

Universidad de las Ciencias Informáticas

Facultad 4



**SISTEMA DE APOYO A LA GESTIÓN DE LA INFORMACIÓN
ACADÉMICA**

**Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
Informáticas**

Autor(es): Ernesto Mastrapa Serrano

Tutor(es): Ing. Yusnier Reyes Dixson

La Habana, Cuba

Junio 2014

Declaración de autoría

Por este medio, declaro que soy el autor oficial del presente Trabajo de Diploma, y otorgo a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales del mismo, en virtud de que pueda ser utilizado según los intereses y necesidades de la institución.

Para que así conste, firmo la presente a los _____ días del mes de _____ del 20_____

Ernesto Mastrapa Serrano

Ing. Yusnier Reyes Dixson

Firma del autor

Firma del tutor

Agradezco...

A mis padres Oneida y German por darme la vida y porque son los que más me han apoyado en estos 5 años de arduo trabajo, por sus consejos, perseverancia y por creer en mí en todo momento.

A mi abuela Margarita que más que mi abuela es como mi madre también, por criarme y sembrar valores en mí que me ayudaron a estar donde estoy hoy.

A mi abuelo Jorge donde quiera que este por educarme desde niño y regañarme cuando tenía que hacerlo, nunca lo olvidare.

A mi prima Aliet por ser la hermana menor que nunca tuve, por su cariño y apego a mi desde pequeña.

A mi tío Alberto por su apoyo y preocupación todos estos años.

A mi madrastra Yudit que desde muy pequeño me acogió como su hijo, por su cariño y preocupación no solo estos 5 años, sino de toda la vida.

A mi abuelo German y a mi abuela Efigenia donde quiera que esté por su cariño y preocupación por mí y por mi familia siempre, por alegrarme cada verano y cada fin de año.

A mi novia Yania primero por su amistad hace 5 años y luego por su cariño y apoyo durante esta difícil etapa, a su familia por acogerme como uno más de ellos.

A mis amistades dentro y fuera de la escuela.

A mi tutor Dixon que sin él este trabajo no hubiera sido posible.

A los profesores que de una forma u otra me ayudaron.

A mis antiguos compañeros de apartamento que durante 4 años me brindaron su amistad e hicieron que mi estancia aquí, lejos de mi lugar de origen fuera mejor.

Dedico este trabajo a mis padres y a mi abuela Margarita espero que se sientan orgullosos de mí, ellos son la razón por la cual estoy aquí y son lo más grande que tengo en la vida...

Resumen

En la Universidad de las Ciencias Informáticas se llevan a cabo procesos que a menudo requieren el uso de mecanismos para el manejo y gestión de la información. Uno de sus componentes es el proceso de enseñanza-aprendizaje, el cual es necesario para la formación de profesionales con una vasta preparación para la vida laboral. Actualmente existe un sistema, el cual dedica uno de sus módulos a la gestión de la información académica de los estudiantes, este no contempla evaluaciones frecuentes que el profesor registra, tampoco refleja las habilidades, ni los temas a evaluar en las actividades, otro elemento es que no permite representar gráficamente reportes necesarios que permitan visualizar el estado de los grupos y de los estudiantes, basados en los datos registrados. Es por eso que se realiza un sistema para apoyar la gestión de la información académica. Durante la investigación se realizó un estudio del estado del arte referente a los sistemas automatizados vinculados al objeto de estudio. Para el desarrollo del sistema se utilizó como Sistema de Gestor de Base de Datos: PostgreSQL, se seleccionó como servidor web Apache y como framework de desarrollo Symfony. Todo el proceso de desarrollo del sistema tuvo como guía la metodología XP, generando los artefactos necesarios. Como resultado se obtuvo un sistema que apoya la gestión de la información docente enfocado a la información que necesita el profesor a diario para desempeñar con mayor calidad su labor docente.

Palabras Clave: docente, enseñanza-aprendizaje, gestión de la información.

ÍNDICE

Introducción:	1
Capítulo 1: Fundamentación Teórica	6
1.1 Introducción.....	6
1.2 Conceptos asociados al objeto de estudio.....	6
Evolución de los Sistemas de Información	6
Clasificación de los Sistemas de información.....	8
El diseño y desarrollo de sistemas de información.....	9
Gestión de la Información	10
Sistemas de Gestión de Información	11
Gestión Académica.....	12
Sistemas de Gestión Académica	12
1.3 Soluciones similares.....	13
Ámbito Internacional:	13
Ámbito nacional:	15
1.4 Lenguajes, tecnologías y herramientas de desarrollo de software.....	17
Metodologías	17
Lenguaje de programación	26
Estándares de servicios web	28
Entorno de desarrollo integrado.....	29
Herramienta CASE	30
Marco de Trabajo.....	31
Servidor web.....	32
Sistema gestor de bases de datos (SGBD).....	33
1.5 Conclusiones.....	34
Capítulo 2 Exploración y Planificación	35
2.1 Introducción:.....	35
2.2 Propuesta de solución	35
2.3 Definición de la audiencia.....	36

