en el diseño conceptual de la base de datos.

Tabla 20.

Descripción de las tablas de la base de datos

| Tabla | Descripción |
|------------------------------|--|
| Actividad | Almacena la información de las actividades |
| Docente | Almacena la información de docentes |
| Evidencia | Almacena la información de evidencias |
| Periodo | Almacena la información de los periodos |
| Proyecto | Almacena la información de los proyectos |
| Rol | Almacena la información de los roles |
| Tarea_Administrativa | Almacena la información de la tareas administrativas |
| Tarea_Administrativa_Docente | Almacena la información de la asignación de tareas administrativas |
| Tesis | Almacena la información de los trabajos de titulación |
| Tipo_Evidencia | Almacena la información de los tipos de evidencia |
| Usuario | Almacena la información de los usuarios |
| Usuario_Rol | Almacena la información de los roles que corresponden a cada usuario |

Nota. Descripción de tablas del diagrama físico de la base de datos

Elaborado por: Patricio Ayerve

3.3. Diccionario de datos

En esta sección se mostrará la descripción de los campos de cada tabla en la base de datos.

La tabla Actividad almacena la información de las actividades de los docentes.

Tabla 21.

Diccionario de datos tabla Actividad

| Tabla: | Actividad | |
|--------------------------|---------------------------|---|
| Campo | Tipo de datos | Descripción |
| actividad_id | Serial | Clave principal de la tabla Actividad |
| tad_id | Integer | Clave foránea, identificación única de la tabla Tarea_Administrativa_Docente. |
| tesis_id | Integer | Clave foránea, identificación única de la tabla Tesis |
| actividad_descripcion | Variable Characters(256) | Descripción de una actividad |
| actividad_fecha_registro | Date | Fecha de registro de una actividad en la base de datos |
| actividad_fecha_inicio | Date | Fecha de inicio de una actividad |
| actividad_fecha_fin | Date | Fecha de fin de una actividad |
| actividad_duracion | Integer | Duración de una actividad en minutos |
| actividad_observacion | Variable Characters(2048) | Observaciones para una actividad |
| actividad_estado | Integer | Estado de una actividad: <ul style="list-style-type: none"> • Cero (0): Inactivo • Uno(1): Activo |

Nota. Descripción de campos de la tabla Actividad

Elaborado por: Patricio Ayerve

La tabla Docente almacena la información de docentes.

Tabla 22.

Diccionario de datos tabla Docente

| Tabla: | Docente | |
|------------------------|---------------------------|--|
| Campo | Tipo de datos | Descripción |
| docente_id | Serial | Clave principal de la tabla Docente |
| usuario_id | Integer | Clave foránea, identificación única de la tabla Usuario. |
| docente_nombre | Variable Characters(30) | Nombres del docente |
| docente_apellido | Variable Characters(30) | Apellidos del docente |
| docente_imagen | Variable Characters(2048) | Nombre del archivo de imagen del docente |
| docente_cedula | Char(10) | Cédula del docente |
| docente_numero_celular | Variable Characters(15) | Número de celular del docente |
| docente_estado | Integer | Estado del docente: <ul style="list-style-type: none"> • Cero (0): Inactivo • Uno(1): Activo |

Nota. Descripción de campos de la tabla Docente

Elaborado por: Patricio Ayerve

La tabla Evidencia, almacena la información de evidencias subidas por los docentes.

Tabla 23.

Diccionario de datos tabla Evidencia

| Tabla: | Evidencia | |
|-----------------------|---------------------------|--|
| Campo | Tipo de datos | Descripción |
| evidencia_id | Serial | Clave principal de la tabla Evidencia |
| tipo_id | Integer | Clave foránea, identificación única de la tabla Tipo_Evidencia. |
| actividad_id | Integer | Clave foránea, identificación única de la tabla Actividad. |
| evidencia_nombre | Variable Characters(150) | Nombre de la evidencia |
| evidencia_fecha | Date | Fecha de registro de la evidencia |
| evidencia_descripcion | Variable Characters(512) | Descripción de la evidencia |
| evidencia_archivo | Variable Characters(2048) | Nombre del archivo de la evidencia |
| evidencia_estado | Integer | Estado de la evidencia: <ul style="list-style-type: none"> • Uno(1): Activo |

Nota. Descripción de campos de la tabla Evidencia

Elaborado por: Patricio Ayerve

La tabla Periodo, almacena la información de los periodos que son registrados por el administrador.

Tabla 24.

Diccionario de datos tabla Periodo

| Tabla: | Periodo | |
|----------------------|-------------------------|---|
| Campo | Tipo de datos | Descripción |
| periodo_id | Serial | Clave principal de la tabla Periodo |
| periodo_nombre | Variable Characters(50) | Nombre del periodo |
| periodo_fecha_inicio | Date | Fecha de inicio del periodo |
| periodo_fecha_fin | Date | Fecha de fin del periodo |
| periodo_estado | Integer | Estado del periodo: <ul style="list-style-type: none"> • Cero(0): Inactivo • Uno(1): Activo |

Nota. Descripción de campos de la tabla Periodo

Elaborado por: Patricio Ayerve

La tabla Proyecto, almacena la información de los proyectos registrados por el administrador.

Tabla 25.

Diccionario de datos tabla Proyecto

| Tabla: | Proyecto | |
|-----------------------|--------------------------|--|
| Campo | Tipo de datos | Descripción |
| proyecto_id | Serial | Clave principal de la tabla Proyecto |
| tarea_id | Integer | Clave foránea, identificación única de la tabla Tarea_Administrativa. |
| proyecto_nombre | Variable Characters(256) | Nombre del proyecto |
| proyecto_fecha_inicio | Date | Fecha de inicio del proyecto |
| proyecto_fecha_fin | Date | Fecha de fin del proyecto |
| proyecto_hora_inicio | Time | Hora inicio del proyecto |
| proyecto_hora_fin | Time | Hora fin del proyecto |
| proyecto_estado | Integer | Estado del proyecto: <ul style="list-style-type: none"> • Cero(0): Inactivo • Uno(1): Activo |

Nota. Descripción de campos de la tabla Proyecto

Elaborado por: Patricio Ayerve

La tabla Rol, almacena los roles disponibles para la asignación a cada usuario y además la información ha sido precargada al crear la base de datos.

Tabla 26.

Diccionario de datos tabla Rol

| Tabla: | Rol | |
|------------|-------------------------|---------------------------------|
| Campo | Tipo de datos | Descripción |
| rol_id | Serial | Clave principal de la tabla Rol |
| rol_nombre | Variable Characters(75) | Nombre del rol |

Nota. Descripción de campos de la tabla Rol

Elaborado por: Patricio Ayerve

La tabla Tarea_Administrativa, almacena la información de las tareas administrativas registradas por el administrador.

Tabla 27.

Diccionario de datos tabla Tarea_Administrativa

| Tabla: | Tarea_Administrativa | |
|--------------|-------------------------|--|
| Campo | Tipo de datos | Descripción |
| tarea_id | Serial | Clave principal de la tabla Tarea_Administrativa |
| tarea_nombre | Variable Characters(60) | Nombre de la tarea administrativa |
| tarea_estado | Integer | Estado de la tarea administrativa: <ul style="list-style-type: none"> • Cero(0): Inactivo • Uno(1): Activo |

Nota. Descripción de campos de la tabla Tarea_Administrativa

Elaborado por: Patricio Ayerve

La tabla Tarea_Administrativa_Docente, es un detalle de las tareas administrativas asignadas a los docentes, esta información es registrada por el administrador.

Tabla 28.

Diccionario de datos tabla Tarea_Administrativa_Docente

| Tabla: | Tarea_Administrativa_Docente | |
|-----------------|------------------------------|---|
| Campo | Tipo de datos | Descripción |
| tad_id | Serial | Clave principal de la tabla Tarea_Administrativa_Docente |
| proyecto_id | Integer | Clave foránea, identificación única de la tabla Proyecto. |
| docente_id | Integer | Clave foránea, identificación única de la tabla Docente. |
| periodo_id | Integer | Clave foránea, identificación única de la tabla Periodo. |
| tarea_id | Integer | Clave foránea, identificación única de la tabla Tarea_Administrativa. |
| horas_asignadas | Integer | Horas asignadas para la tarea administrativa |
| horario | Variable Characters(1024) | Horario en texto para la tarea administrativa |

Nota. Descripción de campos de la tabla Tarea_Administrativa_Docente

Elaborado por: Patricio Ayerve

La tabla Tesis, almacena la información de los trabajos de titulación y su respectiva asignación a un docente y es registrada por el administrador.

Tabla 29.

Diccionario de datos tabla Tesis

| Tabla: | Tesis | |
|----------------|---------------------------|---|
| Campo | Tipo de datos | Descripción |
| tesis_id | Serial | Clave principal de la tabla Tesis |
| docente_id | Integer | Clave foránea, identificación única de la tabla Docente. |
| tesis_titulo | Variable Characters(2048) | Título del trabajo de titulación |
| tesis_tesistas | Variable Characters(256) | Tesistas que desarrollan el trabajo de titulación |
| tesis_estado | Integer | Estado del trabajo de titulación: <ul style="list-style-type: none"> • Cero(0): Inactivo • Uno(1): Activo |

Nota. Descripción de campos de la tabla Tesis

Elaborado por: Patricio Ayerve

La tabla Tipo_Evidencia, almacena los diferentes tipos de evidencia para las evidencias que subirán los docentes por medio de la aplicación.

Tabla 30.

Diccionario de datos tabla Tipo_Evidencia

| Tabla: | Tipo_Evidencia | |
|-------------------|-------------------------|--|
| Campo | Tipo de datos | Descripción |
| tipo_evidencia_id | Serial | Clave principal de la tabla Tipo_Evidencia |
| tipo_descripcion | Variable Characters(50) | Nombre del tipo de evidencia |

Nota. Descripción de campos de la tabla Tipo_Evidencia

Elaborado por: Patricio Ayerve

La tabla Usuario, almacena la información de usuario y contraseña para ingresar a la aplicación y es registrada por el administrador.

Tabla 31.

Diccionario de datos tabla Usuario

| Tabla: | Usuario | |
|------------------|--------------------------|---|
| Campo | Tipo de datos | Descripción |
| usuario_id | Serial | Clave principal de la tabla Usuario |
| usuario_nick | Variable Characters(120) | Usuario para el ingreso al sistema |
| usuario_password | Variable Characters(120) | Contraseña para el ingreso al sistema |
| usuario_estado | Integer | Estado del usuario: <ul style="list-style-type: none"> • Cero(0): Inactivo • Uno(1): Activo |

Nota. Descripción de campos de la tabla Usuarios

Elaborado por: Patricio Ayerve

La tabla Usuario_Rol, almacena la información del identificador del rol para cada usuario.

Tabla 32.

Diccionario de datos tabla Usuario_Rol

| Tabla: | Usuario_Rol | |
|----------------|---------------|--|
| Campo | Tipo de datos | Descripción |
| usuario_rol_id | Serial | Clave principal de la tabla Usuario_Rol |
| usuario_id | Integer | Clave foránea, identificación única de la tabla Usuario. |
| rol_id | Integer | Clave foránea, identificación única de la tabla Rol. |

Nota. Descripción de campos de la tabla Usuario_Rol

Elaborado por: Patricio Ayerve

3.4. Construcción

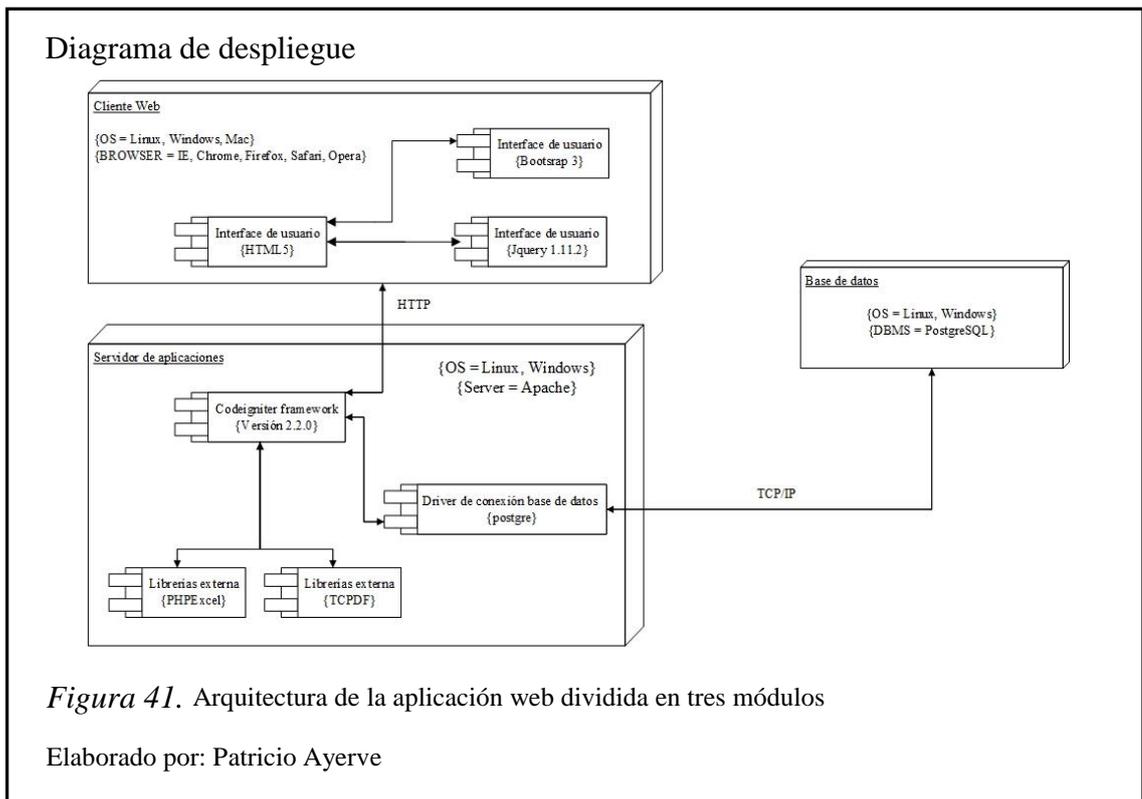
Dentro de este subcapítulo se muestra los aspectos más importantes de la programación del sistema, se explica cómo se encuentran organizadas las clases que constituyen el programa, las páginas HTML y las configuraciones más importantes.

Cabe mencionar que el sistema ha sido desarrollado bajo PHP 5.3.3, se ha utilizado el IDE de desarrollo Netbeans 8.0.2; como sistema gestor de base de datos PostgreSQL 9.1, como servidor de aplicaciones Apache Server 2.2.15 y como ruta de la aplicación la carpeta: `./var/www/html/fichas` con una dirección IP pública: 190.15.136.10, la cual permite conectarse por medio de un navegador de Internet.

