



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**“AUTOMATIZACIÓN DEL CONTROL DE ASISTENCIA DE
DOCENTES UTILIZANDO EL PATRÓN DE ARQUITECTURA DE
SOFTWARE MODELO VISTA CONTROLADOR EN LA
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA (ESPOCH)”**

Trabajo de titulación presentado para optar al grado académico de:
INGENIERO EN SISTEMAS INFORMÁTICOS

AUTORES: FAUSTO FABIAN BAUTISTA TORRES
JOSÉ RODRIGO ROMÁN ZURITA

TUTOR: DR. JULIO SANTILLÁN

Riobamba-Ecuador

2017

@2017, Fausto Fabián Bautista Torres y José Rodrigo Román Zurita

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal del Trabajo de Titulación certifica que: la “AUTOMATIZACIÓN DEL CONTROL DE ASISTENCIA DE DOCENTES UTILIZANDO EL PATRÓN DE ARQUITECTURA DE SOFTWARE MODELO VISTA CONTROLADOR EN LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA (ESPOCH)”, de responsabilidad de los señores Fausto Fabián Bautista Torres y José Rodrigo Román Zurita, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Ing. Washington Luna Encalada DECANO FACULTAD INFORMATICA Y ELECTRONICA	_____	_____
Ing. Patricio Moreno Costales DIRECTOR ESCUELA INGENIERIA EN SISTEMAS	_____	_____
Dr. Julio Santillán Castillo DIRECTOR DE TRABAJO DE TITULACIÓN	_____	_____
Ing. Washington Luna Encalada MIEMBRO DEL TRIBUNAL	_____	_____

Nosotros, Fausto Fabián Bautista Torres y José Rodrigo Román Zurita somos responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del mismo pertenece a la Escuela Superior Politécnica de Chimborazo.

FAUSTO FABIÁN BAUTISTA TORRES

JOSÉ RODRIGO ROMÁN ZURITA

DEDICATORIA

Este trabajo se lo dedico a mis padres Betty Torres y Fausto Bautista, por ser el pilar fundamental y por el apoyo incondicional que me han brindado, así como a mis hermanos Leonardo, Jessenia y Tiffany que constantemente me dan apoyo, además a Guadalupe Zambrano por ser el soporte, llenarme de fuerzas y brindarme su compañía a lo largo de todo este camino.

Fausto

Dedico esta tesis principalmente a mi madre, por haberme dado la vida a apoyarme incondicionalmente en las buenas y las malas y permitirme llegar a una de mis metas. Ella ha sido padre y madre para mí. A mi familia y amigos debido que con todos ellos hemos formado un gran lapso de amistad

José

AGRADECIMIENTO

Agradezco a Dios por llenarme de bendición a lo largo de mi camino, a mis padres Fausto y Betty, a mis hermanos Leonardo, Jessenia y Tiffani, a mi abuelita Aida Castillo y a Guadalupe gracias a su apoyo incondicional y paciencia.

Fausto

Agradezco a mi madre, por ser una mujer luchadora y nunca abandonarme, estar en las buenas y las malas y poder educarme con responsabilidad y ahínco a mi hermana y demás familiares por el empujón dado a lo largo de mi vida estudiantil

José

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE GRÁFICOS.....	xi
ÍNDICE DE ANEXOS.....	xii
RESUMEN.....	xiii
ABSTRACT.....	xiv
INTRODUCCIÓN.....	1
CAPITULO I	
1 MARCO TEÓRICO REFERENCIAL.....	8
1.1 Control de Asistencia.....	8
1.2 Sistema Informático.....	8
1.3 Patrón de arquitectura de software Modelo Vista Controlador.....	9
1.3.1 <i>Ventajas y desventajas del patrón de arquitectura MVC</i>	11
1.3.2 <i>Análisis del patrón MVC</i>	12
1.4 Lenguaje de Programación Java.....	12
1.5 Base de datos PostgreSQL.....	13
1.6 Lector de Tarjetas Magnéticas.....	14
1.7 Norma ISO/IEC 9126.....	16
1.8 Metodologías de desarrollo Ágil.....	17
1.8.1 <i>Cuadro comparativo entre metodologías de desarrollo de software</i>	17
1.8.2 <i>Análisis de la Metodología Ágil</i>	18
1.8.3 <i>Metodología SCRUM</i>	19
1.9 Cuadro comparativo de herramientas de desarrollo de software.....	21
CAPITULO II	
2 MARCO METODOLÓGICO.....	23
2.1 Registro de Asistencia.....	23
2.2 SCRUM.....	24
2.2.1 <i>El Equipo Scrum (Scrum Team)</i>	24
2.2.2. <i>Estudio Preliminar</i>	24
2.2.2.1. <i>Arquitectura del Sistema</i>	27
2.2.2.2. <i>Tipos y Roles del Usuario</i>	27
2.2.3. <i>Requerimientos</i>	28
2.2.4. <i>Sprint 1</i>	32

2.2.5.	<i>Sprint 2 - 12</i>	34
2.2.5.1.	<i>Historias de Usuario</i>	34
2.2.6.	<i>Sprint 13</i>	37
CAPITULO III		
3	MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS DE RESULTADOS	38
3.1	Gestión del proyecto	38
3.2.	Análisis de Resultados	39
3.2.1.	<i>Mejora en los procesos</i>	39
3.2.2.	<i>Métricas</i>	39
3.2.2.1.	<i>Muestra</i>	39
3.2.2.2.	<i>Medidas</i>	40
3.2.3.	<i>Norma ISO/IEC 9126</i>	43
CONCLUSIONES		51
RECOMENDACIONES		52
GLOSARIO		
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-1: Metodología de desarrollo Ágil.....	17
Tabla 2-1: Herramientas para el desarrollo de software	21
Tabla 1-2: Equipo Scrum.....	24
Tabla 2-2: Herramientas de desarrollo.....	25
Tabla 3-2: Materiales para el lector de tarjetas magnéticas	26
Tabla 4-2: Tipos y Roles de Usuario	28
Tabla 5-2: Product Backlog	28
Tabla 6-2: Product Backlog por Sprint	30
Tabla 7-2: Formato Historia de Usuario	34
Tabla 8-2: Formato Tarea de Ingeniería	35
Tabla 9-2: Formato Prueba de Aceptación	35
Tabla 1-3: Tiempo por escuela.....	40
Tabla 2-3: Tiempos a comparar	41
Tabla 3-3: Tiempos a comparar con Wilcoxon.....	42
Tabla 4-3: Características de la ISO/IEC 9126	43
Tabla 5- 3: Uso de Recursos	50

ÍNDICE DE FIGURAS

Figura 1-1: Arquitectura del Patrón MVC.....	10
Figura 2-1: Norma de Evaluación ISO/IEC 9126.....	17
Figura 1-2: Diagrama de Proceso de control de asistencia docente.....	23
Figura 2-2: Diagrama Conceptual de la base de datos.....	33
Figura 3-2: Diagrama de Caso de Uso del usuario docente.....	36
Figura 4-2: Diagrama de Caso de Uso del usuario Administrador.....	37
Figura 5-2: Diagrama de Caso de Uso del usuario estudiante.....	37
Figura 1-3: Ram antes del reporte por Facultad.....	45
Figura 2-3: Pico en reporte por Facultad.....	45
Figura 3-3: Ram antes de solicitar los Feriados.....	46
Figura 4-3: Pico en obtener los Feriados.....	46
Figura 5-3: Ram antes del solicitar el Horario de una Materia.....	47
Figura 6-3: Pico en obtener el Horario.....	47
Figura 7-3: Ram antes del solicitar el Reporte.....	48
Figura 8-3: Pico en obtener el Reporte.....	48
Figura 9-3: Backup completo.....	49
Figura 10-3: Cantidad de Usuarios.....	49
Figura 11-3: Tamaño Total.....	49

ÍNDICE DE GRÁFICOS

Gráfico 1-3: Burndown Chart	38
------------------------------------------	-----------

ÍNDICE DE ANEXOS

ANEXO A. Diccionario de datos

ANEXO B. Diagrama de clases

ANEXO C. Documento de levantamiento de información

ANEXO D. Manual Técnico

ANEXO E. Sistemas de Control de Acceso Físico Basados en Tarjetas Magnéticas

RESUMEN

El objetivo del trabajo fue el desarrollo del Sistema de control de asistencia docente para la Facultad de Informática y Electrónica de la Escuela Superior Politécnica de Chimborazo (ESPOCH), para la identificación de problemas se utilizaron técnicas como la entrevista y la observación, aplicadas a los principales actores de cada proceso tales como: Directores de Escuelas y Decano de la Facultad, luego de lo cual se utilizó la metodología de desarrollo ágil SCRUM para la creación del sistema siguiendo el patrón de desarrollo Modelo Vista Controlador (MVC) con el lenguaje de programación JAVA, la tecnología Java Server Faces (JSF), el framework primefaces y una base de datos en postgresQL, los cuales se comunican con los servicios web del sistema OASIS de la ESPOCH para consumir información necesaria, el desarrollo produjo el Sistema de Control de Asistencia Docente con un total de 1486 líneas de código y un peso total de 137 Mb, el cual a través de las pruebas correspondientes reveló una mejora del 99.993% en el tiempo de generación de reportes sobre el seguimiento por facultad y una mejora del 99.977% del tiempo invertido en generar el documento por Escuela, volviendo así más eficiente al proceso con respecto al tiempo, además al ser implementado en un servidor web en permanente funcionamiento y accesible desde internet se logra mejorar la disponibilidad de la información; por los resultados favorables obtenidos se recomienda que el sistema sea implementado a nivel de toda la ESPOCH.

PALABRAS CLAVE: <TECNOLOGÍA CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA DE SOFTWARE>, <JAVA SERVER FACES (JSF)>, <FRAMEWORK PRIMEFACES>, <METODOLOGÍA SCRUM>, <LENGAJE DE PROGRAMACIÓN JAVA>

ABSTRACT

The objective of this work was to develop a teacher attendance control system for the Informatics and Electronics Faculty of the Escuela Superior Politécnica de Chimborazo (ESPOCH). For the identification of problems, techniques like interviews and observation were used; applied to the main actors of each process such as: School Directors, Faculty Deans and Vice Deans, for which the methodology of agile development SCRUM was used for the creation of the system following the pattern of development Model View Controller (MVC) with the JAVA program language, JAVA Server Faces (JSF) technology, the PrimeFaces framework and a database in postgresSQL, which communicate with the web services of the OASIS system of the ESPOCH to utilize necessary information. Its development produced the Teacher Assistance Control System with a total of 1 486 lines of code and a total weight of 137Mb which, through the corresponding tests, revealed an improvement of 99.993% in the generating time of reports on the faculty follow-up and an improvement of 99.977% of the time invested in generating the document by each school, making the process more efficient with respect to time. In addition to being implemented in a web server in permanent functioning and accessible to the internet. It is possible to improve the availability of the information. For the favorable results obtained, it is recommended that the system be implemented at the ESPOCH-wide level.

KEYWORDS: <ENGINEERING SCIENCES TECHNOLOGY>, <SOFTWARE ENGINEERING>, <JAVA SERVER FACES (JSF)>, < PRIMEFACES FRAMEWORK>, <SCRUM METHODOLOGY>, <JAVA LANGUAGE PROGRAM>

INTRODUCCIÓN

En países en vías de desarrollo, se ha vuelto cotidiano el control a las instituciones y en especial el seguimiento a sus procesos, como ejemplo se pueden mencionar las instituciones educativas y específicamente las instituciones de educación superior, las cuales actualmente se encuentran permanentemente controladas, reguladas y evaluadas por las instituciones correspondientes, en nuestro país son el CEAACES y el SNIESE.

En este entorno particular, como ejemplo se establece a la ESPOCH, misma que ha visto afectado su funcionamiento y su evaluación por una inoportuna toma de decisiones en algunos de sus procesos regulares, se menciona al proceso de control de asistencia docente, el cual degenera en problemas de presupuesto así como de cumplimiento a los estatutos y reglamentos establecidos por las entidades reguladoras antes mencionadas.

Respecto al aspecto educativo, el incumplimiento del horario de clases por parte de los docentes implica una falla en el proceso fundamental de la institución superior que es el preparar a los estudiantes de una manera correcta, misma que no se puede dar en caso de una inasistencia o un atraso por parte de los docentes.

Económicamente, al ser la ESPOCH una institución del estado, esta maneja recursos públicos, los cuales se ven mal gestionados si se les paga a docentes que no están cumpliendo de manera correcta sus horarios de clase y las labores que deben desempeñar dentro de la institución.

Por otra parte, en el ámbito tecnológico aparecen variadas y novedosas soluciones software para la creación de aplicaciones empresariales, mismas que se enfocan a facilitar el proceso y optimizar los resultados alcanzados, principio que se intentará implementar al proceso estudiado durante el presente proyecto.

El lenguaje de programación JAVA ha ido colocándose como una alternativa a tener en cuenta sobre todo por su compatibilidad, razón por la cual gran cantidad de frameworks han sido desarrollados para mejorar las prestaciones de JAVA durante un desarrollo y cada una orientada a una necesidad particular.

En cuanto a tendencias de tecnología, hasta hace relativamente poco tiempo, se notaba una enorme preferencia por parte de los usuarios hacia el uso de herramientas almacenadas localmente o también llamadas de escritorio, por una sensación de desconfianza colectiva que se tenía a

entregar sus datos a través de la red, efecto que ha ido disminuyendo radicalmente dado el auge de servicios en la nube y la invasión de dispositivos móviles que han mostrado las ventajas de esta tecnología, por lo que actualmente la tendencia se inclina totalmente a la aplicaciones web, por sus beneficios como la alta disponibilidad en los datos y la accesibilidad geográfica que esta representa.

Con la base argumental explicada, se plantea el desarrollo del Sistema de Control de Asistencia Docente para la Facultad de Informática y Electrónica (ESPOCH), mismo que se apoyará sobre los datos ya existentes en los servicios web del Orion Academic System with Internet Services de ahora en adelante OASIS, utilizando a JAVA como lenguaje de programación y el framework de primefaces para la mejora de los apartados visuales de la tecnología JSF siguiendo la metodología de desarrollo ágil SCRUM, todo esto utilizando el IDE Netbeans en su versión estable más actual al momento del desarrollo, así como PostgreSQL como motor de base de datos, cabe mencionar que el desarrollo no se restringirá a estas herramientas, dado que a medida que vayan siendo necesarias se podrán integrar otras o a su vez reemplazar algunas de las planteadas.

Se definen los siguientes 3 capítulos para organizar el desarrollo del proyecto:

Capítulo I: Este capítulo está enfocado al problema, a su definición, análisis y comprensión, así como a la definición de los conceptos herramientas y técnicas a implementar.

Capítulo II: Comprende los procesos involucrados en el tema a automatizar, por lo cual contendrá una gran base documental que apoye a las decisiones tomadas, además que abordará a la metodología SCRUM para plantear las bases sobre las cuales se elaborará la solución de la manera que la metodología lo establezca para este caso.

Capítulo III: Se enfoca en la implementación de todo lo establecido en el Capítulo II para la construcción de la solución y el uso de la misma para el levantamiento de información requerido cumpliendo con la comprobación de los objetivos establecidos durante el planteamiento del problema.

ANTECEDENTES

La Escuela de Ingeniería de Sistemas de la Facultad de Informática y Electrónica (FIE) es una unidad académica comprometida con los más altos intereses de la sociedad. En todos los ámbitos institucionales y de la opinión pública se reconoce la importancia y la influencia en la formación de recursos humanos, producción y divulgación de conocimiento científico técnico. Es un espacio académico de formación de profesionales de tercer nivel en la rama específica, cuya función es formar profesionales idóneos, capaces, competentes para insertarse en el desarrollo integral del país.

Durante los últimos años los avances tecnológicos y los requerimientos de mejorar la calidad en la educación ha hecho que se implementen sistemas que ayuden a mejorar procesos que tradicionalmente eran realizados por seres humanos ahora son gracias a las mejoras tecnológicas.

Debido a los constantes cambios tecnológicos y las disposiciones vigentes dispuesta en cada uno de las instituciones por el interés de monitorear y llevar a cabo un control del cumplimiento de las jornadas laborales, es necesarios presentar nuevas alternativas que satisfagan estas exigencias de manera que se puede controlar, automatizar y realizar de manera estructurada esta actividad.

Dentro de la amplia gama de posibles actividades que pueden automatizarse, aquella relacionada con la capacidad para establecer la identidad de los individuos ha cobrado importancia y como consecuencia directa, la biometría se ha transformado en un área emergente.

Los sistemas de control de asistencia modernos se basan en tecnologías de identificación automática, uno de estos es el sistema de sensores magnéticos, siendo este no más que una parte de la solución debido a que el componente principal es fundamentalmente el software de control de asistencia debido que los datos capturados con dicho lector requieren ser procesados para convertirlos en información.

FORMULACIÓN DEL PROBLEMA

¿Con el desarrollo de un sistema informático para el control de asistencias a docentes en la Facultad de Informática y Electrónica (ESPOCH), se podrá contar con información disponible y reducir el tiempo de respuestas para consultar datos, reportes e informes referente a las jornadas laborales de los docentes?

SISTEMATIZACIÓN DEL PROBLEMA

¿Cómo se realiza actualmente el proceso de control de asistencia de docentes de la Facultad de Informática y Electrónica?

¿Con el desarrollo del Sistema informático se llevara un mejor control de la asistencia de los docentes?

¿Con el desarrollo del Sistema informático utilizando los datos obtenidos por el lector de tarjetas magnéticas contribuirá a mejorar la gestión de la información del personal docente?

¿Con el desarrollo del Sistema informático se podrá facilitar la gestión de la información y procesamiento de datos de forma rápida y segura?

¿Con el desarrollo del Sistema informático se podrá generar reportes de forma rápida y confiable respecto a las jornadas laborales del personal docente?

JUSTIFICACIÓN

Se establecen las bases documentales y reglamentarias así como los fundamentos sobre los que se orientará el presente desarrollo.

Justificación teórica

El reglamento de acreditación del CEAACES (R MODELO GENERAL PARA LA EVALUACIÓN DE CARRERAS CON FINES DE ACREDITACIÓN, 2011), especifica que como institución educativa bajo sus reglamentos debe entregar cuando se lo solicite la información que este considere conveniente para evaluar y dar seguimiento a los procesos que viene efectuando como parte de su funcionamiento, tanto en el ámbito administrativo como en el ámbito escolástico, que es el punto en que se enfoca este proyecto.

Tomaremos como objeto de estudio la Escuela Superior Politécnica de Chimborazo, que como institución educativa de nivel superior se ve sujeta a CEAACES, por cuanto debe cumplir lo establecido en el párrafo anterior, lo cual ha producido problemas relacionados a la disponibilidad, acceso y calidad de la información que posee sobre las funcionalidades que fueron ejecutadas.

Una de las actividades que se llevan a cabo en el proceso de enseñanza y aprendizaje es la de controlar la asistencia docente, mismo que debe ser controlado permanentemente por el personal responsable generando así la inversión de recursos tanto humanos como económicos.

A nivel de país, al artículo 81 de la Constitución Política de la República establece que la participación democrática se realiza en base a la información a la que se debe tener acceso, de la misma manera el Pacto Internacional de Derechos Civiles y Políticos así como la Convención Interamericana de Derechos Humanos, garantizan el acceso a la información de calidad de manera oportuna.

Actualmente la Facultad de Informática y Electrónica no cuenta con un sistema que permita llevar el control de asistencia a docentes automatizado, este control es realizado de forma manual lo cual provoca la recolección y proceso de información sean realizados de manera lenta, lo que se requiere de un determinado personal para realizar dicha actividad, derivando en un control no exacto de las horas laboradas, además de convertirse en una tarea tediosa y lenta a la hora de realizar consultas y obtener reportes de asistencias.

Por los problemas mencionados, se propone como una posible solución la realización de un sistema informático que permita automatizar el proceso de control de asistencia a los docentes.

Justificación aplicativa

Enfocándose a los problemas antes mencionados, y teniendo en cuenta que la información producida por la asistencia docente, tiene el fin de ser consultada a futuro, se decide la implementación una base de datos relacionales, mismos que mantendrá la información perdurable en el tiempo y logrará homogenizar los datos al convertirse en una sola fuente de información.

Contará con un módulo para la gestión del horario de clase por parte del docente.

Para el proceso de seguimiento, se utilizará un dispositivo hardware original, mismo que a través de credenciales magnéticas identificara al docente y almacenará su hora de llegada o salida, esos valores estarán disponibles en tiempo real para su consulta

El administrador del sistema muestra la información de configuración, migración y reporte sobre la información almacenada en la base de datos utilizada, para el correcto funcionamiento del sistema.

Módulos Sistema de Control de Asistencia Docente

Módulo	Descripción
Generación	Este módulo permitirá generar un listado de las fechas y horas en que se debe impartir una materia en el transcurso del semestre.
Seguimiento	Este módulo será el encargado de censar y almacenar los datos sobre la asistencia de los docentes.
Administrar/Seguridad	El sistema tendrá un módulo de administración que cumplirá la función de gestión de las configuraciones del sistema así como de los permisos existentes y la asignación de credenciales a los docentes para su identificación durante el control de asistencia, además de toda la información recolectada generará reportes organizados y de fácil interpretación.

Realizado por: Bautista Fausto y Román José. 2017

Como consecuencia el sistema se enfocará en optimizar:

- Registro del horario semestral.
- Revisión e Interpretación de los listados de asistencia.
- Generación de reportes de cumplimiento.

Y con su implementación se pretenden las siguientes ventajas:

- Disminución de tiempos en los Procesos de Generación y Revisión de la asistencia docente.
- Accesibilidad a la Información.
- Mejora en la Integridad de datos.
- Apoyo a la toma de decisiones con información oportuna.
- Mejora de tiempos y recursos en el Proceso de Control de Asistencia Docente.

OBJETIVOS

Objetivo general

Desarrollar el sistema informático para automatizar el control de asistencia de los docentes de la Facultad de Informática y Electrónica (ESPOCH).

Objetivos específicos

- Analizar la tecnología que se va a usar en el desarrollo del Sistema.
- Desarrollar el Sistema de Control de Asistencia de docentes por medio de un lector de tarjetas magnéticas en la Escuela de Ingeniería en Sistemas.
- Analizar los resultados obtenidos y comparar el tiempo empleado con el proceso de asistencia anterior, después del desarrollo del Sistema de Control de Asistencia de Docentes.
- Demostrar la eficiencia del Sistema de Control de Asistencia de Docentes mediante la norma ISO/IEC 9126.

CAPITULO I

1 MARCO TEÓRICO REFERENCIAL

1.1 Control de Asistencia

En su publicación “¿Qué es el registro y control de asistencia?” el autor lo define como:

El proceso administrativo, cuya función principal es la puesta en práctica de un conjunto de instrumentos que se interrelacionan entre sí para cumplir un objetivo en común, con el fin de realizar el registro y llevar un control del personal que labora en una determinada empresa o institución.

Para que cada una de las acciones o actividades realizadas en las instituciones se cumplan de manera satisfactoria, es de gran importancia que exista un adecuado registro y control del capital intelectual, permitiendo de esta forma asegurar que cada una de las unidades que conforman la organización marche conforme a lo planificado y previsto. De tal forma que el proceso debe ser realizado con el fin de controlar las entradas y salida del personal, mismo que cumple con un determinado horario de trabajo. (López, 2013, p.24)

1.2 Sistema Informático

Los sistemas informáticos son considerados como subsistemas que apoyan la información que circula en una determinada empresa o institución, mismo que está consolidado por un conjunto de recursos necesarios para dar respuesta a un determinado tratamiento automático de la información y de aquellos otros actores que posibiliten la comunicación de la misma. (López et al., 2004: p.37)

Los recursos de un Sistema Informático mencionados por (Alegsa, 2016) son:

- **Recurso de hardware:** computadoras, impresoras, escáneres, memorias, lectores de código de barras, estructura física de una red de computadoras, etc.
- **Recurso de software:** manuales de uso, sistema operativo, archivos, documentos, aplicaciones, firmware, bases de datos, información de una red de computadoras, etc.

- **Recurso humano:** son todas las personas que forman parte del sistema, como ser los operadores del sistema, los técnicos que lo mantienen y los usuarios finales.

La finalidad u objetivo de un sistema de informático que señala Hernández (2003, p. 12), son detallados a continuación:

- Apoyar con los objetivos y estrategias a través de sus funcionalidades, generando información necesaria para que instituciones o empresas tengan un correcto funcionamiento.
- Proporcionar información de cada una de las actividades realizadas por la empresa o institución.
- Adaptarse a las necesidades requeridas por la institución, de manera es éste pueda responder a su constante evolución conforme vaya creciendo y evolucionando, debido que en algunos casos surgen nuevas necesidades que han de ser satisfechas por el sistema.
- Tener una buena interacción con cada uno de los actores para los cuales se destina el sistema, de modo que se satisfagan sus necesidades de modo eficaz.

1.3 Patrón de arquitectura de software Modelo Vista Controlador

En su publicación “¿Qué es MVC?” el autor lo define como:

Una guía para el diseño de arquitecturas de aplicaciones en proceso de desarrollo, mismas que ofrecen una robusta interactividad con cada uno de los usuarios. Éste patrón se encarga de organizar la aplicación de forma que este separada en tres modelos (**Figura 1-1**), el primero es el modelo que representa las reglas del negocio y representa los datos que son parte de la aplicación, el segundo es un grupo de vistas representados por los formularios que son el mecanismo de entrada y salida de información y el tercero es un grupo de controladores que son los encargados de receptor las diferentes peticiones de los usuarios controlando de esta forma el flujo de ejecución del sistema.(Gutiérrez, 2006, pp.2-4)

A continuación, en el **Figura 1-1** se muestra la Arquitectura del Patrón MVC.

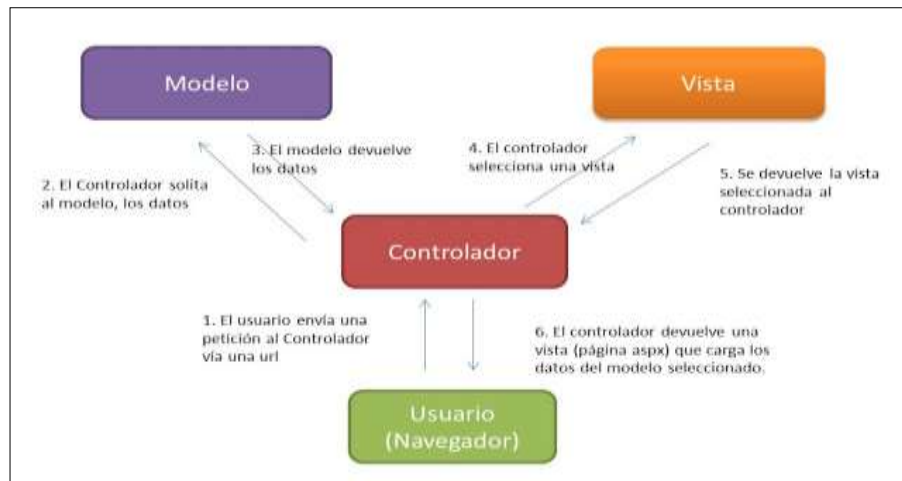


Figura 1-1: Arquitectura del Patrón MVC

Fuente: (Mercado, 2016, <http://vista-controlador.blogspot.com/>)

En su publicación “Entendiendo el Modelo-Vista-Controlador” (Fernández y Díaz, 2012: pp. 49-50) define a las partes o conceptos del patrón MVC de la siguiente manera:

➤ La capa del Modelo

El modelo es considerado como el objeto que se encarga de la representación del programa. Maneja a su vez el conjunto de datos y controla todas sus transformaciones. Está separado totalmente y no tiene referencia alguna con cada una de las actividades que son realizadas por el controlador y las vistas, debido que es propio del sistema mantener entrelazados cada uno de estos componentes.

Cuenta con las siguientes funcionalidades:

- Acceder a la capa de almacenamiento de datos.
- Definir las reglas del negocio (funcionalidades del sistema).
- Notificar a las vistas los cambios que en los datos pueda producir una agente externo.

➤ La capa de la Vista

La Vista es considerada como el objeto que maneja la presentación visual del conjunto de datos que son representados en el componente Modelo. Es la encargada de mostrar los datos que son solicitados por los usuarios debido que interactúa de forma directa con el

Controlador recibiendo cada uno de los datos de entrada para que luego sean convertidos en información de salida a través de cada uno de los formularios.

Siendo estas responsables de:

- Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado.
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo.

➤ La capa del Controlador

El controlador es como el componente que cumple la función de proporcionar significado a casa una de las órdenes del usuario, mismo que actúa sobre los datos que son representados en el considerado modelo, centrando de esta forma toda la interacción entre la Vista y el Modelo.

Cumple con funcionalidades específicas de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos. Estas acciones pueden suponer peticiones al modelo o a las vistas.

1.3.1 Ventajas y desventajas del patrón de arquitectura MVC

Ventajas

- Facilidad en la reutilización de código.
- Permite la integración de frameworks mismas que incluyen librerías para enriquecer los formularios.
- Fácil comprensión en el código.
- La lógica de negocio está separada y si existe alguna modificación no afecta a los demás componentes.
- Las modificaciones realizadas a las vistas no afectan a las demás lógicas del modelo.
- Simplifica las labores de mantenimiento.

Desventajas

- Mayor dedicación en los tiempos iniciales de desarrollo debido a que nos exige la creación de un gran número de clases.
- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos que se desarrollan y se mantienen se aumenta considerablemente.

1.3.2 Análisis del patrón MVC en el desarrollo del sistema de control de asistencia

De acuerdo a las ventajas que proporciona el patrón de arquitectura de software MVC, en la etapa de desarrollo del sistema de control de asistencia, permitió crear de forma estructurada cada una de las sentencias, mismas que se encuentran clasificadas en sus diferentes módulos (Modelo-Vista-Controlador), las cuales se correlacionan entre sí para cumplir con los diferentes objetivos que demanda la aplicación. A medida que surgieron inconvenientes en cuanto a los errores de codificación en la lógica de negocio, las vistas no se vieron afectadas al realizar las respectivas modificaciones. Las librerías del framework Java Server Faces (JSF) se integró de forma correcta con el patrón de arquitectura, permitiendo enriquecer los formularios en cuanto al diseño y validaciones de cada uno de los campos. Debido a que el trabajo fue desarrollado por dos programadores cada una de las funcionalidades fue distribuida, facilitando de esta forma la integración y comprensión del código por parte de ambas partes. Además cabe mencionar que la aplicación está estructurada de tal forma que facilita la unificación de nuevas funcionalidades y simplifica los trabajos de mantenimiento.

1.4 Lenguaje de Programación Java

Como lo menciona su página oficial, JAVA es un lenguaje de programación diferente a JAVASCRIPT, se la define como una tecnología sencilla y útil, dado que su condición multiplataforma lo hace convertirse en una opción a tener en cuenta para cualquier tipo de proyecto. (Oracle, 2017)

A continuación Garrido (2015, pp. 2-6) menciona algunos de los objetivos perseguidos a comienzo del desarrollo del proyecto del lenguaje Java:

- **Familiar:** su sintaxis es similar al lenguaje de programación C o C++, facilitando que los desarrolladores se familiaricen y se acoplen de manera rápida a dicho lenguaje.
- **Sencillo:** Lo aprendido sirve para el desarrollo en distintos dispositivos para lo que comúnmente se utilizan herramientas y lenguajes diferentes entre sí.

- **Multiplataforma:** fue diseñado para que cumpla la función de escribir (y compilar) solo una vez en una plataforma y tener la posibilidad de poder ejecutar en otra plataforma sin necesidad de modificar el código.
- **Alto rendimiento:** Java no es tan rápido en comparación a lenguajes que se compilan de forma nativa para una plataforma concreta, pero a su vez no es tan lenta al momento de compilar e ir interpretando línea por línea, comprobando cada una de la sintaxis.
- **Robusto:** al momento de realizar la compilación se van comprobando cada una de las sintaxis, logrando de esta forma dar lugar a resultados inesperados.
- **Orientado a Objetos:** siguiendo este paradigma facilita todo ciclo de vida del software, desde el análisis, diseño, mantenimiento e implementación.
- **Distribuido:** cuenta con un conjunto de librerías que permite la comunicación entre los distintos componente que maneja.

De la misma manera Sznajdleder (2013, pp. 2-15) menciona algunas de sus características:

- Es multiplataforma lo que provoca que las aplicaciones que se han desarrollado con este lenguaje puedan ser desplegadas en cualquier sistema operativo y hardware. Posicionándose como el lenguaje de programación más versátil en los últimos años.
- JAVA, debido a que es un lenguaje de Programación Orientado a Objetos (POO), los desarrolladores requieren del conocimiento de diferentes técnicas (Herencia, Polimorfismo, Abstracción, Encapsulamiento, etc.), siendo importante para que los mismo logren aplicaciones flexibles, mantenibles, robustas, y escalables.
- Las sintaxis con las que se codifican en JAVA es casi similar a las del lenguaje de programación C++.
- Enfocándonos en el ámbito empresarial, las aplicación en JAVA pueden interactuar con lo que son las bases de datos, servidores de mensajería asincrónica, servicios de nombres y directorio, Consumos de Servicios Web, facilitando de esta manera las tareas de integración y el desarrollo de sistemas distribuidos.

1.5 Base de datos PostgreSQL

Es un sistema gestor de base de datos objeto-relacional, que se encuentra distribuido bajo licencia BSD y cuenta con el código disponible libremente, lo cual lo convierte en el en una de las bases de datos con código abierto más potente en el mercado. Además utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos, mismo que le garantiza

que el sistema se encuentre estable, lo cual permite que si existe un fallo en algunos de los procesos el resto no se verá afectado y seguirá en funcionamiento si ningún inconveniente. (Zea y Molina, 2017: p.12)

De acuerdo con (Kasián y Reyes, 2012: p. 4), menciona que PostgreSQL es un potente motor de bases de datos relacionales, principalmente es conocido por su fiabilidad, integridad de datos y cumple con un correcto desempeño, sin dejar de lado su alta portabilidad hacia los distintos sistemas operativos debido que al momento de utilizarlos no presenta ningún tipo de problema. Por motivos que es de código abierto puede ser sujeto a su uso y modificación sin restricción alguna.

Según el sitio oficial de PostgreSQL, (Zea y Molina, 2017: p.12) detallan a continuación varias de las características:

- Es una base de datos 100% ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad)
- Brinda procesos de PITR orientados a puntos de restauración
- Soporta copias de seguridad en caliente tanto locales como remotas
- Codificación Unicode
- Métodos de autenticación variados
- SSL

En cuanto a sus limitaciones se puede mencionar:

- Máximo tamaño de tabla: 32 TB
- Máximo tamaño de fila: 1.6 TB
- Máximo tamaño de campo: 1 GB
- Máximo número de columnas por tabla: 250 - 1600 (dependiendo del tipo)

1.6 Lector de Tarjetas Magnéticas

Es un dispositivo cuyo mecanismo es actuar como interfaz de usuario y el sistema, en la cual interviene una tarjeta magnética misma que permite enviar información almacenada en la misma. Cabe mencionar que durante todo este proceso interviene la base de datos cuya función es reconocer la información, autenticar los datos y luego procesarlos con ayuda del sistema, luego de que la tarjeta fue leída ya sea por contacto o proximidad. (Vallejo, 2014, p. 22)

Alguno de los beneficios señalados por (Vallejo, 2014, pp. 22), son:

- Automatización de los procesos.
- Lectura rápida y fácil de la información.
- Incrementa su productividad.
- Elimina el error humano.