2.4 Personas relacionadas con el sistema.....	36
2.5 Historia de usuario.....	37
2.6 Estimación en esfuerzos por HU	40
2.7 Plan de iteraciones	42
2.8 Plan de duración de las iteraciones	43
2.9 Plan de entregas	44
2.10 Diseño de la base de datos	46
2.11 Conclusiones.....	47
Capítulo 3 Implementación y pruebas del sistema	49
3.1 Introducción.....	49
3.2 Estándares de codificación.....	49
3.3 Tarjetas CRC.....	50
3.4 Implementación	53
Iteración 1	53
Iteración 2.....	54
Iteración 3.....	56
Iteración 4.....	57
3.5 Diagrama de despliegue.....	57
3.6 Patrón arquitectónico.....	58
3.7 Patrones de diseño.....	60
3.8 Pruebas.....	61
Pruebas unitarias	61
Pruebas de aceptación	63
3.9 Conclusiones.....	70
Conclusiones generales.....	71
Recomendaciones	72
Referencias Bibliográficas.....	73
Anexo I	77
Anexo II	83

Anexo III	92
-----------------	----

Introducción:

Los Sistemas de Información (SI) son una de las muchas herramientas que nos brindan las Tecnologías de la Información y la Comunicación y no son pocas las entidades, ya sean privadas o estatales, que hoy en día requieren el uso de mecanismos rápidos y eficientes para el manejo de la información. Se le llama SI a un conjunto de componentes interrelacionados que incluyen hardware, software y orgware¹ que capturan, almacenan, procesan y distribuyen la información para apoyar la toma de decisiones, el control, análisis y visión en una institución (1). Su gran impacto hoy en día ha cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos y suministran una plataforma de información necesaria para la toma de decisiones.

En Cuba, hace algunos años a causa de la creciente corriente tecnológica se ha optado también por el uso de sistemas de información, que gestionen los principales procesos que se llevan a cabo en la mayoría de las instituciones.

La Gestión de la Información (GI) deviene como uno de los elementos más destacados dentro de los SI, el investigador Woodman plantea que: *“la gestión de información es todo lo que tiene que ver con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado para tomar la acción precisa”* (2).

Las universidades constituyen organizaciones de importante responsabilidad social, en ellas se genera y trasmite gran parte del conocimiento que apoya el desarrollo económico de cualquier sociedad. La trascendencia del encargo social de las universidades y el alto costo de la enseñanza en las mismas sobre todo las relacionadas con las ramas tecnológicas, demandan eficiencia, eficacia y calidad en los procesos que en estas se desarrollan. Para conseguir este propósito, la gestión de los procesos debe ser efectiva, basada ampliamente en el uso de las tecnologías y con métodos sujetos a constante perfeccionamiento. Con el objetivo de apoyar y mejorar la gestión, algunos investigadores proponen que las instituciones de educación superior utilicen de forma organizada la información y el conocimiento que se acumula sobre sus procesos (3). Es con este fin que la gestión de la información y el conocimiento

¹ orgware: Componente estructural de un sistema tecnológico especialmente concebido para integrar al hombre y sus competencias profesionales y asegurar el funcionamiento del hardware y software del sistema así como la interacción de este con otros elementos y con otros sistemas de naturaleza diferente.

traen consigo herramientas importantes en la dirección de las universidades. El proceso de formación, es fundamental en una universidad (4) y es complejo debido a la gran variabilidad en las características de los estudiantes que son el objeto a transformar, y a la variedad de condiciones que confluyen en dicho proceso para desarrollar esta transformación (5). Por lo que la gestión de la información debe ser un elemento importante para asegurar que la información que se obtiene pueda ser utilizada como soporte a las acciones y estrategias que el profesor y los directivos desarrollan continuamente.

En muchas de estas instituciones se hace complejo el control de la información de su principal componente: el proceso de enseñanza-aprendizaje. No obstante, a menudo este se ve influido por la información disponible debido entre otros elementos a la forma en que se gestiona la información académica de los estudiantes, de modo que en diversas ocasiones los profesores no toman una correcta decisión en cuanto a sus actividades docentes, por carecer de datos necesarios o porque aun contando con ellos, no son útiles por estar demasiado desordenados o difíciles de asimilar, o simplemente porque no se dispone de tiempo suficiente para procesarlos e interpretarlos.

En la Universidad de las Ciencias Informáticas (UCI) la cual tiene como principal misión la de formar profesionales de la informática con un alto nivel académico y comprometidos con el proceso revolucionario del país, la gestión de la información académica se realiza principalmente a través de los registros de asistencia y evaluación en copia dura o digital que mantienen los profesores. Además, se cuenta actualmente con el Sistema de Gestión Universitaria que entre sus módulos cuenta con uno dedicado a la gestión de la información académica de los estudiantes, a pesar de que sirve de gran ayuda a este proceso, desde el punto de vista del profesor no realiza una gestión profunda de dicha información. Esto se ve reflejado en que el sistema no contempla algunos de los elementos que el profesor registra sobre sus estudiantes, entre estos se pueden mencionar: las evaluaciones frecuentes (preguntas escritas, seminarios, talleres, entre otros) sobre estas evaluaciones no permite reflejar elementos adicionales como son las habilidades y temas a evaluar en dichas actividades. Otro elemento a tener en cuenta es que dicho sistema no es capaz de representarle mediante elementos gráficos los reportes que puede necesitar el profesor, dígame reportes de asistencias y evaluaciones, por lo que la gestión realizada por dicho sistema es incompleta y no se encuentra enfocada a las necesidades primarias de información que necesita el profesor a diario para desempeñar con mayor calidad su labor docente.

Teniendo en cuenta la **situación problemática** descrita, se evidencia la necesidad de mejorar la gestión que se realiza de la información académica de los estudiantes, por lo que se propone como **problema a resolver**: ¿Cómo contribuir al proceso de gestión de información académica en la labor docente del profesor?