3.4.1. Arquitectura

El sistema es una aplicación web que dispone de una conexión a base de datos y a la cual se accederá a través de un navegador de Internet. Se ha establecido tres módulos para el sistema:

- Cliente web
- Servidor de aplicaciones
- Base de datos



3.4.1.1. Cliente web

Será el computador de los usuarios que accederán al sistema. El sistema operativo que estará instalado puede ser Microsoft Windows, GNU/Linux, Mac OSX y el navegador de Internet podría ser Internet Explorer (versión 9 o superior), Google Chrome (versión 41 o superior), Mozilla Firefox (versión 36 o superior), Safari (versión 8.1 o superior) u Opera (versión 27 o superior), los componentes del cliente serán:

- Bootstrap 3: Se instalará dentro de la aplicación para aplicar estilos a las diferentes interfaces de usuario.
- JQuery 1.11.2: Se instalará dentro de la aplicación el framework de Javascript y sus diferentes plugins para dar interactividad a la aplicación.

3.4.1.2. Servidor de aplicaciones

Es el lugar donde se desplegará la aplicación, podrá ser un servidor Microsoft Windows, GNU/Linux o Mac OSX y el servidor que contendrá la aplicación será Apache Server 2.

La aplicación será desarrollada bajo un modelo MVC (modelo, vista, controlador) y se ha decidido utilizar componentes adicionales para obtener una aplicación más robusta. Los componentes que conformarán la aplicación son los siguientes:

- Extensiones para el fichero php.ini: se instalara las respectivas extensiones para el correcto funcionamiento del lenguaje de programación PHP.
- Fichero httpd.conf: el fichero permitirá la ejecución de rutas en las URL'S y también el acceso a la carpeta que contiene la aplicación.
- Codeigniter 2.2.0: se utilizará el framework de desarrollo bajo la tecnología PHP 5.3.3.
- Librerías externas: se utilizará las librerías PHPExcel y TCPDF para la generación de reportes.

3.4.1.3. Base de datos

El gestor de base de datos que se seleccione para almacenar la información se ha dejado abierto para que el cliente determine el que más le convenga. Igualmente para el sistema operativo se ha contemplado que sea Microsoft Windows, GNU/Linux o Mac OSX.

Tanto el servidor de aplicaciones, como la base de datos podrían funcionar sobre el mismo computador. En el desarrollo se ha optado por utilizar PostgreSQL como gestor de base de datos, la versión utilizada en el servidor de aplicaciones es la 9.1.

3.4.2. Estructura de carpetas y clases

La aplicación web se desarrollará utilizando la tecnología PHP con el framework Codeigniter 2.2.0, utilizando el patrón MVC (modelo, vista, controlador) para crear una separación lógica entre cada una de las clases que conforman el sistema.

Las clases han sido agrupadas en carpetas dependiendo de la función que cumplen, estas carpetas son las siguientes:

- application
- system
- application/controllers
- application/models
- application/views
 - application/views/administracion

- application/views/direccion
- application/views/docente

A continuación se define el contenido de las carpetas:

- Application: se encuentran todos los archivos del proyecto y es donde se realizará el desarrollo.
- System: aquí se encuentran todas las librerías del framework. No es nada recomendable modificarlos.

En la carpeta Application están las siguientes carpetas:

- Controllers: almacena los controladores de la aplicación.
- Models: almacena los modelos de la aplicación.
- Views: almacena las vistas de la aplicación y esta puede contener subcarpetas.
- Helpers: se encuentran aquellas funciones que facilitarán la escritura de código.
- Libraries: se utiliza para la creación de funciones complicadas, que involucren varias tablas o que sencillamente no sea conveniente alojarla en el modelo. La intención de esto es aislar la lógica de la aplicación con el acceso a la base de datos.
- Third_party: son las librerías descargadas para ampliar funcionalidades de PHP, como podría ser leer PDF, Excel o generarlos. Se alojarían todos aquellos plugins creados por personas ajenas a la aplicación.
- Config: se encuentran los archivos más comunes de configuración.

Cabe mencionar que en la carpeta Config se han realizado configuraciones en sus archivos tales como:

- Config.php: se ha modificado el lenguaje de la aplicación a español ya que por defecto estaba en inglés, también la clave encriptación de la información se estableció con md5, así como las variables de sesión que tendrá cada usuario al momento de conectarse a la aplicación.

Configuraciones archivo config.php

```
$config['language']      = 'spanish'; => idioma
$config['encryption_key'] = 'md5';   => tipo de encriptación
$config['sess_cookie_name'] = 'ci_session'; => variable de sesión
$config['sess_expiration'] = 1800;   => tiempo máximo de sesión 30 min.
$config['sess_expire_on_close'] = TRUE; => terminar sesión al cerrar el navegador
$config['sess_encrypt_cookie'] = TRUE; => encriptar cookies
$config['sess_use_database'] = FALSE; => base de datos para la sesión true/false
$config['sess_table_name'] = 'ci_sessions'; => nombre tabla de sesiones
$config['sess_match_ip'] = FALSE;   => guardar ip true/ false
$config['sess_match_useragent'] = TRUE; => lectura de datos de sesión
$config['sess_time_to_update'] = 300; => tiempo de actualización 300 minutos
```

Figura 42. Parámetros de configuración del fichero config.php

Elaborado por: Patricio Ayerve

- Routes.php: se ha modificado el controlador inicial por defecto el cual era index pero se cambió a login para mostrar la pantalla de autenticación del sistema.

Configuración archivo routes.php

```
$route['default_controller'] = "login";
```

Figura 43. Parámetro de configuración del fichero routes.php

Elaborado por: Patricio Ayerve

- Database.php: este archivo permite la conexión a la base de datos con los parámetros configurados mostrados en la siguiente figura.

Configuración archivo database.php

```
$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'postgres';
$db['default']['password'] = '';
$db['default']['database'] = 'FRUPS_BDD_V5';
$db['default']['dbdriver'] = 'postgre';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
$db['default']['port'] = 5432;
```

Figura 44. Parámetros de configuración del fichero database.php

Elaborado por: Patricio Ayerve

3.4.3. Configuración URL

La carpeta controllers contiene todos los controladores utilizados por la aplicación, un controlador en CodeIgniter es un archivo que contiene el código de una clase de programación orientada a objetos. Tiene un nombre cualquiera, que se asociará con una URL de la aplicación web.

Por ejemplo, esta podría ser una URL la aplicación:

`http://190.15.136.10/fichas/index.php/index_admin`

Por defecto Codeigniter incluye en sus rutas URL un index.php para el correcto funcionamiento de las rutas como se puede observar en el anterior ejemplo, pero para tener una URL más limpia se ha optado por incluir en la aplicación un archivo de configuración llamado .htaccess y modificar los permisos de escritura en el archivo httpd.conf de Apache.

Archivo .htaccess

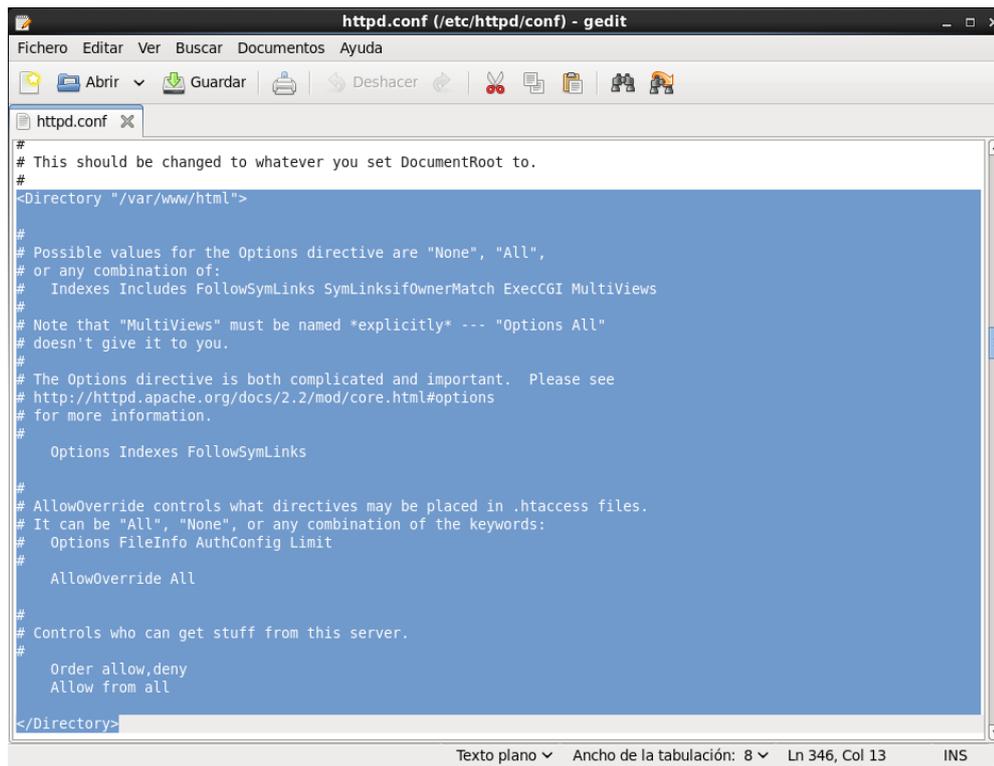
```
RewriteEngine on
RewriteCond $1 !^(index.php|resources|robots.txt)
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1 [L,QSA]
```

Figura 45. Parámetros de configuración archivo .htaccess

Elaborado por: Patricio Ayerve

El archivo .htaccess se debe incluir en la raíz del proyecto para que este sea ejecutado al iniciar la aplicación, el archivo httpd.conf permite la escritura de reglas en el servidor para lo cual se ha modificado sus permisos de escritura como muestra la siguiente figura:

Configuración archivo httpd.conf



```
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "/var/www/html">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.2/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride All
#
# Controls who can get stuff from this server.
#
Order allow,deny
Allow from all
</Directory>
```

Figura 46. Parámetros de configuración archivo httpd.conf

Elaborado por: Patricio Ayerve

La URL resultante después de aplicar los pasos anteriores es:

http://190.15.136.10/fichas/index_admin

Una vez realizada la configuración las URL para los controladores aparecen en el navegador sin index.php lo cual facilita la navegación de las páginas web.

3.4.4. Clases y métodos

En este apartado se mostrará el patrón MVC con cada una de las clases de la aplicación, cabe aclarar que cada controlador y modelo son clases, las vistas se comunican con los controladores y realizan la interacción con los usuarios.

3.4.4.1. Login

El controlador login se encarga de la autenticación a la aplicación web, utiliza el modelo login_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista login_view.

Modelo

Tabla 33.

Modelo login_model

| Archivo | Método | Descripción |
|-----------------|--------------------------------|--|
| login_model.php | login_user(username, password) | Consulta a la base de datos si el usuario y contraseña ingresados son correctos. |

Nota. Descripción de métodos de login_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 34.

Controlador login

| Archivo | Método | Descripción |
|-----------|-------------|---|
| login.php | index() | Muestra la vista login_view, si el usuario no está autenticado, caso contrario redirige a la vista correspondiente del usuario. |
| | user() | Se encarga de la autenticación a la aplicación. |
| | token() | Crea un token aleatorio a cada usuario al iniciar sesión. |
| | logout_ci() | Finaliza la sesión de los usuarios. |

Nota. Descripción de métodos de login.php

Elaborado por: Patricio Ayerve

3.4.4.2. Index_admin

El controlador index_admin, utiliza el modelo admin_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista admin_view, pertenece al módulo de administración y genera reportes.

Modelo

Tabla 35.

Modelo admin_model

| Archivo | Método | Descripción |
|-----------------|--------------------------|--|
| admin_model.php | get_dashboard(\$periodo) | Consulta a la base de datos la información del dashboard para index_admin, utiliza como parámetro el identificador del periodo activo. |
| | get_periodo() | Consulta el nombre del periodo activo. |
| | get_periodo_id() | Consulta el identificador del periodo activo. |
| | reporte_excel(\$periodo) | Consulta la información para generar el reporte de Excel. |

Nota. Descripción de métodos de admin_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 36.

Controlador index_admin

| Archivo | Método | Descripción |
|-----------------|-------------------------------|---|
| index_admin.php | index() | Muestra la vista admin_view, que es la pantalla de bienvenida del administrador y provee una condición de seguridad para que solo los usuarios con rol de Administrador puedan visualizar la información. |
| | perfil_admin() | Muestra la vista perfil_admin, que es la pantalla de perfil del administrador y pueda cambiar la información de su perfil. |
| | _create_thumbnail(\$filename) | Crea una miniatura de la imagen que el usuario haya subido en su perfil y la guarda en el servidor. |
| | generar_pdf() | Genera un PDF de la información mostrada en el Dashboard. |
| | generar_excel() | Genera un Excel de la información mostrada en el Dashboard. |
| | acerca_de() | Muestra la vista acerca_de_view, esta contiene la información de créditos del autor y director del trabajo de titulación. |
| | ayuda() | Permite descargar el manual de usuario. |

Nota. Descripción de métodos de index_admin.php

Elaborado por: Patricio Ayerve

3.4.4.3. Gestion_tareas

El controlador gestion_tareas, utiliza el modelo gtareas_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista gestion_tareas_view, pertenece al módulo de administración y genera reportes.

Modelo

Tabla 37.

Modelo gestion_tareas

| Archivo | Método | Descripción |
|-------------------|---------------------|--|
| gtareas_model.php | get_tareas() | Consulta a la base de datos la lista de tareas administrativas. |
| | get_by_id(\$tarea) | Consulta a la base de datos la información de una tarea específica por su identificador. |
| | add_tarea() | Permite insertar una nueva tarea administrativa a la base de datos. |
| | edit_tarea(\$tarea) | Permite editar una tarea administrativa de la base de datos. |
| | reporte_excel() | Consulta la información para generar el reporte de Excel. |

Nota. Descripción de métodos de gestión_tareas.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 38.

Controlador gestion_tareas

| Archivo | Método | Descripción |
|--------------------|-----------------------|--|
| gestion_tareas.php | index() | Muestra la vista gestion_tareas_view, que es la pantalla que provee una condición de seguridad para que solo los usuarios con rol de Administrador puedan visualizar la información de las tareas administrativas. |
| | mis_reglas() | Valida el formulario para agregar y modificar tareas administrativas. |
| | agregar_tarea() | Permite agregar una tarea administrativa a través de la vista agregar_tarea_view. |
| | modificar_tarea(\$id) | Permite modificar una tarea administrativa a través de la vista agregar_tarea_view. |
| | generar_pdf() | Permite generar un PDF con la información de la lista de tareas administrativas. |
| | generar_excel() | Permite generar un Excel con la información de la lista de tareas administrativas. |

Nota. Descripción de métodos de gestión_tareas.php

Elaborado por: Patricio Ayerve

3.4.4.4. Gestion_proyectos

El controlador gestion_proyectos, utiliza el modelo gproyectos_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista gestion_proyectos_view, pertenece al módulo de administración y genera reportes.