El lector de tarjetas magnéticas utilizado en el presente proyecto utiliza tecnología RFID (Radio Frequency Identification), misma que (Tenorio y Mejía, 2012: p. 48), la definen como: una tecnología para identificar objetos sin la necesidad que exista contacto, además es considerado como un método encargado de almacenar y recuperar de forma remota los datos, haciendo uso de etiquetas o “tags” la cual contiene información.

A continuación, (Tenorio y Mejía, 2012: pp. 48-49) detallan tres elementos que componen el sistema RFID:

- **Etiqueta RFID:** conocida como tag o transpondedor (transmisor y receptor). Misma que siempre va a tener un portador, para que éste a través de la información que se encuentra almacenada, pueda realizar sus actividades respectivas. Consta de un microchip que almacena los datos y una antena para la comunicación con el lector.
- **Lector o interrogador:** responsable de transmitir energía a la etiqueta y proceder a leer los datos que esta le envía.
- **Un ordenador, host o controlador:** desarrolla la aplicación RFID. Encargada de recibir la información de uno o varios lectores y es interpretada por todo el sistema de información.

Alguna de las aplicaciones más frecuente que señala (Vallejo, 2014, p. 23) en su estudio, son:

- Control de acceso: registro de entrada y salida.
- Identificación de persona: lectura de credenciales.
- Tarjetas de puntos y monedero electrónico.
- Consulta de saldos y conectividad con base de datos.
- Peaje de vehículos.
- Centro de pago.

1.7 Norma ISO/IEC 9126

La Norma ISO/IEC 9126 fue desarrollada como un estándar para la ingeniería de software, la cual define la usabilidad como una propiedad considerablemente independiente de la calidad del software, recalcando puntos como el diseño, la evaluación de la interfaz de usuario y su respectiva interacción. Además es considerado como el conjunto de características que tienen repercusión en el esfuerzo necesario para el uso. (Granollers et al., 2005: p.97)

Jakob Nielsen, uno de los pioneros en cuanto a todo lo referente con la usabilidad, indica que la usabilidad es un término multidimensional. Señala que un sistema usable debe contar con las siguientes características: capacidad en cuanto el aprendizaje, eficiencia en el uso, facilidad de memorizar, tolerante a errores y subjetivamente satisfactorio. (Nielsen, 1993; citado en Granollers et al., 2005: p.98)

En su publicación “NORMA DE EVALUACIÓN ISO/IEC 9126”, (Ruiz et al., 2006: pp. 27-28) categoriza seis características (**Figura 2-1**) presentadas a continuación:

- **Funcionalidad:** considerada como la capacidad del software para brindar funciones que cumplan con las necesidades previamente establecidas e implícitas, en momentos que el software es utilizado bajo condiciones específicas.
- **Confiabilidad:** es la capacidad que posee el sistema para mantener su nivel bajo condiciones específicas.
- **Usabilidad:** el software debe de ser entendible, fácil de aprender, utilizable y que sea de agrado para el usuario, en condiciones específicas.
- **Eficiencia:** el software debe de estar en la capacidad de brindar un desempeño apropiado, en relación a la cantidad de recursos destinados, bajo condiciones específicas.
- **Facilidad de Mantenimiento:** debe de tener la capacidad de ser modificado, puede incluir correcciones, mejoras o adaptaciones.
- **Portabilidad:** es la posibilidad de que el sistema pueda ser removido de un ambiente a otro sin presentar mucha complicación.



Figura 2-1: Norma de Evaluación ISO/IEC 9126

Fuente: (Borbón, 2013, <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>)

1.8 Metodologías de desarrollo Ágil

Las metodologías ágiles son las encargadas de resolver problemas que han surgido, posteriormente, a la masificación del uso de la computadora personal, dado que es necesario cumplir con cada una de las expectativas y las necesidades que surgen por parte de los usuarios, siendo éstas cada vez más urgentes y frecuentes. Es por eso que a inicio de los 90 surgieron metodologías que obtienen resultados más rápidos en el desarrollo de software sin disminuir su calidad, entre las más utilizadas se pueden mencionar a XP, CRYSTAL y SCRUM. (Orjuela y Rojas, 2008: p.162)

El término de metodologías ágiles se genera en respuesta a los problemas que presentan las metodologías tradicionales y se basa en dos aspectos principales, el retrasar las decisiones y la planificación adaptativa; permitiendo de esta forma potenciar aún más el proceso de desarrollo de software a una gran escala. (Figuerola et al., 2008: p.5)

1.8.1 Cuadro comparativo entre metodologías de desarrollo de software

El enfoque comparativo entre las metodologías de desarrollo de software (**Tabla 1-1**) se centra en detallar cada una de las ventajas y describir a breves rasgos las diferencias con las que cuentan.

Tabla 1-1: Metodología de desarrollo Ágil

Cuadro comparativo de Metodologías de Desarrollo Ágil		
Metodologías	SCRUM	XP
Semejanzas	<ul style="list-style-type: none"> Están consideradas en el grupo de las metodologías ágiles. El estilo de documentación de los requerimientos lo haces a través de las historias de usuario. 	

	<ul style="list-style-type: none"> • Continuamente se realizan entregas de software en cortos periodos de tiempo. • Las reuniones entre el equipo de trabajo son constantes. 	
Diferencias	Los entregables de cada una de las versiones del sistema son de dos a cuatro semanas, conocidos como sprint.	Los entregables de cada una de las versiones del sistema son de una a cuatro semanas.
	El equipo de desarrollo sigue el orden de prioridad que establece el dueño del producto en relación a las tareas a realizar, pero con opción a cambiar para bien de todos.	El equipo de desarrollo sigue estrictamente el orden de prioridad que establece el cliente en relación a las tareas a realizar, pero sirven como guía para decidir.
	Basada en la administración del proyecto.	Se centra en la programación y la creación del producto.
	Cada miembro del equipo trabaja de forma individual.	Los miembros del equipo trabajan en parejas de dos.

Realizado por: Bautista Fausto, Román José. 2017

1.8.2 Análisis de la Metodología Ágil seleccionada en el desarrollo del sistema de control de asistencia

Las metodologías ágiles mencionadas, cuentan con características importantes, las cuales permiten tomar ventajas durante la ejecución del proyecto, dependiendo el marco en el que se trabaje, de modo que, se optó por la utilización de la metodología SCRUM, por lo que se recalcan puntos importantes en comparación a la otra metodología que van en beneficio a nuestro proyecto. Dicha metodología se base en sí, en la gestión administrativa de todo el proyecto y no estrictamente dedicada solo a la programación, que si bien es cierto, es de gran importancia el producto terminado, pero es necesario seguir un conjunto de etapas que permitan tener una guía acerca de cómo se está estructurando el mismo. Por motivos que el equipo de desarrollo está conformado por dos integrantes es necesario seguir de forma estricta una de las características con la que cuenta la metodología, la cual menciona, que cada miembro del equipo de desarrollo trabaja de forma individual, permitiendo de esta forma distribuir entre ambas partes las funcionalidades con la que cuenta el sistema, logrando disminuir el tiempo del cumpliendo a cabalidad con todo lo que demanda el producto. Cabe recalcar, que al estar dentro del grupo de las metodologías ágiles, nos da la facilidad para que como grupo de trabajo sepamos responder y adaptarnos a posibles cambios que se puedan suscitar en tiempo real. La metodología SCRUM es de gran importancia, debido que nos ayuda que el dueño del producto tenga conocimiento de los avances que se han realizado de forma paulatina a través de los Sprints realizados durante un lapso de tiempo, convirtiéndose éstos en una versión funcional entregable del sistema, sin dejar de lado, que las tareas realizadas, tienen una correcta documentación, plasmados a través de las

historias de usuario, mecanismo utilizado, que sirve para guía de personas inmersas y ajenas en el proyecto.

1.8.3 Metodología SCRUM

El termino SCRUM en la gestión de proyectos es definida como “*una estrategia flexible y holística de desarrollo de productos, donde un equipo de desarrollo trabaja como una unidad para alcanzar un objetivo en común*”. Se enfoca en la filosofía de que los requerimientos están en sus inicios sin perfeccionar y son poco claros. Además se centra a la capacidad que debe de tener el equipo de desarrollo para observar y adaptarse a las nuevas exigencias. (Takeuchi y Nonaka, 1986; citado en Chong, 2016: p.11)

SCRUM también es considerado como un proceso ágil y liviano cuya función es la de administrar y controlar el proceso de desarrollo de software, mismo que es desarrollado de forma interactiva e incremental. Recalcando que cada ciclo finalizado se convierte en una parte ejecutable del software y funcional. (Figuerola et al., 2008: p.7)

El equipo de trabajo SCRUM, mencionado por Rúa (2007, p.2) son:

- **Cliente:** encargado de facilitar los requerimientos y orientar a la funcionalidad esperada del sistema.
- **El Dueño del producto (Product Owner):** propietario del producto.
- **Equipo de Desarrollo (Development Team):** equipo encargado del desarrollo de las funcionalidades del sistema.
- **Scrum Master:** responsable de gestionar el proyecto software y liderar al equipo de desarrollo.

Los eventos de SCRUM, mencionados por Pérez (2011, pp. 11-13) son:

a. El Sprint

Ésta fase está comprendida por un lapso de tiempo de 30 días, en donde se efectúa el proceso de desarrollo de software y donde se llevan a cabo las reuniones, se pueden mencionar algunas subfases: elaborar, integrar, revisar y ajustar, estas subfases no son estrictas obedeciendo a prácticas ya mencionadas.

b. Planificación de Sprint

Es de gran importancia ésta jornada de trabajo, debido que una mala planificación puede provocar arruinar todo el sprint. Interviene el propietario del producto el cual explica las prioridades y dudas que tenga el equipo de trabajo, para lo cual es necesario estimar el esfuerzo de cada uno de los requisitos, realizando una lista de los responsables y nivel de dedicación.

c. Reunión Diaria

Está comprendida por un periodo de tiempo de 15 a 30 minutos de duración, siempre en un mismo lugar y a la misma hora, precedida por el SCRUM Manager y solo puede intervenir el equipo SCRUM, respondiendo a las siguientes interrogantes: ¿Qué hiciste ayer?, ¿Cuál es el trabajo para hoy? y ¿Qué necesitas?

d. Revisión de Sprint

Corresponde al incremento, se revisa el Sprint si es el caso se añaden nuevos ítems a la pila del producto, este proceso es repetido hasta que el producto quede listo para la fase de cierre.

e. Retrospectiva del Sprint

Después de haber culminado cada Sprint se reúne el equipo de trabajo por un tiempo de 4 horas máximo, la misma que sirve para expresar sus opiniones del Sprint culminado, con la finalidad de mejorar los procesos, siendo esta una reunión de evaluación y de mejoramiento.

Como herramientas para el seguimiento de la metodología, Rua (2007, p.4) se plantea diversos artefactos, entre ellos menciona:

a. Lista de Producto (Product Backlog)

Corresponde al listado de cada una de las funcionalidades que serán desarrolladas a lo largo de todo el proyecto, mismas que son definidas y detalladas de forma correcta al inicio del trabajo, deben de ordenadas con una prioridad que estime el equipo.

b. Lista de Pendientes del Sprint (Sprint Backlog)

Es el conjunto de funcionalidades que se crearan con el fin de obtener un producto funcional presentado al cliente, a su vez debe de ser desarrollado de forma incremental relativa al cumplimiento del Backlog anterior (si existe)

1.9 Cuadro comparativo de herramientas de desarrollo de software

En el siguiente cuadro comparativo (**Tabla 2-1**) se enlistan cada una de las herramientas que permiten la elaboración de un proyecto software, mencionando ventajas y desventajas propias de cada una.

Tabla 2-1: Herramientas para el desarrollo de software

Especificación Técnica	Nombre	Ventajas	Desventajas
Motor de base de datos	MySQL	<ul style="list-style-type: none"> Es un software de código abierto. Soporta gran variedad de sistemas operativos. Utiliza la mínima cantidad de recurso para ser ejecutado sin ningún problema. 	<ul style="list-style-type: none"> No es intuitivo como otros programas. Un gran porcentaje de funcionalidades no se encuentran documentadas.
	PostgreSQL	<ul style="list-style-type: none"> Ideal para tecnologías web. Sintaxis SQL estándar y fácil de aprender. Multiplataforma. 	<ul style="list-style-type: none"> Reducida cantidad de tipo de datos. Consume gran cantidad de recursos.
Entorno de desarrollo integrado (IDE)	Eclipse	<ul style="list-style-type: none"> Editor de texto con resaltador de sintaxis. Integración de diversas tecnologías. 	<ul style="list-style-type: none"> Consumo de recursos. Hay que integrar los plugins y módulos independientes.
	NetBeans	<ul style="list-style-type: none"> Entorno más agradable e intuitivo. Cuenta con plugins y módulos integrados. 	<ul style="list-style-type: none"> Una mala implementación de un programa puede derivar en algo muy lento.
Servidor web	Glasfish	<ul style="list-style-type: none"> Superioridad en la administración y la supervisión. Integración total con NetBeans. 	<ul style="list-style-type: none"> Sacrifica velocidad para buscar ser más robusto. Difícil integración con Eclipse.
	Apache	<ul style="list-style-type: none"> Altamente configurable. Multiplataforma. 	<ul style="list-style-type: none"> Falta de integración. Formatos no estandarizados.
Sistema Operativo	Windows	<ul style="list-style-type: none"> Uso del sistema operativo mucho más sencillo. Cuenta con gran cantidad de software desarrollado. 	<ul style="list-style-type: none"> El costo es muy alto. Las versiones requieren muchos recursos.
	Linux Centos	<ul style="list-style-type: none"> Totalmente gratuito. Mayor estabilidad para ser usado en servidores. 	<ul style="list-style-type: none"> No es tan sencillo de utilizar.
Modelador de diagramas UML	Start UML	<ul style="list-style-type: none"> Software libre. Facilidad de creación de nuevos diagramas. 	<ul style="list-style-type: none"> No presenta un resaltador de sintaxis. Sin versión para Linux.
	Microsoft Visio	<ul style="list-style-type: none"> Entorno integrado y fácil de utilizar. 	<ul style="list-style-type: none"> No es gratuito. Falta compatibilidad con otros lenguajes.

		<ul style="list-style-type: none"> • Compatible con otras herramientas. 	
Navegador	Mozilla Firefox	<ul style="list-style-type: none"> • Gran cobertura y bloqueo de Malware. • Funciona en cualquier sistema operativo. 	<ul style="list-style-type: none"> • Extensiones perjudican la velocidad de navegación.
	Google Chrome	<ul style="list-style-type: none"> • Interfaz agradable. • Robusto y seguro. 	<ul style="list-style-type: none"> • Ocupa gran cantidad de memoria RAM.
Entorno de desarrollo integrado (IDE)	Arduino IDE	<ul style="list-style-type: none"> • Multiplataforma. • Entorno de programación simple y directo. 	<ul style="list-style-type: none"> • La configuración cambia dependiendo del tipo del dispositivo Arduino utilizado.
	Minibloq	<ul style="list-style-type: none"> • Crea programas complejos con poco esfuerzo. • Código abierto. 	<ul style="list-style-type: none"> • Difícil identificación de errores de sintaxis.

Realizado Por: Bautista Fausto y Román José. 2017

CAPITULO II

2 MARCO METODOLÓGICO

2.1 Registro de Asistencia

Se lo define como un proceso a cargo y responsabilidad del docente de la asignatura, mismo que se complementa con una persona encargada de la verificación del cumplimiento del mismo durante cada hora de clase durante todo el semestre, esta información se obtiene de una entrevista no estructurada mantenida con el docente tutores del proyecto.

La **Figura 1-2** representa el diagrama del proceso efectuado para el control de la asistencia de los docentes.

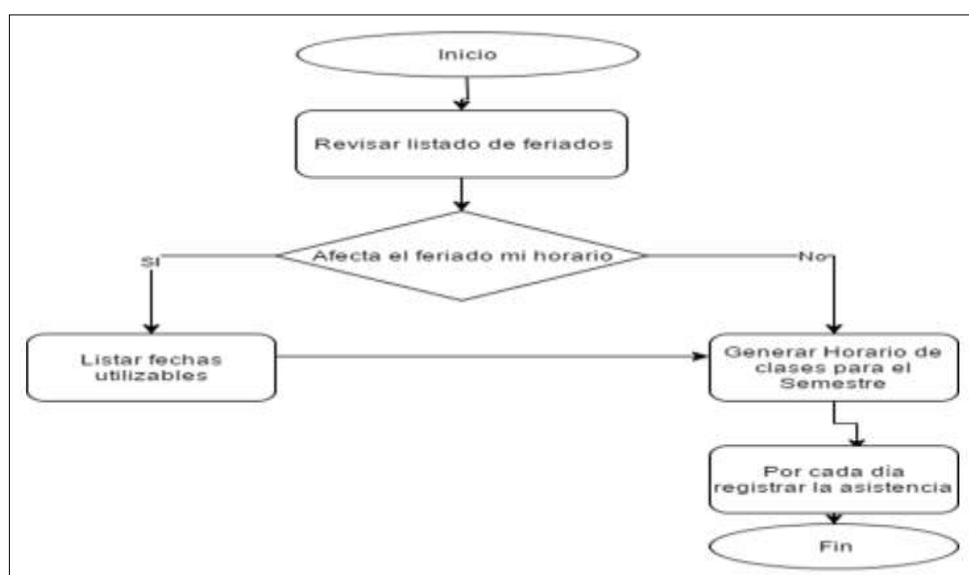


Figura 2-1: Diagrama de Proceso de control de asistencia docente

Realizado Por: Bautista Fausto y Román José. 2017

El gráfico nos demuestra que el proceso de control de asistencia a docentes empieza con la revisión del listado de los feriados involucrados en el periodo académico vigente con el fin de tomar las medidas necesarias para la preparación de las clases. Seguido de la comprobación del respectivo feriado se toma las decisiones, en este caso si el feriado afecta al horario de clases se realiza el listado de las fechas utilizables para las clases, caso contrario se generara el horario de clases para el semestre y registrando la asistencia para cada día de asistencia.

2.2 SCRUM

2.2.1 El Equipo Scrum (Scrum Team)

La **Tabla 1-2** expone al equipo de trabajo SCRUM, definiendo sus funcionalidades como lo sugiere la metodología.

Tabla 1-2: Equipo Scrum

Rol	Persona	Funciones
El Dueño del producto (Product Owner)	Rector de la Escuela Superior Politécnica de Chimborazo	Aportar los recursos humanos y económicos para el desarrollo del proyecto
Clientes	Dr. Julio Santillán	Experto en el proceso que se convierte en la fuente de requisitos
Equipo de Desarrollo (Development Team)	Román José Bautista Fausto	Desarrollar el proyecto
Scrum Master	Dr. Julio Santillán	Asesor experto en el proceso SCRUM

Realizado Por: Bautista Fausto y Román José. 2017

2.2.2. Estudio Preliminar

Con el fin de precautelar los recursos humanos y económicos a invertir por parte tanto de los desarrolladores como del dueño del producto, se decide optar por un estudio de factibilidad que analice si el proyecto está en condiciones de ser ejecutado, se puede consultar en el manual técnico entregado al responsable de los proyectos desarrollados para la Facultad de Informática y Electrónica de la ESPOCH, mismo que se encuentra en el **Anexo D** del presente documento.

El estudio antes mencionado establece que, dado que existe disponibilidad en cuanto al Hardware requerido para el desarrollo por parte de los tesisistas, y que la Facultad de Informática y Electrónica de la ESPOCH está dispuesta a facilitar el hardware requerido para la puesta en producción del sistema, el proyecto se declara factible técnicamente.

Se utiliza un dispositivo lector de tarjetas magnéticas que recibe información almacenada en una tarjeta magnética al deslizar por el módulo RFID. En el **Anexo E** Se explica su funcionamiento, radio de acción operacional, protección de datos, costos, metodología y componentes.

Por otro lado, se estableció un presupuesto de 2204,30 dólares para el desarrollo del sistema, los cuales se distribuyen entre el personal encargado, dado que el presente proyecto servirá como trabajo de titulación para los desarrolladores, los costos generados serán asumidos en su totalidad por el equipo de desarrollo, declarando así al proyecto factible económicamente.

Por cuanto el proyecto se declara como factible tanto técnicamente como económicamente por lo cual se decide iniciar con las fases correspondientes al desarrollo del mismo como apoyo, en la **Tabla 2-2** se exponen las Herramientas a utilizar durante el desarrollo y los materiales utilizados en la realización del lector de tarjetas magnéticas en la **Tabla 3-2**.

Tabla 2- 1: Herramientas de desarrollo

Nombre	Especificación Técnica	Incidencia en el desarrollo del sistema
PostgreSQL 9.4	Motor de base de datos	No se incurrió en gasto para su utilización, debido a que está desarrollado bajo código abierto, por motivos que es multiplataforma no existió inconvenientes al implementarla en Centos7 minimal. A pesar que el tamaño de la base y la cantidad de datos que maneja es grande, presenta estabilidad y un rendimiento constante en la recuperación de información. Cabe recalcar que para realizar todo el esquema de la bases de datos se utilizó el administrador de la bases de datos pgAdmin III, siendo esta una herramienta gráfica y completa, de gran ayuda para la creación de cada uno de los componentes que intervienen en la misma.
Netbeans 8.1	Entorno de desarrollo integrado (IDE)	La utilización de NetBeans en el desarrollo de la aplicación, es por su facilidad de uso y la velocidad con que se puede generar código mediante la combinación de teclas, permitiendo de esta forma tener mayor fluidez en la resolución de cada una de las tareas.
Glassfish 4.0	Servidor web	El servidor de GlasFish se integra de forma correcta y sin complicación alguna con el entorno de desarrollo NetBeans, que a diferencia de otros, se dificulta considerablemente su configuración, además cuenta con una interfaz gráfica para administrar cada una de las aplicaciones a desplegar.
CentOS 7.0	Sistema Operativo	El despliegue del sistema se lo realizó mediante la utilización del sistema operativo Linux CentOS minimal, las configuraciones se las realizaron a través de comandos, pero como ventaja se tiene que existe gran cantidad de documentación en la web, lo cual permitió cumplir con el objetivo, además al ser totalmente gratuito no se tuvo que incurrir en gastos, sin dejar de lado que es uno de los más utilizados por los servidores web.
Star Uml 2.7.0	Modelador de diagramas UML	Existen herramienta para el modelamiento de software basado en los estándares UML más potentes y con una amplia variedad en comparación a la seleccionada, pero fue elegida debido que cumple con cada una de las expectativas al momento de realizar cada uno de los

Mozilla Firefox, Google Chrome	Navegador	modelos requeridos en el proyecto, además de que es un instrumento gratis y su utilización no es de desconocimiento para los desarrolladores. Los dos navegadores web señalados pueden ser utilizados debido que no presentan ningún inconveniente en interpretar cada uno de los componen que conforman los formularios de la aplicación.
Arduino IDE	Entorno de desarrollo integrado (IDE)	La herramienta cuenta con un entorno de desarrollo simple, además utiliza librerías de C++, lo cual permitió sentirnos familiarizados y construir el código para que cada uno de los componentes que integran el dispositivo del lector de tarjetas magnéticas, interpreten en conjunto cada una de las peticiones externas, logrando cumplir con el objetivo en común que es la detección de información guardada en una tarjeta magnética y con ayuda del sistema es guardada en la base de datos.

Realizado Por: Bautista Fausto y Román José. 2017

Tabla 3-2: Materiales para el lector de tarjetas magnéticas

Nombre	Especificación Técnica
Arduino Nano	<ul style="list-style-type: none"> • RESET automático al descargar el programa • LED en la base para indicar el encendido • LED (TX), (RX) y (L) • Regulador de tensión integrado • Conector mini-B USB para programación y comunicación serie integrado en placa • Conector ICSP para programación • Pines con espaciado de 0.1" DIP para insertarlo directamente sobre una protoboard • Botón de reset integrado • Bootloader integrado que permite programarlo con el entorno de desarrollo Arduino sin necesidad de un programador externo
Módulo Wifi ESP8266	<ul style="list-style-type: none"> • Protocolos soportados: 802.11 b/g/n. • Wi-Fi Direct (P2p), Soft Access Point. • Stack TCP/IP integrado. • PLL, reguladores y unidades de manejo de energía integrados. • Potencia de salida: +19.5dBm en modo 802.11b. • Sensor de temperatura integrado. • Consumo en modo de baja energía: <10 uA.
Módulo RFID-RC522 RF	<ul style="list-style-type: none"> • Modelo: MF522-ED. • Corriente de operación: 13-26mA a 3.3V. • Corriente de stand by: 10-13mA a 3.3V. • Corriente de sleep-mode: <80uA. • Corriente máxima: 30mA. • Frecuencia de operación: 13.56Mhz. • Distancia de lectura: 0 a 60mm. • Protocolo de comunicación: SPI.

La **Tabla 4-2** detalla los Tipos de Usuario relacionándolos con las funciones que cumple cada uno de estos.

Tabla 4- 2: Tipos y Roles de Usuario

Tipo de Usuario	Rol
Administrador	<ul style="list-style-type: none"> • Gestionar Períodos. • Gestionar Feriados. • Gestionar Metadatos • Gestionar Identificador de Tarjeta para cada uno de los Docentes. • Migrar Datos • Generar Reportes
Docente Estudiante	<ul style="list-style-type: none"> • Registrar tutorías • Gestionar el horario de clases para todo el semestre por asignatura. • Generación de Documento del horario de clases por asignatura • Subir evidencia sobre las clases de ese día en curso, en caso de que se le sea solicitado.

Realizado Por: Bautista Fausto y Román José. 2017

Dentro del rol administrador, cabe mencionar la diferenciación interna que se hará entre el administrador general y el administrador de cada escuela, siendo los 2 similares en funciones pero no en el grupo a manejar.

2.2.3. *Requerimientos*

Basados en lo expresado en la **Tabla 1-2**, se definen 3 Clientes mismos que se encargarán de expresar las necesidades requeridas o deseadas para el sistema, los cuales con el apoyo de los desarrolladores asesorarán a la encaminación de los mismos para que se conviertan en requerimientos del sistema.

Siguiendo las especificaciones de SCRUM, el Product Backlog se utiliza como artefacto para el control del trabajo a realizar, la misma contiene 27 funcionalidades solicitadas por el usuario apoyado de los desarrolladores, por lo cual la **Tabla 5-2** muestra la lista de los mismos en el orden que fueron apareciendo.

Tabla 5-2: Product Backlog

Historia	Nombre de la Historia
1	Iniciar sesión a los usuarios.
2	Registrar los datos de un feriado.
3	Modificar los datos de un feriado.

4	Eliminar los datos de un feriado.
5	Registrar los metadatos del sistema.
6	Modificar los metadatos del sistema.
7	Eliminar los metadatos del sistema.
8	Registrar los datos de las tutorías por asignatura.
9	Registrar identificador de tarjeta magnética del docente.
10	Modificar identificador de tarjeta magnética del docente.
11	Eliminar identificador de tarjeta magnética del docente.
12	Registrar los datos de horarios de clase.
13	Modificar datos de horario de clase.
15	Registrar evidencias de clases en caso de ser solicitadas.
16	Generar reporte de asistencia por docente.
17	Generar reporte de asistencia por escuela.
18	Generar reporte de asistencia por facultad.
19	Generar reporte de asistencia por carrera.
20	Generar reporte de asistencia por materia.
21	Generar el reporte del horario de clases.
22	Registrar los datos de un período.
23	Modificar los datos de un período.
24	Eliminar los datos de un período.
25	Registrar justificaciones de inasistencias
26	Modificar justificaciones de inasistencias
27	Eliminar justificaciones de inasistencias

Realizado Por: Bautista Fausto y Román José. 2017

El siguiente artefacto que menciona SCRUM es el Sprint Backlog o Pila del Sprint.

Como técnica de estandarización, se utilizan los puntos de función, para conseguir medir el avance del proyecto en relación al esfuerzo y no al tiempo, dado que por condiciones de los desarrolladores el horario de trabajo puede variar, lo que heterogenizaría los valores de estimaciones tanto como de avances.

Por experiencia de los desarrolladores se opta por el método denominado talla de camiseta sobre otros como COCOMO, dado que la metodología expresa que se oriente a la optimización del tiempo y estudiar la metodología COCOMO hubiese requerido un tiempo que al utilizar talla de camiseta fue ahorrado.

Como lo menciona el método seleccionado se inicia desde la parte más baja de la escala determinando así como talla S 5 puntos de función, a continuación se establece la siguiente talla que será, M con 10 puntos, decisión que se toma por la facilidad matemática de llevar estos cálculos, y como talla L se establecen 20 puntos que de la misma manera son el doble que su talla antecesora, como lo sugiere el método se tomará un Sprint como 1 tallas L y 1 talla M o su correspondiente variación con tallas más pequeñas.

Para el desarrollo, un punto de función será equivalente a 90 minutos de trabajo, los días laborables para el desarrollo fueron: lunes, martes, miércoles, jueves y viernes, el horario de trabajo se establece de 15:00- 18:00, lo que representa 3 horas laborables por cada día, resultando 15 horas laborables a la semana, el tamaño del Sprint de 30 puntos de función o 45 horas de trabajo, se establece la duración del Sprint en 3 semanas.

Como siguiente artefacto tenemos al Sprint Backlog el cual representa al trabajo a realizar durante un Sprint, en el proyecto resultan 13 Sprint Backlog, los cuales se organizan para ajustarse a lo que la metodología establece, debido a que cada Sprint debe tener la misma duración y tratar temas relacionados.

Con orden cronológico y de necesidad se establece tratar las metáforas del sistema en el primer y último Sprint, quedando los restantes netamente al desarrollo del sistema, estimando en total 390 puntos de función. La **Tabla 6-2**, muestra todos los Sprint Backlog que contiene el proyecto.

Tabla 6-2: Product Backlog por Sprint

Sprint	Historia	Nombre de la Historia	Puntos
1	MT_01	Diseño de la base de datos	5
	MT_02	Definición de la arquitectura del sistema.	5
	MT_03	Diseño de las interfaces.	5
	HU_01	Iniciar sesión a los usuarios.	5
	HU_02	Registrar los datos de un feriado.	10
2	HU_03	Modificar los datos de un feriado.	10
	HU_05	Registrar los datos de horarios de clase.	20
3	HU_04	Eliminar los datos de un feriado.	10
	HU_06	Registrar los metadatos del sistema.	20

4	HU_07	Modificar los metadatos del sistema.	10
	HU_08	Eliminar los metadatos del sistema.	10
	HU_09	Registrar los datos de un período.	10
5	HU_10	Modificar los datos de un período.	10
	HU_11	Eliminar los datos de un período.	20
6	HU_12	Registrar los datos de las tutorías por asignatura.	20
	HU_13	Registrar identificador de tarjeta magnética del docente.	10
7	HU_14	Modificar identificador de tarjeta magnética del docente.	10
	HU_15	Registrar justificaciones de inasistencias	10
	HU_16	Registrar evidencias de clases en caso de ser solicitadas.	10
8	HU_17	Modificar datos de horario de clase.	20
	HU_18	Generar el reporte del horario de clases	10
9	HU_19	Generar el reporte de asistencia por materias.	10
	HU_20	Generar el reporte de asistencia por docente.	20
10	HU_21	Generar el reporte de asistencia por escuela.	10
	HU_22	Generar el reporte de asistencia por facultad.	10
	HU_23	Generar el reporte de asistencia por carrera.	10
11	HU_24	Realizar lector de tarjetas magnéticas.	30
12	HU_25	Modificar justificaciones de inasistencias	20
	HU_26	Eliminar justificaciones de inasistencias	10
13	MT_04	Desplegar el sistema en el servidor de aplicaciones de la facultad	20
	MT_05	Implantar la base de datos en el servidor de base de datos de la facultad	10

Realizado Por: Bautista Fausto y Román José. 2017

Se estiman 390 puntos de función, distribuidos uniformemente en 13 Sprints, tal como lo menciona el **Análisis de Riesgos** del manual técnico el tiempo estimado puede verse influenciado por factores como los cambios constantes en los requerimientos así como la dificultad presentada en el manejo de las tecnologías involucradas, lo cual establece 400 puntos de función como una estimación pesimista.

Ya terminada la requerimentación el siguiente punto a implementar según la metodología es la fase de desarrollo de cada uno de los Sprint, la información completa sobre ésta se establece en el Manual Técnico entregado al encargado de los proyectos de la Facultad, mismo que se encuentra en el **Anexo D** del presente documento, a continuación se demuestra los resultados obtenidos durante la implementación de la misma.

2.2.4. *Sprint 1*

Como se estableció anteriormente, este Sprint estará enfocado a las Metáforas del Sistema, las cuales dieron como resultado la base de datos a utilizar, se tiene la **Figura 2-2**, que representa el modelo conceptual de la misma, resultado que fue obtenido por el desarrollo de la Metáfora del Sistema **HT_01**, las tareas de ingeniería, así como las pruebas de aceptación y demás artefactos relacionados que se produjeron, se encuentran detallados en el Manual Técnico mismo que se encuentra en el **Anexo D** del presente documento, los demás entregables de esta historia como el diccionario de datos, se lo puede encontrar en el **Anexo A**.