A partir de lo planteado anteriormente, la parte de la ciencia que será **objeto de estudio** es el proceso de gestión de la información. Especificándose como **campo de acción** los sistemas de gestión de la información académica.

Se plantea como **idea a defender**: el desarrollo de un sistema de apoyo a la gestión de la información académica contribuye al desarrollo de la labor docente del profesor.

Para dar solución al problema de investigación se propone como **objetivo general** de la investigación: Desarrollar un sistema de apoyo a la gestión de la información académica para contribuir en el desarrollo de la labor docente del profesor, siendo los **objetivos específicos** los siguientes:

- ✓ Efectuar una exploración bibliográfica de los principales conceptos relacionados con el objeto de estudio para comprender mejor la evolución y las tendencias actuales del mismo.
- ✓ Realizar el análisis y diseño de las principales funcionalidades del sistema propuesto para satisfacer las necesidades del cliente.
- ✓ Implementar las principales funcionalidades del sistema de apoyo a la gestión de la información académica.
- ✓ Realizar las pruebas necesarias al sistema informático propuesto con el objetivo de verificar su correcto funcionamiento.

Para darle cumplimiento a los objetivos específicos se realizarán las siguientes **tareas de investigación**:

- ✓ Análisis bibliográfico sobre los sistemas de gestión de la información.
- ✓ Estudio y aplicación de los conceptos sobre los sistemas de gestión de la información.
- ✓ Análisis de las soluciones similares para obtener variantes de desarrollo y características útiles a incorporar.

- ✓ Selección de tecnologías y herramientas que permitan el desarrollo del sistema propuesto.
- ✓ Definición de las funcionalidades del software.
- ✓ Descripción de la propuesta de solución y actores relacionados con el sistema.
- ✓ Implementación de las funcionalidades del software.
- ✓ Realización de las pruebas pertinentes para demostrar que las funciones del software son operativas.

Entre los métodos de investigación utilizados se destacan:

Métodos Teóricos

Analítico-sintético: Durante la investigación se analizan documentos, páginas de Internet y otros artículos de donde se extraen los elementos más importantes relacionados con sistemas de gestión de la información. Además, se realizan resúmenes y valoraciones de las características más relevantes de estos sistemas.

Análisis histórico-lógico: Permite desarrollar el estudio del arte previo al desarrollo de la investigación. Servirá para definir las tendencias actuales en el desarrollo de sistemas de gestión de la información, así como para determinar las características principales de estos.

Inductivo-Deductivo: Es aquel que permite partir de datos generales aceptados como válidos, llegar a una conclusión de tipo particular y viceversa. En la investigación se toman las características de los sistemas de gestión estudiados aplicables al objeto de estudio.

Métodos Empíricos

Entrevistas abiertas: Se utilizó la variante no formal, para conocer el funcionamiento de los métodos y herramientas que utilizan los profesores en la universidad en el manejo de las evaluaciones diarias de los estudiantes así como para el análisis del estado del arte a nivel nacional.

Estructura Capítular

El presente trabajo, está estructurado en 3 capítulos, distribuidos de la siguiente forma:

Capítulo 1. Fundamentación Teórica: Se abordan conceptos importantes sobre los sistemas de gestión de la información, se presenta un estado del arte de las funcionalidades de los sistemas asociados a estos, en Cuba y el mundo. Se realiza un estudio de los principales lenguajes, tecnologías y herramientas para el desarrollo del sistema.

Capítulo 2. Exploración y Planificación: En este capítulo se describen las fases de Planificación y Exploración definidas en el ciclo de vida de la metodología de desarrollo XP, a través de los principios, prácticas y técnicas que se proponen para estas fases.

Capítulo 3. Implementación y prueba del sistema: Se realiza el análisis y diseño del sistema, elaborando los artefactos para esta fase, como son las tarjetas CRC y las tareas de ingeniería. Además, se realiza la implementación del mismo, definiendo el modelo de despliegue, estándares y patrones así como las pruebas pertinentes para comprobar la factibilidad de la aplicación.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se presentan elementos teóricos que respaldan la investigación, se realiza un análisis de los sistemas homólogos en Cuba y en el mundo, así como un estudio de los aspectos más importantes de los lenguajes, tecnologías y herramientas para la solución del problema planteado.

1.2 Conceptos asociados al objeto de estudio

Son muchos los autores que han definido un SI en la actualidad, y es que el término resulta polisémico en grado sumo, ya que se utiliza para designar a entidades muy diversas. En su generalidad no es más que un conjunto de procedimientos organizados que, cuando se ejecutan, proporcionan información para la toma de decisiones y el control de la organización (6). Otra propuesta que algunos la consideran una de las más precisas es la que define un SI como: *“conjunto formal de procesos que, operando sobre una colección de datos estructurada de acuerdo con las necesidades de la empresa, recopila, elabora y distribuyen selectivamente la información necesaria para la operación de dicha empresa y para las actividades de dirección y control correspondientes, apoyando, al menos en parte, los procesos de toma de decisiones necesarios para desempeñar funciones de negocio de la empresa de acuerdo con su estrategia”* (7).