Modelo

Tabla 39.

Modelo gproyectos_model

| Archivo | Método | Descripción |
|------------------------|--|---|
| gproyectos_model.php | get_proyectos(\$id) | Consulta a la base de datos el listado de los proyectos por el tipo de tarea administrativa. |
| | get_proyectos_tipo(\$id) | Consulta el listado de los proyectos por el tipo de tarea administrativa para generar un PDF. |
| | get_nombre_tipo(\$id) | Consulta el nombre de la tarea administrativa. |
| | get_tipo_tareas() | Consulta el tipo de tareas administrativas a la base de datos. |
| | get_tipo_tareas_active() | Consulta el tipo de tareas administrativas al agregar o modificar un proyecto. |
| | get_by_id(\$id) | Consulta la información de un proyecto. |
| | pro_periodo_folder() | Consulta el nombre del periodo activo. |
| | get_folder_proyecto(\$id) | Consulta el nombre del proyecto. |
| | get_periodo_actual() | Consulta el nombre del periodo activo. |
| | add_proyecto() | Inserta un nuevo proyecto. |
| | edit_proyecto(\$id) | Modifica un proyecto. |
| reporte_excel(\$tarea) | Consulta a la base de datos la lista de proyectos para generar un Excel. | |

Nota. Descripción de métodos de gproyectos_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 40.

Controlador gestion_proyectos

| Archivo | Método | Descripción |
|-----------------------|--|--|
| gestion_proyectos.php | index() | Muestra la vista gestion_proyectos_view, que es la pantalla que provee una condición de seguridad para que solo los usuarios con rol de Administrador puedan visualizar la información de los proyectos registrados. |
| | get_data_proyectos(\$id) | Genera el listado de proyectos por tipo de tarea administrativa. |
| | mis_reglas() | Valida el formulario para agregar y modificar proyectos. |
| | confirma_inicio (\$proyecto_fecha_inicio) | Validación de fecha inicio al insertar o modificar un proyecto. |
| | confirma_fecha (\$proyecto_fecha_fin) | Validación de la fecha fin al insertar o modificar un proyecto. |
| | confirma_hora_inicio (\$proyecto_hora_inicio) | Validación de la hora al insertar o modificar un proyecto. |
| | agregar_proyecto() | Permite agregar un nuevo proyecto a través de la vista agregar_proyecto_view |
| | modificar_proyecto(\$id) | Permite modificar un proyecto a través de la vista agregar_proyecto_view |
| | crear_carpeta_proyecto (\$pro_name) | Permite la creación de una carpeta con el nombre del proyecto. |
| | editar_carpeta_proyecto (\$per, \$nom, \$newnom) | Permite modificar el nombre de la carpeta creada por el método crear_carpeta_proyecto(). |
| | generar_pdf(\$id) | Permite generar un PDF del listado de proyectos. |
| | generar_excel(\$id) | Permite generar un Excel del listado de proyectos. |

Nota. Descripción de métodos gestión_proyectos.php

Elaborado por: Patricio Ayerve

3.4.4.5. Gestion_usuarios

El controlador gestion_usuarios, utiliza el modelo gusuarios_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista gestion_usuarios_view, pertenece al módulo de administración y genera reportes.

Modelo

Tabla 41.

Modelo gusuarios_model

| Archivo | Método | Descripción |
|---------------------|---|---|
| gusuarios_model.php | get_usuarios() | Consulta la lista de usuarios. |
| | get_detalle_usuarios() | Consulta de usuarios para generar un PDF. |
| | get_rols() | Consulta los roles disponibles. |
| | get_by_id(\$id) | Consulta la información de un usuario. |
| | get_by_id_rol(\$id) | Consulta la información del rol de usuario. |
| | add_usuario() | Inserta un nuevo usuario. |
| | edit_usuario(\$id) | Modifica a un usuario. |
| | res_usuario(\$id) | Modifica la contraseña del usuario. |
| reporte_excel() | Consulta de usuarios para generar un Excel. | |

Nota. Descripción de métodos de gusuarios_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 42.

Controlador gestion_usuarios

| Archivo | Método | Descripción |
|----------------------|-------------------------|--|
| gestion_usuarios.php | index() | Muestra la vista gestion_usuarios_view, que es la pantalla que provee una condición de seguridad para que solo los usuarios con rol de Administrador puedan visualizar la información de los usuarios registrados. |
| | mis_reglas() | Valida el formulario para agregar y modificar usuarios. |
| | agregar_usuario() | Agrega un nuevo usuario. |
| | modificar_usuario(\$id) | Modifica un usuario. |
| | restablece_clave(\$id) | Restablece la contraseña del usuario. |
| | generar_pdf() | Genera un PDF de la lista de usuarios. |
| | generar_excel() | Genera un Excel de la lista de usuarios. |

Nota. Descripción de métodos de gestión_usuarios.php

Elaborado por: Patricio Ayerve

3.4.4.6. Gestion_docentes

El controlador gestion_docentes, utiliza el modelo gdocentes_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista gestion_docentes_view, pertenece al módulo de administración y genera reportes.

Modelo

Tabla 43.

Modelo gdocentes_model

| Archivo | Método | Descripción |
|---------------------------------|--|--|
| gdocentes_model.php | get_docentes() | Consulta a la base de datos la lista de docentes. |
| | get_usuario_docente(\$id) | Consulta el usuario correspondiente a de un docente. |
| | get_by_id(\$id) | Consulta la información de un docente. |
| | add_docente() | Inserta un nuevo docente. |
| | edit_docente(\$id) | Modifica un docente. |
| | get_doc_tareas(\$id) | Consulta el nombre completo del docente. |
| | get_doc_tareas_asignar(\$id) | Consulta las tareas administrativas asignadas a un docente. |
| | get_tarea_by_id(\$id) | Método que consulta la información de un docente. |
| | get_tareas() | Consulta el listado de tareas administrativas. |
| | get_proyectos(\$id) | Consulta el listado de proyectos activos. |
| | get_proyectos_lista(\$id) | Consulta el listado de tareas administrativas correspondientes a un docente. |
| | get_periodos() | Consulta los periodos activos. |
| | asignar_tarea(\$id) | Asigna tareas administrativas a un docente. |
| | editar_asignar_tarea(\$id) | Modifica tareas administrativas a un docente. |
| | remover_tarea(\$id) | Elimina una tarea administrativa asignada. |
| | docente_tarea(\$id) | Retorna el identificador único del docente. |
| | detalle_tareas(\$id) | Consulta el detalle de tareas asignadas a un docente. |
| | res_usuario(\$id) | Modifica la contraseña del docente. |
| | reporte_excel_docente() | Consulta de docentes para generar un Excel. |
| reporte_excel_tareas(\$docente) | Consulta de tareas asignadas a un docente para generar un Excel. | |

Nota. Descripción de métodos de gdocentes_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 44.

Controlador gestion_docentes

| Archivo | Método | Descripción |
|---------|-------------------------------|---|
| | index() | Muestra la vista gestion_docentes_view, esta pantalla provee una condición de seguridad para que solo los usuarios con rol de Administrador puedan visualizar la información de los docentes registrados. |
| | mis_reglas() | Valida el formulario para agregar y modificar docentes. |
| | mis_reglas_drop() | Valida el formulario para asignar y modificar tareas administrativas de los docentes. |
| | asignar_tareas(\$id) | Permite asignar tareas administrativas a los docentes. |
| | get_proyectos(\$id) | Carga la lista de proyectos disponibles por el tipo de tarea administrativa. |
| | get_proyectos_modificar(\$id) | Carga la lista de proyectos disponibles por el tipo de tarea administrativa a modificar de un docente. |
| | modificar_tarea(\$id) | Permite modificar una tarea administrativa asignada a un docente |
| | eliminar_tarea(\$id) | Elimina una tarea asignada a un docente. |

| | | |
|----------------------|---------------------------------|--|
| gestion_docentes.php | detalle_tareas(\$id) | Muestra la vista detalle_tareas_view, esta pantalla provee una condición de seguridad para que solo los usuarios con rol de Administrador puedan visualizar la información de las tareas administrativas asignadas a un docente. |
| | agregar_docente() | Permite agregar un nuevo docente. |
| | modificar_docente(\$id) | Permite modificar a un docente. |
| | restablece_clave(\$id) | Permite restablecer la contraseña de un docente. |
| | generar_pdf() | Permite generar un PDF con la lista de docente. |
| | generar_pdf_tareas(\$id) | Permite generar un PDF de las tareas administrativas asignadas a un docente. |
| | generar_excel() | Permite generar un Excel con la lista de docente. |
| | generar_excel_tareas(\$docente) | Permite generar un Excel de las tareas administrativas asignadas a un docente. |

Nota. Descripción de métodos de gestión_docente.php

Elaborado por: Patricio Ayerve

3.4.4.7. Gestion_periodos

El controlador gestion_periodos, utiliza el modelo gperiodos_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista gestion_periodos_view y pertenece al módulo de administración.

Modelo

Tabla 45.

Modelo gperiodos_model

| Archivo | Método | Descripción |
|---------------------|--------------------------|--|
| gperiodos_model.php | get_periodos() | Consulta a la base de datos el listado de periodos registrados. |
| | get_folder_periodo(\$id) | Retorna el nombre de un periodo y usa como parámetro su identificador. |
| | get_by_id(\$id) | Consulta la información de un periodo por su identificador. |
| | add_periodo() | Permite insertar un nuevo periodo en la base de datos. |
| | edit_periodo(\$id) | Permite modificar un periodo previamente registrado. |

Nota. Descripción de métodos de gperiodos_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 46.

Controlador gestion_periodos

| Archivo | Método | Descripción |
|----------------------|---|---|
| gestion_periodos.php | index() | Muestra la vista gestion_periodos_view, esta pantalla provee una condición de seguridad para que solo los usuarios con rol de Administrador puedan visualizar la información de los periodos registrados. |
| | mis_reglas() | Valida el formulario para agregar y modificar periodos. |
| | agregar_periodo() | Agrega un nuevo periodo. |
| | modifiar_periodo(\$id) | Modifica un periodo. |
| | crear_carpeta_periodo(\$nom) | Crea una carpeta para un periodo |
| | editar_carpeta_periodo(\$nom, \$newnom) | Edita la carpeta de un periodo registrado. |

Nota. Descripción de métodos gestión_periodos.php

Elaborado por: Patricio Ayerve

3.4.4.8. Gestion_tesis

El controlador gestion_tesis, utiliza el modelo gtesis_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista gestion_tesis_view, pertenece al módulo de administración y genera reportes.

Modelo

Tabla 47.

Modelo gtesis_model

| Archivo | Método | Descripción |
|------------------|------------------|---|
| gtesis_model.php | get_tesis() | Consulta a la base de datos el listado de trabajos de titulación registrados. |
| | get_docentes() | Consulta el nombre completo de un docente. |
| | get_by_id(\$id) | Consulta la información de un trabajo de titulación por su identificador. |
| | add_tesis() | Inserta un nuevo trabajo de titulación en la base de datos. |
| | edit_tesis(\$id) | Permite modificar un trabajo de titulación. |
| | reporte_excel() | Consulta el listado de trabajos de titulación y para generar un Excel. |

Nota. Descripción de métodos de gtesis_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 48.

Controlador gestion_tesis

| Archivo | Método | Descripción |
|-------------------|-----------------------|--|
| gestion_tesis.php | index() | Muestra la vista gestion_tesis_view, esta pantalla provee una condición de seguridad para que solo los usuarios con rol de Administrador puedan visualizar la información de los trabajos de titulación registrados. |
| | mis_reglas() | Valida el formulario para agregar y modificar trabajos de titulación. |
| | agregar_tesis() | Permite agregar un nuevo trabajo de titulación. |
| | modificar_tesis(\$id) | Permite modificar un trabajo de titulación existente. |
| | generar_pdf() | Permite generar un documento PDF del listado de trabajos de titulación. |
| | generar_excel() | Permite generar un documento de Excel del listado de trabajos de titulación. |

Nota. Descripción de métodos de gestión_tesis

Elaborado por: Patricio Ayerve

3.4.4.9. Tipos_evidencia

El controlador tipos_evidencia, utiliza el modelo tevidencia_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista tipos_evidencia_view y pertenece al módulo de administración.

Modelo

Tabla 49.

Modelo tevidencia_model

| Archivo | Método | Descripción |
|----------------------|------------------|---|
| tevidencia_model.php | get_evidencias() | Consulta a la base de datos el listado de tipos de evidencia disponibles. |
| | get_by_id(\$id) | Consulta la información de un tipo de evidencia por su identificador. |
| | add_tipo() | Inserta un nuevo tipo de evidencia en la base de datos. |
| | edit_tipo(\$id) | Modifica un tipo de evidencia. |

Nota. Descripción de métodos de tevidencia_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 50.

Controlador tipos_evidencia

| Archivo | Método | Descripción |
|---------------------|----------------------|---|
| tipos_evidencia.php | index() | Muestra la vista tipos_evidencia_view, esta pantalla provee una condición de seguridad para que solo los usuarios con rol de Administrador puedan visualizar la información de los tipos de evidencia registrados |
| | mis_reglas() | Valida el formulario para agregar y modificar tipos de evidencia. |
| | agregar_tipo() | Permite agregar un nuevo tipo de evidencia. |
| | modificar_tipo(\$id) | Permite modificar un tipo de evidencia. |

Nota. Descripción de métodos de tipos_evidencia.php

Elaborado por: Patricio Ayerve

3.4.4.10. Index_docente

El controlador index_docente, utiliza el modelo docente_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista detalle_global_view, pertenece al módulo de docentes y genera la ficha de responsabilidad.

Para el caso de los docentes se ha optado por utilizar un único modelo para las consultas en la base de datos y no generar demasiadas clases.

Modelo

Tabla 51.