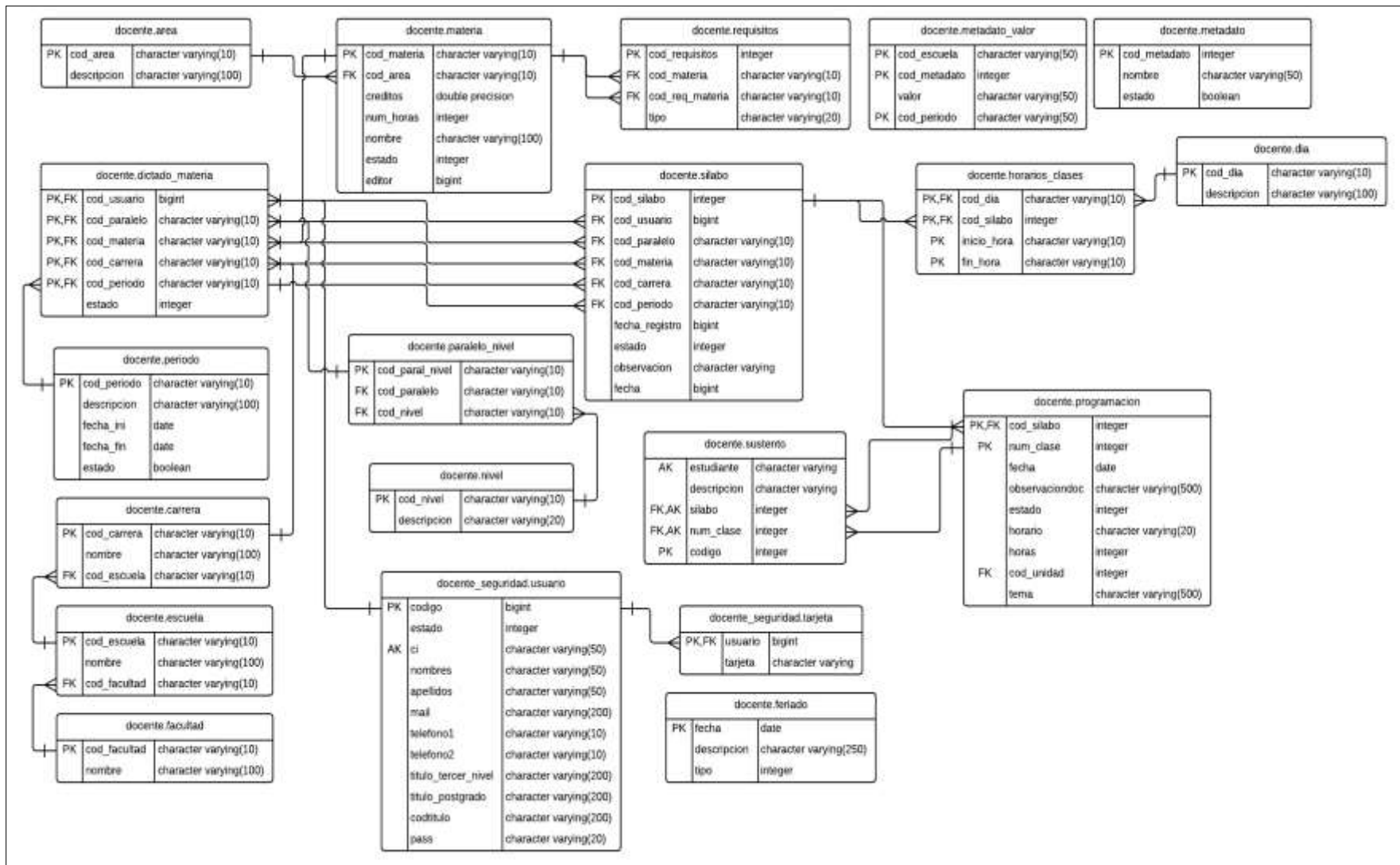


Figura 2-2: Diagrama Conceptual de la base de datos

Realizado Por: Bautista Fausto y Román José. 2017

2.2.5. *Sprint 2 - 12*

Los Sprints ubicados entre 2 y el 12 se enfocaron al desarrollo en sí del sistema (codificación) logrando así como lo establece la metodología en cada uno de estos una versión funcional de los requerimientos tratados hasta ese momento, a continuación se explican los artefactos involucrados en cada paso del desarrollo y los adicionales se ubican en el manual técnico correspondiente, mismo que se encuentra en el **Anexo D** del presente documento.

2.2.5.1. *Historias de Usuario*

Se utilizan los prefijos HU_ y HT_ para representar las Historias de Usuario y las Metáforas de Sistema respectivamente seguidas del número correspondiente y el nombre representativo referente a la funcionalidad a implementar.

Cada Historia de usuario esta descrita por los elementos mencionados en la **Tabla 7-2**, especificando que todas las generadas siguen los mismos parámetros y se encuentran en el Manual Técnico, mismo que se encuentra en el **Anexo D** del presente documento.

Tabla 7-2: Formato Historia de Usuario

Historia de Usuario	
HT_01	Diseñar Base de Datos
Modifica:	
Usuario: Desarrollador	Sprint Asignada: 1
Prioridad Negocio: Alta	Puntos Estimados: 2.5
Riesgo Desarrollo: Media	Puntos Reales: 2.5
Descripción: Como desarrollador deseo diseñar una base de datos para mantener los datos disponibles en el transcurso del tiempo.	
Observaciones:	
<ul style="list-style-type: none">• Utilizar motor de base de datos postgresQL.• Generar diagrama entidad relación de la base terminada.• Generar diccionario de datos correspondiente.	

Realizado Por: Bautista Fausto y Román José. 2017

Considerando lo especificado por la metodología, Las Historias se convierten en tareas al comienzo del Sprint.

Tareas de ingeniería.- Se definen como la unidad fundamental en que se descompone una Historia de Usuario.

En la **Tabla 8-2** se ejemplifica la TI_05 correspondiente a la HT_01 como muestra del formato que se siguió para cada una de las adjuntas en el **manual técnico**, mismo que se encuentra en el **Anexo D** del presente documento.

Tabla 8-2: Formato Tarea de Ingeniería

TAREA DE INGENIERÍA	
Historia de Usuario: MS_01 Diseño de la Base de Datos	
TI_2	Nombre de Tarea: Diccionario de datos
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Programador Responsable: Bautista Fausto	
Descripción: La creación del diccionario de datos con sus respectivos tipos de datos y sus restricciones.	
Pruebas de Aceptación; Que java soporte todos los tipos de datos definidos o sean compatibles.	

Realizado Por: Bautista Fausto y Román José. 2017

Como se puede observar al final de la tarea de Ingeniería aparece un término nuevo denominado Prueba de Aceptación, misma que se ejemplifica en la **Tabla 9-2**, y se consideran los lineamientos que debe seguir una tarea de ingeniería para considerarse correctamente realizada.

Tabla 9-2: Formato Prueba de Aceptación

Prueba de Aceptación	
PA_05	Historia de Usuario: Diseño de Base de Datos
Nombre: Que java soporte todos los tipos de datos definidos o sean compatibles	
Responsable: Román José	
Descripción: Todos los tipos de datos evidenciados en el diccionario de datos deben ser compatibles o existir en JAVA.	

Condiciones de Ejecución: El responsable comparará visualmente cada uno de los tipos de datos con los soportados de java
Pasos de ejecución: Ordenar alfabéticamente las tablas en el diccionario. Ordenar los campos por tipo de dato. Comparar en orden la igualdad entre los elementos.
Resultado esperado: Todos los tipos de datos son soportados.
Evaluación de la prueba: Exitosa.

Realizado Por: Bautista Fausto y Román José. 2017

Continuando con la metodología, ésta establece a los casos de uso como herramientas para el análisis de los requerimientos a desarrollar y que se deben crear al inicio del Sprint, por cuanto se tiene en las **Figuras 3-2, 4-2 y 5-2** los procesos de cada uno de los actores del sistema.

Se ha elegido como herramienta los casos de uso y como lo sugiere se han tomado las siguientes consideraciones:

- Los actores están representados por un monigote con el nombre del actor al pie del gráfico, estos representan un rol que el usuario adquiere con respecto al sistema.
- Los casos de uso están representados por una elipse y en su interior el texto con el nombre de la operación específica.

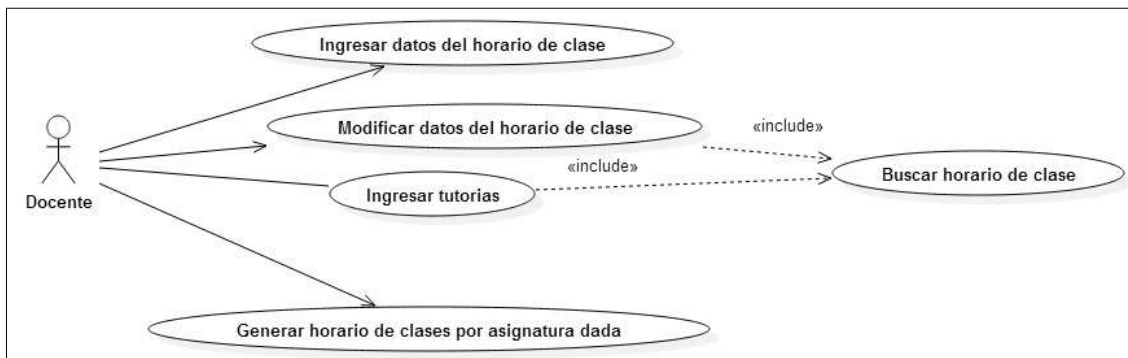


Figura 3-2: Diagrama de Caso de Uso del usuario docente

Realizado Por: Bautista Fausto y Román José. 2017

CAPITULO III

3 MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS DE RESULTADOS

Con el sistema en el estado explicado en el **CAPÍTULO II**, se procede inicialmente al análisis de los resultados obtenidos desde el punto de vista técnico, en cuanto a la implementación de la metodología, para posteriormente enfocarse al levantamiento de información para el análisis en uso.

3.1. Gestión del Proyecto

El **Gráfico 1-3**, representa el artefacto sugerido por la metodología para este análisis, el Burndown Chart nos muestra el avance del esfuerzo restante a implementar para el desarrollo del proyecto, contrastándolo con el estimado, representado así el éxito o fracaso tanto de la estimación como del proyecto en sí mismo.

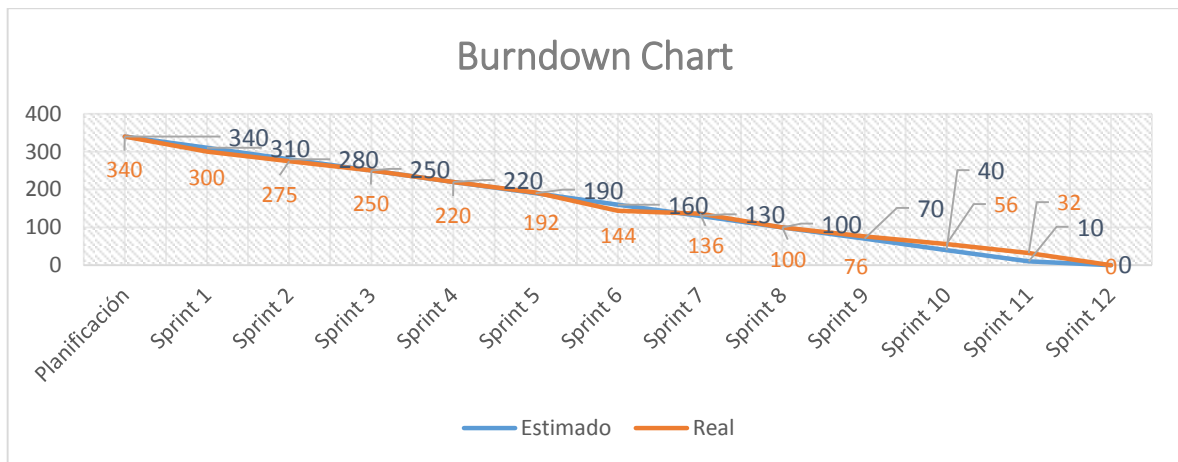


Gráfico 1-3: Burndown Chart

Realizado Por: Bautista Fausto y Román José. 2017

Como lo explica el **Gráfico 1-3**, el proyecto se encuentra totalmente terminado en sus 390 puntos de función iniciales al momento 0, luego de lo cual presenta valores representativos relacionales al avance del desarrollo, se determina una buena estimación inicial por cuanto a la cercanía entre las 2 líneas graficadas.

3.2. Análisis de Resultados

3.2.1. *Mejora en los procesos*

Para la demostración se plantea un entorno de pruebas controlado en el cual se contrastarán las 2 maneras de realizar el proceso, una por parte de los desarrolladores y otra por parte del responsable de realizarlos.

Cabe mencionar que por solicitud expresa del Vicedecano de la Facultad de Informática y Electrónica la prueba de campo y el levantamiento de información se realizan exclusivamente a la escuela de Ingeniería en Sistemas, autorizando que con esta se realicen los cálculos matemáticos necesarios para extrapolarla a las otras escuelas de la Facultad.

Se establecen los siguientes parámetros con el fin contrastar la información generada por las 2 formas de realizar el proceso:

- **Objeto de Experimentación:** Sistema de Control de Asistencia Docente.
- **Sujetos de Experimentación:** Docentes de la Escuela de Ingeniería en Sistemas con clase en los laboratorios seleccionados en el período de tiempo especificado
- **Población:** 10 Docentes

3.2.2. *Métricas*

Para el levantamiento de información se realiza una encuesta al responsable de la generación de los informes de asistencia docente en la cual como lo evidencia el **Anexo C**, declara que el tiempo que se toma en realizar el reporte de la escuela de Ingeniería en Sistemas es de 10 horas aproximadamente, por lo cual estadísticamente se define una significancia de 5% que será llevada a través de toda la medición.

3.2.2.1. *Muestra*

Dado que el Vicedecano de la facultad expresamente solicitó que el estudio se realice en 1 laboratorio de la Escuela de Ingeniería en Sistemas y que los sujetos de experimentación sean todos los docentes

que durante la jornada de los días de prueba laboren ahí, se establece como muestra arbitraria por solicitud externa un total de 10 docentes.

3.2.2.2. Medidas

Como se menciona anteriormente, los cálculos fuera de la escuela de Ingeniería en Sistemas se los realizará por métodos matemáticos de extrapolación, por cuanto se tienen los siguientes datos.

Tiempo requerido para generar el reporte de asistencia docente por escuela 600 minutos

Cantidad de créditos en la escuela 267

Semanas de clase por semestre 16

Con lo cual se procede a establecer que durante el semestre se imparten $267 * 16 = 4272$ horas de clase, por lo cual el encargado en 600 minutos de trabajo analiza las incidencias de 4272 clases, lo que nos especifica un total de 7.12 segundos por hora de clase analizada, teniendo en cuenta que este proceso excluye la clasificación y la toma de información.

De lo anterior se concluye que por cada hora de clase del semestre es necesario en promedio 7.12 segundos, por lo cual se establecen los valores para cada una de las escuelas siguiendo esta tendencia.

La **Tabla 1-3** especifica los valores resultantes en relación al número de créditos en la malla de cada escuela

Tabla 1-3: Tiempo por escuela

	Escuela	Semanas	Créditos	Tiempo (s)	Tiempo Entero
1	E. Telecomunicaciones	16	226	25745,92	25746
2	E. Diseño Grafico	16	221	25176,32	25176
3	E. Sistemas	16	267	30416,64	30417
4	E. Control	16	231	26315,52	26316

Realizado Por: Bautista Fausto y Román José. 2017

Tomando como base el cálculo anterior que especifica, que por cada clase del semestre se requieren 8.43 segundos para generar el reporte y utilizando la función techo que es la recomendada para manejo de tiempos se obtiene la columna Tiempo Entero que será utilizada para el procesamiento de la información, un dato extra implica que la suma de todos estos valores representa el valor requerido para generar el reporte de toda la Facultad, así como de igual manera contando la cantidad de créditos que imparte un docente en cada una de sus materias tendríamos el equivalente a generar el reporte de cada docente de la Escuela.

Por lo antes mencionado, la **Tabla 2-3** establece los valores a comparar con el uso del Sistema.

Tabla 2-3: Tiempos a comparar

	Tiempo Entero	Tiempo Sistema
Reporte por Facultad	107654	6.40
Reporte para la escuela de Telecomunicaciones	25746	3.20
Reporte para la escuela de Diseño Gráfico	25176	3.00
Reporte para la escuela de Sistemas	30417	4.20
Reporte para la escuela de Control	26316	3.60

Realizado Por: Bautista Fausto y Román José. 2017

Como se puede observar en la columna tiempo sistema, no conlleva una relación directamente proporcional a la cantidad de créditos de la malla curricular, así que se estima que éste conjunto de datos no siguen una distribución normal, por tanto se decide optar por una herramienta de medición no probabilística para realizar el análisis dado que al implementar una como Anova o T-Student orientada a datos probabilísticos no nos resulte indicada por la característica de los datos, lo que incurriría en una pérdida de recursos, por experiencia personal se decide por la prueba de los pares relacionados de Wilcoxon, dado que cumple las características requeridas como son, es una prueba no probabilística y funciona para la determinación estadística de la existencia de una diferencia real entre 2 grupos de valores relacionados.

Para la implementación de la prueba, se define:

H0 = los datos entre los procesos no presentan diferencia significativa (hipótesis nula)

H1 = los datos entre los procesos si presentan diferencia significativa (hipótesis del investigador)

La **Tabla 3-3**, muestra de forma organizada los valores requeridos para el análisis de Wilcoxon

Tabla 3- 1: Tiempos a comparar con Wilcoxon

Tiempo Entero (s)	Tiempo Sistema (s)	Signo 1 + -1 = -	Diferencia absoluta (seg)	Rango	Rango con signo
107654	6.40	1	107647.6	5	5
25746	3.20	1	25742.8	2	2
25176	3.00	1	25173	1	1
30417	4.20	1	30412.8	4	4
26316	3.60	1	26312.4	3	3

Realizado Por: Bautista Fausto y Román José. 2017

Implementamos en la herramienta Wilcoxon la tabla resultante y obtenemos los valores significativos de Sum of pos. Ranks 15 y Z-value = -2.0226, valores que son los que debemos utilizar según las indicaciones de la herramienta que indican.

En caso de tener menos de 20 pares de datos, la comprobación se realizará tomando en cuenta el valor de Z-Valué, por tanto, éste valor debe ser comparado con la significancia establecida, en nuestro caso 0.05, con lo cual se tiene que:

$$-2.0226 < 0.05$$

Por lo tanto al ser una afirmación verdadera significa que debemos aceptar la hipótesis del investigador y rechazar la hipótesis nula, por lo que se define que si existe una diferencia significativa entre los 2 conjuntos de pares de datos, por cuanto se procede a calcular el valor de la misma.

Para el cálculo de eficiencia en cuanto al reporte de Facultad se toma la fila 1, la que establece:

107654,4segundos de ahorro en relación a los 107654 segundos originales, lo que representa un ahorro del 99,993% del tiempo requerido.

Para el seguimiento por escuelas se utilizará la escuela de Ingeniería en Sistemas como valor referencial dado que en esta se levantó la información Real.

En este caso se tiene:

30416,64 segundos de ahorro en relación a 30417 segundos originales, lo que representa un 99,977% de ahorro en cuanto al tiempo requerido en la generación del reporte.

Cabe mencionar adicionalmente que este proceso no toma en cuenta el proceso de organización de la información, debido que aquí también se tiene un ahorro no medido pero existente, dado que por definición una base de datos sirve para mantener los datos organizados y perdurables en el tiempo.

Además cabe mencionar que el reporte por docente en la actualidad no se realiza así que al no existir algún dato inicial con el cual comparar este proceso no puede ser realizado.

3.2.3. Norma ISO/IEC 9126

La norma ISO/IEC 9126 evalúa la calidad de uso del software, mismo que comprende 6 parámetros mencionados anteriormente, pero en este caso de estudio se pretende evaluar la Eficiencia del software de control de asistencia de docentes.

Los puntos que se toman en consideración para evaluar la eficiencia son:

- Utilización de Recursos
- Comportamiento de Tiempos

Obteniendo así la **Tabla 4-3**, que representa las preguntas generales de las características a medir.

Tabla 4-3: Características de la ISO/IEC 9126

Característica	Pregunta	Subcaracterística	Pregunta
Eficiencia	¿Es más rápido y eficiente en cuanto al uso de recursos?	Utilización de Recursos	¿Cuántos recursos utiliza en RAM y disco duro?

Realizado Por: Bautista Fausto y Román José. 2017

Para el punto Utilización de recursos se utilizan 4 funcionalidades con la que cuenta el sistema, mismas que serán tomadas en el servidor de producción en el que se encuentra actualmente funcionando a través de una conexión ssh al mismo y haciendo uso de la línea de comandos para no generar un gasto extra de recursos destinados a sostener una Interfaz Gráfica de Usuario (GUI), cada una de las mediciones será evidenciada con una imagen adjunta del valor evaluado.

- Reporte de asistencias por Facultad (el más extenso)
- Carga de Feriados
- Generación del Horario de Clases del Docente
- Reporte de planificación semestral

Para la Utilización de Recursos, se toman como parámetros a medir la Memoria RAM y el espacio en disco utilizado por los datos almacenados.

Tomando en cuenta lo anterior mencionado, siendo las 9 am del día 24 de junio de 2017 en la Escuela Superior Politécnica de Chimborazo en el Edificio de la Escuela de Ingeniería en Sistemas conectados a la red Wifi “ESPOCH-OPEN” se toman las medidas evidenciadas a continuación.

Uso de memoria RAM

- Reporte de Asistencia por Facultad

La **Figura 1-3**, nos muestra que antes de la ejecución de la prueba se tenían 299820 kb libres de memoria RAM, posteriormente la **Figura 2-3**, nos muestra que durante el tiempo comprendido entre la solicitud del reporte y la entrega del mismo se alcanza un pico de 292504 kb libres, por lo cual el mayor uso de RAM en este proceso es de 7316 kb, mismo que en el caso de mayor uso del sistema con un reporte general de las 12 Facultades existentes alcanzaría un máximo de 87792 kb, cantidad que se encuentra disponible y es manejable por el servidor dado que representa un 0.387% del total disponible y un 2.44% del total utilizable.

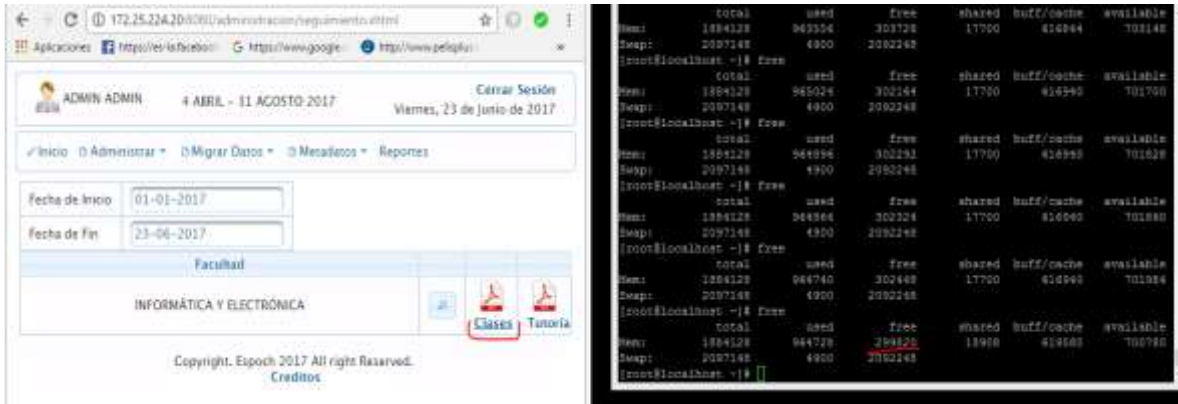


Figura 1-3: RAM antes del reporte por Facultad

Realizado Por: Bautista Fausto y Román José. 2017

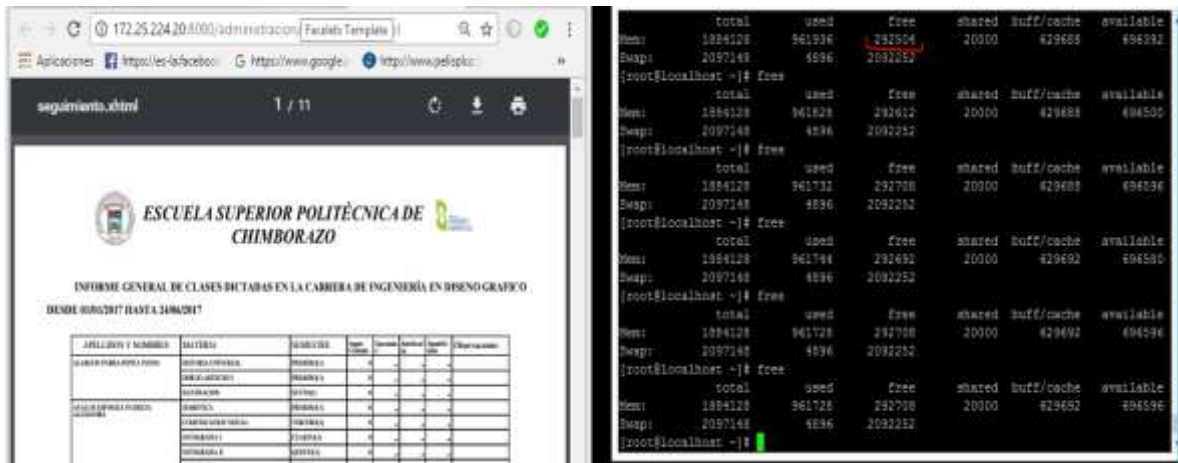


Figura 2-3: Pico en reporte por Facultad

Realizado Por: Bautista Fausto y Román José. 2017

- Carga de Feriados

La **Figura 3-3**, nos muestra que antes de la ejecución de la prueba se tenían 292708 kb libres de memoria RAM, posteriormente la **Figura 4-3**, nos muestra que durante el tiempo comprendido entre la solicitud y la entrega de la información se alcanza un pico de 289744 kb libres, por lo cual el mayor uso de RAM en este proceso es de 2964 kb, misma que es referente a todos los feriados existentes en el sistema para el período actual en la facultad.

```

root@localhost:~# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961744        292692        20000        629692        696580
Swap:          2097148           4896        2092252

[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961728        292708        20000        629692        696596
Swap:          2097148           4896        2092252

[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961728        292708        20000        629692        696596
Swap:          2097148           4896        2092252

[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961888        292712        20032        629528        696628
Swap:          2097148           4896        2092252

```

Figura 3-3: RAM antes de solicitar los Feriados

Realizado Por: Bautista Fausto y Román José. 2017

```

root@localhost:~# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961744        292692        20000        629692        696580
Swap:          2097148           4896        2092252

[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961728        292708        20000        629692        696596
Swap:          2097148           4896        2092252

[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961728        292708        20000        629692        696596
Swap:          2097148           4896        2092252

[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961888        292712        20032        629528        696628
Swap:          2097148           4896        2092252

[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961900        292820        20120        629408        696800
Swap:          2097148           4896        2092252

[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:           1884128        961944        289744        20140        632440        698832
Swap:          2097148           4892        2092252

```

Figura 4-3: Pico en obtener los Feriados

Realizado Por: Bautista Fausto y Román José. 2017

- Generación del Horario de Clase del Docente

La **Figura 5-3**, nos muestra que antes de la ejecución de la prueba se estaban utilizando 961904 kb libres de memoria RAM, posteriormente la **Figura 6-3**, nos muestra que durante el tiempo comprendido entre la solicitud y la entrega de la información se alcanza un pico de 961896 kb utilizados, por lo cual el mayor uso de RAM en este proceso es de 8 kb, misma que es referente a todas las clases que deberá impartir el docente en la materia seleccionada durante las 16 semanas que durará el semestre.

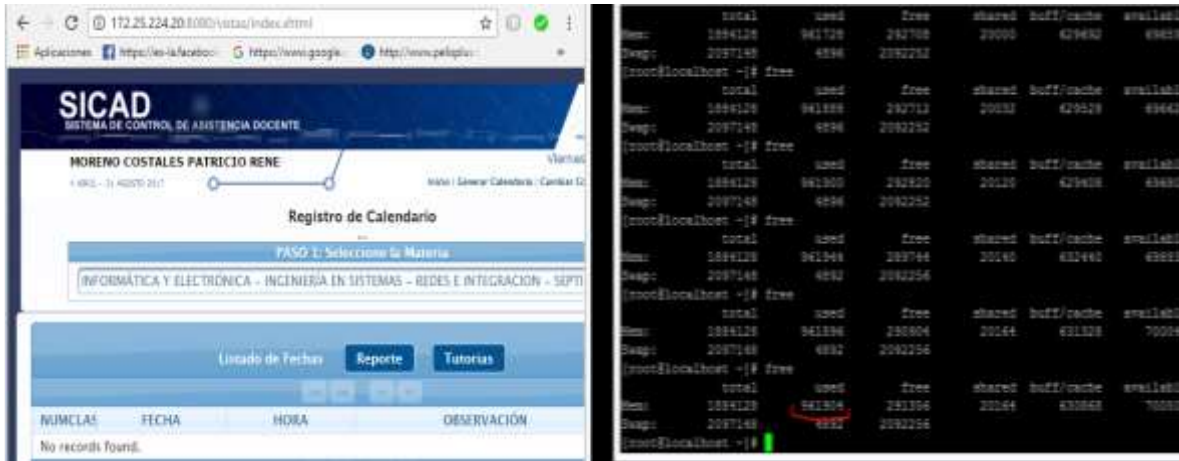


Figura 5-3: RAM antes del solicitar el Horario de una Materia

Realizado Por: Bautista Fausto y Román José. 2017

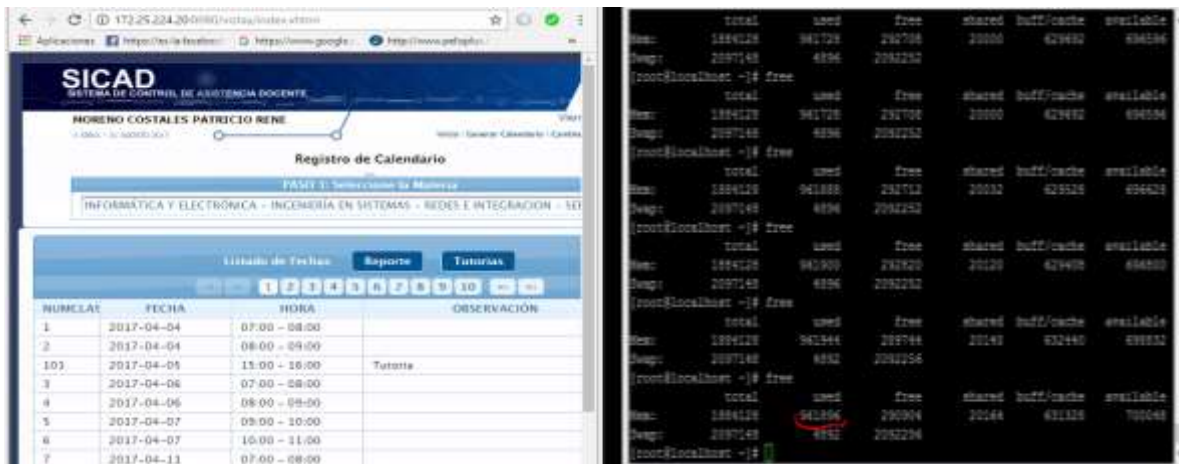


Figura 6-3: Pico en obtener el Horario

Realizado Por: Bautista Fausto y Román José. 2017

- Generación del Reporte de Planificación Semestral

La **Figura 7-3**, nos muestra que antes de la ejecución de la prueba se estaban utilizando 962032 kb libres de memoria RAM, posteriormente la **Figura 8-3**, nos muestra que durante el tiempo comprendido entre la solicitud y la entrega de la información se alcanza un pico de 961904 kb utilizados, por lo cual el mayor uso de RAM en este proceso es de 128 kb, misma que es referente a todas las clases que deberá impartir el docente en la materia seleccionada durante las 16 semanas que durará el semestre y todas las observaciones agregadas por el docente a cada una de las clases.

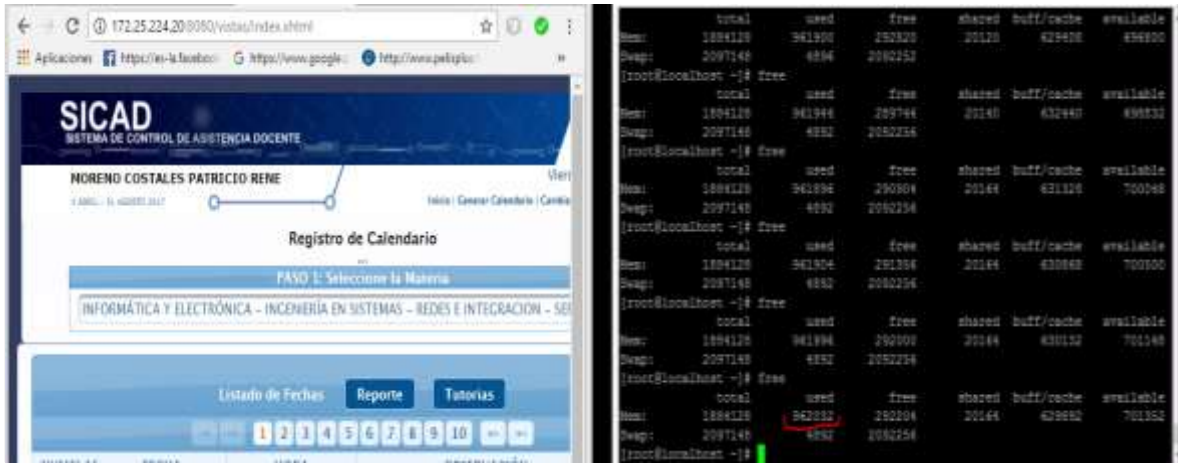


Figura 7-3: RAM antes del solicitar el Reporte

Realizado Por: Bautista Fausto y Román José. 2017

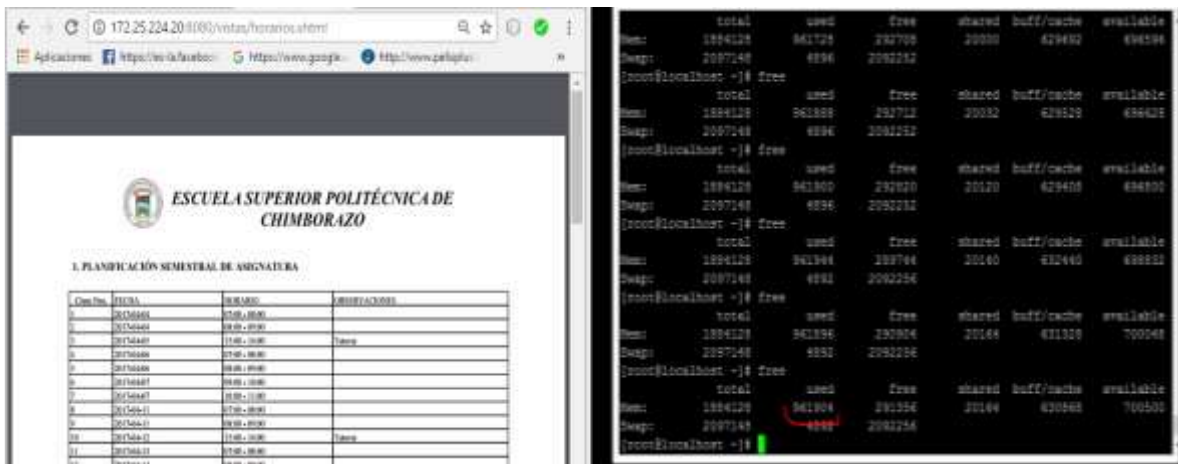


Figura 8-3: Pico en obtener el Reporte

Realizado Por: Bautista Fausto y Román José. 2017

- Uso de Espacio en Disco Duro

En el ambiente de pruebas se genera un backup plano de toda la información almacenada en la base de datos como se puede observar en la **Figura 9-3**, con el nombre asisdocente, mismo que contiene la información de los 145 Usuario **Figura 10-3** y toda la información relacionada a ellos y al negocio como se puede apreciar en la **Figura 11-3**, resultando con un tamaño total de 998 kb.