La capacidad para el acceso inmediato a los datos comprensivos es crítica para el proceso de toma de decisiones y requiere la recopilación, almacenamiento y análisis de grandes cantidades de información. Las instituciones hoy en día ven más hacia el futuro planteándose nuevas estrategias de aprovechamiento de los recursos humanos e informáticos, con una adaptación casi inmediata a los requerimientos que imponen las nuevas tecnologías de una manera más flexible, considerando que la automatización de dichas partes ha sido un principio indispensable para su desenvolvimiento y desarrollo eficiente (1).

Evolución de los Sistemas de Información

Los Sistemas de Información han ido evolucionando durante los últimos años hasta constituir los denominados sistemas de información estratégicos. Primeramente los Sistemas de Información empresariales eran considerados como un instrumento simplificador de las distintas actividades de la

Capítulo 1: Fundamentación Teórica

empresa, una herramienta con la cual se facilitaban los trámites y reducía la burocracia. Su finalidad era básicamente llevar la contabilidad y el procesamiento de los documentos a nivel operativo.

Posteriormente el desarrollo de la informática y las telecomunicaciones permitieron incrementar la eficacia en la realización de las tareas, ahorrar tiempo en el desarrollo de las actividades y almacenar la mayor cantidad de información en el menor espacio posible, lo cual aumentó en las organizaciones el interés en los sistemas de información. Con el transcurrir del tiempo las empresas fueron observando, cómo las tecnologías y sistemas de información permitían a la empresa obtener mejores resultados que sus competidores, constituyéndose por sí mismas como una fuente de ventaja competitiva y una poderosa arma que permitía diferenciarse de sus competidores y obtener mejores resultados que estos. De este modo los sistemas de información se constituyeron como una de las cuestiones estratégicas de la empresa, que ha de considerarse siempre en todo proceso de planificación empresarial (8).

La clasificación de la evolución de los sistemas de información se agrupa en cuatro grandes etapas (7):

- ✓ **Introducción de la informática en la organización:** los sistemas de información se aplicaban para simplificar y automatizar los procesos administrativos. Se usan las computadoras y los sistemas informáticos para mejorar el proceso de contabilidad, elaborar nóminas y facturación buscando sobre todo el ahorro de costes y tiempo en la realización de dichas operaciones. Existe una carencia de formación por parte de los empleados de la organización en dichos sistemas y no hay profesionales que puedan resolver dichos problemas dentro de la compañía.
- ✓ **Etapa de contagio de las aplicaciones informáticas:** tras observar como la aplicación de los sistemas informáticos en algunas áreas de la empresa originan importantes mejoras, estos se van difundiendo por los diferentes departamentos de la empresa. Dicho contagio se desarrolla sin ninguna planificación por parte de la organización, con lo cual se produce un alto incremento de los costes. Aumenta la formación del personal en las tecnologías de información y en las aplicaciones informáticas, existiendo ya en la organización personal capaz de solucionar los problemas planteados en el manejo del sistema de información.
- ✓ **Coordinación de los Sistemas de Información y los objetivos de la empresa:** los sistemas de información son utilizados en la totalidad de la organización y ya son tenidos en cuenta por parte de la dirección como un elemento fundamental de la empresa. Se empiezan a elaborar

Capítulo 1: Fundamentación Teórica

procedimientos de planificación de los sistemas de información y aparece la necesidad de usar los sistemas de información como un medio de cumplimiento de los objetivos de la empresa.

- ✓ **Aparición de los Sistemas Estratégicos de información:** los sistemas de información son valorados como una fuente de ventaja competitiva sostenible, de tal modo que al elaborar la estrategia general de la compañía se establece la planificación y desarrollo de los sistemas de información como otros de los aspectos claves dentro del proceso directivo.

Clasificación de los Sistemas de información

Indiscutiblemente, los SI son necesarios en la actualidad al igual que su evaluación, por eso es preciso indicar que existen muchos tipos con diferente valor para cada usuario final, para el que fue diseñado, a continuación se muestra un resumen general de los existentes en la actualidad, así como sus características más representativas (1).

Tabla 1. Clasificación de los sistemas de información

Tipo	Entrada de información	Procesamiento	Salidas de información	Usuarios
TPS ²	Transacciones de rutina diaria	Intenso, secuencial	Informes con mucho detalles, grandes volúmenes	Empleados operativos
DSS ³	Grandes cantidades	Interactivo, flexible	Información para la toma de decisiones, diferentes alternativas	Profesionistas, administradores generales
ESS ⁴	Interna, externa (incertidumbre)	Gráficas avanzadas, resúmenes, simulaciones	Proyecciones, gráficas	Alta administración
KWS ⁵	Conocimiento de expertos	Simulaciones	Gráfico, modelos	Profesionistas
(Integrales) ERP ⁶	No estructurada, interna, externa, grandes cantidades	Intenso, interactivo, secuencial, flexible	Reportes con muchos detalles, ejecutivos, gráficas	Todos pueden ser usuarios

² TPS: Transaction Processing System (Sistemas de Procesamiento Transaccional)

³ DSS: Decision Support System (Sistemas de Apoyo a las Decisiones)

⁴ ESS: Executive Support System (Sistemas de Apoyo a los Ejecutivos)

⁵ KWS: Knowledge Work System (Sistemas de Trabajo del Conocimiento)

⁶ ERP: Enterprise Resource Planning (Planeación de Recursos Empresariales)