Modelo docente_model

| Archivo | Método | Descripción |
|---------|----------------------------------|---|
| | periodo_id() | Consulta el identificador del periodo activo. |
| | periodo_nombre() | Consulta el nombre del periodo activo. |
| | verifica_periodo() | Verifica si existe un solo periodo activo. |
| | tesis_habilitar(\$id) | Token para habilitar el botón de trabajos de titulación en la página principal del docente. |
| | get_doc_tareas_asignar(\$id) | Consulta el nombre completo del docente. |
| | info_docente(\$id) | Consulta la información del perfil del docente. |
| | detalle_tareas(\$id, \$pa) | Consulta las tareas administrativas asignadas a un docente en el periodo activo. |
| | top_ten_actividades(\$id, \$per) | Consulta las diez últimas actividades del docente. |
| | consulta_grafico(\$id, \$per) | Consulta la información para el gráfico de barras. |
| | detalle_actividades(\$id, \$per) | Consulta las actividades del docente dentro de sus tareas administrativas asignadas. |
| | ruta_carpeta_proyecto(\$id) | Devuelve la ruta de la carpeta para cada proyecto. |
| | info_evidencia(\$actividad) | Devuelve la información de la evidencia por el identificador de actividad. |

| | | |
|--|---|---|
| docente_model.php | desc_tarea_proyecto(\$id) | Devuelve el nombre del proyecto, el nombre de la tarea administrativa y el estado del proyecto. |
| | desc_tesis_tesistas(\$id) | Devuelve la información de un trabajo de titulación. |
| | detalle_actividades_tesis(\$id) | Devuelve el listado de actividades de un docente por trabajo de titulación. |
| | detalle_actividades_tesis_pdf(\$id) | Consulta las actividades en los trabajos de titulación para generar un PDF. |
| | add_actividad(\$id) | Permite insertar una actividad para una tarea administrativa asignada. |
| | consulta_tesis(\$id) | Consulta el listado de trabajos de titulación asignados a un docente. |
| | add_actividad_tesis(\$id) | Inserta una actividad para un trabajo de titulación asignado. |
| | subir_perfil(\$cel, \$img, \$id) | Permite actualizar el perfil del docente. |
| | subir_perfil_cel(\$cel, \$id) | Permite actualizar el perfil del docente, pero solamente su número de celular. |
| | act_actividad(\$id) | Modifica el estado de una actividad. |
| | des_actividad(\$id) | Modifica el estado de una actividad. |
| | comprueba_evidencia(\$id) | Permite verificar evidencias subidas o no. |
| | evidencia_data(\$id) | Consulta la información de una evidencia. |
| | tar_pro_act(\$id) | Devuelve el nombre, tesistas, actividades de un trabajo de titulación. |
| | t_pro_act(\$id) | Devuelve el nombre del proyecto, tareas y actividades de una tarea administrativa. |
| | tipo_evidencia() | Consulta los tipos de evidencia disponibles. |
| | add_evidencia_tesis(\$id, \$archivo) | Inserta una evidencia por actividad de un trabajo de titulación y el nombre del archivo. |
| | add_evidencia_tareas(\$id, \$archivo) | Inserta una evidencia por actividad por tarea administrativa y el nombre del archivo. |
| | evidencia_archivo(\$id) | Retorna el nombre del archivo de una evidencia. |
| | tareas_ficha(\$id) | Consulta la información de las actividades por tarea administrativa de un docente. |
| tareas_administrativas(\$per, \$sus, \$id) | Consulta las tareas administrativas asignadas a un docente. | |

Nota. Descripción de métodos de docente_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 52.

Controlador index_docente

| Archivo | Método | Descripción |
|-------------------|----------------|--|
| index_docente.php | index() | Muestra la vista detalle_global_view, esta pantalla provee una condición de seguridad para que solo los usuarios con rol de Docente puedan visualizar la su información. |
| | data_grafico() | Es el encargado de obtener la información necesaria para generar el grafico de barras. |
| | perfil_docente | Muestra la vista perfil_docente_view, que es la pantalla de perfil del docente y pueda cambiar la información de su perfil |

| | | |
|--|-------------------------------|--|
| | _create_thumbnail(\$filename) | Crea una miniatura de la imagen que el usuario haya subido en su perfil y la guarda en el servidor. |
| | genera_ficha() | Genera la ficha de responsabilidad del docente. |
| | acerca_de() | Muestra la vista acerca_de_view, esta contiene la información de créditos del autor y director del trabajo de titulación |
| | ayuda() | Permite descargar el manual de usuario. |

Nota. Descripción de métodos de index_docente.php

Elaborado por: Patricio Ayerve

3.4.4.11. Tareas_docente

El controlador tareas_docente, utiliza el modelo docente_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista tareas_docente_view, pertenece al módulo de docentes.

Controlador

Tabla 53.

Controlador tareas_docente

| Archivo | Método | Descripción |
|--|--|--|
| tareas_docente.php | index | Muestra la vista tareas_docente_view, esta pantalla provee una condición de seguridad para que solo los usuarios con rol de Docente puedan visualizar la su información. |
| | mis_reglas() | Valida el formulario para agregar actividades. |
| | confirma_inicio (\$actividad_fecha_inicio) | Valida las fecha de una actividad. |
| | confirma_fin (\$actividad_fecha_fin) | Valida la fecha de inicio de una actividad. |
| | detalle_actividades(\$id) | Muestra el detalle de actividades de las tareas administrativas asignadas. |
| | agregar_actividad(\$id) | Permite agregar una actividad |
| | act_actividad(\$id, \$stad) | Activa una actividad. |
| | des_actividad(\$id, \$stad) | Desactiva una actividad. |
| | evidencia_tareas (\$id, \$starea, \$estado) | Muestra la pantalla para agregar una evidencia por medio de la vista agregar_evidencia_tareas_view, |
| descarga_evidencia (\$actividad, \$starea) | Permite descargar la evidencia de una actividad. | |

Nota. Descripción de métodos de tareas_docente.php

Elaborado por: Patricio Ayerve

3.4.4.12. Tesis_docente

El controlador tesis_docente, utiliza el modelo docente_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista tesis_docente_view, pertenece al módulo de docentes.

Controlador

Tabla 54.

Controlador tesis_docente

| Archivo | Método | Descripción |
|-------------------|---|---|
| tesis_docente.php | index | Muestra la vista tesis_docente_view, esta pantalla provee una condición de seguridad para que solo los usuarios con rol de Docente puedan visualizar la su información. |
| | mis_reglas() | Valida el formulario para agregar actividades. |
| | confirma_inicio (\$actividad_fecha_inicio) | Valida las fecha de una actividad. |
| | confirma_fin (\$actividad_fecha_fin) | Valida la fecha de inicio de una actividad. |
| | detalle_actividades(\$id) | Muestra el detalle de actividades de las tareas administrativas asignadas. |
| | agregar_actividad(\$id) | Permite agregar una actividad |
| | act_actividad_tesis(\$id, \$tesis) | Activa una actividad. |
| | des_actividad_tesis(\$id, \$tesis) | Desactiva una actividad. |
| | evidencia_tesis (\$id, \$tesis, \$estado) | Muestra la pantalla para agregar una evidencia por medio de la vista agregar_evidencia_tesis_view, |
| | descarga_evidencia (\$actividad) | Permite descargar la evidencia de una actividad. |
| | genera_ficha(\$id) | Permite generar un documento PDF con las actividades por trabajo de titulación. |

Nota. Descripción de métodos de tesis_docente.php

Elaborado por: Patricio Ayerve

3.4.4.13. Index_direccion

El controlador index_direccion, utiliza el modelo docente_model detallado anteriormente y dirección_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza la vista de dircarrera_view, pertenece al módulo de Dirección de Carrera y genera reportes por actividad, docentes y trabajos de titulación.

Para el caso de la Dirección de Carrera se ha optado por utilizar dos modelos para las consultas en la base de datos y no generar demasiadas clases.

Modelo

Tabla 55.

Modelo direccion_model

| Archivo | Método | Descripción |
|-----------------------------|---|---|
| direccion_model.php | periodo_id() | Consulta el identificador del periodo activo. |
| | periodo_nombre() | Consulta el nombre del periodo activo. |
| | verifica_perido() | Verifica el periodo activo. |
| | info_docente(\$id) | Consulta la información del director para su perfil. |
| | detalle_tareas(\$pa) | Consulta las tareas administrativas asignadas en el periodo activo. |
| | detalle_tareas_docente(\$pa) | Consulta las tareas administrativas por docente en el periodo activo. |
| | detalle_tesis_docente() | Consulta los trabajos de titulacion asignadas a los docentes. |
| | consulta_grafico(\$per) | Consulta la información para generar el grafico de barras. |
| | detalle_actividades(\$id, \$per) | Consulta las actividades por tarea administrativa de cada docente. |
| | ruta_carpeta_proyecto(\$id) | Retorna la ruta de la carpeta que contiene las evidencias de cada docente. |
| | desc_tarea_proyecto(\$id) | Consulta el nombre del proyecto, el nombre de la tarea y el estado del proyecto. |
| | desc_tesis_tesistas(\$id) | Consulta el título, tesistas, estado de los trabajos de titulación. |
| | detalle_actividades_tesis(\$id) | Consulta el detalle de actividades por trabajos de titulación. |
| | consulta_tesis(\$id) | Consulta el detalle de los trabajos de titulación. |
| | subir_perfil(\$cel, \$img, \$id) | Permite actualizar el perfil del docente. |
| | subir_perfil_cel(\$cel, \$id) | Permite actualizar el perfil del docente, pero solamente su número de celular. |
| | comprueba_evidencia(\$id) | Permite verificar las evidencias subidas y no subidas. |
| | evidencia_data(\$id) | Consulta la información de una evidencia. |
| | evidencia_archivo(\$id) | Consulta el nombre del archivo de una evidencia. |
| | tareas_ficha(\$id) | Consulta las tareas administrativas para generar un PDF. |
| | tareas_administrativas(\$per, \$us, \$id) | Consulta las tareas administrativas consolidadas para generar un PDF. |
| | actividades_tareas_docente(\$id) | Consulta las tareas administrativas de un docente para genera un PDF. |
| | actividades_tesis_docente(\$id) | Consulta las actividades de los trabajos de titulación para generar un PDF. |
| | get_doc_tareas_asignar(\$id) | Consulta el nombre completo del docente. |
| | get_doc_tesis_asignar(\$id) | Consulta el nombre completo del docente. |
| | reporte_actividades_tarea() | Consulta para genera un archivo Excel con el listado de tareas administrativas asignadas. |
| | reporte_actividades_docente() | Consulta las actividades de los docentes para generar un archivo Excel. |
| reporte_actividades_tesis() | Consulta las actividades por trabajos de titulación y genera una archivo Excel, | |

Nota. Descripción de métodos de direccion_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 56.

Controlador *index_direccion*

| Archivo | Método | Descripción |
|---------------------|---|---|
| index_direccion.php | index() | Muestra la vista <i>dircarrera_view</i> , esta pantalla provee una condición de seguridad para que solo los usuarios con rol de Dirección de Carrera puedan visualizar la su información. |
| | data_grafico() | Permite la creación del gráfico de barras. |
| | perfil_director() | Muestra la vista <i>perfil_dir_view</i> , que es la pantalla de perfil del director y pueda cambiar la información de su perfil |
| | _create_thumbnail(\$filename) | Crea una miniatura de la imagen que el usuario haya subido en su perfil y la guarda en el servidor. |
| | detalle_actividades_docente(\$id) | Permite desplegar la vista <i>detalles_docente_view</i> y mostrar el listado de actividades de un docente. |
| | detalle_tesis_docente(\$id) | Permite desplegar la vista <i>detalles_tesis_view</i> y mostrar el listado de actividades de los trabajos de titulación. |
| | genera_ficha(\$usuario) | Permite generar un documento PDF con la ficha de responsabilidad del docente. |
| | genera_ficha_tesis(\$tesis) | Permite generar un documento PDF con la ficha de actividades por trabajo de titulación del docente. |
| | pdf_actividades_tareas() | Permite generar un documento PDF de las tareas administrativas asignadas. |
| | pdf_actividades_docente() | Permite generar un documento PDF por docente con sus tareas administrativas asignadas. |
| | pdf_actividades_tesis() | Permite generar un documento PDF de las actividades por trabajo de titulación. |
| | pdf_detalle_actividades_tareas(\$id) | Permite generar un documento PDF con el detalle de las actividades de un docente. |
| | pdf_detalle_actividades_tesis(\$id) | Permite generar un documento PDF con el detalle de las actividades de un docente por trabajo de titulación. |
| | excel_actividades_tareas() | Permite generar un archivo Excel de las tareas administrativas asignadas. |
| | excel_actividades_docente() | Permite generar un archivo Excel por docente con sus tareas administrativas asignadas. |
| | excel_actividades_tesis() | Permite generar un archivo Excel de las actividades por trabajo de titulación. |
| | evidencia_tareas(\$id, \$tarea, \$user) | Muestra la pantalla para descargar una evidencia por medio de la vista <i>descargar_evidencia_tareas_view</i> , |
| | evidencia_tesis(\$id, \$tesis) | Muestra la pantalla para descargar una evidencia por medio de la vista <i>descargar_evidencia_tesis_view</i> , |
| | acerca_de | Muestra la vista <i>acerca_de_view</i> , esta contiene la información de créditos del autor y director del trabajo de titulación |
| ayuda() | Permite descargar el manual de usuario. | |

Nota. Descripción de métodos *index_direccion.php*

Elaborado por: Patricio Ayerve

3.4.4.14. Cambio_password

El controlador cambio_password, utiliza el modelo gpass_model para generar las consultas SQL a la base de datos, la interacción con el usuario lo realiza las vistas cambio_password_admin_view, cambio_password_doc_view, y finalmente la vista cambio_password_dir_view, cada una por el rol de Administrador, Docente y Dirección de Carrera.

Modelo

Tabla 57.

Modelo gpass_model

| Archivo | Método | Descripción |
|-----------------|---|--|
| gpass_model.php | get_current_password(\$usuario) | Consulta la contraseña actual del usuario. |
| | cambiar_password(\$usuario, \$nuevo_password) | Modifica la contraseña del usuario. |

Nota. Descripción de métodos de gpass_model.php

Elaborado por: Patricio Ayerve

Controlador

Tabla 58.

Controlador cambio_password

| Archivo | Método | Descripción |
|---------------------|---|---|
| cambio_password.php | index() | Invoca al método modificar_password y provee una verificación de seguridad. |
| | mis_reglas() | Valida el formulario para cambiar de contraseña. |
| | modificar_password(\$id) | Permite cambiar la contraseña del usuario por una nueva. |
| | vista_cambio_password() | Permite mostrar la vista respectiva para cada rol de usuario. |
| | comprueba_password(\$usuario_password) | Verifica si la contraseña a cambiar es correcta. |
| | confirma_password(\$usuario_password_confirmar) | Verifica si la nueva contraseña coincide para modificar la anterior contraseña. |

Nota. Descripción de métodos de cambio_password.php

Elaborado por: Patricio Ayerve

3.4.5. Estructura de las vistas

Una vista dentro de Codeigniter es una página web o un fragmento de ella, como un encabezado, un pie de página, una barra lateral, las vistas pueden ser flexiblemente embebidas dentro de otras vistas.