Figura 9-3: Backup completo

Realizado Por: Bautista Fausto y Román José. 2017

codigo	estado	ci	apellido	apellido	mail
108	1	1	RODRIGO DEL PILAR	RODRIGO DEL PILAR	
109	1	1	FRANCISCO MARTIN	FRANCISCO MARTIN	
120	1	1	MARCELO DE JESUS	MARCELO DE JESUS	
121	1	1	MARCELO ALFARERA	MARCELO ALFARERA	
122	1	1	FABRIZIO ANTONI	FABRIZIO ANTONI	
123	1	1	DIEGO FERRARO	DIEGO FERRARO	
124	1	1	GIORGIO ANTONI	GIORGIO ANTONI	
125	1	1	FABRIZIO ANTONI	FABRIZIO ANTONI	
126	1	1	FRANCESCO ANTONI	FRANCESCO ANTONI	
127	1	1	DIEGO	DIEGO	
128	1	1	JOSÉ LUIS	JOSÉ LUIS	
129	1	1	MARCELO DE JESUS	MARCELO DE JESUS	
130	1	1	JOSÉ LUIS	JOSÉ LUIS	
131	1	1	ANGELICA GONZALEZ	ANGELICA GONZALEZ	
132	1	1	ANGELICA	ANGELICA	
133	1	1	MARCELO	MARCELO	
134	1	1	JOSÉ LUIS	JOSÉ LUIS	
135	1	1	DIEGO FERRARO	DIEGO FERRARO	
136	1	1	JOSÉ LUIS	JOSÉ LUIS	
137	1	1	ANGELICA	ANGELICA	
138	1	1	ANGELICA	ANGELICA	
139	1	1	ANGELICA	ANGELICA	
140	1	1	ANGELICA	ANGELICA	
141	1	1	ANGELICA	ANGELICA	
142	1	1	ANGELICA	ANGELICA	
143	1	1	ANGELICA	ANGELICA	
144	1	1	ANGELICA	ANGELICA	
145	1	1	ANGELICA	ANGELICA	

Figura 10-3: Cantidad de Usuarios

Realizado Por: Bautista Fausto y Román José. 2017

Nombre	Fecha de modifica...	Tipo	Tamaño
asisdocente.backup	23/06/2017 10:57	Archivo BACKUP	998 KB

Figura 11-3: Tamaño Total

Realizado Por: Bautista Fausto y Román José. 2017

Tomando en cuenta todo lo mencionado, se resume en la **Tabla 5-3** el uso de recursos de los procesos analizados.

Tabla 5-3: Uso de Recursos

Subcaracterística	Recursos disponibles	Uso de Recursos	Uso de Recursos (porcentaje)
Reporte de asistencias por Facultad (el más extenso)	299820 kb en RAM	7316 kb en RAM	2.44%
Carga de Feriados	292708 kb en RAM	2964 kb en RAM	1.01%
Generación del Horario de Clases del Docente	961904 kb en RAM	8 kb en RAM	0.00083%
Reporte de planificación semestral	962032 kb en RAM	128 kb en RAM	0.01%
Tamaño de base de datos	1887436.8 kb en Disco Duro	998 kb en Disco Duro	0.52%

Realizado Por: Bautista Fausto y Román José. 2017

CONCLUSIONES

- Se utiliza tecnologías como: patrón de arquitectura de software MVC, Frameworks JSF (Java Server Faces), metodología SCRUM para guiar el proceso de desarrollo de software, CentOS 7 como sistema operativo, Netbeans 8.1 entorno de desarrollo, lenguaje de programación JAVA, servidor Glassfish 4.0, base de datos PostgreSQL 9.4 con su respectivo administrador pgAdmin III, Arduino IDE como entorno de desarrollo de la programación del arduino, Start UML 2.7 encargado de realizar los diagramas de caso de uso, navegadores web Google Chrome y Mozilla Firefox. Estas tecnologías fueron analizadas acorde a las necesidades y exigencias del entorno de trabajo, lo cual permite integrarlas entre sí para desarrollar el proyecto software.
- Para el sistema de control de asistencia, se integra un dispositivo de lector de tarjetas magnéticas que permite registrar el tiempo de entrada y salida de horas clase de los docentes, una vez que los datos son receptados pasan a ser procesados por el sistema convirtiéndolos en reportes, proceso que fue realizado en el levantamiento de información en el laboratorio de computo de la Escuela de Ingeniería en Sistemas a los docentes de las diferentes materias.
- Dentro del análisis en base a los resultados obtenidos en las pruebas pilotos del sistema y su correcto uso, existe una tendencia de mejora en el proceso de control de asistencia docente, mismo que está orientado al ahorro en tiempos, siendo de un 99.993% en relación a los reportes por facultad y de un 99.977% en relación a los reportes por escuela, permitiendo a la parte administrativa optimizar recursos.
- Mediante la Norma ISO/IEC 9126 se analiza los recursos utilizados por cada una de las funcionalidades en cuanto a memoria RAM tenemos: en la generación de Reporte de Asistencia por facultad representa el 2.44%, en la Asignación de feriado es de 1.01%, Generación de Horario de Clase del Docente es de 0.00083%, Reporte de Planificación Semestral 0.01%, valores que no afectan significativamente en cuanto a la utilización de RAM disponible en el servidor. De igual manera la base de datos quedo con un valor de 998 kb en el actual semestre que en comparación de los 50Gb de capacidad del servidor no representa un mayor uso en espacio de memoria. De esta forma se comprueba la eficiencia del software en cuanto a la utilización de recursos.

RECOMENDACIONES

- Es conveniente tener puntos de acceso inalámbricos en el área donde se instale el lector de tarjetas magnéticas permitiendo que el dispositivo funcione de forma óptima y el proceso de marcar el tiempo de entrada y salida por parte del docente se desarrolle sin ningún inconveniente.
- Tener en cuenta las resistencias a utilizar para regular la corriente y así evitar el daño de los elementos que conforman el circuito en especial el módulo wifi, y conectar en los pines que corresponden en el nano Arduino.
- Implementar una aplicación móvil que se integre con el sistema de control de asistencia a docentes, para que los diferentes actores involucrados puedan interactuar con cada una de las funcionalidades con la que este cuenta.
- Debido a que el dispositivo fue creado como prototipo para comprobar la funcionalidad del sistema informático. Es importante tener en cuenta a futuros investigadores realizar mejoras y/o cambios en cuanto al dispositivo haciendo un estudio y partiendo del manual que consta de los materiales y elaboración del mismo, adaptar un componente que permita conectarse a una red Ethernet, ayudando a mitigar el problema que existió en cuanto a la red wifi.
- En base a los resultados obtenidos se plantea a las autoridades pertinentes que el sistema informático sea implementado a nivel de toda la ESPOCH, debido que cumple con las actividades que involucra el proceso de control de asistencia de docentes, permitiendo optimizar recursos económicos y mejorar el tiempo de respuesta de la información.

BIBLIOGRAFÍA

ALEGSA, L. *Definición de Sistema informático (SI)*. [En línea]. 2016 [Consulta: 10 marzo 2017].
Disponible en: http://www.alegsa.com.ar/Dic/sistema_informatico.php.

BORBÓN, N. *Norma de Evaluación ISO/IEC 9126*. actividadreconocimiento-301569-8 [blog] marzo 2013. [Consulta: 6 marzo 2017]. Disponible en: <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>

CEAACES, Modelo General Para La Evaluación De Carreras Con Fines De Acreditación, 2011.

CHONG, Yap K. *¡Guía definitiva de prácticas ágiles esenciales de Scrum!* [en línea]. The Blokehead: Babelcube, Inc, 2016. [Consulta: 22 de marzo 2017]. Disponible en: <https://books.google.com.ec/books?id=T24eDQAAQBAJ&printsec=frontcover&dq=SCRUM&hl=es&sa=X&ved=0ahUKEwjqm4PWuNnUAhWDZiYKHelZBaYQ6AEIITAA#v=onepage&q=SCRUM&f=true>

ESPOCH, Reglamento de régimen académico institucional de grado, 06 de mayo de 2014,17-24.

FERNÁNDEZ, Yenisleidy R.; DÍAZ, Yanette G. “Patrón Modelo-Vista-Controlador”. *Revista Digital de las Tecnologías de la Información y las Comunicaciones* [en línea], 2012, (La Habana) 11(1), pp. 47-57. [Consulta: 10 febrero 2017]. ISSN 1729-3804. Disponible en: <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10>

FIGUEROA, Roberth G.; SOLÍS, Camilo J. *Metodologías Tradicionales vs. Metodologías Agiles* [en línea]. Ecuador: Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación, 2008. [Consulta: 20 febrero 2017]. Disponible en: <http://www.academia.edu/download/41231515/articulo-metodologia-de-sw-formato.doc>

GARRIDO, Pablo P. *Comenzando a Programar con JAVA* [en línea]. Elche: Universidad Miguel Hernández de Elche, 2015. [Consulta: 20 de marzo 2017]. Disponible en: <https://books.google.es/books?id=4v8QCgAAQBAJ&pg=PA107&dq=que+es+JAVA&hl=es>

&sa=X&ved=0ahUKEwixmLC_oNfUAhXB0iYKHYkOAhkQ6AEIPzAE#v=onepage&q&f=false

GRANOLLERS, Toni; et al. *Diseño de sistemas interactivos centrados en el usuario* [en línea]. Barcelona-España: Eureka media SL, 2005. [Consulta: 25 de marzo 2017]. Disponible en: <https://books.google.com.ec/books?id=Bk5Uv0Aais0C&pg=PA97&dq=%22ISO/IEC+9126%22&hl=es&sa=X&ved=0ahUKEwjx9vi84dfUAhVCOSYKHbCDDnkQ6AEITzAH#v=onepage&q=%22ISO%2FIEC%209126%22&f=true>

HERNÁNDEZ, A. *Los Sistemas de Información: Evolución y Desarrollo* [en línea]. Zaragoza: Departamento de Economía y Dirección de Empresas, Universidad de Zaragoza, 2004. [Consulta: 15 de marzo 2017]. Disponible en: https://www.researchgate.net/profile/Alejandro_Hernandez-Trasobares/publication/28253512_Los_sistemas_de_informacion_evolucion_y_desarrollo/links/02e7e5343c8b0b5031000000.pdf

GUTIÉRREZ, J. *¿Qué es un framework web?* [en línea]. Sevilla: Departamento de Lenguajes y Sistemas Informáticos, 2006. [Consulta: 14 de marzo 2017]. Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

KASIÁN, F.; REYES, N. *Búsqueda por similitud en PostgreSQL* [en línea]. San Luis-Argentina: Departamento de Informática, Universidad Nacional de San Luis, 2012. [Consulta: 20 de marzo 2017]. Disponible en: http://sedici.unlp.edu.ar/bitstream/handle/10915/23754/Documento_completo.pdf?sequence=1

LÓPEZ GONZALES, Carlos Augusto. *Control de asistencia al personal administrativo de la UNSM utilizando biometría* [En Línea] (tesis). Universidad Nacional de San Martín, Tarapoto, Perú. 2013. p.24. [Consulta: 15 marzo 2017]. Disponible en: <http://tesis.unsm.edu.pe/jspui/bitstream/11458/239/1/Carlos%20Augusto%20L%C3%B3pez%20Gonzales.pdf>

LÓPEZ, Joaquín; et al. *Informática y Comunicaciones en la empresa* [en línea]. Madrid-España: Universidad Rey Juan Carlos, ESIC Editorial, 2004. [Consulta: 24 febrero 2017]. Disponible en: <https://books.google.com.ec/books?id=U0MXWtqjxtsC&pg=PA33&dq=sistema+informatico&hl=es&sa=X&ved=0ahUKEwj2tJSu3tnUAhXFJCYKHaVbAOIQ6AEIJTAA#v=onepage&q=sistema%20informatico&f=true>

MERCADO, J. *Entendiendo el Modelo - Vista - Controlador*. vista-controlador [blog] 25 abril 2016. [Consulta: 6 marzo 2017]. Disponible en <http://vista-controlador.blogspot.com/>

PÉREZ, Oiver A. “Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM”. *Revista Inventum* [en línea], 2011, (Bogotá) (10), pp. 64-78. [Consulta: 15 febrero 2017]. ISSN 1909-2520. <http://biblioteca.uniminuto.edu/ojs/index.php/Inventum/article/view/9/9>

ORACLE CORPORATION. *¿Qué es Java?* [en línea]. Estados Unidos. [Consulta: 2 marzo 2017]. Disponible en: https://www.java.com/es/about/whatis_java.jsp

ORJUELA, Ailin D.; ROJAS, Mauricio C. “Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo”. *Red de Revistas Científicas de América Latina y el Caribe, España y Portugal* [en línea], 2008, (Medellín) 5(2), pp. 160-172. [Consulta: 13 febrero 2017]. ISSN 1657-7663. Disponible en: <http://www.redalyc.org/html/1331/133115027022/>

RUA, Wilson B. *SCRUM – Metodología de Desenvolvimento Ágil* [en línea]. Brasil: Tecnología em Análise e Desenvolvimento de Sistemas, Centro Universitário de Maringá, 2007. [Consulta: 21 de marzo 2017]. Disponible en: <http://revistas.bvs-vet.org.br/campodigital/article/view/30944/33947>

RUIZ, Gustavo A; et al. “Modelo de Evaluación de Calidad de Software Basado en Lógica Difusa, Aplicada a Métricas de Usabilidad de Acuerdo con la Norma ISO/IEC 9126”. *Red de Revistas Científicas de América Latina y el Caribe, España y Portugal* [en línea], 2006, (Colombia) 3(2), pp. 25-29. [Consulta: 15 febrero 2017]. ISSN 1657-7663. Disponible en: <http://www.redalyc.org/html/1331/133114988005/>

SZNAJDLEDER, Pablo. *Java a Fondo: Estudio del Lenguaje y Desarrollo de Aplicaciones* [en línea]. 2ª ed. Buenos Aires: Alfaomega Grupo Editor Argentino, 2013. [Consulta: 20 de marzo 2017]. Disponible en: https://books.google.es/books?id=WcL2DQAAQBAJ&pg=PT114&dq=que+es+JAVA&hl=es&sa=X&ved=0ahUKEwixmLC_oNfUAhXB0iYKHYkOAhkQ6AEIJTAA#v=onepage&q=que%20es%20JAVA&f=false

TENORIO CASTILLO, Omar Stalin, & MEJÍA QUINTEROS, Darling Ramiro. Diseño de un sistema híbrido para controlar el acceso a los estacionamientos utilizando tarjetas magnéticas y con tecnología bluetooth del teléfono celular e implementación de un prototipo en el parqueadero de la UNACH [En Línea] (tesis). Universidad Nacional de Chimborazo. Riobamba, Ecuador. 2012. pp. 48-49. [Consulta: 25 marzo 2017]. Disponible en: <http://dspace.unach.edu.ec/bitstream/51000/737/1/UNACH-EC-IET-2012-0003.pdf>

VALLEJO SALAZAR, Carlos Ernesto. Análisis, diseño y construcción de un sistema prototipo para control de ingreso de socios, consumos y facturación. Mediante el uso de tarjetas magnéticas en el club de empleados del Banco del Pacífico de la ciudad de Quito [En Línea] (tesis). Universidad Politécnica Salesiana. Quito, Ecuador. 2014. pp. 22-23. [Consulta: 20 marzo 2017]. Disponible en: <http://dspace.ups.edu.ec/bitstream/123456789/7582/1/UPS-ST001318.pdf>

ZEA, Mariuxi P.; MOLINA, Jimmy R. *Administración de Base de Datos con PostgreSQL* [en línea]. Alcoy-España: Área de innovación y desarrollo, S.L, 2017. [Consulta: 20 febrero 2017]. Disponible en: <https://books.google.es/books?id=5-mkDgAAQBAJ&printsec=frontcover&dq=Base+de+datos+Postgresql&hl=es&sa=X&ved=0ahUKEwiVseX2udfUAhVBNSYKHeZuDioQ6AEIJTAB#v=onepage&q&f=true>

GLOSARIO

MVC

Patrón de desarrollo de software que separa de manera física o lógica los componentes del proyecto agrupándolos según su función, las siglas significan Modelo Vista Controlador siendo estos la lógica del negocio, las interfaces y el intermediario correspondientemente.

Punto de Función

Es un método utilizado en ingeniería del software para medir el tamaño del software. Fue definida por Allan Albrecht, de IBM, en 1979 y pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software, y también ser útil en cualquiera de las fases de vida del software, desde el diseño inicial hasta la implantación y mantenimiento.

SCRUM

Metodología de desarrollo de software ágil.

UML

Lenguaje Unificado de Modelado (Unified Modeling Language). Es un lenguaje de modelado de sistemas o de algún software en específico. El lenguaje gráfico, estandarizado por el OMG (Object Management Group), de mucha ayuda en las tareas de visualizar, especificar, construir y documentar un sistema o algún otro proyecto de desarrollo de software.

ANEXOS

ANEXO A. Diccionario de datos

Esquema docente

Tabla	Columna	Tipo de Dato	Longitud	Nombre UDT
Área	cod_area	character varying	10	Varchar
Área	descripcion	character varying	100	varchar
Carrera	cod_carrera	character varying	10	varchar
Carrera	nombre	character varying	100	varchar
Carrera	cod_escuela	character varying	10	varchar
descripcion_asignatura	cod_descrip_asignatura	integer		integer
descripcion_asignatura	ident_problema	character varying	2000	varchar
descripcion_asignatura	contibucion_asignada	character varying	2000	varchar
descripcion_asignatura	objetivo_general_asig	character varying	2000	varchar
descripcion_asignatura	ambiente_aprendizaje	character varying	2000	varchar
descripcion_asignatura	uso_tecnologia	character varying	2000	varchar
descripcion_asignatura	estrategia_metodologica	character varying	2000	varchar
descripcion_asignatura	cod_silabo	integer		integer
dia	cod_dia	character varying	10	varchar
dia	descripcion	character varying	100	varchar
dictado_materia	cod_usuario	bigint		int8
dictado_materia	cod_paralelo	character varying	10	varchar
dictado_materia	cod_materia	character varying	10	varchar
dictado_materia	cod_carrera	character varying	10	varchar
dictado_materia	cod_periodo	character varying	10	varchar
dictado_materia	estado	integer		integer
escuela	cod_escuela	character varying	10	varchar
escuela	nombre	character varying	100	varchar
escuela	cod_facultad	character varying	10	varchar
facultad	cod_facultad	character varying	10	varchar
facultad	nombre	character varying	100	varchar
feriado	fecha	date		date
feriado	descripcion	character varying	250	varchar
feriado	tipo	integer		integer
horarios_clases	cod_dia	character varying	10	varchar
horarios_clases	cod_silabo	integer		integer
horarios_clases	inicio_hora	character varying	10	varchar
horarios_clases	fin_hora	character varying	10	varchar
informacion_general	cod_inf_general	integer		integer
informacion_general	sede	character varying	250	varchar
informacion_general	modalidad	integer		integer

informacion_general	cod_silabo	integer		integer
materia	cod_materia	character varying	10	varchar
materia	cod_area	character varying	10	varchar
materia	creditos	double precision		float8
materia	num_horas	integer		integer
materia	nombre	character varying	100	varchar
materia	estado	integer		integer
materia	editor	bigint		int8
metadato	cod_metadato	integer		integer
metadato	nombre	character varying	50	varchar
metadato	estado	boolean		boolean
metadato_valor	cod_escuela	character varying	50	varchar
metadato_valor	cod_metadato	integer		integer
metadato_valor	valor	character varying	50	varchar
metadato_valor	cod_periodo	character varying	50	varchar
nivel	cod_nivel	character varying	10	varchar
nivel	descripcion	character varying	20	varchar
paralelo	cod_paralelo	character varying	10	varchar
paralelo	nombre	character	1	bpchar
paralelo_nivel	cod_paral_nivel	character varying	10	varchar
paralelo_nivel	cod_paralelo	character varying	10	varchar
paralelo_nivel	cod_nivel	character varying	10	varchar
periodo	cod_periodo	character varying	10	varchar
periodo	descripcion	character varying	100	varchar
periodo	fecha_ini	date		date
periodo	fecha_fin	date		date
periodo	estado	boolean		boolean
programacion	cod_silabo	integer		integer
programacion	num_clase	integer		integer
programacion	fecha	date		date
programacion	observaciondoc	character varying	500	varchar
programacion	estado	integer		integer
programacion	horario	character varying	20	varchar
programacion	horas	integer		integer
programacion	cod_unidad	integer		integer
programacion	tema	character varying	500	varchar
registro_tema	codigo	bigint		int8
registro_tema	cod_tema	integer		integer
registro_tema	fecha	bigint		int8
registro_tema	validacion	boolean		boolean
registro_tema	observacionesdoc	character varying	500	varchar

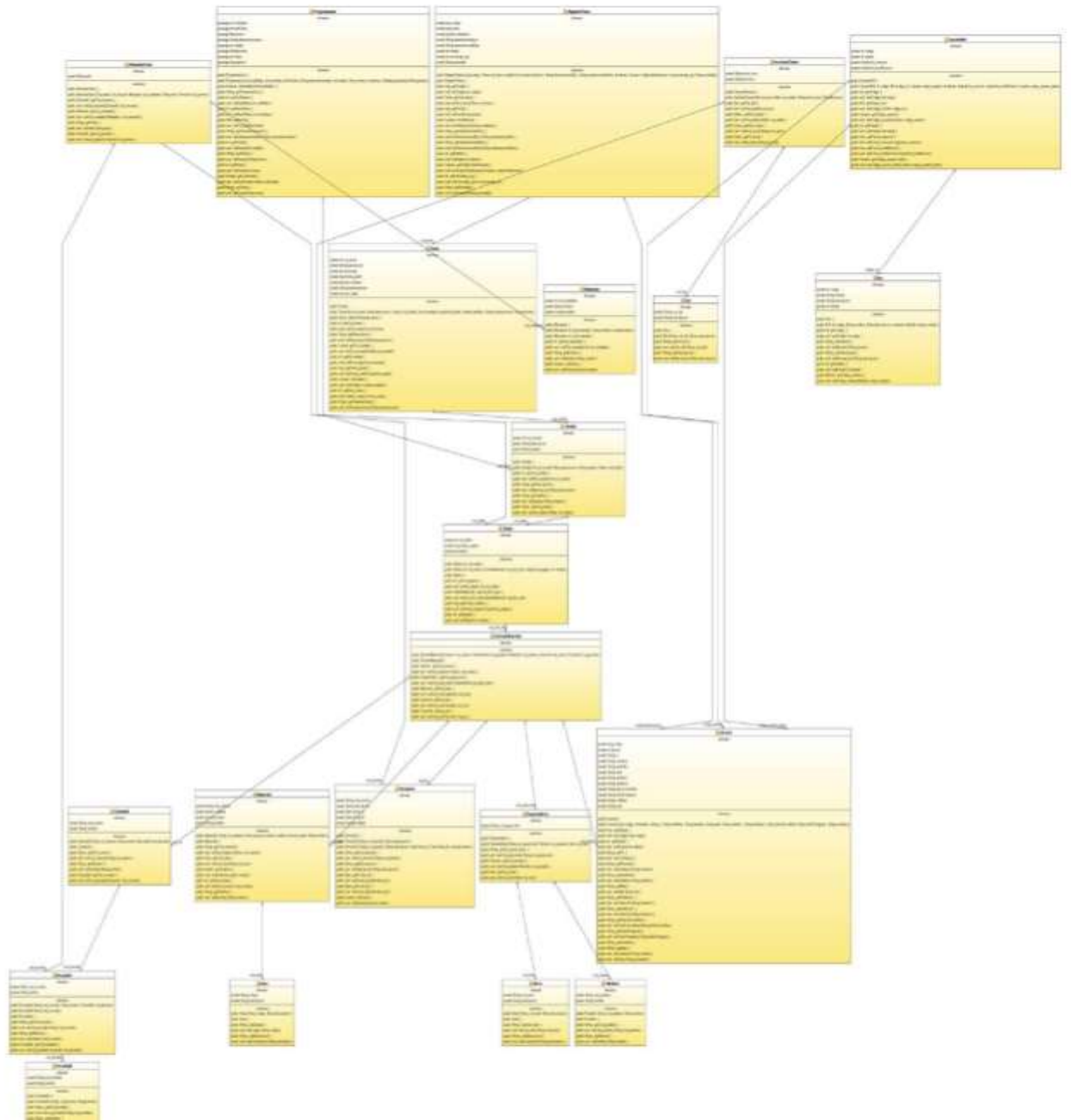
registro_tema	observacionesestu	character varying	500	varchar
registro_tema	estado	integer		integer
registro_tema	codigovalidausuario	bigint		int8
registro_tema	porcentaje_eje	integer		integer
registro_tema	actividad	character varying	100	varchar
requisitos	cod_requisitos	integer		integer
requisitos	cod_materia	character varying	10	varchar
requisitos	cod_req_materia	character varying	10	varchar
requisitos	tipo	character varying	20	varchar
silabo	cod_silabo	integer		integer
silabo	cod_usuario	bigint		int8
silabo	cod_paralelo	character varying	10	varchar
silabo	cod_materia	character varying	10	varchar
silabo	cod_carrera	character varying	10	varchar
silabo	cod_periodo	character varying	10	varchar
silabo	fecha_registro	bigint		int8
silabo	estado	integer		integer
silabo	observacion	character varying		varchar
silabo	fecha	bigint		int8

Esquema docente_seguridad

table_name	column_name	data_type	str_length	udt_name
rol	codigo	integer		integer
rol	nombre	character varying	50	varchar
rol	descripcion	character varying	200	varchar
rol	estado	integer		integer
rol	codigo_modulo	integer		integer
usuario	codigo	bigint		int8
usuario	estado	integer		integer
usuario	ci	character varying	50	varchar
usuario	nombres	character varying	50	varchar
usuario	apellidos	character varying	50	varchar
usuario	mail	character varying	200	varchar
usuario	telefono1	character varying	10	varchar
usuario	telefono2	character varying	10	varchar
usuario	titulo_tercer_nivel	character varying	200	varchar
usuario	titulo_postgrado	character varying	200	varchar
usuario	codtitulo	character varying	200	varchar
usuario	pass	character varying	20	varchar
usuario_rol	codigo	integer		integer
usuario_rol	codigo_rol	integer		integer

usuario_rol	codigo_usuario	bigint		int8
usuario_rol	estado	integer		integer
usuario_rol	fecha_creacion	bigint		int8
usuario_rol	fecha_modificacion	bigint		int8
usuario_rol	codigo_usuario_trans	bigint		int8

ANEXO B. Diagrama de clases



ANEXO C. Documento de levantamiento de información

Riobamba 14 de junio del 2017

Ingeniero

Raúl Cuzco

ENCARGADO DE RECOGER LA ASISTENCIA DE LA EIS

De nuestra consideración

Nosotros, **Fausto Fabián Bautista Torres** y **José Rodrigo Román Zurita** nos dirigimos a usted como desarrolladores del trabajo de titulación de la Escuela de Ingeniería en Sistemas, que tiene como título: "AUTOMATIZACIÓN DEL CONTROL DE ASISTENCIA DE DOCENTES UTILIZANDO EL PATRÓN DE ARQUITECTURA DE SOFTWARE MODELO VISTA CONTROLADOR EN LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA (ESPOCH)", para solicitarle información relacionada a los tiempos de generación de reportes a automatizar con el sistema mencionado, los cuales deben de ser detallados en la tabla adjunta.

Tipo de Reporte	Tiempo	Observación
Reporte por carreras	10 horas	esperamos en los próximos días recibir todo de todos los

En la certeza que este pedido será atendido favorablemente anticipamos nuestro agradecimiento.



Fausto Bautista

Sr. José Román

MANUAL TÉCNICO

SISTEMA INFORMÁTICO PARA AUTOMATIZAR EL CONTROL DE ASISTENCIA DE
LOS DOCENTES PARA LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
(ESPOCH)

Presentado por:

Román Zurita José Rodrigo y

Bautista Torres Fausto Fausto

Produce:

SISTEMA INFORMÁTICO PARA AUTOMATIZAR EL CONTROL DE ASISTENCIA
DOCENTE

Celular: 0969356186

Email josejavier-r@hotmail.com

Celular: 0967692050

Email: fabian.fabian.fabian10@gmail.com

TABLA DE CONTENIDOS

1. ANTECEDENTES.....	66
1.1. <i>Misión</i>	67
1.2. <i>Visión</i>	67
2. OBJETIVOS.....	67
3. DESARROLLO.....	68
3.1. <i>Estudio de Factibilidad</i>	68
3.2. <i>Análisis de Riesgos</i>	72
3.3. <i>Metodología de desarrollo</i>	82
3.4. <i>Personas y Roles del Proyecto</i>	83
3.5. <i>Gestión del proyecto</i>	83
3.5.1. Requerimientos	83
3.5.2. Diseño	87
3.5.3. Desarrollo	97
3.5.3.1. Sprint	97
3.5.3.1.1. Historias de Usuario	97
3.5.3.1.2. Tareas de Ingeniería	113
3.5.3.1.3. Pruebas de Aceptación	139
3.5.3.1.4. Servicios web consumidos del sistema OASIS	200
3.5.4. Implementación	202

1. ANTECEDENTES

La Escuela Superior Politécnica de Chimborazo (ESPOCH), tiene su origen en el Instituto tecnológico Superior de Chimborazo, creado mediante Ley No.6090, expedida por el Congreso Nacional, el 18 de abril de 1969. Inicia sus actividades académicas el 2 de mayo de 1972 con las Escuelas de Ingeniería Zootécnica, Nutrición y Dietética e Ingeniería Mecánica. Se inaugura el 3 de abril de 1972.

En el 2000, mediante resolución No. Del H.C.P. se crea la Facultad de Informática y Electrónica la misma que agrupa a las escuelas de Ingeniería en Sistemas, Ingeniería Electrónica y Tecnología en Computación y Diseño Gráfico.

La Facultad de Informática y Electrónica se propone conformar una unidad académica administrativa que involucrando áreas afines (informática, electrónica y diseño gráfico) brinde un servicio eficiente y eficaz a la sociedad, apoyada activamente en modelos de calidad en la gestión. Específicamente, optimizando los procesos administrativos y académicos de las Escuelas involucradas y diversificando carreras y programas de estudios en diferentes modalidades, tanto a nivel de pre-grado como de post grado. Dentro de sus propósitos se tienen:

- Generar una administración alternativa enmarcados dentro de la misión y visión universitaria.
- Optimizar los servicios de bienestar estudiantil y generar nuevas alternativas que permitan su desarrollo.
- Proyectar la Institución a nuevos retos de la tecnología, basados principalmente en el uso de la Informática, Electrónica e Internet.
- Proponer nuevos modelos de cooperación internacional.
- Proponer nuevos modelos para la vinculación empresa-universidad.
- Crear un Centro de Capacitación Informática.
- Fortalecer los proyectos de creación de unidades de producción en el campo de la electrónica, informática y diseño gráfico.

1.1. Misión

"Ser una institución universitaria líder en la Educación Superior y en el soporte científico y tecnológico para el desarrollo socioeconómico y cultural de la provincia de Chimborazo y del país, con calidad, pertinencia y reconocimiento social".

1.2. Visión

"Formar profesionales competitivos, emprendedores, conscientes de su identidad nacional, justicia social, democracia y preservación del ambiente sano, a través de la generación, transmisión, adaptación y aplicación del conocimiento científico y tecnológico para contribuir al desarrollo sustentable de nuestro país".

2. OBJETIVOS

- Desarrollar el sistema informático para automatizar el control de asistencia de los docentes de la Escuela de Ingeniería en Sistemas (ESPOCH).
- Analizar la tecnología que se va a usar en el desarrollo del Sistema.
- Desarrollar el Sistema de control de Asistencia de docentes por medio de un lector de tarjetas magnéticas en la Escuela de Ingeniería en Sistemas.
- Demostrar la eficiencia del sistema de control de asistencia de docentes mediante la norma ISO/IEC 9126.

3. DESARROLLO

3.1. Estudio de Factibilidad

El estudio de Factibilidad permite recabar información relevante sobre el desarrollo del Proyecto para poder tomar decisiones en base a esta, para así poder decidir confiablemente si se procede con el desarrollo o no. Los tipos de factibilidad que se considera para el desarrollo del Sistema Informático para automatizar el control de asistencia de los docentes de la Facultad de Informática y Electrónica (ESPOCH), se describirán a continuación.

Factibilidad Técnica

Nos permite determinar si la solución propuesta puede ser implementada con los Recursos hardware, software y humanos disponibles. Por lo antes mencionado, se concluye que para el desarrollo de la aplicación web y el lector de tarjetas magnéticas se cuenta con la mayoría de los recursos de hardware y software necesarios, a continuación, se detallará los recursos de hardware, software y humanos requeridos.

- **Hardware Existente**

Fuente: Bautista Fausto y Román José

CANTIDAD	DESCRIPCIÓN	OBSERVACIÓN
1	Laptop LENOVO ideapad flex 15	Herramienta para el desarrollo de la Aplicación web y su respectiva documentación.
1	Flash Memory SanDisk 4GB	Herramienta para movilidad de la información.
1	Infraestructura de Red	Ya existente en el área de trabajo donde se llevará a cabo el desarrollo.

- **Hardware Requerido**

Fuente: Bautista Fausto y Román José

CANTIDAD	DESCRIPCIÓN	OBSERVACIÓN
1	Impresora	Impresión de los informes o documentos necesarios.

- **Software Existente**

Fuente: Bautista Fausto y Román José

NOMBRE	DESCRIPCIÓN	ESTADO	OBSERVACIÓN
Ubuntu 16.04 64 bits	Sistema operativo	Legal	Licencia Libre
Microsoft Office 2013	Suite de ofimática	Legal	Instalado a través de wine con licencia de regalo a nombre del desarrollador
Netbeans 8.2	Plataforma de desarrollo	Legal	Licencia Libre
PostgreSQL 9.4	Gestor de Base de Datos	Legal	Licencia Libre
JDK 1.8	Kit de Desarrollo para JAVA	Legal	Licencia Libre
Mozilla Firefox, Google Chrome	Navegador	Legal	Aplicación que opera a través de Internet, interpretando archivos y sitios web desarrollados.

- **Software Requerido**

Fuente: Bautista Fausto y Román José

NOMBRE	DESCRIPCIÓN	ESTADO	OBSERVACIÓN
Servidor Glassfish	Servidor Web	Legal	Licencia Libre
Primefaces	Framework para JSF	Legal	Licencia Libre
Arduino IDE	Entorno de desarrollo integrado (IDE)	Legal	Programa que permite ingresar códigos a la placa.

- **Recurso Humano Requerido**

Fuente: Bautista Fausto y Román José

FUNCIÓN	FORMACIÓN	EXPERIENCIA
Desarrollador	Estudiante de la Escuela de Ingeniería en Sistemas	Desarrollo web en PHP, JAVA
Administrador de Base de Datos	Estudiante de la Escuela de Ingeniería en Sistemas.	PostgreSQL, MySQL

Factibilidad Operativa

Ésta factibilidad permite determinar la probabilidad de que el nuevo Sistema se use de forma correcta por parte de los usuarios del Sistema.

- **Recursos humanos**

El recurso humano que participara en la utilización del Sistema son:

Usuarios Directos: Individuos quienes van a utilizar el Sistema.

Personal a Capacitar: Las personas a capacitar son las siguientes:

Fuente: Bautista Fausto y Román José

NOMBRE	FUNCIÓN
Administrador del Sistema	Vicedecano (o representante) de la Facultad de Informática y Electrónica.
Docentes	Registrar tutorías, gestionar el horario de clases para todo el semestre por asignatura y generación de Documento del horario de clases por asignatura.
Estudiantes	Subir evidencia sobre las clases de ese día en curso, en caso de que se le sea solicitado.

Factibilidad Legal

El desarrollo del sistema web y el lector de tarjetas magnéticas tienen reservado todos los derechos de autor, por lo que cualquier copia parcial o total debe ser autorizada por el autor según la Ley de Propiedad Intelectual. Según lo estipula en la legislación ecuatoriana que consta en el código penal con la ley de comercio electrónico, firmas electrónicas y mensajes de datos.

Esto asegura que el Sistema que se está proponiendo es legalmente factible, debido que actualmente no existe ningún impedimento para la realización del mismo.