Capítulo 1: Fundamentación Teórica

MIS ⁷	Datos resumidos, grandes cantidades	Reportes de rutina, bajo nivel de análisis	Informes resumidos y periódicos	Directivos de nivel medio
Ofimática	Documentos (texto, números)	Administración de documentos y multimedia	Documentos	Empleados generales
GDSS ⁸	No estructurada, procedimientos	En base a modelos	Resúmenes ejecutivos, gráficas	Todos pueden ser usuarios
EIS ⁹	Conocimiento del experto en un área	Ingeniería del conocimiento, reglas heurísticas	Bases de conocimiento, simulación del razonamiento de un experto	Empleados operativos
SIS ¹⁰	No estructurada	Intenso, secuencial, simulaciones, análisis	Detalles abundante, gráficas, informes ejecutivos	Todos pueden ser usuarios
eBIS ¹¹	Estructurada, grandes cantidades, externa	Intenso, interactivo, secuencial, flexible	Reportes con muchos detalles, ejecutivos, gráficas	Todos pueden ser usuarios

El diseño y desarrollo de sistemas de información

El diseño y desarrollo de sistemas de información sigue una metodología, que según varios autores es "...un enfoque para organizar, dirigir y realizar las actividades del ciclo de vida de un sistema de información." (9). Todas las metodologías muestran cuatro fases genéricas, las cuales son (10):

1. **Planificación preliminar:** analizar el problema de información en el contexto organizativo, definir los objetivos del sistema y organizar el proyecto y equipo de trabajo.
2. **Fase de diseño preliminar:** encontrar una solución a los problemas de información, mediante la recogida de datos, la definición de necesidades y medios, el análisis de diseños alternativos y la selección del diseño definitivo.

⁷ MIS: Management Information System (Sistemas de Información Administrativos)

⁸ GDSS: Group Decision Support System (Sistema de Apoyo a la Decisión en Grupo)

⁹ EIS: Expert Information System (Sistemas de Información Expertos)

¹⁰ SIS: Strategic Information System (Sistemas de Información Estratégicos)

¹¹ eBIS: e-Business Information System (Sistemas de Información de Negocios electrónicos)

Capítulo 1: Fundamentación Teórica

3. **Fase de desarrollo:** diseño de los elementos del sistema, entradas, salidas, procesos, controles, archivos y ficheros.
4. **Fase de instalación y desempeño:** puesta en marcha y desempeño del sistema.

Gestión de la Información

Son muchos los autores que han definido la gestión de la información, entre ellos se destaca una intelectual de nuestro país, quien define que:

“La gestión de la información es un proceso mediatizado por un conjunto de actividades que permiten la obtención de información, lo más pertinente, relevante y económica posible, para ser usada en el desarrollo y el éxito de una organización” (11)

La gestión de información tiene el objetivo de:

- ✓ Maximizar el valor y los beneficios derivados del uso de la información.
- ✓ Minimizar el costo de adquisición, procesamiento y uso de la información.
- ✓ Determinar responsabilidades para el uso efectivo, eficiente y económico de información.
- ✓ Asegurar un suministro continuo de la información (12).

Funciones

Las principales funciones de la gestión de la información son:

- ✓ Determinar las necesidades internas de información (relativas a las funciones, actividades y procesos administrativos de la organización) y satisfacerlas competitivamente.
- ✓ Determinar las necesidades de inteligencia externa de la organización y satisfacerlas competitivamente.
- ✓ Desarrollar la base informacional de la organización y garantizar su accesibilidad.
- ✓ Optimizar el flujo organizacional de la información y el nivel de las comunicaciones.
- ✓ Desarrollar la estructura informacional de la organización y garantizar su operatividad.
- ✓ Manejar eficientemente los recursos institucionales de información, mejorar las inversiones sucesivas en ellos, y optimizar su valor y su aprovechamiento organizacional.
- ✓ Garantizar la integridad y accesibilidad a la memoria corporativa.

Capítulo 1: Fundamentación Teórica

- ✓ Evaluar periódicamente la calidad e impacto del soporte informacional para la gestión y el desarrollo de la organización.
- ✓ Optimizar el aprovechamiento de la base y la estructura informacionales de la organización para incrementar su productividad o el rendimiento de la inversión.
- ✓ Establecer, aplicar y supervisar los procedimientos relativos a la seguridad de la información organizacional.
- ✓ Entrenar a los miembros de la organización en el manejo o la utilización, de los Recursos informacionales de la organización.
- ✓ Contribuir a modernizar u optimizar las actividades organizacionales y los procesos administrativos, relacionados con ellas.
- ✓ Garantizar la calidad de los productos informacionales de la organización, y asegurar su dimensión efectiva (13).

Sistemas de Gestión de Información

Los Sistemas de Gestión de Información permiten la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez, aprovechando al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones organizacional a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa (14).

La implementación de un sistema de gestión de la información brinda algunas ventajas como son:

- ✓ Control efectivo de las actividades de la organización.
- ✓ Integración de nuevas tecnologías y herramientas.
- ✓ Disponibilidad de mayor y mejor información para los usuarios en tiempo real.
- ✓ Elimina la barrera de la distancia trabajando con un mismo sistema en puntos distantes.
- ✓ Disminuye errores, tiempo y recursos (15).

Gestión Académica

La gestión académica es vital para el logro de la misión de una institución, así como sus metas de formación, por eso los profesores y los estudiantes son actores centrales en el diseño, implantación y desarrollo de procesos claves en la gestión académica, como son la elaboración y revisión de planes de estudio; las normas y los procedimientos de evaluación del desempeño de los estudiantes; la planificación de las actividades para la generación y aplicación del conocimiento; la selección de estudiantes y administración de procesos relacionados con ellos y la selección de los medios instrumentales y bibliográficos apropiados para la enseñanza, el aprendizaje y la investigación.