Las vistas nunca son llamadas directamente, ellas deben ser cargadas por un controlador.

En la siguiente tabla se detalla cada una de las páginas que conforman el sistema.

Autenticación

Tabla 59.

Vistas autenticación

| Módulo | Página | Descripción |
|---------------|----------------|--|
| Autenticación | login_view.php | Página de autenticación para el sistema. |

Nota. Descripción de la vista de autenticación

Elaborado por: Patricio Ayerve

Administración

Tabla 60.

Vistas administración

| Módulo | Página | Descripción |
|--------------------|--------------------------------------|---|
| Administración | admin_view.php | Página de bienvenida del administrador. |
| | gestion_tareas_view.php | Página para visualizar las tareas administrativas registradas. |
| | agregar_tarea_view.php | Página para agregar o modificar una tarea administrativa. |
| | gestion_proyectos_view.php | Página para visualizar los proyectos registrados. |
| | agregar_proyecto_view.php | Página para agregar o modificar un proyecto. |
| | gestion_usuarios_view.php | Página para visualizar los usuarios registrados en el sistema. |
| | agregar_usuario_view.php | Página para agregar o modificar un usuario. |
| | gestion_docentes_view.php | Página para visualizar los docentes registrados. |
| | agregar_docente_view.php | Página para agregar o modificar un docente. |
| | detalle_tareas_view.php | Página para visualizar las tareas administrativas asignadas a los docentes. |
| | asignar_tareas_view.php | Página para asignar las tareas administrativas a los docentes. |
| | modificar_tareas_view.php | Página para modificar las tareas administrativas asignadas a los docentes. |
| | gestion_periodos_view.php | Página para visualizar los periodos registrados. |
| | agregar_periodo_view.php | Página para agregar o modificar un periodo. |
| | gestion_tesis_view.php | Página para visualizar los trabajos de titulación asignados. |
| | agregar_tesis_view.php | Página para agregar o modificar un trabajo de titulación. |
| | tipos_evidencia_view.php | Página para visualizar los tipos de evidencia registrados. |
| | agregar_tipo_view.php | Página para agregar o modificar los tipos de evidencia. |
| | perfil_admin_view.php | Página para visualizar el perfil del administrador. |
| | cambio_password_admin_view.php | Página para cambiar la clave de usuario. |
| acerca_de_view.php | Página para visualizar los créditos. | |

Nota. Descripción de las vistas para la administración

Elaborado por: Patricio Ayerve

Docentes

Tabla 61.

Vistas docentes

| Módulo | Página | Descripción |
|----------|-------------------------------------|--|
| Docentes | detalle_global_view.php | Página de bienvenida del docente. |
| | tareas_docente_view.php | Página para visualizar las tareas administrativas asignadas a un docente. |
| | tesis_docente_view.php | Página para visualizar los trabajos de titulación de los docentes. |
| | detalle_actividades_view.php | Página para visualizar las actividades registradas por tarea administrativa. |
| | agregar_actividad_view.php | Página para agregar una actividad por tarea administrativa. |
| | agregar_evidencia_tareas_view.php | Página para agregar evidencias por cada actividad de una tarea administrativa. |
| | descargar_evidencia_tareas_view.php | Página para descargar evidencias por cada actividad de una tarea administrativa. |
| | detalle_actividades_tesis_view.php | Página para visualizar las actividades registradas por trabajo de titulación. |
| | agregar_actividad_tesis_view.php | Página para agregar una actividad trabajo de titulación |
| | agregar_evidencia_tesis_view.php | Página para agregar evidencias por cada actividad de un trabajo de titulación. |
| | descargar_evidencia_tesis_view.php | Página para descargar evidencias por cada actividad de un trabajo de titulación. |
| | perfil_docente_view.php | Página para visualizar el perfil del docente. |
| | cambio_password_doc_view.php | Página para cambiar la clave de usuario. |
| | acerca_de_view.php | Página para visualizar los créditos. |

Nota. Descripción de vistas para los docentes

Elaborado por: Patricio Ayerve

Dirección de Carrera

Tabla 62.

Vistas dirección de carrera

| Módulo | Página | Descripción |
|----------------------|-------------------------------------|---|
| Dirección de Carrera | dircarrera_view.php | Página de bienvenida de la Dirección de Carrera. |
| | detalles_docente_view.php | Página para visualizar las actividades de los docentes por cada tarea administrativa. |
| | detalles_tesis_view.php | Página para visualizar las actividades de los docentes por trabajo de titulación. |
| | descargar_evidencia_tareas_view.php | Página para descargar la evidencia de cada docente por actividad. |
| | descargar_evidencia_tesis_view.php | Página para descargar la evidencia de cada trabajo de titulación por actividad. |
| | perfil_dir_view.php | Página para visualizar el perfil del docente. |
| | cambio_password_dir_view.php | Página para cambiar la clave de usuario. |
| | acerca_de_view.php | Página para visualizar los créditos. |

Nota. Descripción de vistas para la dirección de carrera

Elaborado por: Patricio Ayerve

3.5. Pruebas

Una vez concluida la construcción de la aplicación se la sometió a dos tipos de pruebas: contra requerimientos y pruebas de rendimiento.

3.5.1. Pruebas de caja negra

Se realizó las pruebas con cada uno de los usuarios del sistema para verificar sus requerimientos funcionales, requerimientos específicos y otros requisitos.

Administrador

Tabla 63.

Pruebas de caja negra administrador

| No. | Prueba | Historia | Pasos | OK | Observaciones |
|-----|-------------------------------------|----------|---|----|--|
| 1 | Ingreso del usuario al sistema | 1 | 1. Ingresar a 190.15.136.10/fichas/login 2. Ingresar usuario y contraseña. 3. Presionar el botón <i>Ingresar</i> . | SI | |
| 2 | Cambio de contraseña | 2 | 1. Acceder a la opción de <i>Cambiar contraseña</i> , ubicado en el menú de usuario. 2. Ingresar la contraseña actual, la nueva contraseña y la confirmación. 3. Presionar el botón <i>Guardar</i> si desea el cambio caso contrario presionar <i>Cancelar</i> . | SI | La contraseña por defecto es el correo del usuario. |
| 3 | Modificar perfil | 2 | 1. Acceder a la opción de <i>Perfil</i> , ubicado en el menú de usuario. 2. Ingresar un número de celular y una imagen. 3. Presionar el botón <i>Guardar</i> si desea el cambio caso contrario presionar <i>Cancelar</i> . | SI | El archivo para subir debe ser una imagen. |
| 4 | Registro de usuarios | 3 | 1. Acceder a la opción de <i>Gestión de usuarios</i> , ubicado en el menú lateral izquierdo. 2. Presionar el botón <i>Agregar usuario</i> . 3. Ingresar la información. 4. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 5 | Registro de docentes | 3 | 1. Acceder a la opción de <i>Gestión de docentes</i> , ubicado en el menú lateral izquierdo. 2. Presionar el botón <i>Agregar docente</i> . 3. Ingresar la información. 4. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 6 | Registro de tareas administrativas. | 4 | 1. Acceder a la opción de <i>Gestión de tareas</i> , ubicado en el menú lateral izquierdo. 2. Presionar el botón <i>Agregar tarea</i> . 3. Ingresar la información. 4. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| | | | 1. Acceder a la opción de <i>Gestión de proyectos</i> , ubicado en el menú lateral izquierdo. | | 1. Cada proyecto debe tener un tipo de tarea administrativa. |

| | | | | | |
|----|---|-------|---|----|---|
| 7 | Registro de proyectos. | 6 | 2. Presionar el botón <i>Agregar proyecto</i> . 3. Ingresar la información. 4. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | 2. Al registrar un nuevo proyecto se crea una carpeta con el nombre del mismo dentro de la carpeta del periodo activo en el servidor. |
| 8 | Registro de tipos de evidencia para archivos. | 8 | 1. Acceder a la opción de <i>Tipos de evidencia</i> , ubicado en el menú lateral izquierdo. 2. Presionar el botón <i>Nuevo tipo de evidencia</i> . 3. Ingresar la información. 4. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 9 | Registro de periodos | 9 | 1. Acceder a la opción de <i>Gestión de periodos</i> , ubicado en el menú lateral izquierdo. 2. Presionar el botón <i>Agregar periodo</i> . 3. Ingresar la información. 4. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | Al registrar un nuevo periodo se crea una carpeta con el nombre del mismo en el servidor. |
| 10 | Registro de trabajos de titulación. | 10 | 1. Acceder a la opción de <i>Trabajos de titulación</i> , ubicado en el menú lateral izquierdo. 2. Presionar el botón <i>Agregar trabajo</i> . 3. Ingresar la información. 4. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 11 | Asignar tareas administrativas a docente. | 5 y 7 | 1. Acceder a la opción de <i>Gestión de docentes</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de docentes 3. Presionar el botón <i>Asignar tareas</i> . 4. Ingresar la información. 5. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 12 | Restablecer contraseña usuario. | 3 | 1. Acceder a la opción de <i>Gestión de usuarios</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de usuarios. 3. Presionar el botón <i>Restablecer clave</i> . 4. Aparece un mensaje de notificación. | SI | Al restablecer la contraseña del usuario, el valor de la misma es su correo electrónico. |
| 13 | Restablecer contraseña docente. | 3 | 1. Acceder a la opción de <i>Gestión de docentes</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de docentes. 3. Presionar el botón <i>Restablecer clave</i> . 4. Aparece un mensaje de notificación. | SI | Al restablecer la contraseña del docente, el valor de la misma es su correo electrónico. |
| 14 | Modificar usuarios | 3 | 1. Acceder a la opción de <i>Gestión de usuarios</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de usuarios. 3. Presionar el botón <i>Editar</i> . 4. Ingresar la información. 5. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 15 | Modificar docentes | 3 | 1. Acceder a la opción de <i>Gestión de docentes</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de docentes. | SI | |

| | | | | | |
|----|--|-------|--|----|--|
| | | | 3. Presionar el botón <i>Editar</i> . 4. Ingresar la información. 5. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | | |
| 16 | Modificar tareas administrativas. | 4 | 1. Acceder a la opción de <i>Gestión de tareas</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de tareas administrativas. 3. Presionar el botón <i>Editar</i> . 4. Ingresar la información. 5. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 17 | Modificar proyectos. | 6 | 1. Acceder a la opción de <i>Gestión de proyectos</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de proyectos. 3. Presionar el botón <i>Editar</i> . 4. Ingresar la información. 5. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | La carpeta delo proyecto cambia de nombre al ser modificada. |
| 18 | Modificar tipos de evidencia para archivos. | 8 | 1. Acceder a la opción de <i>Tipos de evidencia</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de tipos de evidencia. 3. Presionar el botón <i>Editar</i> . 4. Ingresar la información. 5. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 19 | Modificar periodos | 9 | 1. Acceder a la opción de <i>Gestión de periodos</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de periodos. 3. Presionar el botón <i>Editar</i> . 4. Ingresar la información. 5. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | La carpeta del periodo cambia de nombre al ser modificada. |
| 20 | Modificar trabajos de titulación. | 10 | 1. Acceder a la opción de <i>Trabajos de titulación</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de trabajos de titulación. 3. Presionar el botón <i>Editar</i> . 4. Ingresar la información. 5. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 21 | Modificar tareas administrativas asignadas a un docente. | 5 y 7 | 1. Acceder a la opción de <i>Gestión de docentes</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de docentes 3. Presionar el botón <i>Asignar tareas</i> . 4. Revisar las tareas administrativas asignadas. 5. Presionar el botón <i>Editar</i> . 6. Ingresar la información. 7. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 22 | Eliminar tareas administrativas | 5 y 7 | 1. Acceder a la opción de <i>Gestión de docentes</i> , ubicado en el menú lateral izquierdo. 2. Revisar la lista de docentes 3. Presionar el botón <i>Asignar tareas</i> . | SI | |

| | | | | | |
|----|------------------------------|--|--|----|--|
| | asignadas a un docente. | | 4. Revisar las tareas administrativas asignadas. 5. Presionar el botón <i>Eliminar</i> . 6. Confirmar o cancelar. | | |
| 23 | Exportar tablas detalladas. | | 1. Acceder al menú <i>Acciones</i> 2. Presionar el botón <i>Generar PDF</i> . 3. Presionar el botón <i>Generar Hoja de Cálculo</i> . | SI | Genera un documento PDF o Excel de la información requerida. |
| 24 | Descargar manual de usuario. | | 1. Presionar en la opción de <i>Ayuda</i> , ubicado en el menú de usuario. 2. Se descarga el manual de usuario. | SI | Permite la descarga del manual de administración en formato PDF. |

Nota. Descripción de pruebas de caja negra para la administración de la aplicación web

Elaborado por: Patricio Ayerve

Docente

Tabla 64.

Pruebas de caja negra docente

| No. | Prueba | Historia | Pasos | OK | Observaciones |
|-----|--|----------|--|----|---|
| 1 | Ingreso del usuario al sistema | 1 | 1. Ingresar a 190.15.136.10/fichas/login 2. Ingresar usuario y contraseña. 3. Presionar el botón <i>Ingresar</i> . | SI | |
| 2 | Cambio de contraseña | 2 | 1. Acceder a la opción de <i>Cambiar contraseña</i> , ubicado en el menú de usuario. 2. Ingresar la contraseña actual, la nueva contraseña y la confirmación. 3. Presionar el botón <i>Guardar</i> si desea el cambio caso contrario presionar <i>Cancelar</i> . | SI | La contraseña por defecto es el correo del usuario. |
| 3 | Modificar perfil | 2 | 1. Acceder a la opción de <i>Perfil</i> , ubicado en el menú de usuario. 2. Ingresar un número de celular y una imagen. 3. Presionar el botón <i>Guardar</i> si desea el cambio caso contrario presionar <i>Cancelar</i> . | SI | El archivo para subir debe ser una imagen. |
| 4 | Descargar manual de usuario. | | 1. Presionar en la opción de <i>Ayuda</i> , ubicado en el menú de usuario. 2. Se descarga el manual de usuario. | SI | Permite la descarga del manual de docente en formato PDF. |
| 5 | Visualizar tareas administrativas asignadas. | | 1. Presionar el botón <i>Tareas Administrativas</i> . 2. Se carga una tabla con las tareas administrativas asignadas. | SI | |
| 6 | Detalle de actividades. | | 1. Presionar el botón <i>Detalle</i> . 2. Se carga una tabla con el detalle de las actividades registradas. | SI | |
| 7 | Activar/desactivar actividad. | | 1. Presionar el botón <i>Activa</i> , para activar la actividad o el botón <i>Inactiva</i> para inhabilitarla. | SI | |
| 8 | Agregar actividad. | | 1. Presionar el botón <i>Agregar actividad</i> . 2. Ingresar la información 3. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | SI | |
| 9 | Agregar evidencia. | | 1. Presionar el botón <i>Subir evidencia</i> . 2. Ingresar la información | SI | Se sube un archivo al servidor. |

| | | | | | |
|----|-----------------------------|--|--|----|---|
| | | | 3. Presionar el botón <i>Guardar</i> si desea guardar caso contrario presionar <i>Cancelar</i> . | | |
| 10 | Descargar evidencia. | | 1. Presionar el botón <i>Descargar evidencia</i> . 2. Visualizar información 3. Presionar el link de descarga del archivo. | SI | Se descarga el archivo en el ordenador del usuario. |
| 11 | Exportar tablas detalladas. | | 1. Acceder al menú <i>Acciones</i> 2. Presionar el botón <i>Generar PDF</i> . 3. Presionar el botón <i>Generar Hoja de Cálculo</i> . | SI | Genera una ficha de actividades en PDF. |

Nota. Descripción de pruebas de caja negra para los docentes en la aplicación web

Elaborado por: Patricio Ayerve

Dirección de Carrera

Tabla 65.