Factibilidad Económica

Fuente: Bautista Fausto y Román José

COSTOS	VALOR
Costos de Desarrollo	2 204,30
Costo de Personal	
Costo de Desarrollador	2 000,00
Costo de Hardware y Software	
Hardware: Impresora	150,00
Software: Software a Utilizar	00,00

Costo desarrollo de lector de tarjeta magnética	
Arduino Nano	17,00
Módulo Wifi ESP8266	15,00
Módulo RFID-RC522 RF	15,00
2 Led (Amarillo y Rojo)	0,20
2 Resistencias(Color: Rojo Rojo Marrón Naranja)	0,20
2 Diodos Rectificadores	1,40
1 Tarjeta magnética con chip inteligente	1,50
Multi cargador 3 entradas USB 3.1 mAh Fast Charger	4,00
Costo de Instalación	00,00
La Escuela de Ingeniería en Sistemas cuenta con todos los recursos necesarios para implantar el Sistema.	00,00
Costo de Operación	00,00
Costos de personal de operación	00,00
Costo de Mantenimiento	00,00
Costo de suministros de operación	00,00
TOTAL	2 204,30

3.2. Análisis de Riesgos

Nomenclatura utilizada:

RP: Riesgo del Proyecto

RT: Riesgo Técnico

RN: Riesgo del Negocio

IDENTIFICACIÓN DE RIESGOS

Fuente: Realizado por Bautista Fausto y Román José.

Nº	Riesgo	Categoría	Consecuencia
R1	Las herramientas utilizadas para el desarrollo del proyecto se vuelvan complejas de aprender.	RT	Retraso en el proyecto
R2	El Ingeniero de asesoramiento del proyecto ya no pueda continuar.	RP	Retraso en el proyecto
R3	El tiempo estimado para terminar el proyecto es muy poco y no se logre concluirlo.	RT	Demora en la entrega del proyecto, Software inadecuado
R4	La computadora donde se estaba elaborando el proyecto se dañó y la información no se puede recuperar.	RP	Cancelación del proyecto
R5	El hardware requerido para el equipo de lectura de datos del docente tiene un costo elevado al implementarlo por unidad.	RP, RT	Retraso en el proyecto.
R6	Al no definir correctamente los requerimientos, la solución software no cumplirá con las expectativas del cliente por lo que nuevamente se debe plantear los requerimientos.	RT	Planificación inadecuada
R7	Las interfaces resultaron difíciles de entender y los usuarios se rehúsan a continuar utilizando el Sistema.	RN	Pérdida de recursos, Proyecto rechazado
R8	Se cambian continuamente los requerimientos.	RP	Demora en la entrega del proyecto

CATEGORIZACIÓN DEL RIESGO

Determinación de la probabilidad:

PORCENTAJE	DESCRIPCIÓN	VALOR
1% - 33%	Bajo	1
34% - 67%	Medio	2
68% - 99%	Alto	3

Fuente: Bautista Fausto y Román José.

Determinación del Impacto:

IMPACTO	COSTO	RETRASO	IMPACTO TÉCNICO	VALOR
Bajo	< 1%	1 Semana	Ligero impacto en el desarrollo del proyecto.	1
Medio	< 5%	2 Semanas	Moderado impacto en el desarrollo del proyecto.	2
Alto	< 10%	1 Mes	Severo impacto en el desarrollo del proyecto.	3
Critico	> 10%	> 1 Mes	No se puede terminar el proyecto.	4

Fuente: Bautista Fausto y Román José.

Determinación de la Exposición de riesgo

IMPACTO	BAJO=1	MEDIO=2	ALTO=3	CRÍTICO=4
	Alta=3	3	6	9
Media=2	2	4	6	8
Baja=1	1	2	3	4

Fuente: Bautista Fausto y Román José.

Código de Colores según la Exposición de Riesgo

EXPO. RIESGO	VALOR	COLOR
Baja	1 – 2	Verde
Media	3 – 4	Amarillo
Alta	>= 6	Rojo

Determinación de prioridad de riesgos

Riesgo	PROBABILIDAD			IMPACTO		EXPO. RIESGO	
	%	Valor	Prob.	Valor	Impacto	Valor	Expo.
R7	40%	2	Media	3	Alto	6	Alta
R1	35%	2	Media	3	Alto	6	Alta
R4	40%	2	Media	3	Alto	6	Alta
R5	50%	1	Media	3	Alto	4	Medio
R2	20%	1	Baja	3	Alto	3	Medio
R8	20%	1	Baja	3	Alto	3	Medio
R3	10%	1	Baja	3	Alto	3	Medio
R6	10%	1	Baja	2	Medio	2	Baja

HOJAS DE RIESGO

HOJA DE INFORMACION DEL RIESGO									
ID Riesgo: R1		Fecha: 25 de enero del 2017							
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alta Valor: 6	Prioridad: 1						
Descripción: Las herramientas utilizadas para el desarrollo del proyecto se vuelvan complejas de aprender o utilizar.									
Refinamiento: Causas: Poca información de las herramientas a utilizar, falta de interés por parte de los desarrolladores del proyecto.									
Consecuencia: Retraso en la entrega del Proyecto.									
Reducción y supervisión: Buscar información acerca de las herramientas, pedir ayuda a terceras personas, poner empeño en aprender a utilizar las nuevas herramientas.									
Gestión: Preparar al personal del proyecto mediante cursos, información bibliográfica y consultas acerca de las herramientas.									
Estado actual: <table><tr><td>Fase de Reducción Iniciada</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Fase de Supervisión</td><td><input type="checkbox"/></td></tr><tr><td>Fase de Gestión</td><td><input type="checkbox"/></td></tr></table>				Fase de Reducción Iniciada	<input checked="" type="checkbox"/>	Fase de Supervisión	<input type="checkbox"/>	Fase de Gestión	<input type="checkbox"/>
Fase de Reducción Iniciada	<input checked="" type="checkbox"/>								
Fase de Supervisión	<input type="checkbox"/>								
Fase de Gestión	<input type="checkbox"/>								
Responsable: Bautista Fausto y Román José									

HOJA DE INFORMACION DEL RIESGO			
ID Riesgo: R2		Fecha: 25 de enero del 2017	
Probabilidad: Baja	Impacto: Alto	Exposición: Alta	Prioridad: 1

Valor: 1	Valor: 3	Valor: 3	
Descripción: El Ing. de Asesoramiento del proyecto ya no puede continuar.			
Refinamiento:			
Causas: Renuncia por parte del Ingeniero, enfermedad o disposición de las autoridades de la Escuela.			
Consecuencia: Cambio del proyecto o pérdida de tiempo.			
Reducción y supervisión: Cooperación simultanea entre desarrollador y personal implicado en el proyecto. Informar a todas las autoridades acerca de la situación del proyecto.			
Gestión: Buscar todas soluciones posibles para que el Ingeniero Asesor del proyecto no abandone.			
Estado actual:			
Fase de Reducción Iniciada		<input checked="" type="checkbox"/>	
Fase de Supervisión		<input type="checkbox"/>	
Fase de Gestión		<input type="checkbox"/>	
Responsable: Bautista Fausto y Román José			

HOJA DE INFORMACION DEL RIESGO			
ID Riesgo: R3		Fecha: 25 de enero del 2017	
Probabilidad: Baja	Impacto: Alto	Exposición: Medio	Prioridad: 1
Valor: 1	Valor: 3	Valor: 3	
Descripción: El tiempo estimado para la terminar el proyecto es muy poco y no se logre concluirlo.			
Refinamiento:			
Causas: Mala planificación del proyecto.			

Consecuencia: Demora en la entrega del proyecto.
Reducción y supervisión: Revisión y refinamiento de la planificación, cooperación entre todo el personal del proyecto para mejorar los tiempos de desarrollo.
Gestión: Informar a los dueños del proyecto y establecer la nueva fecha de entrega del proyecto.
Estado actual: <p style="text-align: center;"> Fase de Reducción Iniciada <input type="checkbox"/> Fase de Supervisión <input checked="" type="checkbox"/> Fase de Gestión <input type="checkbox"/> </p>
Responsable: Bautista Fausto y Román José

HOJA DE INFORMACION DEL RIESGO			
ID Riesgo: R4		Fecha: 25 de enero del 2017	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alta Valor: 6	Prioridad: 1
Descripción: La computadora donde se estaba elaborando el proyecto se daña y la información no se puede recuperar.			
Refinamiento: Causas: Corto circuito, daño en el disco duro, caída de la computadora al suelo.			
Consecuencia: Cancelación del Proyecto.			
Reducción y supervisión: Respaldo del proyecto en un disco externo.			
Gestión: Adquirir un Disco Extraíble para poder guardar el proyecto mediante backups.			

Estado actual:	
Fase de Reducción Iniciada	<input checked="" type="checkbox"/>
Fase de Supervisión	<input type="checkbox"/>
Fase de Gestión	<input type="checkbox"/>
Responsable: Bautista Fausto y Román José	

HOJA DE INFORMACION DEL RIESGO			
ID Riesgo: R5		Fecha: 25 de enero del 2017	
Probabilidad: Media Valor: 1	Impacto: Alto Valor: 3	Exposición: Alta Valor: 4	Prioridad: 1
Descripción: El hardware requerido para el equipo de lectura de datos del docente tiene un costo elevado al implementarlo por unidad.			
Refinamiento:			
Causas: Excesivo costo de implementación del proyecto.			
Consecuencia: No implementación del proyecto.			
Reducción y supervisión: Definir un límite de costo del desarrollo del dispositivo.			
Gestión: Reemplazar el dispositivo por una opción alternativa de menor costo.			
Estado actual:			
Fase de Reducción Iniciada	<input type="checkbox"/>		
Fase de Supervisión	<input type="checkbox"/>		
Fase de Gestión	<input checked="" type="checkbox"/>		
Responsable: Bautista Fausto y Román José			

HOJA DE INFORMACION DEL RIESGO

ID Riesgo: R6		Fecha: 25 de enero del 2017	
Probabilidad: Baja Valor: 1	Impacto: Medio Valor: 2	Exposición: Baja Valor: 2	Prioridad: 1
Descripción: Al no definir correctamente los requerimientos, la solución software no cumplirá con las expectativas del cliente por lo que nuevamente se debe plantear los requerimientos.			
Refinamiento: Causas: Mala Planificación del Proyecto.			
Consecuencia: Retraso en la elaboración del proyecto.			
Reducción y supervisión: Tener claro lo que el cliente realmente quiere que el proyecto solucione.			
Gestión: Al inicio del proyecto definir claramente lo requisitos, los que realmente necesita el cliente.			
Estado actual: <div style="text-align: center;"> <p>Fase de Reducción Iniciada <input checked="" type="checkbox"/></p> <p>Fase de Supervisión <input type="checkbox"/></p> <p>Fase de Gestión <input type="checkbox"/></p> </div>			
Responsable: Bautista Fausto y Román José			

HOJA DE INFORMACION DEL RIESGO			
ID Riesgo: R7		Fecha: 25 de enero del 2017	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alta Valor: 6	Prioridad: 1
Descripción: Las interfaces resultaron difíciles de entender y los usuarios se rehúsan a continuar utilizando el Sistema.			

<p>Refinamiento:</p> <p>Causas: Mal diseño de las interfaces de usuario.</p>
<p>Consecuencia: Pérdida de tiempo, obtención de un producto no deseado.</p>
<p>Reducción y supervisión: Crear prototipos de interfaz para ir mostrando al dueño del producto según como avance el proyecto.</p>
<p>Gestión: Incluir parámetros de usabilidad al proyecto, para mejorar su calidad.</p>
<p>Estado actual:</p> <p style="text-align: center;"> Fase de Reducción Iniciada <input checked="" type="checkbox"/> </p> <p style="text-align: center;"> Fase de Supervisión <input type="checkbox"/> </p> <p style="text-align: center;"> Fase de Gestión <input type="checkbox"/> </p>
<p>Responsable: Bautista Fausto y Román José</p>

HOJA DE INFORMACION DEL RIESGO			
ID Riesgo: R8		Fecha: 25 de enero del 2017	
Probabilidad: Baja Valor: 1	Impacto: Alto Valor: 3	Exposición: Medio Valor: 3	Prioridad: 1
Descripción: Se cambian continuamente los requerimientos.			
<p>Refinamiento:</p> <p>Causas: El cliente del producto no sabe lo que realmente quiere.</p>			
Consecuencia: Pérdida de tiempo, o no se realice el proyecto.			
Reducción y supervisión: Orientar al Cliente del producto de lo que el realmente necesita.			
Gestión: Mantener una constante comunicación con el cliente.			

Estado actual:	
Fase de Reducción Iniciada	<input checked="" type="checkbox"/>
Fase de Supervisión	<input type="checkbox"/>
Fase de Gestión	<input type="checkbox"/>
Responsable: Bautista Fausto y Román José	

3.3. Metodología de desarrollo

Para el desarrollo de este sistema web se hará uso de la metodología ágil SCRUM.

Cabe mencionar que SCRUM no es un proceso o una técnica para construir productos, sino un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Se basa en el control de procesos empírico, es decir que el conocimiento procede de la experiencia y fomenta el tomar decisiones de esta misma manera.

Se soporta de 3 pilares fundamentales:

- **Transparencia:** Se define un estándar común
- **Inspección:** Se debe inspeccionar frecuentemente, (no tanto como para interferir en el trabajo) los artefactos y el progreso hacia un objetivo para detectar variaciones indeseadas, de preferencia lo deben hacer expertos.
- **Adaptación:** Si se detecta una variación considerable, se debe ajustar el producto o material, debe hacerse cuanto antes.

Las ventajas de Scrum sobre otras metodologías ágiles, radica en su orientación a proyectos de toda rama, volviéndolo ampliamente versátil y adaptable a cualquier caso particular.

Se definen como auto-organizados y multifuncionales y se diseñan para optimizar la flexibilidad, la creatividad y la productividad.

3.4. Personas y Roles del Proyecto

El equipo de desarrollo que ha llevado a cabo este proyecto es el siguiente:

Rol	Persona	Responsabilidad
El Dueño del producto (Product Owner)	Rector de la Escuela Superior Politécnica de Chimborazo	Aportar los recursos necesarios para el desarrollo y ejecución del proyecto
Clientes	Dr. Julio Santillán	Se encargan de la entrega de requisitos dado que son expertos en el proceso a automatizar
Equipo de Desarrollo (Development Team)	Román Jose Bautista Fausto	Se encargará del desarrollo del proyecto
Scrum Master	Dr. Julio Santillán	Experto en el proceso SCRUM, que prestara su asesoría

3.5. Gestión del proyecto

3.5.1. Requerimientos

Dada la naturaleza del negocio se tienen 3 Clientes que serán los encargados de definir los requerimientos del sistema (Historias de Usuario), las mismas que fueron obtenidas mediante la técnica de entrevista realizada a cada uno de los ellos de manera individual.

Durante la implementación de la técnica de entrevista, se orientó a no profundizar en descripciones ni procesos logrando así mantener los requerimientos breves y claros, con las cuales no hubo inconvenientes al momento de programar los mismos.

Como lo menciona la metodología se utilizará como artefacto el Product Backlog o pila del producto.

Definida con los requerimientos del cliente, la pila del producto consta de 27 funcionalidades específicas a implementar para construir la visión que detalla el cliente, se la presenta de manera priorizada, el Product Backlog muestra la lista de requisitos planteados por el cliente.

Product Backlog

Historia	Nombre de la Historia
1	Iniciar sesión a los usuarios.
2	Registrar los datos de un feriado.
3	Modificar los datos de un feriado.
4	Eliminar los datos de un feriado.
5	Registrar los metadatos del sistema.
6	Modificar los metadatos del sistema.
7	Eliminar los metadatos del sistema.
8	Registrar los datos de las tutorías por asignatura.
9	Registrar identificador de tarjeta magnética del docente.
10	Modificar identificador de tarjeta magnética del docente.
11	Eliminar identificador de tarjeta magnética del docente.
12	Registrar los datos de horarios de clase.
13	Modificar datos de horario de clase.
15	Registrar evidencias de clases en caso de ser solicitadas.
16	Generar reporte de asistencia por docente.
17	Generar reporte de asistencia por escuela.
18	Generar reporte de asistencia por facultad.
19	Generar reporte de asistencia por carrera.
20	Generar reporte de asistencia por materia.

21	Generar el reporte del horario de clases.
22	Registrar los datos de un período.
23	Modificar los datos de un período.
24	Eliminar los datos de un período.
25	Registrar justificaciones de inasistencias
26	Modificar justificaciones de inasistencias
27	Eliminar justificaciones de inasistencias

La planificación y estimación de esfuerzo ha utilizado como artefacto el Sprint Backlog o Pila del Sprint.

En las estimaciones de esfuerzo asociado a la implementación de las historias se las ha establecido utilizando como medida el punto de función. Para establecer esta estimación y por experiencia del desarrollador en el método de talla de la camiseta, se decide el uso de esta, la cual se implementó de la siguiente manera: como talla S se tiene 5 puntos de función, como talla M se tiene 10 puntos, como talla L 20 puntos y como talla XL 30 puntos, se tomará un Sprint como 1 tallas L y 1 talla M o su correspondiente con tallas más pequeñas.

En el proyecto, un punto de función equivale a 90 minutos de trabajo, los días laborables para el desarrollo fueron: lunes, martes, miércoles, jueves y viernes, con horario de 15:00- 18:00, además se ha establecido que un Sprint tendrá 3 semanas de duración.

Partiendo de las historias de usuario definidas, se generar doce pilas del Sprint (Sprint Backlog) basándose en el tiempo y procurando agrupar la funcionalidad relacionada en el mismo Sprint.

En los Sprint 1 y 13 se han definido las metáforas iniciales del sistema y las historias técnicas enfocadas a la puesta en funcionamiento del sistema respectivamente y los 11 Sprint restantes corresponden a las funcionalidades propias del sistema. Además se ha estimado un total de 400 puntos de función.

Sprint Backlog

Sprint	Historia	Nombre de la Historia	Puntos
1	MT_01	Diseño de la base de datos	5
	MT_02	Definición de la arquitectura del sistema.	5
	MT_03	Diseño de las interfaces.	5
	HU_01	Iniciar sesión a los usuarios.	5
	HU_02	Registrar los datos de un feriado.	10
2	HU_03	Modificar los datos de un feriado.	10
	HU_05	Registrar los datos de horarios de clase.	20
3	HU_04	Eliminar los datos de un feriado.	10
	HU_06	Registrar los metadatos del sistema.	20
4	HU_07	Modificar los metadatos del sistema.	10
	HU_08	Eliminar los metadatos del sistema.	10
	HU_09	Registrar los datos de un período.	10
5	HU_10	Modificar los datos de un período.	10
	HU_11	Eliminar los datos de un período.	20
6	HU_12	Registrar los datos de las tutorías por asignatura.	20
	HU_13	Registrar identificador de tarjeta magnética del docente.	10
7	HU_14	Modificar identificador de tarjeta magnética del docente.	10
	HU_15	Registrar justificaciones de inasistencias	10
	HU_16	Registrar evidencias de clases en caso de ser solicitadas.	10

8	HU_17	Modificar datos de horario de clase.	20
	HU_18	Generar el reporte del horario de clases	10
9	HU_19	Generar el reporte de asistencia por materias.	10
	HU_20	Generar el reporte de asistencia por docente.	20
10	HU_21	Generar el reporte de asistencia por escuela.	10
	HU_22	Generar el reporte de asistencia por facultad.	10
	HU_23	Generar el reporte de asistencia por carrera.	10
11	HU_24	Realizar lector de tarjetas magnéticas.	30
12	HU_25	Modificar justificaciones de inasistencias	20
	HU_26	Eliminar justificaciones de inasistencias	10
13	MT_04	Desplegar el sistema en el servidor de aplicaciones de la facultad	20
	MT_05	Implantar la base de datos en el servidor de base de datos de la facultad	10

3.5.2. Diseño

Para el diseño se ha tomado en cuenta cada uno de los usuarios del sistema web a desarrollar, para lo cual se hace uso de los diagramas de caso de uso desarrollados con el software StarUML.

Diagrama de Caso de Uso del usuario estudiante

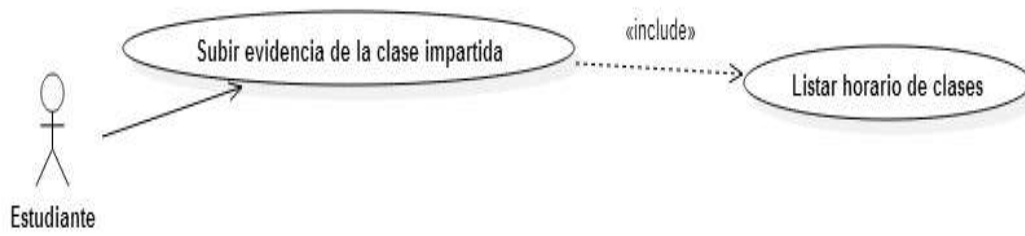


Diagrama de Caso de Uso del usuario docente

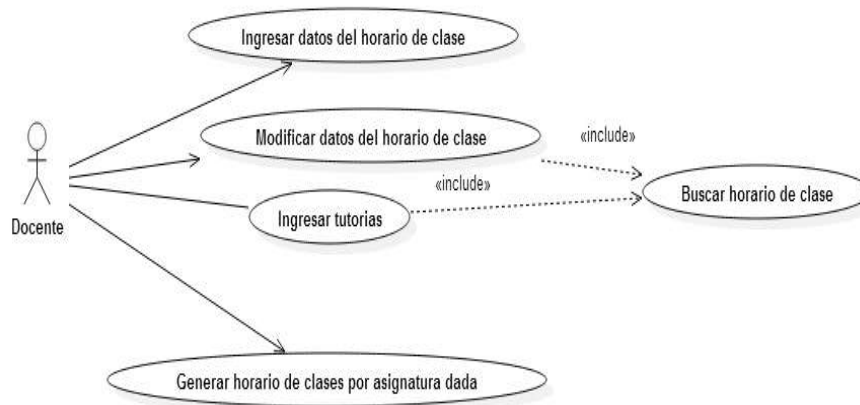
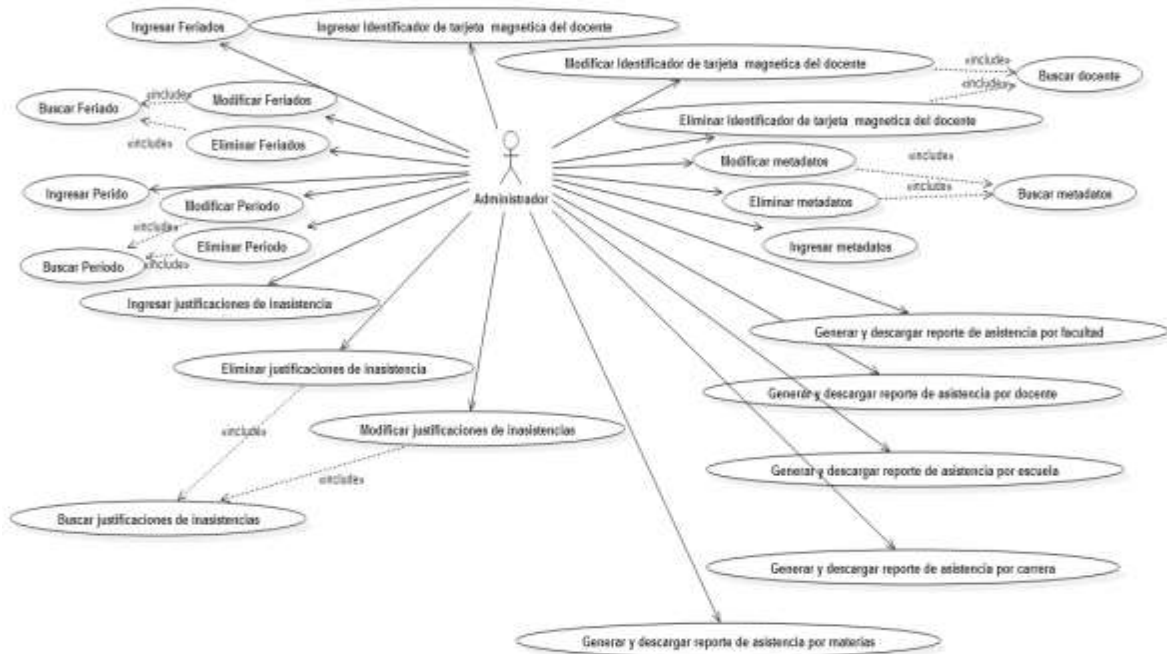


Diagrama de Caso de Uso del usuario Administrador



Además se muestra el diseño conceptual de la base de datos la misma que consta de 20 tablas con sus respectivos campos y tipo de datos, cabe recalcar que para su diseño físico se ha utilizado el motor de base de datos PostgreSQL en la versión 9.5. El diagrama conceptual y el respectivo diccionario de datos de muestran a continuación.

Diccionario de datos

Tabla	Columna	Tipo de Dato	Longitud	Nombre UDT
Área	cod_area	character varying	10	Varchar
Área	descripcion	character varying	100	varchar
Carrera	cod_carrera	character varying	10	varchar
Carrera	nombre	character varying	100	varchar
Carrera	cod_escuela	character varying	10	varchar
descripcion_asignatura	cod_descrip_asignatura	integer		integer
descripcion_asignatura	ident_problema	character varying	2000	varchar
descripcion_asignatura	contibucion_asignada	character varying	2000	varchar
descripcion_asignatura	objetivo_general_asig	character varying	2000	varchar
descripcion_asignatura	ambiente_aprendizaje	character varying	2000	varchar
descripcion_asignatura	uso_tecnologia	character varying	2000	varchar
descripcion_asignatura	estrategia_metodologica	character varying	2000	varchar
descripcion_asignatura	cod_silabo	integer		integer
dia	cod_dia	character varying	10	varchar
dia	descripcion	character varying	100	varchar
dictado_materia	cod_usuario	bigint		int8

dictado_materia	cod_paralelo	character varying	10	varchar
dictado_materia	cod_materia	character varying	10	varchar
dictado_materia	cod_carrera	character varying	10	varchar
dictado_materia	cod_periodo	character varying	10	varchar
dictado_materia	estado	integer		integer
escuela	cod_escuela	character varying	10	varchar
escuela	nombre	character varying	100	varchar
escuela	cod_facultad	character varying	10	varchar
facultad	cod_facultad	character varying	10	varchar
facultad	nombre	character varying	100	varchar
feriado	fecha	date		date
feriado	descripcion	character varying	250	varchar
feriado	tipo	integer		integer
horarios_clases	cod_dia	character varying	10	varchar
horarios_clases	cod_silabo	integer		integer
horarios_clases	inicio_hora	character varying	10	varchar
horarios_clases	fin_hora	character varying	10	varchar
informacion_general	cod_inf_general	integer		integer
informacion_general	sede	character varying	250	varchar

informacion_general	modalidad	integer		integer
informacion_general	cod_silabo	integer		integer
materia	cod_materia	character varying	10	varchar
materia	cod_area	character varying	10	varchar
materia	creditos	double precision		float8
materia	num_horas	integer		integer
materia	nombre	character varying	100	varchar
materia	estado	integer		integer
materia	editor	bigint		int8
metadato	cod_metadato	integer		integer
metadato	nombre	character varying	50	varchar
metadato	estado	boolean		boolean
metadato_valor	cod_escuela	character varying	50	varchar
metadato_valor	cod_metadato	integer		integer
metadato_valor	valor	character varying	50	varchar
metadato_valor	cod_periodo	character varying	50	varchar
nivel	cod_nivel	character varying	10	varchar
nivel	descripcion	character varying	20	varchar
paralelo	cod_paralelo	character varying	10	varchar
paralelo	nombre	character	1	bpchar
paralelo_nivel	cod_paral_nivel	character varying	10	varchar

paralelo_nivel	cod_paralelo	character varying	10	varchar
paralelo_nivel	cod_nivel	character varying	10	varchar
periodo	cod_periodo	character varying	10	varchar
periodo	descripcion	character varying	100	varchar
periodo	fecha_ini	date		date
periodo	fecha_fin	date		date
periodo	estado	boolean		boolean
programacion	cod_silabo	integer		integer
programacion	num_clase	integer		integer
programacion	fecha	date		date
programacion	observaciondoc	character varying	500	varchar
programacion	estado	integer		integer
programacion	horario	character varying	20	varchar
programacion	horas	integer		integer
programacion	cod_unidad	integer		integer
programacion	tema	character varying	500	varchar
registro_tema	codigo	bigint		int8
registro_tema	cod_tema	integer		integer
registro_tema	fecha	bigint		int8
registro_tema	validacion	boolean		boolean
registro_tema	observacionesdoc	character varying	500	varchar
registro_tema	observacionesestu	character varying	500	varchar
registro_tema	estado	integer		integer
registro_tema	codigovalidausuario	bigint		int8

registro_tema	porcentaje_eje	integer		integer
registro_tema	actividad	character varying	100	varchar
requisitos	cod_requisitos	integer		integer
requisitos	cod_materia	character varying	10	varchar
requisitos	cod_req_materia	character varying	10	varchar
requisitos	tipo	character varying	20	varchar
silabo	cod_silabo	integer		integer
silabo	cod_usuario	bigint		int8
silabo	cod_paralelo	character varying	10	varchar
silabo	cod_materia	character varying	10	varchar
silabo	cod_carrera	character varying	10	varchar
silabo	cod_periodo	character varying	10	varchar
silabo	fecha_registro	bigint		int8
silabo	estado	integer		integer
silabo	observacion	character varying		varchar
silabo	fecha	bigint		int8
rol	codigo	integer		integer
rol	nombre	character varying	50	varchar
rol	descripcion	character varying	200	varchar
rol	estado	integer		integer
rol	codigo_modulo	integer		integer
usuario	codigo	bigint		int8

usuario	estado	integer		integer
usuario	ci	character varying	50	varchar
usuario	nombres	character varying	50	varchar
usuario	apellidos	character varying	50	varchar
usuario	mail	character varying	200	varchar
usuario	telefono1	character varying	10	varchar
usuario	telefono2	character varying	10	varchar
usuario	titulo_tercer_nivel	character varying	200	varchar
usuario	titulo_postgrado	character varying	200	varchar
usuario	codtitulo	character varying	200	varchar
usuario	pass	character varying	20	varchar
usuario_rol	codigo	integer		integer
usuario_rol	codigo_rol	integer		integer
usuario_rol	codigo_usuario	bigint		int8
usuario_rol	estado	integer		integer
usuario_rol	fecha_creacion	bigint		int8
usuario_rol	fecha_modificacion	bigint		int8
usuario_rol	codigo_usuario_trans	bigint		int8

3.5.3. Desarrollo

3.5.3.1. Sprint

El Sprint 1 estuvo orientado a las metáforas del sistema principalmente, por lo cual se obtuvo la base de datos a utilizar por el sistema, en cuanto a los sprint del 2 al 13 estuvieron enfocados a la codificación del proyecto, cada uno de estos consistiendo en un entregable alojado en el servidor de pruebas de la facultad disponible para el uso de los docentes destinados a probarlo.

3.5.3.1.1. Historias de Usuario

Las historias de usuario se las utiliza como herramienta para la definición y organización de los requisitos del futuro sistema.

Cada Historia de usuario esta descrita por:

- ID: Es el identificador de la Historia de Usuario seguidas de una numeración, se utilizará HU_ para Historia de Usuario y HT_ para Metáforas del Sistema.
- Nombre de la Historia de Usuario: Nombre descriptivo acordado con el cliente.
- Usuario: Cuál de los miembros es el que necesita este requerimiento.
- Sprint Asignado: Número del Sprint.
- Descripción: Breve explicación del requerimiento.
- Puntos estimados: Estimación del esfuerzo realizada en la fase de planificación.
- Puntos reales: Puntos reales en la realización de la historia de usuario.

Para el desarrollo del proyecto se han identificado un total de 23 historias de usuario, y 5 metáforas del sistema, tras estudiar el sistema y mantener conversaciones con el cliente se han definido las historias de usuario que se visualizan a continuación:

Historia de Usuario	
Número: MT_01	Diseñar Base de Datos
Modificación de historia de usuario:	
Usuario: Desarrollador	Iteración Asignada: 1
Prioridad en el Negocio: Alta	Puntos Estimados: 5
Riesgo en el Desarrollo: Media	Puntos Reales: 5
Descripción: Como desarrollador deseo diseñar una base de datos para mantener los datos disponibles en el transcurso del tiempo.	
Observaciones:	
<ul style="list-style-type: none"> • Utilizar motor de base de datos postgresQL. • Generar diagrama entidad relación de la base terminada. • Generar diccionario de datos correspondiente. 	

Historia de Usuario	
Número: MT_02	Definición de la arquitectura del sistema.
Modificación de historia de usuario:	
Usuario: Desarrollador	Iteración Asignada: 1
Prioridad en el Negocio: Alta	Puntos Estimados: 5
Riesgo en el Desarrollo: Media	Puntos Reales: 5
Descripción: Como desarrollador deseo definir la arquitectura del sistema para ajustar el desarrollo a las necesidades del negocio.	
Observaciones:	
<ul style="list-style-type: none"> • Se utilizará el hardware existente en la facultad. • Se seguirá en la arquitectura MVC. 	

Historia de Usuario	
Número: MT_03	Diseño de las interfaces.
Modificación de historia de usuario:	
Usuario: Desarrollador	Iteración Asignada: 1
Prioridad en el Negocio: Baja	Puntos Estimados: 5
Riesgo en el Desarrollo: Media	Puntos Reales: 2
Descripción: Como desarrollador deseo diseñar las interfaces a utilizar en el sistema para implementar la interacción con el usuario.	
Observaciones:	
<ul style="list-style-type: none"> • Se seguirá el formato establecido por la Escuela de Ingeniería en Sistemas de la ESPOCH. • Se rediseñarán las interfaces del sistema que sean requeridas. 	

Historia de Usuario	
Número: HU_01	Iniciar sesión a los usuarios
Modificación de historia de usuario:	
Usuario: Administrador/Docente/Estudiante	Iteración Asignada: 1
Prioridad en el Negocio: Alta	Puntos Estimados: 5
Riesgo en el Desarrollo: Media	Puntos Reales: 5
Descripción: Como usuario deseo poder ingresar al sistema a través del sistema para poder utilizar la información existente.	
Observaciones:	
<ul style="list-style-type: none"> • Se utilizará la cédula, contraseña y tipo de usuario para acceder al sistema. 	

Historia de Usuario	
Número: HU_02	Registrar los datos de un feriado
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 1
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 7
Descripción: Como administrador deseo poder ingresar feriados al sistema para que estos no sean tomados en cuenta en la planificación.	
Observaciones:	
<ul style="list-style-type: none"> • Se incluirán al inicio del semestre. • Se incluirán únicamente feriados no recuperables. 	

Historia de Usuario	
Número: HU_03	Modificar los datos de un feriado
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 2
Prioridad en el Negocio: Media	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder modificar datos de un feriado para poder corregir errores.	
Observaciones:	
<ul style="list-style-type: none"> • Se cambiará únicamente la fecha del feriado. 	

Historia de Usuario	
Número: HU_04	Eliminar los datos de un feriado
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 3
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 3
Descripción: Como administrador deseo poder eliminar feriados para poder corregir errores.	
Observaciones:	
<ul style="list-style-type: none"> • Se eliminarán feriados desde la misma interfaz de modificación. 	