Otro punto de vista apunta a que dicha gestión académica es un proceso con características fundamentales en el orden administrativo. En una institución contribuye al proceso de formación integral del estudiante, tiene relación directa con la enseñanza, el aprendizaje, y los procesos de admisión de estudiantes, matrículas, pensiones y servicios educativos integrales.

Algunos de los componentes de la Gestión Académica son:

- ✓ Procesos de preinscripción.
- ✓ Selección y admisión.
- ✓ Planes de estudio.
- ✓ Modelos pedagógicos.
- ✓ Ofertas de programas.
- ✓ Información académica y administrativa de los estudiantes y docentes. (16)

Sistemas de Gestión Académica

Realizar los procesos docentes de forma manual puede resultar una tarea compleja y tediosa debido al gran volumen de información involucrado. Afortunadamente la incorporación de las tecnologías de la información ha dado la posibilidad de informatizar estos procesos mediante el empleo de sistemas automatizados para la gestión académica. Estos sistemas constituyen una poderosa herramienta de trabajo que permite hacer más eficientes los procesos docentes y optimizar sus recursos.

Por todo lo planteado, se hace necesario identificar las experiencias efectivas de las instituciones que estén inmersas en el desarrollo de soluciones informáticas y así obtener una solución al problema

existente, por lo que se requiere realizar un estudio sobre los sistemas similares de gestión de información académica principalmente en instituciones de educación superior.

1.3 Soluciones similares

Durante la investigación se analizaron sistemas similares en el ámbito internacional y nacional los cuales servirán como punto de referencia a la hora de desarrollar el sistema propuesto para dar solución a la problemática dada, a continuación se muestra un resumen de algunos de ellos.

Ámbito Internacional:

CENACAD: el Sistema de Censo Académico en Línea para la Automatización de la Evaluación a los Docentes, fue realizado por parte del personal de la Escuela Superior Politécnica del Litoral (ESPOL), (Guayaquil, Ecuador). Nace de la necesidad de brindar una solución informática sobre las evaluaciones docentes en la ESPOL. Cuenta con un módulo de reportes el cual permite a los usuarios la obtención de diferentes tipos de reportes, a continuación se muestra algunos de ellos (17):

- ✓ Listado de mejores docentes en base a los mejores promedios obtenidos en la ESPOL por el período de un año.
- ✓ Promedio de la(s) facultad(es) o instituto(s).
- ✓ Promedios obtenidos en cada área que conforman cada una de las encuestas realizadas a cada materia. Estos promedios deberán ser generales, de toda la facultad o unidad. Además, son presentadas de manera gráfica.

SIGA: Sistema de Información de Gestión Académica de la Universidad Técnica Federico Santa María (Valparaíso, Chile), con el fin de propiciar un crecimiento continuo de la Universidad y agregar valor a la Organización, su misión principal contribuir al desarrollo de las tecnologías de información con el fin de responder a las necesidades de la comunidad universitaria, a través del desarrollo de sistemas, que permitan mejorar la calidad de los servicios que brinda dicha institución. Dentro de sus indicadores más importantes se encuentran (18):

- ✓ Rendimiento Académico Ponderado
- ✓ Índice Académico
- ✓ Avance Académico

Capítulo 1: Fundamentación Teórica

- ✓ Prioridad Académica

Objetivos generales:

- ✓ Desarrollar los Sistemas Institucionales de la Universidad.
- ✓ Fortalecer la Tecnología de Información en la organización como agente de cambio.
- ✓ Dar acceso a la comunidad universitaria, a los recursos de información local y globalmente disponible.
- ✓ Aumentar la interacción de la comunidad para el enriquecimiento del conocimiento, la información, las habilidades y la comunicación.
- ✓ Generar nuevos servicios.

SIU: Sistema de Información Universitario de la Universidad de San Juan (UNSJ), Argentina, es un sistema que colabora con las universidades en el procesamiento de datos para facilitar el análisis de la información. Es utilizado por la UNSJ principalmente para la administración de la gestión referida a aspirantes, alumnos, graduados, docentes, becas y estadísticas. Su objetivo es contribuir a mejorar la gestión de las instituciones, permitiéndoles contar con información segura, íntegra y disponible, optimizar sus recursos y lograr que el software sea aprovechado en toda su potencialidad (19).

Tiene como objetivos:

- ✓ Promover un sistema de información integral que involucre a las universidades y a la Secretaría de Políticas Universitarias.
- ✓ Desarrollar los acuerdos necesarios para lograr el intercambio de conocimientos y llegar a soluciones informáticas comunes a los organismos involucrados.
- ✓ Garantizar la disponibilidad, integridad, seguridad y calidad de la información.
- ✓ Colaborar en el análisis de la información producida y la consecuente definición de estrategias a fin de lograr un adecuado aprovechamiento de los recursos del Estado.
- ✓ Contribuir con la transparencia de la gestión y la rendición de cuentas, asegurando el derecho a la información veraz.