Pruebas de caja negra Dirección de Carrera

| No. | Prueba | Historia | Pasos | OK | Observaciones |
|-----|--------------------------------|----------|--|----|--|
| 1 | Ingreso del usuario al sistema | 1 | 1. Ingresar a 190.15.136.10/fichas/login 2. Ingresar usuario y contraseña. 3. Presionar el botón <i>Ingresar</i> . | SI | |
| 2 | Cambio de contraseña | 2 | 1. Acceder a la opción de <i>Cambiar contraseña</i> , ubicado en el menú de usuario. 2. Ingresar la contraseña actual, la nueva contraseña y la confirmación. 3. Presionar el botón <i>Guardar</i> si desea el cambio caso contrario presionar <i>Cancelar</i> . | SI | La contraseña por defecto es el correo del usuario. |
| 3 | Modificar perfil | 2 | 1. Acceder a la opción de <i>Perfil</i> , ubicado en el menú de usuario. 2. Ingresar un número de celular y una imagen. 3. Presionar el botón <i>Guardar</i> si desea el cambio caso contrario presionar <i>Cancelar</i> . | SI | El archivo para subir debe ser una imagen. |
| 4 | Descargar manual de usuario. | | 1. Presionar en la opción de <i>Ayuda</i> , ubicado en el menú de usuario. 2. Se descarga el manual de usuario. | SI | Permite la descarga del manual de Dirección de Carrera en formato PDF. |
| 5 | Descargar evidencias. | | 1. Presionar el botón <i>Descargar evidencia</i> . 2. Visualizar información 3. Presionar el link de descarga del archivo. | SI | Se descarga los archivos en el ordenador del usuario. |
| 6 | Exportar tablas detalladas. | | 1. Acceder al menú <i>Acciones</i> 2. Presionar el botón <i>Generar PDF</i> . 3. Presionar el botón <i>Generar Hoja de Cálculo</i> . | SI | Genera un reporte en PDF. |

Nota. Descripción de pruebas de caja negra para la dirección de carrera en la aplicación web

Elaborado por: Patricio Ayerve

3.5.2. Pruebas de rendimiento

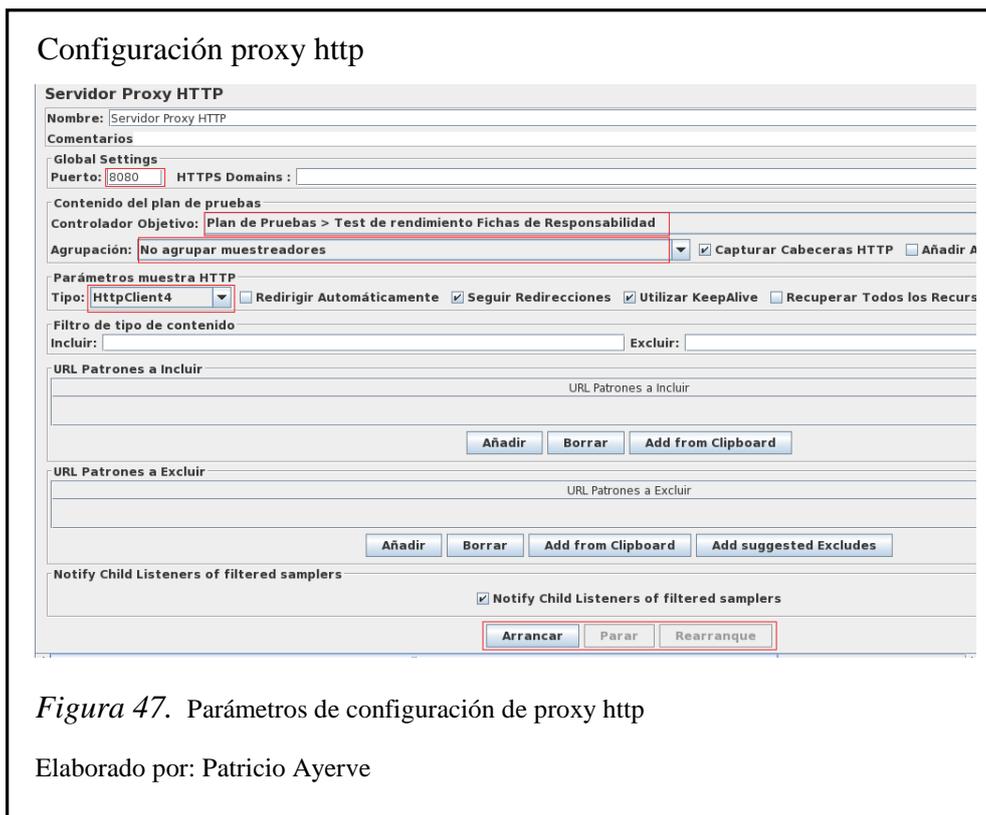
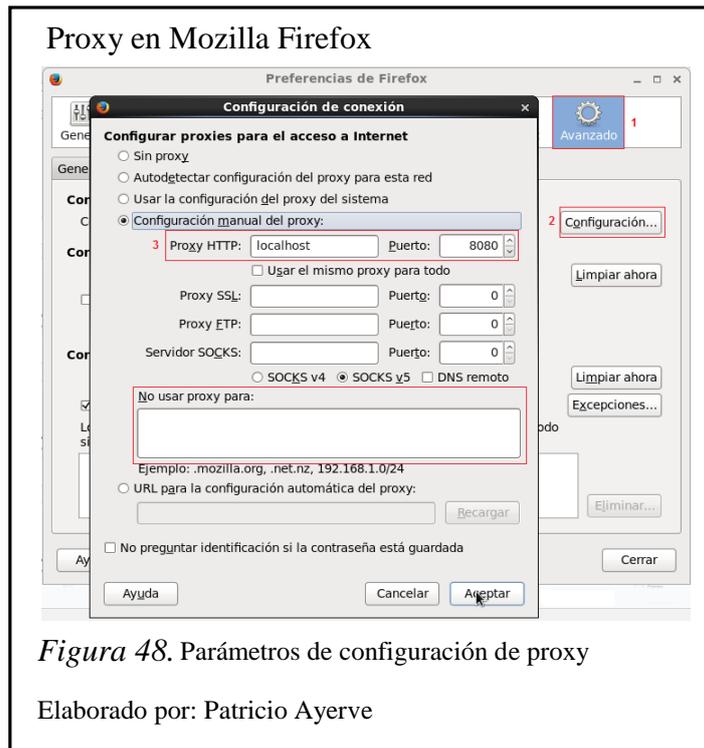
Para someter al sistema a pruebas de rendimiento y conocer la capacidad de respuesta, se realizará una carga con 50 usuarios. Las pruebas serán realizadas en un computador con las siguientes características:

- Procesador Intel Core i7
- CPU 2.60 GHz
- Memoria RAM 8 Gb

A continuación se detallan los pasos realizados para ejecutar las pruebas.

1. Agregar un grupo de hilos para indicar el número de usuarios concurrentes y de repeticiones que posteriormente ejecutarán la navegación, esta será grabada para ejecutar las pruebas. En plan de pruebas hacer clic con el botón derecho del ratón y seleccionar la opción *Añadir » Hilos (Usuarios) » Grupo de hilos*.
2. La aplicación usa cookies, agregar un gestor de cookies al grupo de hilos, para ello seleccionar el grupo de hilos, hacer clic con el botón derecho y seleccionar *Añadir » Elemento de configuración » Gestor de cookies HTTP*.
3. Agregar un gestor de cabeceras al grupo de hilos, para ello seleccionar el grupo de hilos, hacer clic con el botón derecho y seleccionar *Añadir » Elemento de configuración » Gestor de cabecera HTTP*.
4. Agregar un gestor de caché al grupo de hilos, para ello seleccionar el grupo de hilos, hacer clic con el botón derecho y seleccionar *Añadir » Elemento de configuración » Gestor de la cache HTTP*.
5. Agregar los informes para tratar los resultados obtenidos al ejecutar el test de la siguiente manera:
 - *Añadir » Receptor » Gráfico de resultados*.
 - *Añadir » Receptor » Ver resultados en árbol*.
6. Agregar el servidor proxy HTTP, para ello seleccionar el *Banco de trabajo* y luego hacer clic en *Añadir » Elementos NodePrueba » Servidor Proxy HTTP*

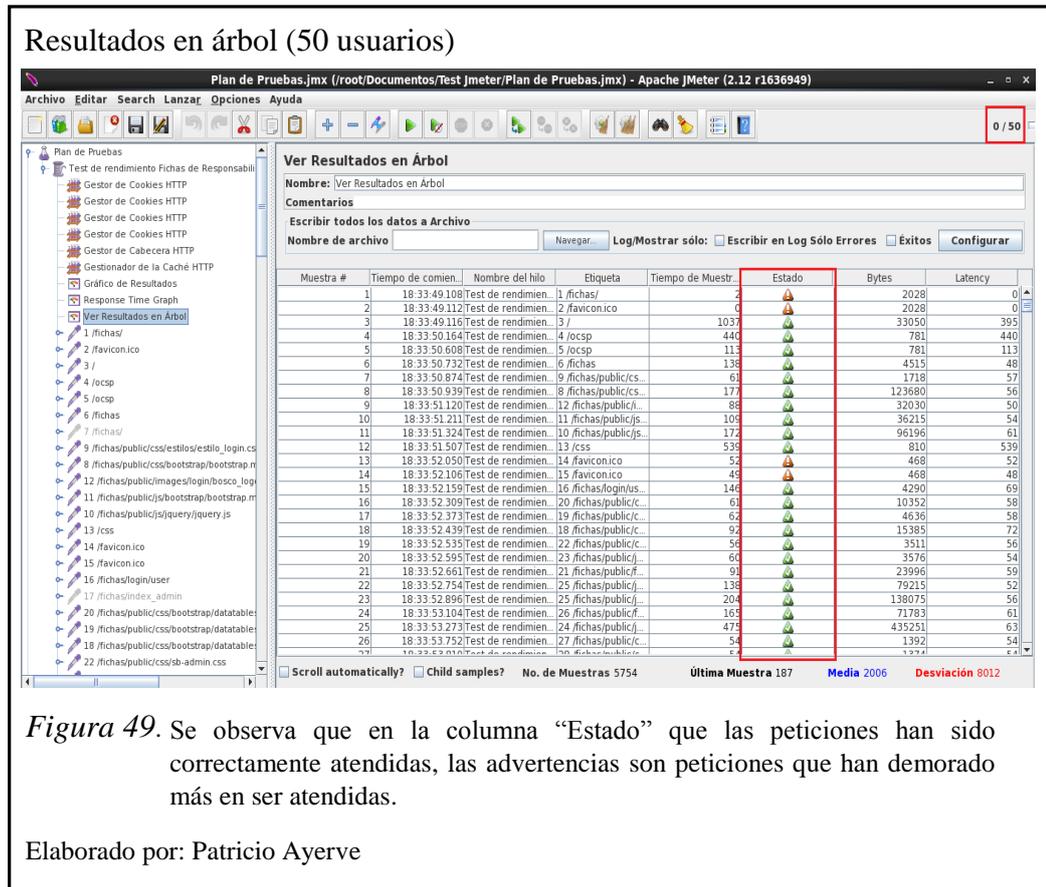
El proxy grabará todas las peticiones HTTP que se realicen al navegador, para ello hacer las configuraciones que se muestran en las siguientes figuras:



Prueba de rendimiento con 50 usuarios concurrentes, con 30 usuarios el rendimiento ha sido superior a continuación se muestra los resultados de las pruebas.