Historia de Usuario	
Número: HU_05	Registrar los datos de horarios de clase
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 2
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Alto	Puntos Reales: 8.5
Descripción: Como administrador deseo poder registrar los datos de horario de clase de las materias que imparten los docentes para poder ajustar la planificación a ellos.	
Observaciones:	
<ul style="list-style-type: none"> • Se obtendrán los datos a ingresar de los servicios web del sistema OASIS. • Se integrara el ingreso de datos al método de migración del sistema OASIS. 	

Historia de Usuario	
Número: HU_06	Registrar los metadatos del sistema
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 3
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 6
Descripción: Como administrador deseo poder cambiar el valor de los metadatos del sistema para complementar la información necesaria en la planificación.	
Observaciones:	
<ul style="list-style-type: none"> • Se integrarán en la interfaz del administrador del sistema. 	

Historia de Usuario	
Número: HU_07	Modificar los metadatos del sistema
Modificación de historia de usuario:	
Usuario: Desarrollador	Iteración Asignada: 4
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 5
Descripción: Como administrador deseo poder cambiar el valor de los metadatos del sistema para complementar la información necesaria en la planificación.	
Observaciones:	
<ul style="list-style-type: none"> • Se integrarán en la interfaz del administrador del sistema. 	

Historia de Usuario	
Número: HU_08	Eliminar los metadatos del sistema
Modificación de historia de usuario:	
Usuario: Desarrollador	Iteración Asignada: 4
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder eliminar el valor de los metadatos del sistema para complementar la información necesaria en la planificación	
Observaciones:	
<ul style="list-style-type: none"> • Se integrarán en la interfaz del administrador del sistema. 	

Historia de Usuario	
Número: HU_09	Registrar los datos de un período
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 4
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 6
Descripción: Como administrador deseo poder registrar los datos de un período para que este pueda ingresar al sistema.	
Observaciones:	
<ul style="list-style-type: none"> • Se utilizará la nueva nomenclatura de áreas. • Solo existirá un coordinador por área en cada escuela. 	

Historia de Usuario	
Número: HU_10	Modificar los datos de un período
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 5
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder modificar los datos de un período para mantener los datos correctos.	
Observaciones:	

Historia de Usuario	
Número: HU_11	Eliminar los datos de un período
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 5
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder eliminar los datos de un período para mantener los datos correctos.	
Observaciones:	

Historia de Usuario	
Número: HU_12	Registrar los datos de las tutorías por asignatura.
Modificación de historia de usuario:	
Usuario: Docente	Iteración Asignada: 6
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 10

Descripción: Como docente deseo poder registrar los datos de las tutorías por asignatura.
Observaciones: <ul style="list-style-type: none"> • Se permitirá seleccionar la asignatura a la cual ingresar las tutorías.

Historia de Usuario	
Número: HU_13	Registrar identificador de tarjeta magnética del docente.
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 6
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Alta	Puntos Reales: 12
Descripción: Como administrador necesito poder registrar el identificador de la tarjeta magnética del docente.	
Observaciones: <ul style="list-style-type: none"> • El identificador debe ser único para cada docente. 	

Historia de Usuario	
Número: HU_14	Modificar identificador de tarjeta magnética del docente
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 7
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 6
Descripción: Como administrador deseo poder modificar los datos de la identificador de tarjeta magnética del docente	
Observaciones:	

Historia de Usuario	
Número: HU_15	Registrar justificaciones de inasistencias
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 7
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 6
Descripción: Como administrador deseo poder Registrar justificaciones de inasistencias para un control de las mismas.	
Observaciones:	

Historia de Usuario	
Número: HU_16	Registrar evidencias de clases en caso de ser solicitadas.
Modificación de historia de usuario:	
Usuario: Estudiante	Iteración Asignada: 7
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Alta	Puntos Reales: 12
Descripción: Como estudiante necesito poder registrar la evidencia de clases en caso de ser solicitada	
Observaciones:	
<ul style="list-style-type: none"> Varios estudiantes pueden registrar la evidencia de clases por materia. 	

Historia de Usuario	
Número: HU_17	Modificar los datos de horarios de clase
Modificación de historia de usuario:	
Usuario: Docente	Iteración Asignada: 8
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Alta	Puntos Reales: 9.5
Descripción: Como docente deseo poder modificar los datos de horario de clase de las materias que imparten los docentes para poder mantener la información actualizada.	
Observaciones:	
<ul style="list-style-type: none"> • Se obtendrán los datos a ingresar de los mismos servicios web que consume el sistema OASIS. 	

Historia de Usuario	
Número: HU_18	Generar el reporte del horario de clases
Modificación de historia de usuario:	
Usuario: Docente	Iteración Asignada: 8
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como docente deseo poder generar un reporte de horario de clases para evidencia y ayudar a la toma de decisiones.	
Observaciones:	
<ul style="list-style-type: none"> • El reporte debe cumplir con el formato solicitado por la escuela. • Se debe generar el reporte con extensión .pdf 	

--

Historia de Usuario	
Número: HU_19	Generar el reporte de asistencia por materia.
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 9
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder generar un reporte de asistencia por materias para evidencias y ayudar a la toma de decisiones.	
Observaciones:	
<ul style="list-style-type: none"> • El reporte debe cumplir con el formato solicitado por la escuela. • Se debe generar el reporte con extensión .pdf 	

Historia de Usuario	
Número: HU_20	Generar el reporte de asistencia por docente
Modificación de historia de usuario:	
Usuario: Docente	Iteración Asignada: 9
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder generar un reporte de asistencia por docente para evidencias y ayudar a la toma de decisiones.	
Observaciones:	
<ul style="list-style-type: none"> • El reporte debe cumplir con el formato solicitado por la escuela. • Se debe generar el reporte con extensión .pdf 	

Historia de Usuario	
Número: HU_21	Generar el reporte de asistencia por escuela

Modificación de historia de usuario:	
Usuario: Docente	Iteración Asignada: 10
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder generar un reporte de asistencia por escuela para evidencias y ayudar a la toma de decisiones.	
Observaciones:	
<ul style="list-style-type: none"> • El reporte debe cumplir con el formato solicitado por la escuela. • Se debe generar el reporte con extensión .pdf 	

Historia de Usuario	
Número: HU_22	Generar el reporte de asistencia por facultad
Modificación de historia de usuario:	
Usuario: Docente	Iteración Asignada: 10
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder generar un reporte de asistencia por facultad para evidencias y ayudar a la toma de decisiones.	
Observaciones:	
<ul style="list-style-type: none"> • El reporte debe cumplir con el formato solicitado por la escuela. • Se debe generar el reporte con extensión .pdf 	

Historia de Usuario	
Número: HU_23	Generar el reporte de asistencia por carrera
Modificación de historia de usuario:	
Usuario: Docente	Iteración Asignada: 10

Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder generar un reporte de asistencia por carrera para evidencias y ayudar a la toma de decisiones.	
Observaciones:	
<ul style="list-style-type: none"> • El reporte debe cumplir con el formato solicitado por la escuela. • Se debe generar el reporte con extensión .pdf 	

Historia de Usuario	
Número: HU_24	Realizar lector de tarjetas magnéticas.
Modificación de historia de usuario:	
Usuario: Docente	Iteración Asignada: 11
Prioridad en el Negocio: Alta	Puntos Estimados: 30
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como docente deseo poder tener un medio de levantamiento de información.	
Observaciones:	
<ul style="list-style-type: none"> • El medio para obtener información debe permanecer encendido las 24 horas. 	

Historia de Usuario	
Número: HU_25	Modificar justificaciones de inasistencias
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 12
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 4

Descripción: Como administrador deseo poder modificar los datos de las justificaciones de inasistencias para mantener los datos correctos.
Observaciones:

Historia de Usuario	
Número: HU_26	Eliminar justificaciones de inasistencias
Modificación de historia de usuario:	
Usuario: Administrador	Iteración Asignada: 12
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo poder eliminar los datos de las justificaciones de inasistencias para mantener los datos correctos.	
Observaciones:	

Historia de Usuario	
Número: MT_04	Desplegar el sistema en el servidor de aplicaciones de la Facultad
Modificación de historia de usuario:	
Usuario: Desarrollador	Iteración Asignada: 13
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 10
Descripción: Como desarrollador deseo poder realizar el despliegue del sistema web en el servidor de pruebas de la facultad.	
Observaciones:	
<ul style="list-style-type: none"> • Instalar el servidor de aplicaciones Glassfish para el sistema. 	

Historia de Usuario	
Número: MT_05	Implantar la base de datos en el servidor de base de datos de la facultad
Modificación de historia de usuario:	
Usuario: Desarrollador	Iteración Asignada: 13
Prioridad en el Negocio: Alta	Puntos Estimados: 10
Riesgo en el Desarrollo: Media	Puntos Reales: 10
Descripción: Como desarrollador deseo poder realizar el despliegue de la base de datos en el servidor de la facultad.	
Observaciones:	
<ul style="list-style-type: none"> • Tener en cuenta el usuario y contraseña de la base de datos. • Tener instalado el motor de base de datos PostgreSQL 	

3.5.3.1.2. Tareas de Ingeniería

Tareas de ingeniería.- Son cada una de las actividades necesarias para cumplir con el correcto desarrollo del requerimiento especificado en la historia de usuario

Las respectivas tareas de ingeniería generadas para cada una de las historias de usuario se presentan a continuación:

TAREA DE INGENIERÍA	
Historia de Usuario: HT_01 Diseñar base de datos	
Número de Tarea: TI_1	Nombre de Tarea: Diseño del modelo conceptual de la base de datos.
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.5
Programador Responsable: Bautista Fausto y Román José	
Descripción: La creación del diseño del esquema conceptual de la base de datos mediante un modelo entidad relación (DER), para tener una idea clara de las entidades, relaciones y atributos de la base de datos.	
Prueba de aceptación: Que tenga todas las tablas y las relaciones tengan bien definidos sus tipos de datos	

TAREA DE INGENIERÍA	
Historia de Usuario: HT_01 Diseño de la base de datos	
Número de Tarea: TI_2	Nombre de Tarea: Diccionario de datos
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Bautista Fausto y Román José	
Descripción: Realizar el respectivo diccionario de datos de la base implantada	
Prueba de aceptación: Verificar que el diccionario de datos cuente con los campos: nombre tabla, nombre columna, tipo de dato, longitud máxima.	

TAREA DE INGENIERÍA	
Historia de Usuario: HT_02 Definición de la arquitectura del sistema.	
Número de Tarea: TI_3	Nombre de Tarea: Diccionario de datos
Tipo de Tarea: Desarrollo	Puntos Estimados: 2.5
Programador Responsable: Bautista Fausto y Román José	
Descripción: Estudio de la arquitectura del sistema debido que es al que se enlazara el sistema a desarrollar.	
Prueba de aceptación: Verificar que se siga la arquitectura definida. Verificar que la documentación de la arquitectura coincida con su aplicación.	

TAREA DE INGENIERÍA	
Historia de Usuario: HT_03 Diseño de las interfaces.	
Número de Tarea: TI_4	Nombre de Tarea: Definición del estándar de diseño de interfaces.
Tipo de Tarea: Desarrollo	Puntos Estimados: 2.5
Programador Responsable: Bautista Fausto y Román José	
Descripción: Definir las características, colores, fuentes, funcionalidades del sistema.	
Prueba de aceptación: Verificar que el diseño cumpla con la identidad corporativa de la facultad.	

TAREA DE INGENIERÍA	
Historia de Usuario: HT_03 Diseño de las interfaces.	
Número de Tarea: TI_5	Nombre de Tarea: Convertir las imágenes vectoriales a formatos de mapas de bits
Tipo de Tarea: Desarrollo	Puntos Estimados: 2.5
Programador Responsable: Bautista Fausto y Román José	

Descripción: Convertir las imágenes vectoriales a formatos de mapas de bits en alta resolución.

Prueba de aceptación: Verificar que las imágenes convertidas tengan una alta resolución.
Verificar que los colores de fondo de las paginas HTML coincidan con las solicitadas.

TAREA DE INGENIERÍA

Historia de Usuario: HU_02 Registrar los datos de un feriado.

Número de Tarea: TI_6

Nombre de Tarea: Mostrar diálogo de nuevo feriado

Tipo de Tarea: Desarrollo

Puntos Estimados: 2.5

Programador Responsable: Bautista Fausto y Román José

Descripción: Diálogo de primefaces que se comunique con el bean correspondiente para ingresar la información del nuevo feriado

Pruebas de Aceptación;

Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos.

El dialogo se cierra con la tecla escape.

TAREA DE INGENIERÍA

Historia de Usuario: HU_02 Registrar los datos de un feriado.

Número de Tarea: TI_7

Nombre de Tarea: Crear función

Tipo de Tarea: Desarrollo

Puntos Estimados: 2.5

Programador Responsable: Bautista Fausto y Román José

Descripción: Dentro del paquete funciones en la clase FFeriado se crea la función Insertar, recibe por parámetro un objeto de la clase Feriado y retorna un entero correspondiente al índice en el que fue insertado, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgresQL con los parámetros esperados.

Probar la función con datos reales correctos

Probar la función con datos incorrectos

TAREA DE INGENIERÍA

Historia de Usuario: HU_03 Modificar los datos de un feriado.

Número de Tarea: TI_8

Nombre de Tarea: Mostrar listado de feriados

Tipo de Tarea: Desarrollo

Puntos Estimados: 2

Programador Responsable: Bautista Fausto y Román José

Descripción: Tabla de primefaces que se comuniquen con el bean correspondiente para mostrar un ArrayList con todos los feriados de área existentes en la base de datos

Pruebas de Aceptación;

La tabla se carga al abrir el formulario

Muestra la cantidad de registros que existan en la base de datos

TAREA DE INGENIERÍA

Historia de Usuario: HU_03 Modificar los datos de un feriado.

Número de Tarea: TI_9

Nombre de Tarea: Mostrar diálogo del feriado seleccionado

Tipo de Tarea: Desarrollo

Puntos Estimados: 2

Programador Responsable: Bautista Fausto y Román José

Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del feriado de la fila correspondiente.

Pruebas de Aceptación;

Todas las filas tienen el botón para mostrar el dialogo de modificación

Al dar clic en el botón se muestra un dialogo con los datos del feriado correspondiente

TAREA DE INGENIERÍA	
Historia de Usuario: HU_03 Modificar los datos de un feriado.	
Número de Tarea: TI_10	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Bautista Fausto y Román José	
Descripción: Dentro del paquete funciones en la clase FFeriado se crea la función Modificar, recibe por parámetro un objeto de la clase feriado retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.	
Pruebas de Aceptación; Probar el Script en postgresQL con los parámetros esperados. Probar la función con datos reales correctos Probar la función con datos incorrectos	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_04 Eliminar los datos de un feriado.	
Número de Tarea: TI_11	Nombre de Tarea: Mostrar listado de feriado
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Programador Responsable: Bautista Fausto y Román José	
Descripción: Tabla de primefaces que se comunice con el bean correspondiente para mostrar un ArrayList con todos los feriados existentes en la base de datos	
Pruebas de Aceptación; La tabla se carga al abrir el formulario Muestra la cantidad de registros que existan en la base de datos	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_04 Eliminar los datos de un feriado.	

Número de Tarea: TI_12	Nombre de Tarea: Mostrar diálogo del feriado seleccionado
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Programador Responsable: Bautista Fausto y Román José	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del feriado de la fila correspondiente.	
Pruebas de Aceptación; Todas las filas tienen el botón para mostrar el dialogo de eliminación Al dar clic en el botón se muestra un dialogo con los datos del feriado correspondiente	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_04 Eliminar los datos de un feriado.	
Número de Tarea: TI_13	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Bautista Fausto y Román José	
Descripción: Dentro del paquete funciones en la clase FFeriado se crea la función Eliminar, recibe por parámetro un entero correspondiente al código del objeto a eliminar y retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.	
Pruebas de Aceptación; Probar el Script en postgresSQL con los parámetros esperados. Probar la función con datos reales correctos Probar la función con datos incorrectos	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_05 Registrar los datos de horarios de clase.	
Número de Tarea: TI_14	Nombre de Tarea: Consumo web
Tipo de Tarea: Desarrollo	Puntos Estimados: 4

Programador Responsable: Bautista Fausto y Román José
Descripción: Dentro de la clase ConsumoWeb se crea la función obtenerHorarios, recibe como parámetro un objeto usuario y un periodo para devolver la lista de horarios que le corresponden al docente.
Pruebas de Aceptación; Verificar con la información del sistema académico OASIS la coincidencia de los días y horas de las clases.

TAREA DE INGENIERÍA	
Historia de Usuario: HU_05 Registrar los datos de horarios de clase.	
Número de Tarea: TI_15	Nombre de Tarea: Crear método para insertar horario
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Programador Responsable: Bautista Fausto y Román José	
Descripción: Dentro del paquete funciones en la clase FHorario se crea la función Insertar, recibe por parámetro un objeto de la clase horario y retorna un entero correspondiente al índice en el que fue insertado, esta función se encargará de la inyección SQL.	
Pruebas de Aceptación; Probar el Script en postgresSQL con los parámetros esperados. Probar la función con datos reales correctos Probar la función con datos incorrectos	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_05 Registrar los datos de horarios de clase.	
Número de Tarea: TI_16	Nombre de Tarea: Integrar el nuevo método al método de migración
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Programador Responsable: Bautista Fausto y Román José	

Descripción: Dentro del proceso de migración implementar el método de ingreso y modificación a cada una de las materias impartidas por los docentes.

Pruebas de Aceptación;

Migrar una materia sin un horario de clases y verificar la inserción del mismo

Migrar una materia con un horario de clases incorrecto y verificar la corrección del mismo

TAREA DE INGENIERÍA

Historia de Usuario: HU_06 Registrar los metadatos del sistema.

Número de Tarea: TI_17

Nombre de Tarea: Mostrar diálogo de nuevo metadato

Tipo de Tarea: Desarrollo

Puntos Estimados: 2

Programador Responsable: Bautista Fausto y Román José

Descripción: Diálogo de primefaces que se comunique con el bean correspondiente para ingresar la información del nuevo metadato

Pruebas de Aceptación;

Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos.

El dialogo se cierra con la tecla escape.

TAREA DE INGENIERÍA

Historia de Usuario: HU_06 Registrar los metadatos del sistema.

Número de Tarea: TI_18

Nombre de Tarea: Crear función

Tipo de Tarea: Desarrollo

Puntos Estimados: 1

Programador Responsable: Bautista Fausto y Román José

Descripción: Dentro del paquete funciones en la clase FHorario se crea la función Insertar, recibe por parámetro un objeto de la clase coordinador y retorna un entero correspondiente al índice en el que fue insertado, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgresQL con los parámetros esperados.

Probar la función con datos reales correctos
Probar la función con datos incorrectos

TAREA DE INGENIERÍA

Historia de Usuario: HU_07 Modificar los metadatos del sistema.

Número de Tarea: TI_19

Nombre de Tarea: Mostrar lista de metadatos

Tipo de Tarea: Desarrollo

Puntos Estimados: 2

Programador Responsable: Bautista Fausto y Román José

Descripción: Tabla de primefaces que se comunice con el bean correspondiente para mostrar un ArrayList con todos los metadatos existentes en la base de datos

Pruebas de Aceptación;

La tabla se carga al abrir el formulario

Muestra la cantidad de registros que existan en la base de datos

TAREA DE INGENIERÍA

Historia de Usuario: HU_07 Modificar los metadatos del sistema.

Número de Tarea: TI_20

Nombre de Tarea: Mostrar diálogo con la información del metadato seleccionado

Tipo de Tarea: Desarrollo

Puntos Estimados: 2

Programador Responsable: Bautista Fausto y Román José

Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del metadato de la fila correspondiente.

Pruebas de Aceptación;

Todas las filas tienen el botón para mostrar el dialogo de modificación

Al dar clic en el botón se muestra un dialogo con los datos del metadato correspondiente

TAREA DE INGENIERÍA	
Historia de Usuario: HU_07 Modificar los metadatos del sistema.	
Número de Tarea: TI_21	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Bautista Fausto y Román José	
Descripción: Dentro del paquete funciones en la clase FMetadato se crea la función Modificar, recibe por parámetro un objeto de la clase Metadato y retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.	
Pruebas de Aceptación; Probar el Script en postgresSQL con los parámetros esperados. Probar la función con datos reales correctos Probar la función con datos incorrectos	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_08 Eliminar los metadatos del sistema.	
Número de Tarea: TI_22	Nombre de Tarea: Mostrar listado de metadatos
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Programador Responsable: Bautista Fausto y Román José	
Descripción: Tabla de primefaces que se comunice con el bean correspondiente para mostrar un ArrayList con todos los metadatos existentes en la base de datos	
Pruebas de Aceptación; La tabla se carga al abrir el formulario Muestra la cantidad de registros que existan en la base de datos	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_08 Eliminar los metadatos del sistema.	

Número de Tarea: TI_23	Nombre de Tarea: Mostrar diálogo de metadato seleccionado
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Programador Responsable: Bautista Fausto y Román José	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del metadato de la fila correspondiente.	
Pruebas de Aceptación; Todas las filas tienen el botón para mostrar el diálogo de eliminación Al dar click en el botón se muestra un dialogo con los datos del metadato correspondiente	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_08 Eliminar los metadatos del sistema.	
Número de Tarea: TI_24	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Bautista Fausto y Román José	
Descripción: Dentro del paquete funciones en la clase FMetadato se crea la función Eliminar, recibe por parámetro un entero correspondiente al código del objeto a eliminar y retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.	
Pruebas de Aceptación; Probar el Script en postgresSQL con los parámetros esperados. Probar la función con datos reales correctos Probar la función con datos incorrectos	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_09 Registrar los datos de un período.	
Número de Tarea: TI_25	Nombre de Tarea: Mostrar diálogo de nuevo período

Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	
Descripción: Diálogo de primefaces que se comunique con el bean correspondiente para ingresar la información del nuevo período	
Pruebas de Aceptación; Al dar click en el botón nuevo se despliega el dialogo con los campos vacíos. El dialogo se cierra con la tecla escape.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_09 Registrar los datos de un período.	
Número de Tarea: TI_26	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	
Descripción: Dentro del paquete funciones en la clase FPeriodo se crea la función Insertar, recibe por parámetro un objeto de la clase período y retorna un entero correspondiente al índice en el que fue insertado, esta función se encargará de la inyección SQL.	
Pruebas de Aceptación; Probar el Script en postgresQL con los parámetros esperados. Probar la función con datos reales correctos Probar la función con datos incorrectos	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_10 Modificar los datos de un período.	
Número de Tarea: TI_27	Nombre de Tarea: Mostrar listado de período
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	
Descripción: Tabla de primefaces que se comunique con el bean correspondiente para mostrar un ArrayList con todos los períodos existentes en la base de datos	

Pruebas de Aceptación;

La tabla se carga al abrir el formulario

Muestra la cantidad de registros que existan en la base de datos

TAREA DE INGENIERÍA

Historia de Usuario: HU_10 Modificar los datos de un período.

Número de Tarea: TI_28

Nombre de Tarea: Mostrar diálogo del período seleccionado

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del período de la fila correspondiente.

Pruebas de Aceptación;

Todas las filas tienen el botón para mostrar el dialogo de modificación

Al dar click en el botón se muestra un dialogo con los datos del período correspondiente

TAREA DE INGENIERÍA

Historia de Usuario: HU_10 Modificar los datos de un período.

Número de Tarea: TI_29

Nombre de Tarea: Crear función

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Dentro del paquete funciones en la clase FPeriodo se crea la función Modificar, recibe por parámetro un objeto de la clase período y retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgresSQL con los parámetros esperados.

Probar la función con datos reales correctos

Probar la función con datos incorrectos

TAREA DE INGENIERÍA	
Historia de Usuario: HU_11 Eliminar los datos de un período.	
Número de Tarea: TI_30	Nombre de Tarea: Mostrar listado de períodos
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	
Descripción: Tabla de primefaces que se comunice con el bean correspondiente para mostrar un ArrayList con todos los períodos existentes en la base de datos	
Pruebas de Aceptación; La tabla se carga al abrir el formulario Muestra la cantidad de registros que existan en la base de datos	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_11 Eliminar los datos de un período.	
Número de Tarea: TI_31	Nombre de Tarea: Mostrar diálogo de período seleccionado
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del período de la fila correspondiente.	
Pruebas de Aceptación; Todas las filas tienen el botón para mostrar el dialogo de eliminación Al dar click en el botón se muestra un dialogo con los datos del período correspondiente	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_11 Eliminar los datos de un período.	
Número de Tarea: TI_32	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José
Descripción: Dentro del paquete funciones en la clase FPeriodo se crea la función Eliminar, recibe por parámetro un entero período al código del objeto a eliminar y retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.
Pruebas de Aceptación; Probar el Script en postgresSQL con los parámetros esperados. Probar la función con datos reales correctos Probar la función con datos incorrectos

TAREA DE INGENIERÍA	
Historia de Usuario: HU_12 Registrar los datos de las tutorías por asignatura.	
Número de Tarea: TI_33	Nombre de Tarea: Mostrar diálogo de nueva tutoría
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	
Descripción: Diálogo de primefaces que se comunique con el bean correspondiente para ingresar la información de la nueva tutoría	
Pruebas de Aceptación; Al dar click en el botón nuevo se despliega el dialogo con los campos vacíos. El dialogo se cierra con la tecla escape.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_12 Registrar los datos de las tutorías por asignatura.	
Número de Tarea: TI_34	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	

Descripción: Dentro del paquete funciones en la clase FTutoria se crea la función Insertar, recibe por parámetro un objeto de la clase tutoría y retorna un entero correspondiente al índice en el que fue insertado, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgresQL con los parámetros esperados.

Probar la función con datos reales correctos

Probar la función con datos incorrectos

TAREA DE INGENIERÍA

Historia de Usuario: HU_13 Registrar los datos de un identificador de tarjeta magnética del docente.

Número de Tarea: TI_35

Nombre de Tarea: Mostrar diálogo de nuevo identificador de tarjeta magnética del docente

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Diálogo de primefaces que se comunique con el bean correspondiente para ingresar la información del nuevo identificador de tarjeta magnética del docente

Pruebas de Aceptación;

Al dar click en el botón nuevo se despliega el dialogo con los campos vacíos.

El dialogo se cierra con la tecla escape.

TAREA DE INGENIERÍA

Historia de Usuario: HU_13 Registrar los datos de un identificador de tarjeta magnética del docente.

Número de Tarea: TI_36

Nombre de Tarea: Crear función

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Dentro del paquete funciones en la clase FIdentificador se crea la función Insertar, recibe por parámetro un objeto de la clase identificador de tarjeta magnética del

docente y retorna un entero correspondiente al índice en el que fue insertado, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgreSQL con los parámetros esperados.

Probar la función con datos reales correctos

Probar la función con datos incorrectos

TAREA DE INGENIERÍA

Historia de Usuario: HU_14 Modificar los datos de un identificador de tarjeta magnética del docente.

Número de Tarea: TI_37

Nombre de Tarea: Mostrar listado de identificador de tarjeta magnética del docente

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Tabla de primefaces que se comunice con el bean correspondiente para mostrar un ArrayList con todos los identificador de tarjeta magnética del docentes existentes en la base de datos

Pruebas de Aceptación;

La tabla se carga al abrir el formulario

Muestra la cantidad de registros que existan en la base de datos

TAREA DE INGENIERÍA

Historia de Usuario: HU_14 Modificar los datos de un identificador de tarjeta magnética del docente.

Número de Tarea: TI_38

Nombre de Tarea: Mostrar diálogo del identificador de tarjeta magnética del docente seleccionado

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del identificador de tarjeta magnética del docente de la fila correspondiente.

Pruebas de Aceptación;

Todas las filas tienen el botón para mostrar el dialogo de modificación

Al dar click en el botón se muestra un dialogo con los datos del identificador de tarjeta magnética del docente correspondiente

TAREA DE INGENIERÍA

Historia de Usuario: HU_14 Modificar los datos de un identificador de tarjeta magnética del docente.

Número de Tarea: TI_39

Nombre de Tarea: Crear función

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Dentro del paquete funciones en la clase FPeriodo se crea la función Modificar, recibe por parámetro un objeto de la clase identificador de tarjeta magnética del docente y retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgreSQL con los parámetros esperados.

Probar la función con datos reales correctos

Probar la función con datos incorrectos

TAREA DE INGENIERÍA

Historia de Usuario: HU_15 Registrar justificaciones de inasistencias

Número de Tarea: TI_40

Nombre de Tarea: Mostrar diálogo de nuevo identificador de tarjeta magnética del docente

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Diálogo de primefaces que se comunique con el bean correspondiente para ingresar la información del nuevo identificador de tarjeta magnética del docente

Pruebas de Aceptación;

Al dar click en el botón nuevo se despliega el dialogo con los campos vacíos.

El dialogo se cierra con la tecla escape.

TAREA DE INGENIERÍA

Historia de Usuario: HU_15 Registrar justificaciones de inasistencias

Número de Tarea: TI_

Nombre de Tarea: Crear función

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Dentro del paquete funciones en la clase FIdentificador se crea la función Insertar, recibe por parámetro un objeto de la clase identificador de tarjeta magnética del docente y retorna un entero correspondiente al índice en el que fue insertado, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgreSQL con los parámetros esperados.

Probar la función con datos reales correctos

Probar la función con datos incorrectos

TAREA DE INGENIERÍA

Historia de Usuario: HU_16 Registrar evidencias de clases en caso de ser solicitadas.

Número de Tarea: TI_40

Nombre de Tarea: Mostrar diálogo de nuevo identificador de tarjeta magnética del docente

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Diálogo de primefaces que se comunique con el bean correspondiente para ingresar la información de la nueva evidencia de clases.

Pruebas de Aceptación;

Al dar click en el botón nuevo se despliega el dialogo con los campos vacíos.

El dialogo se cierra con la tecla escape.

TAREA DE INGENIERÍA

Historia de Usuario: HU_16 Registrar evidencias de clases en caso de ser solicitadas.

Número de Tarea: TI_41

Nombre de Tarea: Crear función

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Dentro del paquete funciones en la clase FEvidencias se crea la función Insertar, recibe por parámetro un objeto de la clase evidencia y retorna un entero correspondiente al índice en el que fue insertado, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgresSQL con los parámetros esperados.

Probar la función con datos reales correctos

Probar la función con datos incorrectos

TAREA DE INGENIERÍA

Historia de Usuario: HU_17 Modificar los datos de un horario de clases.

Número de Tarea: TI_42

Nombre de Tarea: Mostrar listado de horario de clases

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Tabla de primefaces que se comunice con el bean correspondiente para mostrar un ArrayList con todos los horario de clases existentes en la base de datos

Pruebas de Aceptación;

La tabla se carga al abrir el formulario

Muestra la cantidad de registros que existan en la base de datos

TAREA DE INGENIERÍA	
Historia de Usuario: HU_17 Modificar los datos de un horario de clases.	
Número de Tarea: TI_43	Nombre de Tarea: Mostrar diálogo del horario de clases seleccionado
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del horario de clases de la fila correspondiente.	
Pruebas de Aceptación; <p>Todas las filas tienen el botón para mostrar el dialogo de modificación</p> <p>Al dar click en el botón se muestra un dialogo con los datos del horario de clases correspondiente</p>	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_17 Modificar los datos de un horario de clases.	
Número de Tarea: TI_44	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	
Descripción: Dentro del paquete funciones en la clase FHorario se crea la función Modificar, recibe por parámetro un objeto de la clase horario de clases y retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.	
Pruebas de Aceptación; <p>Probar el Script en postgresQL con los parámetros esperados.</p> <p>Probar la función con datos reales correctos</p> <p>Probar la función con datos incorrectos</p>	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_18 Generar el reporte del horario de clases	
Número de Tarea: TI_45	Nombre de Tarea: Generar .pdf
Tipo de Tarea: Desarrollo	Puntos Estimados: 2.5
Programador Responsable: Bautista Fausto y Román José	
Descripción: Permitir generar un archivo pdf en el que contendrá el horario de clases del docente.	
Pruebas de Aceptación; Generar archivo con extensión pdf Probar que se genere correctamente el grafico estadístico.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_19 Generar el reporte de asistencia por materias.	
Número de Tarea: TI_46	Nombre de Tarea: Generar .pdf
Tipo de Tarea: Desarrollo	Puntos Estimados: 2.5
Programador Responsable: Bautista Fausto y Román José	
Descripción: Permitir generar un archivo pdf en el que contendrá el registro de la asistencia del docente por materias	
Pruebas de Aceptación; Generar archivo con extensión pdf Probar que se genere correctamente el grafico estadístico.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_20 Generar el reporte de asistencia por docente	
Número de Tarea: TI_47	Nombre de Tarea: Generar .pdf
Tipo de Tarea: Desarrollo	Puntos Estimados: 2.5
Programador Responsable: Bautista Fausto y Román José	

Descripción: Permitir generar un archivo pdf en el que contendrá el registro de la asistencia del docente por docente

Pruebas de Aceptación;

Generar archivo con extensión pdf

Probar que se genere correctamente el grafico estadístico.

TAREA DE INGENIERÍA

Historia de Usuario: HU_21 Generar el reporte de asistencia por escuela

Número de Tarea: TI_48

Nombre de Tarea: Generar .pdf

Tipo de Tarea: Desarrollo

Puntos Estimados: 2.5

Programador Responsable: Bautista Fausto y Román José

Descripción: Permitir generar un archivo pdf en el que contendrá el registro de la asistencia del docente por escuela

Pruebas de Aceptación;

Generar archivo con extensión pdf

Probar que se genere correctamente el grafico estadístico.

TAREA DE INGENIERÍA

Historia de Usuario: HU_22 Generar el reporte de asistencia por facultad.

Número de Tarea: TI_49

Nombre de Tarea: Generar .pdf

Tipo de Tarea: Desarrollo

Puntos Estimados: 2.5

Programador Responsable: Bautista Fausto y Román José

Descripción: Permitir generar un archivo pdf en el que contendrá el registro de la asistencia del docente por facultad

Pruebas de Aceptación;

Generar archivo con extensión pdf

Probar que se genere correctamente el grafico estadístico.

TAREA DE INGENIERÍA	
Historia de Usuario: HU_23 Generar el reporte de asistencia por carrera.	
Número de Tarea: TI_50	Nombre de Tarea: Generar .pdf
Tipo de Tarea: Desarrollo	Puntos Estimados: 2.5
Programador Responsable: Bautista Fausto y Román José	
Descripción: Permitir generar un archivo pdf en el que contendrá el registro de la asistencia del docente por carrera	
Pruebas de Aceptación; Generar archivo con extensión pdf Probar que se genere correctamente el grafico estadístico.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_24 Realizar lector de tarjetas magnéticas.	
Número de Tarea: TI_51	Nombre de Tarea: Generar .pdf
Tipo de Tarea: Desarrollo	Puntos Estimados: 2.5
Programador Responsable: Bautista Fausto y Román José	
Descripción: Permitir generar un dispositivo hardware que permita recolectar información de la asistencia.	
Pruebas de Aceptación; Comprobar que se haya realizado correctamente la comunicación de los materiales. Verificar que exista comunicación entre e lector de tarjetas magnéticas con la aplicación web.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_25 Modificar justificaciones de inasistencias.	
Número de Tarea: TI_52	Nombre de Tarea: Mostrar listado de justificaciones de inasistencias
Tipo de Tarea: Desarrollo	Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José
Descripción: Tabla de primefaces que se comunice con el bean correspondiente para mostrar un ArrayList con todos los justificaciones de inasistencias existentes en la base de datos
Pruebas de Aceptación; La tabla se carga al abrir el formulario Muestra la cantidad de registros que existan en la base de datos

TAREA DE INGENIERÍA	
Historia de Usuario: HU_25 Modificar justificaciones de inasistencias.	
Número de Tarea: TI_53	Nombre de Tarea: Mostrar diálogo del justificaciones de inasistencias seleccionado
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del justificaciones de inasistencias de la fila correspondiente.	
Pruebas de Aceptación; Todas las filas tienen el botón para mostrar el dialogo de modificación Al dar click en el botón se muestra un dialogo con los datos de justificaciones de inasistencias correspondiente	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_25 Modificar justificaciones de inasistencias.	
Número de Tarea: TI_54	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Puntos Estimados:
Programador Responsable: Bautista Fausto y Román José	

Descripción: Dentro del paquete funciones en la clase FInasistencia se crea la función Modificar, recibe por parámetro un objeto de la clase justificaciones de inasistencias y retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgreSQL con los parámetros esperados.