A modo de resumen se puede concluir que, tras el estudio de las soluciones informáticas expuestas en el ámbito internacional, se arroja como resultado que aunque todas llevan a cabo la informatización de los

Capítulo 1: Fundamentación Teórica

procesos de gestión de información en las distintas instituciones, no pueden ser reutilizados ya que no se tiene acceso al código fuente de dichos sistemas por lo que la posibilidad de la reutilización queda descartada. No obstante, cabe resaltar el CENACAD, el cual maneja algunos indicadores docentes que pudieran tenerse en cuenta para provecho de los desarrolladores en el diseño del sistema propuesto. De los demás sistemas se pudo obtener muy poco pues carecen de funcionalidades importantes que sirvan de beneficio en el sistema a desarrollar.

Ámbito nacional:

Se realizó un estudio a los diferentes sistemas que realizaban la gestión académica en el país, destacándose que en la Facultad de Informática de la Universidad de Matanzas Camilo Cienfuegos, se cuenta con un Sistema para la Gestión Académica (GESTACAD), el cual brinda un grupo de funcionalidades que permiten mantener una información actualizada de estudiantes y profesores del centro, así como la obtención de reportes. Se investigó además en la provincia de La Habana el software Sistema de Gestión de la Nueva Universidad (SIGENU), el cual tiene como fin, ser una herramienta que permita la gestión de toda la información académica vinculada con la educación superior en Cuba. En correspondencia con su carácter nacional y la gran diversidad de sistemas de enseñanza superior con que cuenta la universidad cubana, este sistema ha sido concebido de manera tal que sea capaz de brindar gran seguridad e integridad de la información.

Otro de las soluciones estudiadas es el Sistema Docente Integral (SDI) creado en el Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE), el mismo surge debido a la vital importancia que tiene para este centro el control, planificación y resultado de los procesos docentes. El SDI permite registrar, procesar y crear mecanismos de recuperación de la información relacionada con el proceso docente, tanto de pregrado como de postgrado de forma automatizada. Esto constituye una ventaja ya que con su uso se pueden realizar estos procesos de forma eficiente permitiendo optimizar los recursos del centro de secretaría de las facultades.

Todas estas soluciones mencionadas anteriormente permitieron tener un conocimiento acerca del funcionamiento de la gestión académica en otras universidades del país. Cabe destacar que estos sistemas han permitido, en las instituciones en las que se han utilizado, un mejor desarrollo del proceso de gestión académica de manera general, pero carecen de funcionalidades que son importantes para dar

Capítulo 1: Fundamentación Teórica

solución a la situación problemática planteada, también en algunos de ellos, las funcionalidades están centralizadas a nivel de secretaría.

Es importante resaltar que en todos estos sistemas mencionados anteriormente, no se realizó una investigación más detallada ya que los mismos responden a fines propios de cada universidad a la que corresponden. Además el objetivo de la investigación está enfocado a los procesos de gestión académica propios de la UCI, por lo que como esta cuenta con un sistema de gestión universitaria, se decidió realizar una exploración específicamente a este sistema. A continuación se muestra una descripción del mismo.

XAUGE. SGU: Sistema de Gestión Universitaria (UCI), es un sistema con que cuenta la universidad para manejar la información en las distintas áreas, está dividido por varios módulos uno de ellos es Pregrado el cual se encarga de gestionar todo lo relacionado con los temas de la docencia de los estudiantes. Este módulo gestiona evaluaciones del estudiante, exámenes de premio, bonificaciones y promedio. El usuario es capaz de obtener la información mediante el filtrado de los diferentes componentes y esta información puede ser exportada en formato pdf.

El proceso de evaluación y control docente inicia con la creación de los grupos docentes luego de que la secretaria (general o facultad) recibe notificación del inicio del curso docente o inicio de un período académico y se ha culminado el reajuste de los grupos administrativos y los estudiantes han sido promovidos. La secretaria a partir de los años y períodos activos, crea los grupos docentes por cada asignatura definida en la organización de plan de estudio del período, según el tipo que le corresponda ya sean optativas o básicas y a cada uno de ellos le asocia los profesores que los atenderán. Cuando se crean los grupos docentes se confecciona el registro de asistencia y evaluaciones para cada uno y este se identifica con la carrera, año académico, tipo de curso y curso escolar. Durante todo el período docente el profesor le registra la asistencia y las evaluaciones a los estudiantes de los grupos docentes que tiene asociado. Cuando llega la fecha de culminación del período docente, la secretaría chequea el registro de las evaluaciones definidas para el período en el que se encuentren y teniendo en cuenta que se cumplan, las reglas de promociones establecidas, promueve a los estudiantes a un nuevo período o año académico y luego realiza el cierre del grupo docente (20).

XAUGE brinda algunos elementos importantes como son la gestión de las evaluaciones, el manejo de la asistencia y la exportación de la información en formato pdf. Por otra parte, se identificaron algunos

elementos con los cuales no contaba este sistema como el manejo de las habilidades a superar en cada evaluación para permitir al profesor efectuar un monitoreo más personalizado en cuanto a la trayectoria de los estudiantes. Además, para visualizar el comportamiento de los grupos en las diferentes asignaturas el equipo de desarrollo deliberó que era necesario que el sistema aparte de la gestión de información contara con algunas gráficas para establecer puntos de referencia en cuanto a indicadores de promedios de evaluaciones, asistencia entre otros parámetros para darle valor agregado a dicho sistema, así como contar con enlaces a páginas internas relacionadas con el tema de la docencia para darle mayor accesibilidad al mismo. Por las deficiencias antes descritas queda descartado el sistema XAUGE para dar respuesta a la situación problemática planteada.