Análisis de Resultados

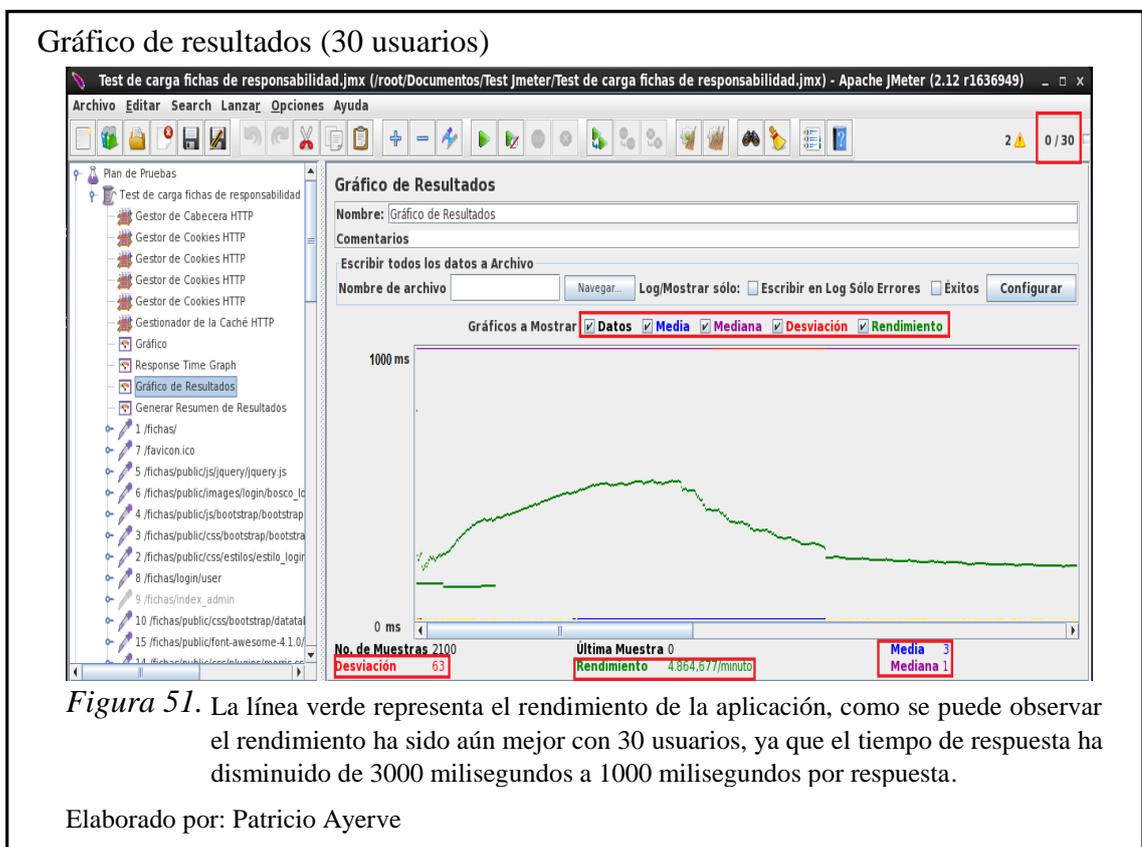
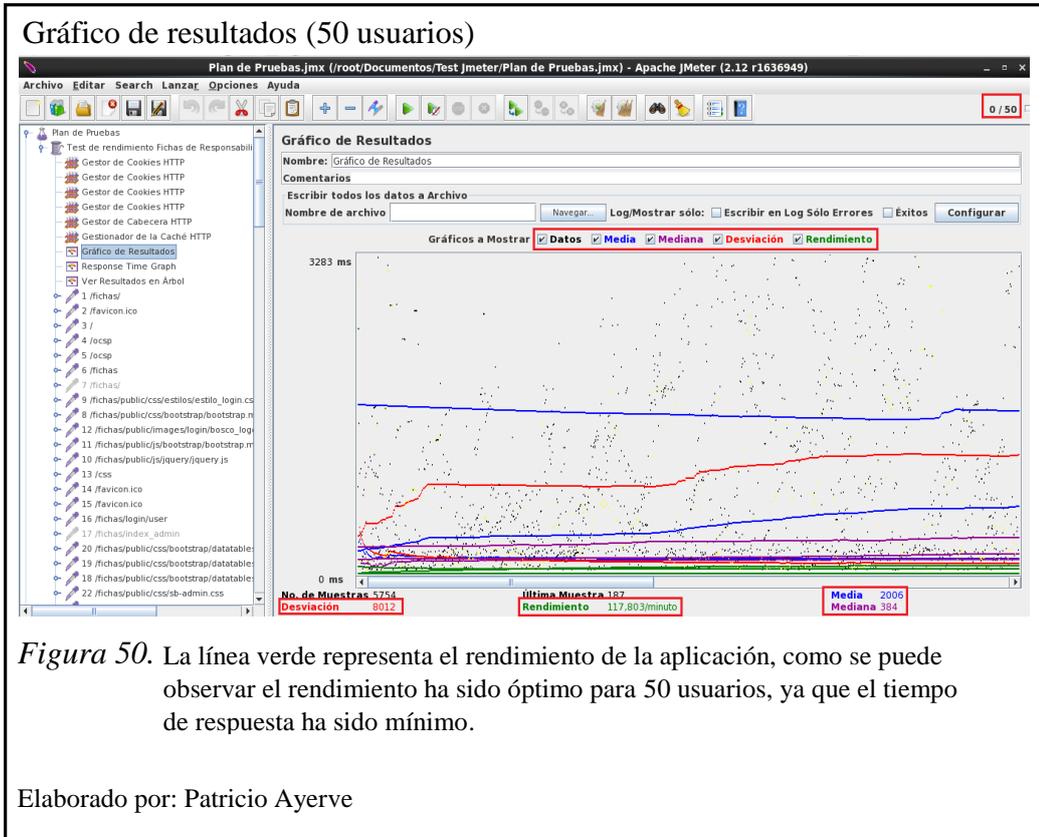
Mediante “Resultados en árbol” se determinó que la aplicación soporta una carga para 50 usuarios correctamente, sin embargo se observa que algunas respuestas son más lentas, luego retoma el tiempo normal aunque vuelve a retardar en algunos usuarios debido a la latencia.



Para los resultados obtenidos con el “Gráfico de resultados” se observa que el rendimiento aumenta según el número de usuarios, sin embargo también la desviación aumenta por lo que crece el tiempo de respuesta a cada petición. Por el conjunto de resultados obtenidos la aplicación soporta 50 usuarios concurrentes sin ningún problema, luego sigue aunque sus respuestas son más lentas, sin embargo no deja de funcionar a continuación la interpretación de los datos de la gráfica.

- Datos - Trama de color negro, muestra el tiempo consumido por los datos.
- Promedio - Trama de color azul, es la media actual de todas las muestras.
- Mediana - Trama de color violeta, indica el valor medio actual de las muestras
- Desviación - Trama de color rojo, indica la desviación estándar.

- Rendimiento - Trama de color verde, indica el rendimiento de las muestras por minuto.



CAPÍTULO 4

IMPLEMENTACIÓN

Para la implementación del sistema fue necesario realizar configuraciones en el servidor de aplicaciones y determinar que extensiones de PHP podrían configurarse, para publicar la aplicación web se utilizó el cliente FileZilla. En los siguientes apartados se explica la configuración realizada.

4.1. Características del servidor

A continuación se muestran las características del servidor de aplicaciones:

- Sistema Operativo: CentOS release 6.6 (Final) 64 bits.
- Kernel: 2.6.32-431.3.1.el6.x86_64 SMP Fri Jan 3 21:39:27 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux.
- PostgreSQL 9.1.14 on x86_g4
- Versión server: Apache/2.2.15 (Unix) built: Oct 16 2014 14:48:21.
- PHP 5.3.3 (cli) (built: Oct 30 2014 20:12:53) Copyright (c) 1997-2010 The PHP Group Zend Engine v2.3.0, Copyright (c) 1998-2010 Zend Technologies.

4.2. Configuración de la base de datos

A continuación se muestra los pasos para la creación y restauración de la base de datos para el sistema:

1. Crear la base de datos en el servidor de aplicaciones con el siguiente comando.

```
Creación de la base de datos
[root@server-pc ~]# su - postgres
Contraseña para el usuario postgres:
-bash-4.1$ psql(9.1.14)
-bash-4.1$ create database FRUPS_BDD_V5;
-bash-4.1$ CREATE DATABASE
```

Figura 52. Comandos para la creación de la base de datos

Elaborado por: Patricio Ayerve

2. Restaurar la base de datos desde un script SQL con el siguiente comando :

Restauración base de datos

```
[root@server-pc fichas]# psql -U postgres -d FRUPS_BDD_V5 -f /var/www/html/fichas/FRUPS_BDD.sql
```

Figura 54. Comandos para la restauración de la base de datos

Elaborado por: Patricio Ayerve

Ubicación archivo de restauración

```
[root@server-pc fichas]# ls -la
. application docentes_images FRUPS_BDD.sql include license.txt system
. ayuda fichas responsabilidad .htaccess index.php public tesis
[root@server-pc fichas]#
```

Figura 53. Archivo de restauración de la base de datos

Elaborado por: Patricio Ayerve

3. Una vez realizado los pasos anteriores se procede a verificar dentro del DBMS.

Base de datos restaurada

The screenshot displays the PostgreSQL Enterprise Manager interface. On the left, the 'Object browser' shows a tree view of the database structure, with 'FRUPS_BDD_V5' selected. The main pane shows the 'Properties' tab for the selected database, listing various configuration parameters such as Name, OID, Owner, ACL, Tablespace, and Collation. Below the properties, the 'SQL pane' shows the SQL commands used to create the database, including a DROP statement and a CREATE DATABASE statement with specific settings for encoding, tablespace, collation, and character type.

| Property | Value |
|----------------------|--------------|
| Name | FRUPS_BDD_V5 |
| OID | 16779 |
| Owner | postgres |
| ACL | |
| Tablespace | pg_default |
| Default tablespace | pg_default |
| Encoding | UTF8 |
| Collation | es_ES.UTF-8 |
| Character type | es_ES.UTF-8 |
| Default schema | public |
| Default table ACL | |
| Default sequence ACL | |
| Default function ACL | |
| Allow connections? | Yes |
| Connected? | Yes |
| Connection limit | -1 |
| System database? | No |
| Comment | |

```
-- Database: "FRUPS_BDD_V5"
-- DROP DATABASE "FRUPS_BDD_V5";
CREATE DATABASE "FRUPS_BDD_V5"
WITH OWNER = postgres
ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'es_ES.UTF-8'
LC_CTYPE = 'es_ES.UTF-8'
CONNECTION LIMIT = -1;
```

Figura 55. Archivo restaurado dentro del DBMS de PostgreSQL

Elaborado por: Patricio Ayerve

Finalmente la base de datos ha sido configurada correctamente, ahora la aplicación podrá comunicarse con ella.

4.3. Instalación

La instalación de la aplicación web se realiza con el cliente Filezilla, la Dirección de Carrera ha provisto de un usuario y contraseña con permisos para depositar los archivos necesarios para ejecutar del sistema. A continuación se detalla la configuración realizada.

1. Autenticar con el servidor por medio de Filezilla, con la dirección IP del servidor, puerto 21, ingresar el usuario y contraseña y conectar.

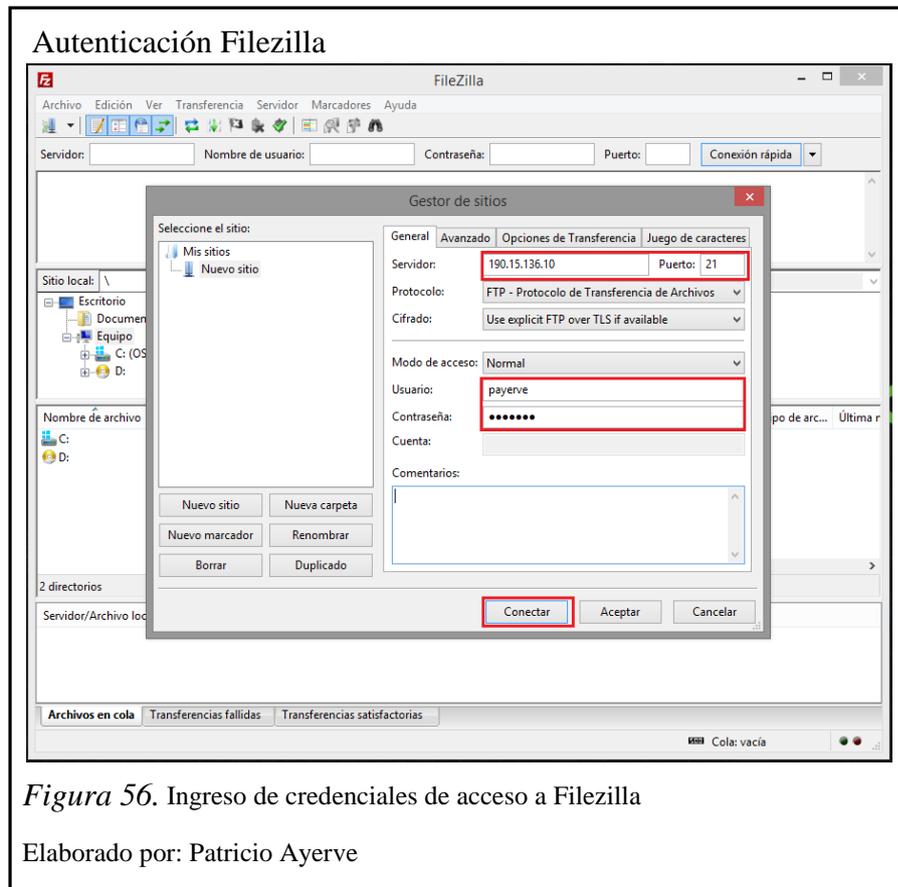
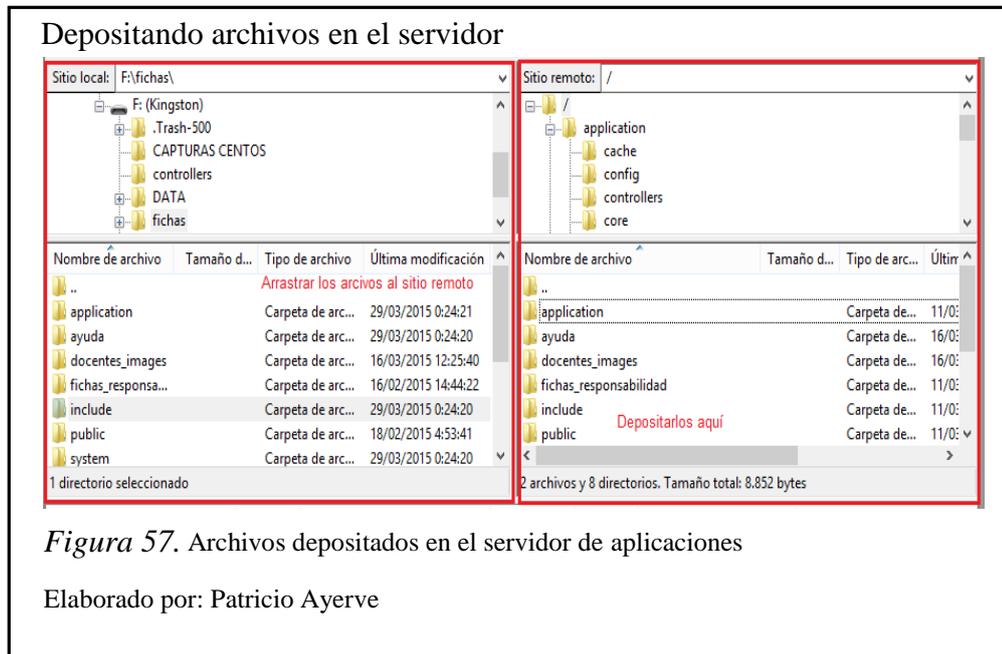


Figura 56. Ingreso de credenciales de acceso a Filezilla

Elaborado por: Patricio Ayerve

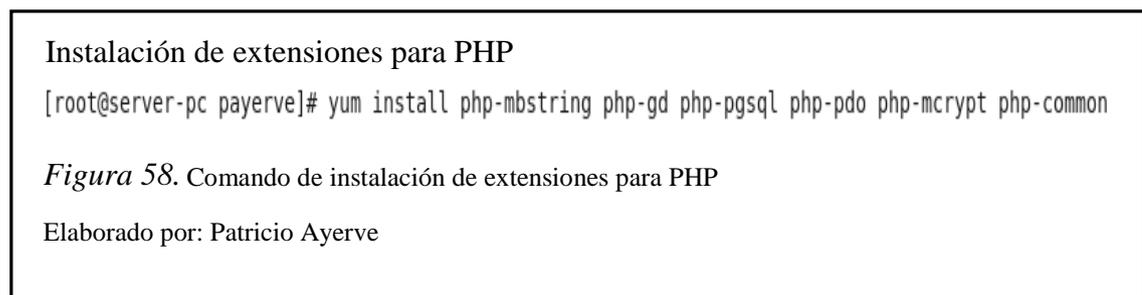
2. La ubicación del directorio en el cual se depositaran los archivos es: `./var/www/html/fichas`, desde el ordenador buscar la ruta que contiene los archivos y arrastrarlos a la carpeta del servidor.