Probar la función con datos reales correctos

Probar la función con datos incorrectos

TAREA DE INGENIERÍA

Historia de Usuario: HU_26 Eliminar justificaciones de inasistencias

Número de Tarea: TI_55

Nombre de Tarea: Mostrar listado de períodos

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Tabla de primefaces que se comunice con el bean correspondiente para mostrar un ArrayList con todos los períodos existentes en la base de datos.

Pruebas de Aceptación;

La tabla se carga al abrir el formulario

Muestra la cantidad de registros que existan en la base de datos

TAREA DE INGENIERÍA

Historia de Usuario: HU_26 Eliminar justificaciones de inasistencias

Número de Tarea: TI_56

Nombre de Tarea: Mostrar diálogo de período seleccionado

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el dialogo, el cual debe contener los datos del período de la fila correspondiente.

Pruebas de Aceptación;

Todas las filas tienen el botón para mostrar el diálogo de eliminación

Al dar click en el botón se muestra un diálogo con los datos de la justificación de inasistencia correspondiente

TAREA DE INGENIERÍA

Historia de Usuario: HU_26 Eliminar justificaciones de inasistencias

Número de Tarea: TI_57

Nombre de Tarea: Crear función

Tipo de Tarea: Desarrollo

Puntos Estimados:

Programador Responsable: Bautista Fausto y Román José

Descripción: Dentro del paquete funciones en la clase FInasistencia se crea la función Eliminar, recibe por parámetro un entero período al código del objeto a eliminar y retorna un boolean correspondiente al éxito del proceso, esta función se encargará de la inyección SQL.

Pruebas de Aceptación;

Probar el Script en postgresql con los parámetros esperados.

Probar la función con datos reales correctos

Probar la función con datos incorrectos

3.5.3.1.3. Pruebas de Aceptación

Pruebas de Aceptación.- Condiciones que debe cumplir la tarea de ingeniería para considerarse correctamente desarrollada.

A continuación se describen las respectivas pruebas de aceptación generadas para cada una de las tareas de ingeniería.

Prueba de Aceptación

Código: PA_01

Historia de Usuario: MS_02 Diseño de la Base de Datos

Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un nuevo feriado vacíos.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase feriado y la tabla feriado de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de feriado. Se dará clic al botón nuevo feriado. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla feriado y de la clase feriado.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_02	Historia de Usuario: HU_02 Registrar los datos de un feriado
Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un nuevo feriado vacíos.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase feriado y la tabla feriado de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de feriado. Se dará clic al botón nuevo feriado. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla feriado y de la clase feriado.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_03	Historia de Usuario: HU_02 Registrar los datos de un feriado
Nombre: El dialogo se cierra con la tecla escape.	

Responsable: Bautista Fausto y Román José
Descripción: Al presionar la tecla Esc, el diálogo crear nuevo feriado debe cerrarse.
Condiciones de Ejecución: La tecla Esc esté funcionando correctamente.
Pasos de ejecución: Presionar la tecla Esc. Visualizar que el diálogo Nuevo Feriado se cierre.
Resultado esperado: El diálogo se cierre una vez que se presione la tecla Esc
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_04	Historia de Usuario: HU_02 Registrar los datos de un feriado.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiara el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_05	Historia de Usuario: HU_02 Registrar los datos de un feriado.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic al botón Nuevo Feriado para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de aceptación
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_06	Historia de Usuario: HU_02 Registrar los datos de un feriado.
Nombre: Probar la función con datos incorrectos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario incorrecto a ingresar Se hará clic al botón Nuevo Feriado para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de error correspondiente	
Resultado esperado: Se emite un mensaje correspondiente al error producido.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_07	Historia de Usuario: HU_03 Modificar los datos de un feriado.
Nombre: La tabla se carga al abrir el formulario	

Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los feriados existentes.	
Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Feriados Se verificará visualmente que se muestre la lista de feriados en una tabla	
Resultado esperado: Un formulario con una tabla llena con la información correspondiente a los feriados existentes en la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_08	Historia de Usuario: HU_03 Modificar los datos de un feriado.
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los feriados existentes.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Feriados Se verificará visualmente que la cantidad de registros de los feriados	
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_09	Historia de Usuario: HU_03 Modificar los datos de un feriado.

Nombre: Todas las filas tienen el botón para mostrar el diálogo de modificación	
Responsable: Bautista Fausto y Román José	
Descripción: Todas las filas tienen el botón para mostrar el diálogo de edición	
Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el diálogo de edición.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de feriados Verificar visualmente	
Resultado esperado: Cada fila contiene el botón en la columna requerida.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_10	Historia de Usuario: HU_03 Modificar los datos de un feriado.
Nombre: Al dar clic en el botón se muestra un diálogo con los datos del feriado correspondiente	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón editar se debe desplegar un diálogo con los campos correspondientes al feriado seleccionado.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase feriado y la tabla feriado de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de feriados Se dará clic al botón editar. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla feriado y de la clase feriado.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_11	Historia de Usuario: HU_03 Modificar los datos de un feriado.

Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiará el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido modificado de la tabla	
Resultado esperado: La tabla contiene el registro con los datos modificados	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_12	Historia de Usuario: HU_03 Modificar los datos de un feriado.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un feriado correcto a eliminar. Se hará clic al botón editar para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que la lista tenga el registro con los datos seleccionados	
Resultado esperado: La tabla contiene el registro con los nuevos valores asignados	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_13	Historia de Usuario: HU_04 Eliminar los datos de un feriado.
Nombre: La tabla se carga al abrir el formulario	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los feriados existentes.	
Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Feriados Se verificará visualmente que se muestre la lista de feriados en una tabla	
Resultado esperado: Un formulario con una tabla llena con la información correspondiente a los feriados existentes en la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_14	Historia de Usuario: HU_04 Eliminar los datos de un feriado.
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los feriados existentes.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Feriados Se verificará visualmente que la cantidad de registros de los feriados	
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_15	Historia de Usuario: HU_04 Eliminar los datos de un feriado.
Nombre: Todas las filas tienen el botón para mostrar el dialogo de eliminación	
Responsable: Bautista Fausto y Román José	
Descripción: Todas las filas tienen el botón para mostrar el dialogo de eliminación	
Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el dialogo de eliminación.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de feriados Verificar visualmente	
Resultado esperado: Cada fila contiene el botón en la columna requerida.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_16	Historia de Usuario: HU_04 Eliminar los datos de un feriado.
Nombre: Al dar clic en el botón se muestra un dialogo con los datos del feriado correspondiente	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón eliminar se debe desplegar un dialogo con los campos correspondientes al feriado seleccionado.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase feriado y la tabla feriado de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de feriados Se dará clic al botón eliminar. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla feriado y de la clase feriado.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_17	Historia de Usuario: HU_04 Eliminar los datos de un feriado.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiará el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_18	Historia de Usuario: HU_04 Eliminar los datos de un feriado.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic icono de la lupa y se mostrara un diálogo Se hará clic en el botón Eliminar Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_19 **Historia de Usuario:** HU_04 Eliminar los datos de un feriado.

Nombre: Probar la función con datos incorrectos.

Responsable: Bautista Fausto y Román
José

Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.

Pasos de ejecución:

Crear método main en la clase Feriado

Invocar al método fferiado.eliminar enviando como parámetro -1

Resultado esperado: En la excepción se genere el mensaje correspondiente al error producido.

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_20 **Historia de Usuario:** HU_05 Registrar los datos de un horario de clases

Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos

Responsable: Bautista Fausto y Román
José

Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un nuevo horario de clases vacíos.

Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase horario de clases y la tabla horario de clases de la base de datos.

Pasos de ejecución: Se navegará hasta la interfaz de gestión de horario de clases.

Se dará clic al botón nuevo horario de clases.

Comparar en orden la igualdad entre ambos elementos.

Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla horario de clases y de la clase horario de clases.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_21	Historia de Usuario: HU_05 Registrar los datos de un horario de clases
Nombre: El dialogo se cierra con la tecla escape.	
Responsable: Bautista Fausto y Román José	
Descripción: Al presionar la tecla Esc, el diálogo crear nuevo horario de clases debe cerrarse.	
Condiciones de Ejecución: La tecla Esc esté funcionando correctamente.	
Pasos de ejecución: Presionar la tecla Esc. Visualizar que el diálogo Nuevo Horario de clases se cierre.	
Resultado esperado: El diálogo se cierre una vez que se presione la tecla Esc	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_22	Historia de Usuario: HU_05 Registrar los datos de un horario de clases.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiara el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla	

Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_23	Historia de Usuario: HU_05 Registrar los datos de un horario de clases.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic al botón Nuevo Horario de clases para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_24	Historia de Usuario: HU_05 Registrar los datos de un horario de clases.
Nombre: Probar la función con datos incorrectos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	

<p>Pasos de ejecución:</p> <p>Se utilizará la interfaz para asegurarnos que se tenga un usuario incorrecto a ingresar</p> <p>Se hará clic al botón Nuevo Horario de clases para que se muestre el diálogo</p> <p>Se hará clic en el botón aceptar del diálogo</p> <p>Se verificará visualmente que se muestre el mensaje de error correspondiente</p>
<p>Resultado esperado: Se emite un mensaje correspondiente al error producido.</p>
<p>Evaluación de la prueba: Exitosa.</p>

Prueba de Aceptación	
Código: PA_25	Historia de Usuario: HU_17 Modificar los datos de un horario de clases.
Nombre: La tabla se carga al abrir el formulario	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los horario de clases existentes.	
Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador	
<p>Pasos de ejecución:</p> <p>Se hará clic al botón Administrar</p> <p>Se hará clic en el botón Horario de clases</p> <p>Se verificará visualmente que se muestre la lista de horario de clases en una tabla</p>	
Resultado esperado: Un formulario con una tabla llena con la información correspondiente a los horario de clases existentes en la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_26	Historia de Usuario: HU_17 Modificar los datos de un horario de clases.
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	

Descripción: Se visualiza la tabla con los datos correspondientes a los horario de clases existentes.
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Horario de clases Se verificará visualmente que la cantidad de registros de los horario de clases
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_27	Historia de Usuario: HU_17 Modificar los datos de un horario de clases.
Nombre: Todas las filas tienen el botón para mostrar el dialogo de modificación	
Responsable: Bautista Fausto y Román José	
Descripción: Todas las filas tienen el botón para mostrar el dialogo de edición	
Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el dialogo de edición.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de horario de clases Verificar visualmente	
Resultado esperado: Cada fila contiene el botón en la columna requerida.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_28	Historia de Usuario: HU_17 Modificar los datos de un horario de clases.

Nombre: Al dar clic en el botón se muestra un dialogo con los datos del horario de clases correspondiente	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón editar se debe desplegar un dialogo con los campos correspondientes al horario de clases seleccionado.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase horario de clases y la tabla horario de clases de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de horario de clases Se dará clic al botón editar. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla horario de clases y de la clase horario de clases.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_29	Historia de Usuario: HU_17 Modificar los datos de un horario de clases.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiara el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido modificado de la tabla	
Resultado esperado: La tabla contiene el registro con los datos modificados	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_30	Historia de Usuario: HU_17 Modificar los datos de un horario de clases.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un horario de clases correcto a eliminar. Se hará clic al botón editar para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que la lista tenga el registro con los datos seleccionados	
Resultado esperado: La tabla contiene el registro con los nuevos valores asignados	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_31	Historia de Usuario: HU_06 Registrar los datos de un horario de clases
Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un nuevo horario de clases vacíos.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase horario de clases y la tabla horario de clases de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de metadatos del sistema	

Se dará clic al botón nuevo metadatos del sistema
Comparar en orden la igualdad entre ambos elementos.
Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla horario de clases y de la clase metadatos del sistema
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_32	Historia de Usuario: HU_06 Registrar los datos de un metadatos del sistema
Nombre: El dialogo se cierra con la tecla escape.	
Responsable: Bautista Fausto y Román José	
Descripción: Al presionar la tecla Esc, el diálogo crear nuevo metadatos del sistema debe cerrarse.	
Condiciones de Ejecución: La tecla Esc esté funcionando correctamente.	
Pasos de ejecución: Presionar la tecla Esc. Visualizar que el diálogo Nuevo metadatos del sistema se cierre.	
Resultado esperado: El diálogo se cierre una vez que se presione la tecla Esc	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_33	Historia de Usuario: HU_06 Registrar los datos de un metadatos del sistema
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiara el script a la interfaz pgadmin	

Se reemplazarán los comodines por información correcta
Se ejecutará el script
Se revisará visualmente que el campo haya sido ingresado de la tabla
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_34	Historia de Usuario: HU_06 Registrar los datos de un metadatos del sistema
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic al botón Nuevo Horario de clases para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_35	Historia de Usuario: HU_06 Registrar los datos de un metadatos del sistema
Nombre: Probar la función con datos incorrectos.	
Responsable: Bautista Fausto y Román José	

Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario incorrecto a ingresar Se hará clic al botón Nuevo Horario de clases para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de error correspondiente
Resultado esperado: Se emite un mensaje correspondiente al error producido.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_36	Historia de Usuario: HU_07 Modificar los datos de un metadatos del sistema
Nombre: La tabla se carga al abrir el formulario	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los metadatos del sistema existentes.	
Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón metadatos del sistema Se verificará visualmente que se muestre la lista metadatos del sistema en una tabla	
Resultado esperado: Un formulario con una tabla llena con la información correspondiente a los metadatos del sistema existentes en la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación

Código: PA_37	Historia de Usuario: HU_07 Modificar los datos de un metadatos del sistema
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los metadatos del sistema existentes.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Metadatos del sistema Se verificará visualmente que la cantidad de registros de los metadatos del sistema	
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_38	Historia de Usuario: HU_07 Modificar los datos de un metadatos del sistema
Nombre: Todas las filas tienen el botón para mostrar el dialogo de modificación	
Responsable: Bautista Fausto y Román José	
Descripción: Todas las filas tienen el botón para mostrar el dialogo de edición	
Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el dialogo de edición.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de metadatos del sistema Verificar visualmente	
Resultado esperado: Cada fila contiene el botón en la columna requerida.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_39	Historia de Usuario: HU_07 Modificar los datos de un metadatos del sistema
Nombre: Al dar clic en el botón se muestra un dialogo con los datos del horario de clases correspondiente	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón editar se debe desplegar un dialogo con los campos correspondientes al horario de clases seleccionado.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase horario de clases y la tabla horario de clases de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de metadatos del sistema Se dará clic al botón editar. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla horario de clases y de la clase metadatos del sistema	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_40	Historia de Usuario: HU_07 Modificar los datos de un metadatos del sistema
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de padmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente padmin con una conexión establecida.	
Pasos de ejecución: Se copiará el script a la interfaz padmin Se reemplazarán los comodines por información correcta Se ejecutará el script	

Se revisará visualmente que el campo haya sido modificado de la tabla
Resultado esperado: La tabla contiene el registro con los datos modificados
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_41	Historia de Usuario: HU_07 Modificar los datos de un metadatos del sistema
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un horario de clases correcto a eliminar. Se hará clic al botón editar para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que la lista tenga el registro con los datos seleccionados	
Resultado esperado: La tabla contiene el registro con los nuevos valores asignados	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_42	Historia de Usuario: HU_08 Eliminar los datos de un metadatos del sistema.
Nombre: La tabla se carga al abrir el formulario	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los metadatos del sistema existente.	

Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Metadatos del sistemas Se verificará visualmente que se muestre la lista de metadatos del sistemas en una tabla
Resultado esperado: Un formulario con una tabla llena con la información correspondiente a los metadatos del sistemas existentes en la base de datos.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_43	Historia de Usuario: HU_08 Eliminar los datos de un metadatos del sistema.
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los metadatos del sistema existente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Metadatos del sistemas Se verificará visualmente que la cantidad de registros de los metadatos del sistemas	
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_44	Historia de Usuario: HU_08 Eliminar los datos de un metadatos del sistema.
Nombre: Todas las filas tienen el botón para mostrar el dialogo de eliminación	

Responsable: Bautista Fausto y Román José	
Descripción: Todas las filas tienen el botón para mostrar el diálogo de eliminación	
Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el diálogo de eliminación.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de metadatos del sistema Verificar visualmente	
Resultado esperado: Cada fila contiene el botón en la columna requerida.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_45	Historia de Usuario: HU_08 Eliminar los datos de un metadatos del sistema.
Nombre: Al dar clic en el botón se muestra un diálogo con los datos del metadatos del sistema correspondiente	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón eliminar se debe desplegar un diálogo con los campos correspondientes al metadatos del sistema seleccionado.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase metadatos del sistema y la tabla metadatos del sistema de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de metadatos del sistema Se dará clic al botón eliminar. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla metadatos del sistema y de la clase metadatos del sistema.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación

Código: PA_46	Historia de Usuario: HU_08 Eliminar los datos de un metadatos del sistema.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiara el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_47	Historia de Usuario: HU_08 Eliminar los datos de un metadatos del sistema.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic icono de la lupa y se mostrara un diálogo Se hará clic en el botón Eliminar Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_48	Historia de Usuario: HU_08 Eliminar los datos de un metadatos del sistema.
----------------------	-----------------------------------------------------------------------------------

Nombre: Probar la función con datos incorrectos.

Responsable: Bautista Fausto y Román José

Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.

Pasos de ejecución:

Crear método main en la clase Metadatos del sistema

Invocar al método fmetadatos del sistema.eliminar enviando como parámetro -1

Resultado esperado: En la excepción se genere el mensaje correspondiente al error producido.

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_49	Historia de Usuario: HU_09 Registrar los datos de un periodo
----------------------	---------------------------------------------------------------------

Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos

Responsable: Bautista Fausto y Román José

Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un nuevo periodo vacíos.

Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase periodo y la tabla periodo de la base de datos.

Pasos de ejecución: Se navegará hasta la interfaz de gestión de periodo.

Se dará clic al botón nuevo periodo.

Comparar en orden la igualdad entre ambos elementos.

Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla periodo y de la clase periodo.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_50	Historia de Usuario: HU_09 Registrar los datos de un periodo
Nombre: El dialogo se cierra con la tecla escape.	
Responsable: Bautista Fausto y Román José	
Descripción: Al presionar la tecla Esc, el diálogo crear nuevo periodo debe cerrarse.	
Condiciones de Ejecución: La tecla Esc esté funcionando correctamente.	
Pasos de ejecución: Presionar la tecla Esc. Visualizar que el diálogo Nuevo Periodo se cierre.	
Resultado esperado: El diálogo se cierre una vez que se presione la tecla Esc	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_51	Historia de Usuario: HU_09 Registrar los datos de un periodo.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiará el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_52 **Historia de Usuario:** HU_09 Registrar los datos de un periodo.

Nombre: Probar la función con datos reales correctos.

Responsable: Bautista Fausto y Román
José

Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.

Pasos de ejecución:

Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar

Se hará clic al botón Nuevo Periodo para que se muestre el diálogo

Se hará clic en el botón aceptar del diálogo

Se verificará visualmente que se muestre el mensaje de aceptación

Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_53 **Historia de Usuario:** HU_09 Registrar los datos de un periodo.

Nombre: Probar la función con datos incorrectos.

Responsable: Bautista Fausto y Román
José

Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.

Pasos de ejecución:

Se utilizará la interfaz para asegurarnos que se tenga un usuario incorrecto a ingresar

Se hará clic al botón Nuevo Periodo para que se muestre el diálogo

Se hará clic en el botón aceptar del diálogo
Se verificará visualmente que se muestre el mensaje de error correspondiente
Resultado esperado: Se emite un mensaje correspondiente al error producido.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_54	Historia de Usuario: HU_10 Modificar los datos de un periodo.
Nombre: La tabla se carga al abrir el formulario	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los periodos existentes.	
Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Periodos Se verificará visualmente que se muestre la lista de periodos en una tabla	
Resultado esperado: Un formulario con una tabla llena con la información correspondiente a los periodos existentes en la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_55	Historia de Usuario: HU_10 Modificar los datos de un periodo.
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los periodos existentes.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se hará clic al botón Administrar	

Se hará clic en el botón Periodos	
Se verificará visualmente que la cantidad de registros de los periodos	
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_56	Historia de Usuario: HU_10 Modificar los datos de un periodo.
Nombre: Todas las filas tienen el botón para mostrar el dialogo de modificación	
Responsable: Bautista Fausto y Román José	
Descripción: Todas las filas tienen el botón para mostrar el dialogo de edición	
Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el dialogo de edición.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de periodos Verificar visualmente	
Resultado esperado: Cada fila contiene el botón en la columna requerida.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_57	Historia de Usuario: HU_10 Modificar los datos de un periodo.
Nombre: Al dar clic en el botón se muestra un dialogo con los datos del periodo correspondiente	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón editar se debe desplegar un dialogo con los campos correspondientes al periodo seleccionado.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase periodo y la tabla periodo de la base de datos.	

<p>Pasos de ejecución: Se navegará hasta la interfaz de gestión de periodos</p> <p>Se dará clic al botón editar.</p> <p>Comparar en orden la igualdad entre ambos elementos.</p>
<p>Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla periodo y de la clase periodo.</p>
<p>Evaluación de la prueba: Exitosa.</p>

Prueba de Aceptación	
Código: PA_58	Historia de Usuario: HU_10 Modificar los datos de un periodo.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
<p>Pasos de ejecución: Se copiara el script a la interfaz pgadmin</p> <p>Se reemplazarán los comodines por información correcta</p> <p>Se ejecutará el script</p> <p>Se revisará visualmente que el campo haya sido modificado de la tabla</p>	
Resultado esperado: La tabla contiene el registro con los datos modificados	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_59	Historia de Usuario: HU_10 Modificar los datos de un periodo.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un periodo correcto a eliminar. Se hará clic al botón editar para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que la lista tenga el registro con los datos seleccionados
Resultado esperado: La tabla contiene el registro con los nuevos valores asignados
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_60	Historia de Usuario: HU_11 Eliminar los datos de un periodo.
Nombre: La tabla se carga al abrir el formulario	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los periodos existentes.	
Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Periodos Se verificará visualmente que se muestre la lista de periodos en una tabla	
Resultado esperado: Un formulario con una tabla llena con la información correspondiente a los periodos existentes en la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_61	Historia de Usuario: HU_11 Eliminar los datos de un periodo.
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los periodos existentes.	

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Periodos Se verificará visualmente que la cantidad de registros de los periodos	
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_62	Historia de Usuario: HU_11 Eliminar los datos de un periodo.
Nombre: Todas las filas tienen el botón para mostrar el dialogo de eliminación	
Responsable: Bautista Fausto y Román José	
Descripción: Todas las filas tienen el botón para mostrar el dialogo de eliminación	
Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el dialogo de eliminación.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de periodos Verificar visualmente	
Resultado esperado: Cada fila contiene el botón en la columna requerida.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_63	Historia de Usuario: HU_11 Eliminar los datos de un periodo.
Nombre: Al dar clic en el botón se muestra un dialogo con los datos del periodo correspondiente	
Responsable: Bautista Fausto y Román José	

Descripción: Al dar clic en el botón eliminar se debe desplegar un dialogo con los campos correspondientes al periodo seleccionado.
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase periodo y la tabla periodo de la base de datos.
Pasos de ejecución: Se navegará hasta la interfaz de gestión de periodos Se dará clic al botón eliminar. Comparar en orden la igualdad entre ambos elementos.
Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla periodo y de la clase periodo.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_64	Historia de Usuario: HU_11 Eliminar los datos de un periodo.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiara el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_65	Historia de Usuario: HU_11 Eliminar los datos de un periodo.
Nombre: Probar la función con datos reales correctos.	

Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic icono de la lupa y se mostrara un diálogo Se hará clic en el botón Eliminar Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_66	Historia de Usuario: HU_11 Eliminar los datos de un periodo.
Nombre: Probar la función con datos incorrectos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Crear método main en la clase Periodo Invocar al método fperiodo.eliminar enviando como parámetro -1	
Resultado esperado: En la excepción se genere el mensaje correspondiente al error producido.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_67	Historia de Usuario: HU_12 Registrar los datos de un tutorías por asignatura

Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un nuevo tutorías por asignatura vacíos.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase tutorías por asignatura y la tabla tutorías por asignatura de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de tutorías por asignatura. Se dará clic al botón nuevo tutorías por asignatura. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla tutorías por asignatura y de la clase tutorías por asignatura.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_68	Historia de Usuario: HU_12 Registrar los datos de un tutorías por asignatura
Nombre: El dialogo se cierra con la tecla escape.	
Responsable: Bautista Fausto y Román José	
Descripción: Al presionar la tecla Esc, el diálogo crear nuevo tutorías por asignatura debe cerrarse.	
Condiciones de Ejecución: La tecla Esc esté funcionando correctamente.	
Pasos de ejecución: Presionar la tecla Esc. Visualizar que el diálogo Nuevo Tutorías por asignatura se cierre.	
Resultado esperado: El diálogo se cierre una vez que se presione la tecla Esc	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación

Código: PA_69	Historia de Usuario: HU_12 Registrar los datos de un tutorías por asignatura.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiara el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_70	Historia de Usuario: HU_12 Registrar los datos de un tutorías por asignatura.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic al botón Nuevo Tutorías para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_71 **Historia de Usuario:** HU_12 Registrar los datos de un tutorías

Nombre: Probar la función con datos incorrectos.

Responsable: Bautista Fausto y Román
José

Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.

Pasos de ejecución:

Se utilizará la interfaz para asegurarnos que se tenga un usuario incorrecto a ingresar

Se hará clic al botón Nuevo Tutorías para que se muestre el diálogo

Se hará clic en el botón aceptar del diálogo

Se verificará visualmente que se muestre el mensaje de error correspondiente

Resultado esperado: Se emite un mensaje correspondiente al error producido.

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_72 **Historia de Usuario:** HU_13 Registrar los datos de un identificador de tarjeta magnética del docente

Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos

Responsable: Bautista Fausto y Román
José

Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un nuevo identificador de tarjeta magnética del docente vacíos.

Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase identificador de tarjeta magnética del docente y la tabla identificador de tarjeta magnética del docente de la base de datos.

<p>Pasos de ejecución: Se navegará hasta la interfaz de gestión de identificador de tarjeta magnética del docente .</p> <p>Se dará clic al botón nuevo identificador de tarjeta magnética del docente .</p> <p>Comparar en orden la igualdad entre ambos elementos.</p>
<p>Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla identificador de tarjeta magnética del docente y de la clase identificador de tarjeta magnética del docente .</p>
<p>Evaluación de la prueba: Exitosa.</p>

Prueba de Aceptación	
Código: PA_73	Historia de Usuario: HU_13 Registrar los datos de un identificador de tarjeta magnética del docente
Nombre: El dialogo se cierra con la tecla escape.	
Responsable: Bautista Fausto y Román José	
Descripción: Al presionar la tecla Esc, el diálogo crear nuevo identificador de tarjeta magnética del docente debe cerrarse.	
Condiciones de Ejecución: La tecla Esc esté funcionando correctamente.	
Pasos de ejecución: Presionar la tecla Esc. Visualizar que el diálogo Nuevo Identificador de tarjeta magnética del docente se cierre.	
Resultado esperado: El diálogo se cierre una vez que se presione la tecla Esc	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_74	Historia de Usuario: HU_13 Registrar los datos de un identificador de tarjeta magnética del docente .
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.
Pasos de ejecución: Se copiara el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_75	Historia de Usuario: HU_13 Registrar los datos de un identificador de tarjeta magnética del docente .
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic al botón Nuevo Identificador de tarjeta magnética del docente para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_76	Historia de Usuario: HU_13 Registrar los datos de un identificador de tarjeta magnética del docente .

Nombre: Probar la función con datos incorrectos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario incorrecto a ingresar Se hará clic al botón Nuevo Identificador de tarjeta magnética del docente para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de error correspondiente	
Resultado esperado: Se emite un mensaje correspondiente al error producido.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_77	Historia de Usuario: HU_14 Modificar los datos de un identificador de tarjeta magnética del docente.
Nombre: La tabla se carga al abrir el formulario	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes al identificador de tarjeta magnética del docente s existente.	
Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Identificador de tarjeta magnética del docente s Se verificará visualmente que se muestre la lista de identificador de tarjeta magnética del docente s en una tabla	

Resultado esperado: Un formulario con una tabla llena con la información correspondiente al identificador de tarjeta magnética del docente s existente en la base de datos.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_78	Historia de Usuario: HU_14 Modificar los datos de un identificador de tarjeta magnética del docente.
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes al identificador de tarjeta magnética de los docentes s existentes.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Identificador de tarjeta magnética del docente s Se verificará visualmente que la cantidad de registros de los identificador de tarjeta magnética del docente s	
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_79	Historia de Usuario: HU_14 Modificar los datos de un identificador de tarjeta magnética del docente.
Nombre: Todas las filas tienen el botón para mostrar el dialogo de modificación	
Responsable: Bautista Fausto y Román José	
Descripción: Todas las filas tienen el botón para mostrar el dialogo de edición	

Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el dialogo de edición.
Pasos de ejecución: Se navegará hasta la interfaz de gestión de identificador de tarjeta magnética del docente s Verificar visualmente
Resultado esperado: Cada fila contiene el botón en la columna requerida.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_80	Historia de Usuario: HU_14 Modificar los datos de un identificador de tarjeta magnética del docente.
Nombre: Al dar clic en el botón se muestra un dialogo con los datos del identificador de tarjeta magnética del docente correspondiente	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón editar se debe desplegar un dialogo con los campos correspondientes al identificador de tarjeta magnética del docente seleccionado.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase identificador de tarjeta magnética del docente y la tabla identificadora de tarjeta magnética del docente de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de identificador de tarjeta magnética del docente s Se dará clic al botón editar. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla identificador de tarjeta magnética del docente y de la clase identificador de tarjeta magnética del docente.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación

Código: PA_81	Historia de Usuario: HU_14 Modificar los datos de un identificador de tarjeta magnética del docente.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiará el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido modificado de la tabla	
Resultado esperado: La tabla contiene el registro con los datos modificados	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_82	Historia de Usuario: HU_14 Modificar los datos de un identificador de tarjeta magnética del docente.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un identificador de tarjeta magnética del docente correcto a eliminar. Se hará clic al botón editar para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo	

Se verificará visualmente que la lista tenga el registro con los datos seleccionados
Resultado esperado: La tabla contiene el registro con los nuevos valores asignados
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_83	Historia de Usuario: HU_15 Registrar los datos de un justificaciones de inasistencias
Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un nuevo justificaciones de inasistencias vacíos.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase justificaciones de inasistencias y la tabla justificaciones de inasistencias de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de justificaciones de inasistencias. Se dará clic al botón nuevo justificaciones de inasistencias. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla justificaciones de inasistencias y de la clase justificaciones de inasistencias.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_84	Historia de Usuario: HU_15 Registrar los datos de un justificaciones de inasistencias
Nombre: El dialogo se cierra con la tecla escape.	
Responsable: Bautista Fausto y Román José	
Descripción: Al presionar la tecla Esc, el diálogo crear nuevo justificaciones de inasistencias debe cerrarse.	
Condiciones de Ejecución: La tecla Esc esté funcionando correctamente.	

Pasos de ejecución: Presionar la tecla Esc. Visualizar que el diálogo Nuevo Justificaciones de inasistencias se cierre.
Resultado esperado: El diálogo se cierre una vez que se presione la tecla Esc
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_85	Historia de Usuario: HU_15 Registrar los datos de un justificaciones de inasistencias.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiara el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_86	Historia de Usuario: HU_15 Registrar los datos de un justificaciones de inasistencias.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic al botón Nuevo Justificaciones de inasistencias para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de aceptación
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_87	Historia de Usuario: HU_15 Registrar los datos de un justificaciones de inasistencias.
Nombre: Probar la función con datos incorrectos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario incorrecto a ingresar Se hará clic al botón Nuevo Justificaciones de inasistencias para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de error correspondiente	
Resultado esperado: Se emite un mensaje correspondiente al error producido.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_88	Historia de Usuario: HU_25 Modificar los datos de un justificaciones de inasistencias.

Nombre: La tabla se carga al abrir el formulario	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los justificaciones de inasistencias s existentes.	
Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Justificaciones de inasistencias s Se verificará visualmente que se muestre la lista de justificaciones de inasistencias s en una tabla	
Resultado esperado: Un formulario con una tabla llena con la información correspondiente a los justificaciones de inasistencias s existentes en la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_89	Historia de Usuario: HU_25 Modificar los datos de un justificaciones de inasistencias.
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los justificaciones de inasistencias s existentes.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Justificaciones de inasistencias s Se verificará visualmente que la cantidad de registros de los justificaciones de inasistencias s	
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.	

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_90	Historia de Usuario: HU_25 Modificar los datos de un justificaciones de inasistencias.
----------------------	-----------------------------------------------------------------------------------------------

Nombre: Todas las filas tienen el botón para mostrar el dialogo de modificación

Responsable: Bautista Fausto y Román José

Descripción: Todas las filas tienen el botón para mostrar el dialogo de edición

Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el dialogo de edición.

Pasos de ejecución: Se navegará hasta la interfaz de gestión de justificaciones de inasistencias
s
Verificar visualmente

Resultado esperado: Cada fila contiene el botón en la columna requerida.

Evaluación de la prueba: Exitosa.

Prueba de Aceptación

Código: PA_91	Historia de Usuario: HU_25 Modificar los datos de un justificaciones de inasistencias.
----------------------	-----------------------------------------------------------------------------------------------

Nombre: Al dar clic en el botón se muestra un dialogo con los datos del justificaciones de inasistencias correspondiente

Responsable: Bautista Fausto y Román José

Descripción: Al dar clic en el botón editar se debe desplegar un dialogo con los campos correspondientes al justificaciones de inasistencias seleccionado.

Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase justificaciones de inasistencias y la tabla justificaciones de inasistencias de la base de datos.

<p>Pasos de ejecución: Se navegará hasta la interfaz de gestión de justificaciones de inasistencias</p> <p>s</p> <p>Se dará clic al botón editar.</p> <p>Comparar en orden la igualdad entre ambos elementos.</p>
<p>Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla justificaciones de inasistencias y de la clase justificaciones de inasistencias.</p>
<p>Evaluación de la prueba: Exitosa.</p>

Prueba de Aceptación	
Código: PA_92	Historia de Usuario: HU_25 Modificar los datos de un justificaciones de inasistencias.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
<p>Pasos de ejecución: Se copiara el script a la interfaz pgadmin</p> <p>Se reemplazarán los comodines por información correcta</p> <p>Se ejecutará el script</p> <p>Se revisará visualmente que el campo haya sido modificado de la tabla</p>	
Resultado esperado: La tabla contiene el registro con los datos modificados	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_93	Historia de Usuario: HU_25 Modificar los datos de un justificaciones de inasistencias.
Nombre: Probar la función con datos reales correctos.	

Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga unas justificaciones de inasistencias correctas a eliminar. Se hará clic al botón editar para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que la lista tenga el registro con los datos seleccionados	
Resultado esperado: La tabla contiene el registro con los nuevos valores asignados	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_94	Historia de Usuario: HU_26 Eliminar los datos de un justificaciones de inasistencias.
Nombre: La tabla se carga al abrir el formulario	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a los justificaciones de inasistencias s existentes.	
Condiciones de Ejecución: Ingresar al sistema con el tipo de usuario Administrador	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Justificaciones de inasistencias s Se verificará visualmente que se muestre la lista de justificaciones de inasistencias s en una tabla	
Resultado esperado: Un formulario con una tabla llena con la información correspondiente a las justificaciones de inasistencias existentes en la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_95	Historia de Usuario: HU_26 Eliminar los datos de un justificaciones de inasistencias.
Nombre: Muestra la cantidad de registros que existan en la base de datos	
Responsable: Bautista Fausto y Román José	
Descripción: Se visualiza la tabla con los datos correspondientes a las justificaciones de inasistencias s existentes.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se hará clic al botón Administrar Se hará clic en el botón Justificaciones de inasistencias s Se verificará visualmente que la cantidad de registros de los justificaciones de inasistencias s	
Resultado esperado: Que la cantidad de registros visualizados en la interfaz coincida con la cantidad de registros de la base de datos.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_96	Historia de Usuario: HU_26 Eliminar los datos de un justificaciones de inasistencias.
Nombre: Todas las filas tienen el botón para mostrar el dialogo de eliminación	
Responsable: Bautista Fausto y Román José	
Descripción: Todas las filas tienen el botón para mostrar el dialogo de eliminación	
Condiciones de Ejecución: El responsable revisará visualmente que la tabla contenga una columna a la derecha de cada fila de la tabla que contenga el botón para mostrar el dialogo de eliminación.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de justificaciones de inasistencias s	

Verificar visualmente
Resultado esperado: Cada fila contiene el botón en la columna requerida.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_97	Historia de Usuario: HU_26 Eliminar los datos de un justificaciones de inasistencias.
Nombre: Al dar clic en el botón se muestra un dialogo con los datos del justificaciones de inasistencias correspondiente	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón eliminar se debe desplegar un dialogo con los campos correspondientes al justificaciones de inasistencias seleccionado.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase justificaciones de inasistencias y la tabla justificaciones de inasistencias de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de justificaciones de inasistencias Se dará clic al botón eliminar. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: descripción, fecha y tipo de la tabla justificaciones de inasistencias y de la clase justificaciones de inasistencias.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_98	Historia de Usuario: HU_26 Eliminar los datos de un justificaciones de inasistencias.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	

Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.
Pasos de ejecución: Se copiará el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_99	Historia de Usuario: HU_26 Eliminar los datos de un justificaciones de inasistencias.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic icono de la lupa y se mostrara un diálogo Se hará clic en el botón Eliminar Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación

Código: PA_100	Historia de Usuario: HU_26 Eliminar los datos de un justificaciones de inasistencias.
Nombre: Probar la función con datos incorrectos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Crear método main en la clase Justificaciones de inasistencias Invocar al método fjustificaciones de inasistencias .eliminar enviando como parámetro -1	
Resultado esperado: En la excepción se genere el mensaje correspondiente al error producido.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_101	Historia de Usuario: HU_12 Registrar los datos de un tutorías por asignatura
Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un nuevo tutorías por asignatura vacíos.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase tutorías por asignatura y la tabla tutorías por asignatura de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de tutorías por asignatura. Se dará clic al botón nuevo tutorías por asignatura. Comparar en orden la igualdad entre ambos elementos.	

Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla tutorías por asignatura y de la clase tutorías por asignatura.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_102	Historia de Usuario: HU_12 Registrar los datos de un tutorías por asignatura
Nombre: El dialogo se cierra con la tecla escape.	
Responsable: Bautista Fausto y Román José	
Descripción: Al presionar la tecla Esc, el diálogo crear nuevo tutorías por asignatura debe cerrarse.	
Condiciones de Ejecución: La tecla Esc esté funcionando correctamente.	
Pasos de ejecución: Presionar la tecla Esc. Visualizar que el diálogo Nuevo Tutorías por asignatura se cierre.	
Resultado esperado: El diálogo se cierre una vez que se presione la tecla Esc	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_103	Historia de Usuario: HU_12 Registrar los datos de un tutorías por asignatura.
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiará el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script	

Se revisará visualmente que el campo haya sido ingresado de la tabla
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_104	Historia de Usuario: HU_12 Registrar los datos de un tutorías por asignatura.
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic al botón Nuevo Tutorías para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_105	Historia de Usuario: HU_12 Registrar los datos de un tutorías
Nombre: Probar la función con datos incorrectos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.	

Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario incorrecto a ingresar Se hará clic al botón Nuevo Tutorías para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de error correspondiente
Resultado esperado: Se emite un mensaje correspondiente al error producido.
Evaluación de la prueba: Exitosa.

Prueba de Aceptación	
Código: PA_106	Historia de Usuario: HU_16 Registrar los datos de un identificador de tarjeta magnética del docente
Nombre: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos	
Responsable: Bautista Fausto y Román José	
Descripción: Al dar clic en el botón nuevo se debe desplegar un dialogo con los campos correspondientes a un Nuevo evidencias de clases en caso de ser solicitadasvacíos.	
Condiciones de Ejecución: El responsable revisará visualmente que el diálogo mostrado tenga los mismos campos que la clase identificador de tarjeta magnética del docente y la tabla identificadora de tarjeta magnética del docente de la base de datos.	
Pasos de ejecución: Se navegará hasta la interfaz de gestión de identificador de tarjeta magnética del docente. Se dará clic al botón nuevo identificador de tarjeta magnética del docente. Comparar en orden la igualdad entre ambos elementos.	
Resultado esperado: El diálogo se muestra con los campos: fecha, descripción y tipo de la tabla identificador de tarjeta magnética del docente y de la clase identificador de tarjeta magnética del docente.	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación

Código: PA_107	Historia de Usuario: HU_16 Registrar los datos de un identificador de tarjeta magnética del docente
Nombre: El dialogo se cierra con la tecla escape.	
Responsable: Bautista Fausto y Román José	
Descripción: Al presionar la tecla Esc, el diálogo crear Nuevo evidencias de clases en caso de ser solicitadas debe cerrarse.	
Condiciones de Ejecución: La tecla Esc esté funcionando correctamente.	
Pasos de ejecución: Presionar la tecla Esc. Visualizar que el diálogo Nuevo evidencias de clases en caso de ser solicitadas se cierre.	
Resultado esperado: El diálogo se cierre una vez que se presione la tecla Esc	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_108	Historia de Usuario: HU_16 Registrar los datos de evidencias de clases en caso de ser solicitadas
Nombre: Probar el Script en postgresql con los parámetros esperados.	
Responsable: Bautista Fausto y Román José	
Descripción: Copiar el script a la interfaz de pgadmin y reemplazar los comodines con los datos estimados.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual y el cliente pgadmin con una conexión establecida.	
Pasos de ejecución: Se copiará el script a la interfaz pgadmin Se reemplazarán los comodines por información correcta Se ejecutará el script Se revisará visualmente que el campo haya sido ingresado de la tabla	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación

Código: PA_109	Historia de Usuario: HU_16 Registrar los datos de evidencias de clases en caso de ser solicitadas
Nombre: Probar la función con datos reales correctos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos reales correctos haciendo uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario correcto a ingresar Se hará clic al botón Nuevo evidencias de clases en caso de ser solicitadas para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de aceptación	
Resultado esperado: La tabla contiene el registro con el código elegido para el ingreso	
Evaluación de la prueba: Exitosa.	

Prueba de Aceptación	
Código: PA_110	Historia de Usuario: HU_16 Registrar los datos de evidencias de clases en caso de ser solicitadas
Nombre: Probar la función con datos incorrectos.	
Responsable: Bautista Fausto y Román José	
Descripción: Probar la función con datos incorrectos sin hacer uso de la interfaz correspondiente.	
Condiciones de Ejecución: Tener funcionando el motor de base de datos postgresql con la base de datos en su versión actual.	
Pasos de ejecución: Se utilizará la interfaz para asegurarnos que se tenga un usuario incorrecto a ingresar Se hará clic al botón Nuevo evidencias de clases en caso de ser solicitadas para que se muestre el diálogo Se hará clic en el botón aceptar del diálogo Se verificará visualmente que se muestre el mensaje de error correspondiente	

Resultado esperado: Se emite un mensaje correspondiente al error producido.

Evaluación de la prueba: Exitosa.

3.5.3.1.4. Servicios web consumidos del sistema OASIS

Métodos de Servicios Web consumidos por cada clase relacionada			
Método	Clase que invoca	Tipo de Respuesta	Observación
getPeriodoActual()	FPeriodo	Periodo	Retorna el objeto marcado como activo en el sistema académico.
getAllPeriodos()	FPeriodo	ArrayOfPeriodo	Devuelve todos los periodos existentes en la base de datos del Oasis
getHorariosDocente(java.lang.String strCodCarrera, java.lang.String strCedula, java.lang.String strCodPeriodo)	FMigracion FLoginUsuario	ArrayOfHorarioClase	Devuelve la lista de horarios que tiene el docente en las materias asignadas durante el periodo activo
getDatosUsuarioCarrera(java.lang.String codCarrera, java.lang.String cedula)	FMigracion FLoginUsuario	Persona	Devuelve los datos personales del docente existe en la tabla docente de la base de datos del Oasis

getDictadosMateria(java.lang.String codCarrera, java.lang.String codMateria)	FMigracion FLoginUsuario FPlanificacion FReporteControlador	ArrayOfDictadoMateria	Devuelve la lista de materias asignadas a un docente en un determinado periodo
getMateriasDocente(java.lang.String codCarrera, java.lang.String cedula, java.lang.String codPeriodo)	FMigracion FLoginUsuario FPlanificacion FReporteControlador	ArrayOfMateria	Devuelve las materias con información adicional de las mismas.
getTodasFacultades()	FMigracion	ArrayOfFacultad	Devuelve todas las facultades de la Espoch
getTodasEscuelasTotales()	FMigracion	ArrayOfEscuela	Devuelve todas las escuelas de la Espoch
getTodasCarrerasTotales()	FMigracion	ArrayOfUnidadAcademica	Devuelve todas las carreras de la Espoch
getRolUsuarioCarrera(java.lang.String login)	FMigracion FLoginUsuario	ArrayOfRolCarrera	Devuelve el rol que desempeña un usuario en una determinada carrera
getMallaCurricularPensumVigenteSinDescripcion(java.lang.String strCodCarrera)	FMigracion	ArrayOfMateriaPensum	Devuelve la malla curricular vigente.

getTitulosDocentes(java.lang.String strCodCarrera, java.lang.String strCedula)	FMigracion FLoginUsuario	ArrayOfTitulosDocente	Devuelve los títulos académicos organizados por niveles del docente seleccionado.
getMateriasEstudianteCarrera(Estudiante objEstudiante, java.lang.String strCodCarrera, java.lang.String strCodPeriodo)	LoginUsuario	ArrayOfParaleloMateria	Devuelve las materias que toma un estudiante en el periodo ingresado

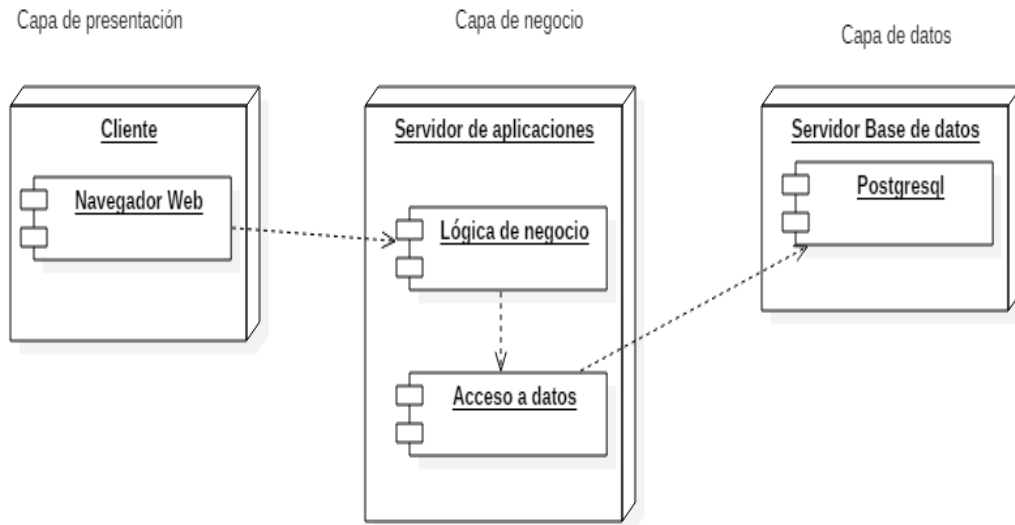
3.5.4. Implementación

3.5.4.1. *Arquitectura del sistema*

El diseño de la arquitectura implementada en el sistema SISPEA es de n capas cuyo objetivo primordial es la separación de la lógica de negocios de la lógica de diseño.

Las capas definidas en el sistema son las siguientes:

- **Capa de presentación:** presenta el sistema al usuario, se encarga de presentar y obtener información del mismo en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio.
- **Capa de negocio:** Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.
- **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

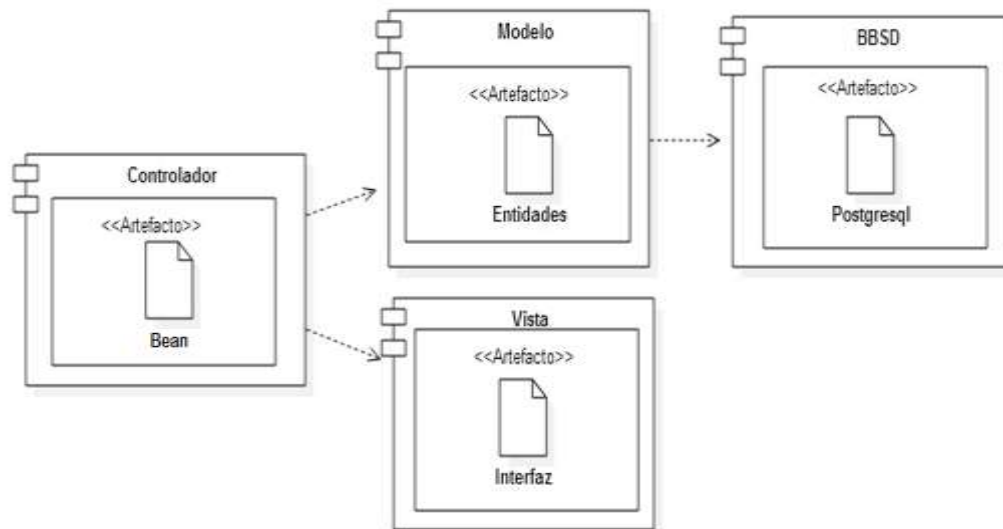


3.5.4.2. Diagrama de Componentes

El sistema web desarrollado está conformado por los siguientes componentes: Modelo, Vista y Controlador. Los controladores son clases Bean estas permiten gestionar todas las acciones que realiza el usuario a través de la vista y las transformara en llamadas a la capa del modelo.

En el modelo tenemos las unidades del sistema.

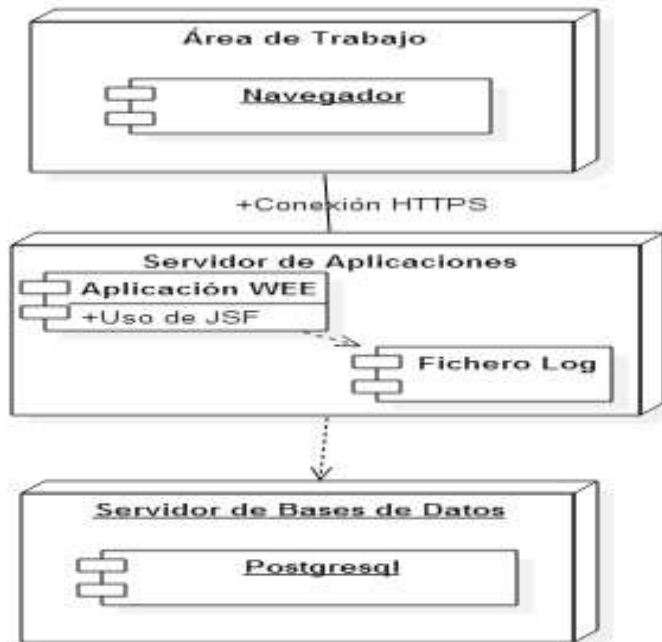
Las vistas estarán conformadas por las paginas JSF, los mensajes propios de la aplicación y los ficheros de configuración.



3.5.4.3. Diagrama de Despliegue

El sistema ha sido desplegado de la siguiente manera:

Para el cliente que lo denominamos area de trabajo se tienen el navegador para la comunicación del usuario con la aplicación. Se para la comunicación entre el servidor de aplicaciones y el area de trabajo se hace uso del protocolo http. Para la base de datos se tiene postgresql.



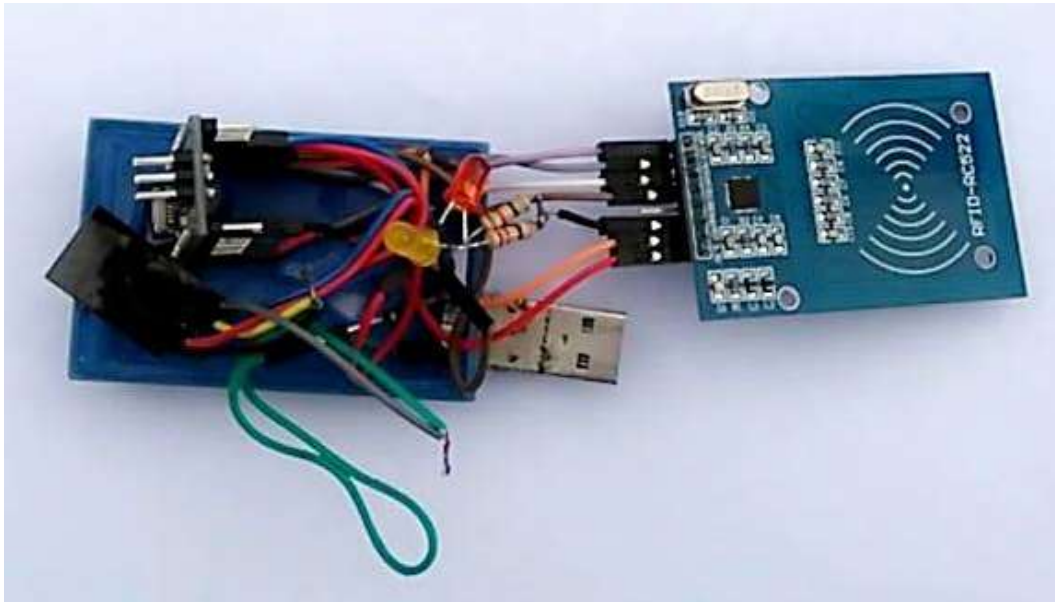
ANEXO E. Sistemas de Control de Acceso Físico Basados en Tarjetas Magnéticas

Sistemas de Control de Acceso Físico Basados en Tarjetas Magnéticas

El sistema de control de asistencia docente es una red coordinada de tarjetas de identificación, lectores electrónicos, bases de datos especializadas, software y computadoras diseñadas para monitorear y controlar el tráfico a través de puntos de acceso.

Generalmente, la tarjeta muestra una foto de su portador. Cada tarjeta almacena información protegida sobre la persona y sobre los privilegios de esta persona.

Sistemas de Control de Acceso Físico Basados en Tarjetas Magnéticas



Tanto las empresas privadas como las agencias de gobierno están implementando cada vez más los sistemas de control de acceso basados en tarjetas inteligentes

- La adopción de sistemas de control de acceso basados en tarjetas inteligentes puede también resultar en otras ventajas para la organización, incluyendo:
- Eliminación de la necesidad de reemplazar las tarjetas cuando los derechos o privilegios cambian.
- Administración centralizada, permitiendo a la organización mantener o aumentar la seguridad mientras se ahorra tiempo, logrando una distribución más completa de la información, manejando cambios globales para privilegios de acceso a partir de un único

punto y reduciendo las complejidades involucradas en la sincronización de sistemas múltiples.

- Flexibilidad para respaldar múltiples funciones dentro de la organización, (por ejemplo seguridad de los establecimientos e Informática) para manejar y controlar aplicaciones separadas en una única tarjeta de identidad inteligente de aplicación múltiple.

Radio de Acción Operacional

Una característica importante de la operación del sistema del control acceso es la distancia del lector en la cual la credencial es efectiva (llamado radio de acción operacional). Esta característica puede afectar la percepción final del usuario sobre la conveniencia de utilizar el sistema. Para los sistemas que utilizan las tarjetas inteligentes de contacto, el radio de acción operacional no es un problema, ya que la tarjeta cuenta con un radio de 60mm o 6cm.

Protección de datos

Uno de los argumentos más fuertes para el uso sistemas basados en tarjeta inteligentes para control de acceso físico es su capacidad de usar mecanismos para mezclar datos o criptografiar para proteger la información tanto en el chip como durante la transmisión. La seguridad y confiabilidad de la información requerida para la identificación de una persona y sus derechos y privilegios es clave para el éxito del sistema de control de acceso físico.

METODOLOGÍA

La metodología utilizada para esta investigación es la deductiva ya que estamos analizando datos tomados por los docentes que necesitan o requieren un dispositivo que les ayude con el control de asistencia docente.

Componentes del Sistema de Control Acceso

El generador produce la corriente eléctrica. Una pila, es un generador. El generador tiene dos polos o bornes. Por uno sale las cargas y por otro entra. De este modo se crea la corriente. En este caso se utilizó una corriente de 5v.

Los cables, conduce la corriente eléctrica desde el generador hasta los demás elementos del circuito. El cable utilizado es UTP categoría 6.

Nombre	Especificación Técnica	
Arduino Nano	<ul style="list-style-type: none"> • RESET automático al descargar el programa • LED en la base para indicar el encendido • LED (TX), (RX) y (L) • Regulador de tensión integrado • Conector mini-B USB para programación y comunicación serie integrado en placa • Conector ICSP para programación • Pines con espaciado de 0.1" DIP para insertarlo directamente sobre una protoboard • Botón de reset integrado • Bootloader integrado que permite programarlo con el entorno de desarrollo Arduino sin necesidad de un programador externo 	
Módulo ESP8266	Wifi	<ul style="list-style-type: none"> • Protocolos soportados: 802.11 b/g/n. • Wi-Fi Direct (P2p), Soft Access Point. • Stack TCP/IP integrado. • PLL, reguladores y unidades de manejo de energía integrados. • Potencia de salida: +19.5dBm en modo 802.11b. • Sensor de temperatura integrado. • Consumo en modo de baja energía: <10 uA.
Módulo RC522 RF	RFID-	<ul style="list-style-type: none"> • Modelo: MF522-ED. • Corriente de operación: 13-26mA a 3.3V. • Corriente de stand by: 10-13mA a 3.3V. • Corriente de sleep-mode: <80uA. • Corriente máxima: 30mA. • Frecuencia de operación: 13.56Mhz. • Distancia de lectura: 0 a 60mm. • Protocolo de comunicación: SPI.
2 Led	mcd	<ul style="list-style-type: none"> • Color: Amarillo * Luminosidad: 8 * Consumo: 10 mA * Diámetro: 3 y 5
	mm	<ul style="list-style-type: none"> • Color: Rojo * Luminosidad: 1.25 mcd * Longitud onda: 660 nm * Consumo: 10 mA * Diámetro: 3 y 5 mm
Cable USB a Mini USB Para Arduino Nano	<ul style="list-style-type: none"> • Conectores macho tipo "A" a macho mini USB de 5 pines. 	
2 Resistencias	<ul style="list-style-type: none"> • Código de colores: Rojo Rojo Marrón Naranja * Banda 1: 2 * Banda 2 onda: 2 * Multiplicador: x 10 ohms 	

		* Tolerancia: 5%
2	Diodos	<ul style="list-style-type: none"> • Polarización directa • Tensión de ruptura para el diodo es de 50 V
Rectificadores		
Multicargador	3	<ul style="list-style-type: none"> • 3 Entradas USB. • Carga rápida y segura. • 3.1 mAh
entradas		

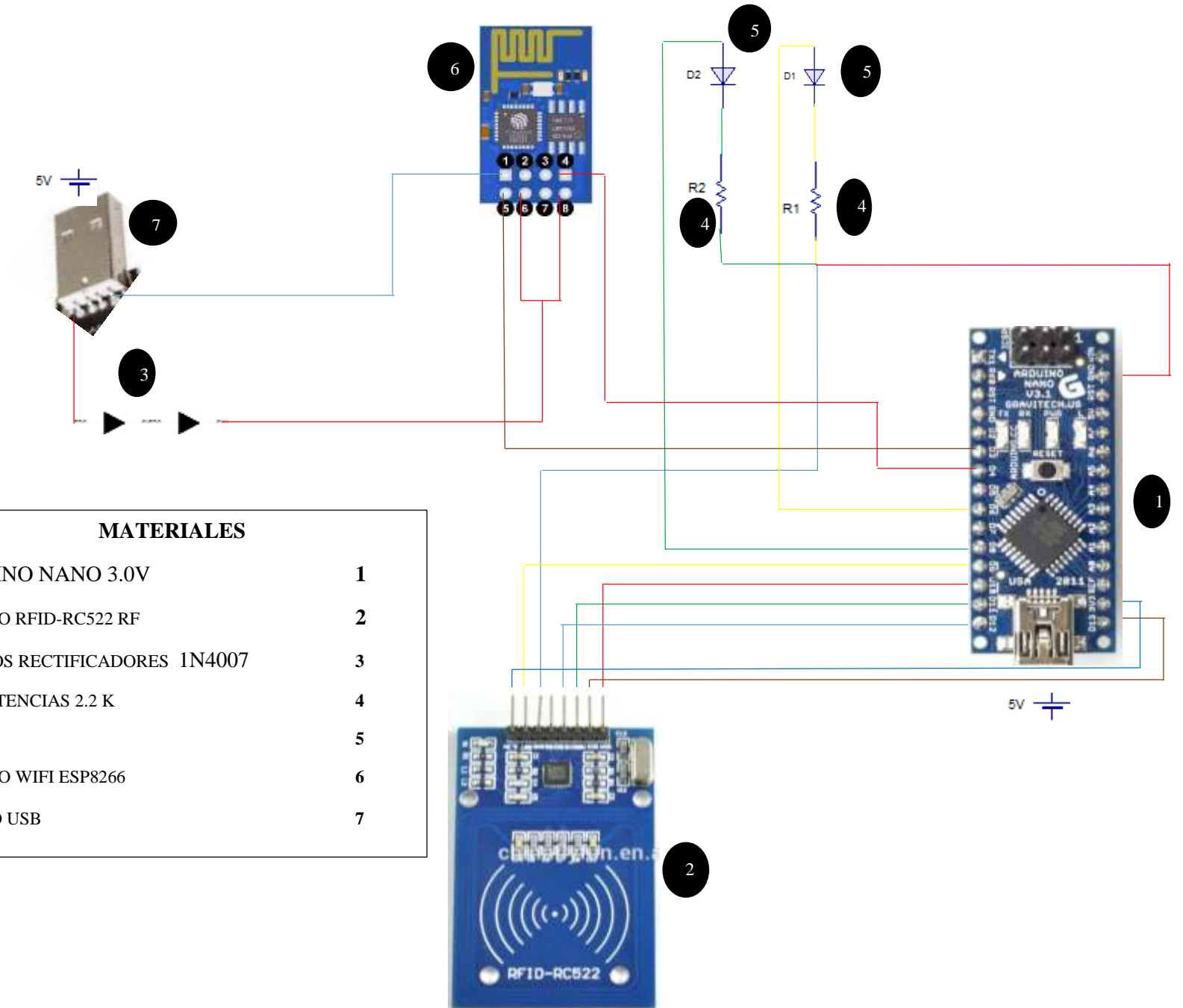
Costo desarrollo de lector de tarjeta magnética

El costo para el desarrollo del lector de tarjetas magnéticas con un total de **54,30** dólares cabe mencionar que este costo es por una unidad, a este detector de tarjetas magnéticas se le pueden asignar las tarjetas magnéticas necesarias para su uso. También mencionamos los posibles proveedores de materia prima con sus direcciones.

Costo desarrollo de lector de tarjeta magnética		
Arduino Nano		17,00
Módulo Wifi ESP8266		15,00
Módulo RFID-RC522 RF		15,00
2 Led (Amarillo y Rojo)		0,20
2 Resistencias(Color: Rojo Rojo Marrón Naranja)		0,20
2 Diodos Rectificadores		1,40
1 Tarjeta magnética con chip inteligente		1,50
Multi cargador 3 entradas USB 3.1 mAh Fast Charger		4,00
Costo de Instalación		00,00
La Escuela de Ingeniería en Sistemas cuenta con todos los recursos necesarios para implantar el Sistema.		00,00
Costo de Operación		00,00
Costos de personal de operación		00,00
Costo de Mantenimiento		00,00
Costo de suministros de operación		00,00
TOTAL		54,30

Información de proveedores		
Descripción	Dirección	Teléfono
MCA ELECTRÓNICA	Matriz: Veloz 24-18 y Larrea	2963-384
ELECTRO SISTEM	Matriz: José Joaquín de Olmedo 17-28 y entre Benalcázar y Alvarado	2969-284 0984484653

DIAGRAMA O PLANO DEL CIRCUITO



MATERIALES

ARDUINO NANO 3.0V	1
MÓDULO RFID-RC522 RF	2
2 DIODOS RECTIFICADORES 1N4007	3
2 RESISTENCIAS 2.2 K	4
2 LED	5
MÓDULO WIFI ESP8266	6
PUERTO USB	7

Funcionamiento del lector de tarjetas magnéticas

El terminal se encuentra preparado para dar paso a la autorización o desautorización de las tarjetas de cada uno de los usuarios que va a leer el lector, teniendo en cuenta que primero se debe conectar el puerto USB y luego el arduino nano a la corriente de 5v

Arduino Nano

Dado que el Arduino es como un pequeño ordenador que ejecuta una serie de códigos que previamente le hemos introducido, necesitaremos un programa para poder meter estos códigos a la propia placa. Este programa se llama IDE, que significa "Integrated Development Environment" ("Entorno de Desarrollo Integrado"). Se utilizó el software libre Arduino 1.8.1 para la programación del nano Arduino.

Código del Arduino Nano

```
#include <SoftwareSerial.h>
#include <SPI.h>
#include <WiFi.h>
#include <MFRC522.h>

int ROJO = 6;
int AMARILLO = 8;
int PULSADOR = 5;
int band = 0;

#define RST_PIN 9 //Pin 9 para el reset del RC522
#define SS_PIN 10 //Pin 10 para el SS (SDA) del RC522
//AT+CWMODE=1 MODO CLIENTE
//AT+CIPSTA="172.25.210.158","172.25.210.1","255.255.254.0" ESTABLECE LA IP
//AT+CIFSR MUESTRA LA IP
MFRC522 mfrc522(SS_PIN, RST_PIN); //Creamos el objeto para el RC522

SoftwareSerial softSerial(3, 4); // RX, TX
String red = "ESPOCH-FIE-EIS"; // CREDENCIALES DE RED ESPOCH-OPEN (name)
String pass = "ESPOCH-EIS-2015";

void setup()
{
  pinMode(ROJO, OUTPUT);
  pinMode(AMARILLO, OUTPUT);
  uint32_t baud = 115200;
  Serial.begin(baud);
  softSerial.begin(baud);
  digitalWrite(ROJO, HIGH);
  softSerial.println("AT+CIOBAUD=9600");
```

```

delay(1000);
baud = 9600;
Serial.begin(baud);
softSerial.begin(baud);
SPI.begin(); //Iniciamos el Bus SPI
mfrc522.PCD_Init(); // Iniciamos el MFRC522
}

void loop()
{
  if (band == 0) {
    softSerial.println("AT");
    digitalWrite(ROJO, LOW);
    delay(1000);
    digitalWrite(ROJO, HIGH);
    softSerial.println("AT+CWMODE=3");
    digitalWrite(ROJO, LOW);
    delay(1000);
    digitalWrite(ROJO, HIGH);
    softSerial.println("AT+CWQAP");
    digitalWrite(ROJO, LOW);
    delay(3000);
    digitalWrite(ROJO, HIGH);
    softSerial.println("AT+CWJAP=\"" + red + "\",\"" + pass + "\"");
    digitalWrite(ROJO, LOW);
    delay(5000);
    digitalWrite(ROJO, HIGH);
    softSerial.println("AT+CIFSR");
    band = 1;
  }

  while (softSerial.available() > 0)
  {
    char a = softSerial.read();
    if (a == '\0')
      continue;
    if (a != '\r' && a != '\n' && (a < 32))
      continue;
    Serial.print(a);
  }

  while (Serial.available() > 0)
  {
    char a = Serial.read();
    Serial.write(a);
    softSerial.write(a);
  }

  if ( mfrc522.PICC_IsNewCardPresent()) {
    //Seleccionamos una tarjeta
    if ( mfrc522.PICC_ReadCardSerial()) {

```

```

char readCard[mfrc522.uid.size] ;
for (byte i = 0; i < mfrc522.uid.size; i++) {
  readCard[i] = mfrc522.uid.uidByte[i];
}
String respuesta = "";
for (int i = 0; i < mfrc522.uid.size; i++) {
  respuesta = respuesta + String(readCard[i], HEX);
}
Serial.println(respuesta);
comunicar(respuesta);
digitalWrite(AMARILLO, HIGH);
delay(500);
digitalWrite(AMARILLO, LOW);
// Terminamos la lectura de la tarjeta actual
mfrc522.PICC_HaltA();
}
}
}

void comunicar(String uuid) {
  digitalWrite(AMARILLO, HIGH);
  String cmd = "AT+CIPSTART=\"TCP\",172.25.224.20,8080"; //del servidor
  softSerial.println(cmd);

  if (softSerial.find("Error"))
  {
    Serial.println("AT+CIPSTART error");
    return;
  }
  String hostt = "GET /index.xhtml?test=" + uuid + " HTTP/1.1";
  hostt += "\r\n";
  hostt += "Host: 172.25.224.20"; //del servidor
  hostt += "\r\n\r\n";
  softSerial.println(hostt);

  cmd = "AT+CIPSEND=";
  cmd += String(hostt.length());
  softSerial.println(cmd);

  if (softSerial.find(">")) {
    softSerial.print(hostt);
  }
  else {
    softSerial.println("AT+CIPCLOSE");
    // alert user
    Serial.println("AT+CIPCLOSE");
  }
  digitalWrite(AMARILLO, LOW);
}
}

```

Fuente: <https://hetpro-store.com/TUTORIALES/modulo-lector-rfid-rc522-rf-con-arduino/>