4.4. Configuración

En este apartado se especifican las configuraciones necesarias para la puesta a producción de la aplicación web.

Instalar las extensiones para PHP 5.3, las cuales proveen de librerías y funcionalidad a las clases del framework Codeigniter con el siguiente comando:



Finalmente reiniciar el servidor y escribir en el navegador: <http://190.15.136.10/fichas>

Reinicio del servidor

```
[root@server-pc payerve]# service httpd restart  
Parando httpd: [ OK ]  
Iniciando httpd: [ OK ]  
[root@server-pc payerve]#
```

Figura 60. Comando de reinicio del servidor de producción

Elaborado por: Patricio Ayerve

Página de autenticación

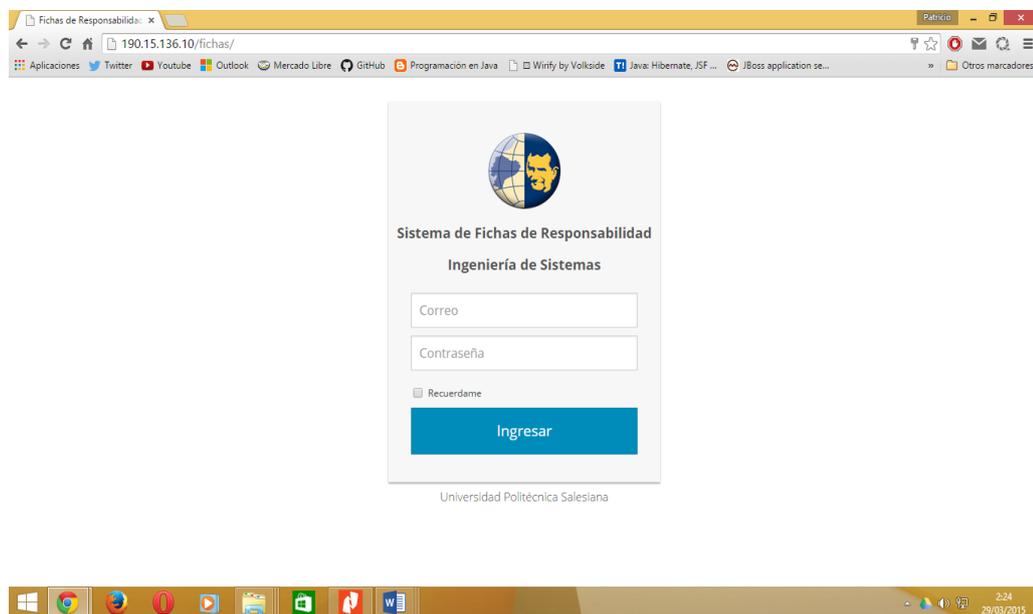


Figura 59. La aplicación web está funcionando en el servidor de producción

Elaborado por: Patricio Ayerve

4.5. Evidencias de funcionamiento

Una vez finalizada la fase de desarrollo e implementación se ha obtenido un sistema para la gestión de las fichas de responsabilidad. A continuación se muestra algunas capturas del sistema en ejecución.

Dashboard administración

| No. | Docente | Proyecto | Tarea | Fecha Inicio | Fecha Fin | Hora Inicio | Hora Fin |
|-----|--------------------------------------|---------------------------------------|--|--------------|------------|-------------|----------|
| 1 | Manolo Acosta Muñoz | Desarrollo y análisis de evaluaciones | Comisión de evaluaciones objetivas (Reactivos) | 2015-03-02 | 2015-03-09 | 10:00:00 | 13:00:00 |
| 2 | Manolo Acosta Muñoz | Test Proyecto académico 3 | Aula virtual | 2015-02-18 | 2015-02-18 | 03:22:00 | 03:22:00 |
| 3 | Manolo Acosta Muñoz | Test Proyecto académico 2 | Acreditación | 2015-02-18 | 2015-02-18 | 03:22:00 | 03:22:00 |
| 4 | Ricardo Humberto Albarracín Zambrano | Test Proyecto académico 5 | Acreditación | 2015-02-23 | 2015-02-27 | 07:00:00 | 11:30:00 |
| 5 | Alfonso René Arevalo Campos | Organización de aulas virtuales | Aula virtual | 2015-03-23 | 2015-03-24 | 12:40:00 | 13:40:00 |

Mostrando 5 de 10 registros

Primera Anterior 1 2 Siguiete Última

Elaborado por: Patricio Ayerve

Figura 61. Dashboard de administración con datos

Elaborado por: Patricio Ayerve

Gestión de proyectos

| No. | Nombre del Proyecto | Fecha de Inicio | Fecha Fin | Hora de Inicio | Hora Fin | Opciones |
|-----|---------------------------|-----------------|------------|----------------|----------|----------|
| 1 | Test Proyecto académico | 2015-02-18 | 2015-02-18 | 03:22:00 | 03:22:00 | [Icon] |
| 2 | Test Proyecto académico 2 | 2015-02-18 | 2015-02-18 | 03:22:00 | 03:22:00 | [Icon] |
| 3 | Test Proyecto académico 5 | 2015-02-23 | 2015-02-27 | 07:00:00 | 11:30:00 | [Icon] |
| 4 | Test proyecto académico 6 | 2015-02-23 | 2015-02-26 | 07:30:00 | 21:30:00 | [Icon] |

Mostrando 4 de 4 registros

Primera Anterior 1 Siguiete Última

+ Agregar proyecto

Elaborado por: Patricio Ayerve

Figura 62. Datos en gestión de proyectos del administrador

Elaborado por: Patricio Ayerve

Gestión de docentes

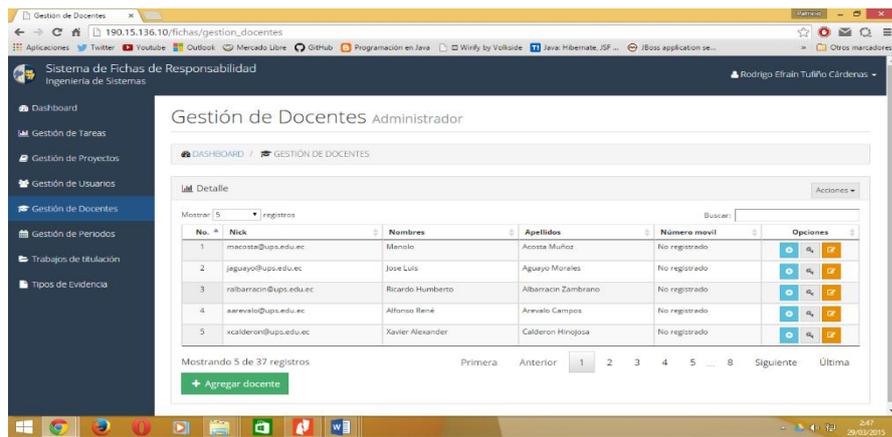


Figura 63. Datos de docentes registrados en la aplicación web

Elaborado por: Patricio Ayerve

Reporte de tareas administrativas asignadas

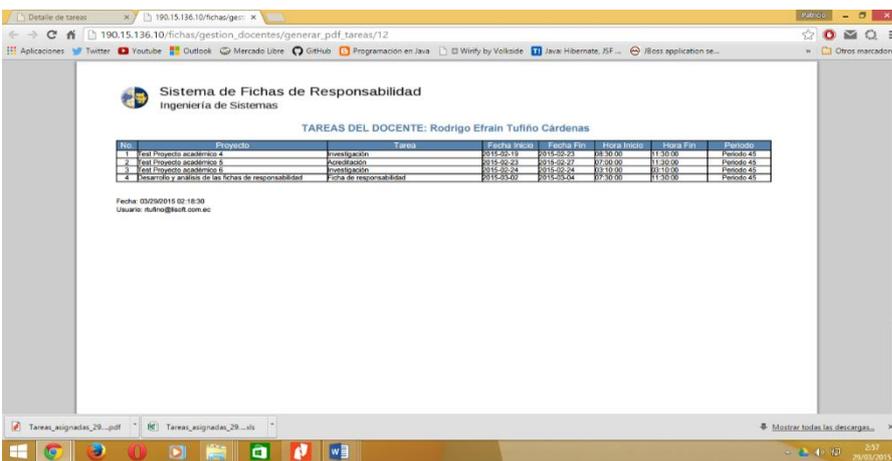


Figura 64. Reportes de tareas administrativas asignadas a docentes

Elaborado por: Patricio Ayerve

Dashboard docentes

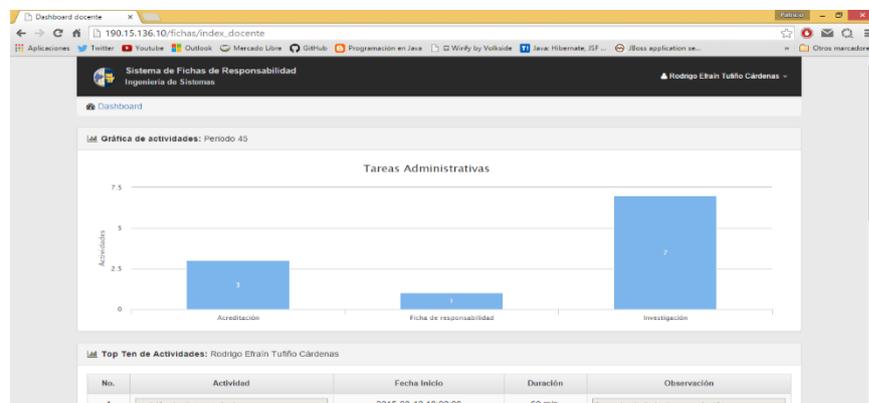


Figura 65. Gráfico de barras de docentes

Elaborado por: Patricio Ayerve

CONCLUSIONES

- El haber realizado un adecuado análisis de requerimientos permitió diseñar, codificar e implementar una aplicación web que se ajusta a las necesidades de la Dirección de Carrera.
- El lenguaje PHP permitió un rápido desarrollo en conjunto con el framework Codeigniter, ya que la curva de aprendizaje ha sido mínima y se logró obtener la aplicación web para la gestión de las fichas de responsabilidad.
- La aplicación web permite obtener reportes que facilitarán la toma de decisiones de la Dirección de Carrera y en base a esta información gestionar correctamente la distribución de la carga horaria de cada docente.
- La metodología XP aplicada en el diseño y construcción del sistema permitió un rápido desarrollo de los módulos sin invertir demasiado tiempo en la documentación ya que se entregaban prototipos funcionales que cambiaron sobre la marcha y han pasado por un proceso de pruebas que garantiza su correcto funcionamiento.

RECOMENDACIONES

- Durante la realización de este sistema, se lanzó una nueva versión de Codeigniter por lo que se recomienda sea probada para mantener actualizada la aplicación y aprovechar las nuevas características que provee el framework.
- Para una nueva versión de la aplicación se recomienda integrarla con una aplicación móvil nativa, ya que mejoraría el tiempo de respuesta del servidor al no cargar demasiados recursos en el cliente.
- Evaluar de manera periódica el uso del sistema e implementar nuevas funcionalidades que permitan dar robustez a la aplicación web.
- Es recomendable realizar respaldos mensuales de la información para reducir la posibilidad de pérdida de información.
- Se recomienda utilizar herramientas de software libre si se desea añadir nuevas funcionalidades a la aplicación para no incrementar el costo de producción y licenciamiento.

LISTA DE REFERENCIAS

- D'Onofrio, L. (01 de Agosto de 2013). Libro gratuito de jQuery en español - Fundamentos de jQuery. Recuperado el 08 de Diciembre de 2014, de Libro gratuito de jQuery en español - Fundamentos de jQuery: <http://librojquery.com/#bienvenidoa>
- Foster, R. (2013). Codeigniter 2 Cookbook. Birmingham, Inglaterra: Packt Publishing Ltd.
- Gauchat, J. D. (2012). El gran libro de HTML5, CSS3 y Javascript. Barcelona, España: Marcombo S.A. Recuperado el 10 de Noviembre de 2014, de <https://books.google.es/books?id=szDMIRzwzuUC>
- Jacob, T., & Otto, M. (07 de Junio de 2014). Bootstrap 3, el manual oficial. Recuperado el 12 de Noviembre de 2014, de Bootstrap 3, el manual oficial: http://librosweb.es/libro/bootstrap_3/
- Obe, R., & Hsu, L. (2012). PostgreSQL: Up and Running. Sebastopol, California, Estados Unidos: O'Reilly Media Inc. Obtenido de <https://books.google.es/books?id=Q8jkIZkMTPcC>
- Pavón Mestras, J. (17 de Febrero de 2014). Facultad de Informática. Universidad Complutense de Madrid. Recuperado el 17 de Noviembre de 2014, de Servidores Web – Apache: <http://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>
- Pérez A., O. A. (2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM. Bogota: Facultad de Ingeniería Uniminuto.
- Ramos Martín, A., & Ramos Martín, J. (2014). Aplicaciones Web. Madrid, España: Ediciones Paraninfo S.A. Recuperado el 13 de Noviembre de 2014, de Google Books: <https://books.google.es/books?id=43G6AwAAQBAJ>
- Universidad de Alicante. (10 de Febrero de 2012). Modelo vista controlador (MVC). Recuperado el 03 de Diciembre de 2014, de Modelo vista controlador (MVC). Servicio de Informática ASP.NET MVC 3 Framework: <http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

ANEXOS

Anexo 1. Manual administrador

El manual del administrador se puede descargar desde la aplicación web, la ruta en el servidor es: `./var/www/html/fichas/ayuda/Administrador.pdf`

Anexo 2. Manual docente

El manual del docente se puede descargar desde la aplicación web, la ruta en el servidor es: `./var/www/html/fichas/ayuda/Docente.pdf`

Anexo 3. Manual dirección de carrera

El manual del director de carrera se puede descargar desde la aplicación web, la ruta en el servidor es: `./var/www/html/fichas/ayuda/Director_de_Carrera.pdf`