A partir del estudio de los sistemas similares se detecta que brindan una serie de funciones que serán tomadas como base para la propuesta de solución, incorporándoles además las que se describen anteriormente que no poseen. Para cumplir el objetivo de esta investigación es imprescindible definir algunas tecnologías y herramientas que posibiliten el desarrollo exitoso del sistema.

1.4 Lenguajes, tecnologías y herramientas de desarrollo de software

Para el desarrollo del sistema se hizo un análisis minucioso sobre las tendencias de los lenguajes, las tecnologías y herramientas de mayor demanda en la actualidad para el desarrollo del software, estableciendo una serie de comparaciones con el propósito de elegir las más idóneas y eficaces para dar solución a la problemática de la investigación.

Metodologías

Las metodologías no son más que una combinación de los modelos de procesos genéricos (cascada, evolutivo, incremental, etc.). Adicionalmente una metodología debería definir con precisión artefactos, roles y actividades involucrados, conjuntamente con las prácticas y técnicas recomendadas. Habitualmente se utiliza el término método para referirse a técnicas, notaciones y guías asociadas, que son aplicables a algunas actividades del proceso de desarrollo. En la mayoría de los casos suele hablarse de métodos de análisis y/o diseño (21).

Las metodologías de desarrollo influyen directamente en el proceso de construcción de software y se elaboran a partir del marco definido por el grupo de trabajo. Por tanto, se puede concluir que para

Capítulo 1: Fundamentación Teórica

desarrollar un proyecto es necesario establecer un enfoque disciplinado y sistemático. Según Piattini”... *una metodología es un conjunto de procedimientos, herramientas y soporte documental que ayuda a los desarrolladores a realizar un nuevo software*” (22).

Metodologías Ágiles

El desarrollo ágil de software son métodos de ingeniería del software basado en el desarrollo iterativo e incremental, donde los requerimientos y soluciones evolucionan mediante la colaboración de grupos auto organizado y multidisciplinario. Existen muchos métodos de desarrollo ágil, la mayoría minimiza riesgos desarrollando software en lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como indisciplinados por la falta de documentación técnica (23).

A continuación se muestran algunas de las metodologías ágiles más usadas:

Extreme Programming (XP):

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre (24).

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste a grandes rasgos en los siguientes pasos (25):

Capítulo 1: Fundamentación Teórica

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma, el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

Según Kent Beck considerado el padre de esta metodología, el ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto (26).

Fase I: Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.

Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Fase II: Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Capítulo 1: Fundamentación Teórica

Fase III: Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Fase IV: Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Fase V: Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Fase VI: Muerte del Proyecto

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Buenas Prácticas en XP

Capítulo 1: Fundamentación Teórica

Según afirman varios autores, la principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione (27). Lo anterior indica que la aplicación de las buenas prácticas nos permite minimizar costos y optimizar el rendimiento del proyecto. A pesar de los beneficios que se pueden obtener del uso de todas las prácticas en conjunto, la metodología no obliga, ni exige la aplicación estricta de todas ellas. A continuación se describen algunas de las principales prácticas que más se utilizan en eXtreme Programming.

El juego de la planificación: Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

Entregas pequeñas: Es la idea de producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero sí que constituyan un resultado de valor para el negocio. Una entrega no debería tardar más de tres meses.

Diseño simple: Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente.

Pruebas: La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse.

Refactorización: La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.

Propiedad colectiva del código: Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.

Capítulo 1: Fundamentación Teórica

Cliente in-situ: El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada.

Estándares de programación: XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación, los mismos mantienen el código legible para los miembros del equipo, facilitando los cambios.

Scrum:

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, estas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración (23).

Metodologías Pesadas

Están guiadas por una fuerte planificación. Centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo. Entre las metodologías robustas se encuentran: MSF (por sus siglas en inglés Microsoft Solution Framework), MÉTRICA 3 y RUP (siglas de Rational Unified Process). Según el criterio de autores este esquema tradicional para abordar el desarrollo del software ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de ceremonia en el proceso. Sin embargo este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exigen reducir drásticamente los tiempos de desarrollo pero manteniendo un alta calidad (28).

Microsoft Solution Framework (MSF):

Capítulo 1: Fundamentación Teórica

Es una herramienta que ayuda a llevar a cabo con éxito los planes de acción creados como parte del proceso de administración de riesgos de seguridad de Microsoft. Se ha diseñado para ayudar a las organizaciones a ofrecer soluciones de tecnología de alta calidad puntuales y según lo presupuestado. MSF sintetiza un enfoque bien disciplinado orientado hacia los proyectos tecnológicos basándose en un conjunto definido de principios, modelos, disciplinas, conceptos, orientaciones y prácticas demostradas de Microsoft. Microsoft Solution Framework propone un conjunto de componentes y buenas prácticas que pueden ser adoptadas e implementados en conjunto o por separado, sus prácticas aplican tanto a los equipos de trabajo como a los procesos de desarrollo de software, cuenta con diferentes disciplinas de administración de proyectos y sus principios pueden ser adoptados en la mayoría de los proyectos de desarrollo. Esto proporciona una gran ventaja puesto que al adoptar este framework se podría incorporar cada uno de sus prácticas en forma incremental para así analizar su impacto, ver el costo y el beneficio que proporciona en los proyectos y así dependiendo de los resultados obtenidos poder incrementar su uso y aplicación en los procesos de desarrollo (29). Esta metodología consta de cinco fases como se muestra a continuación:

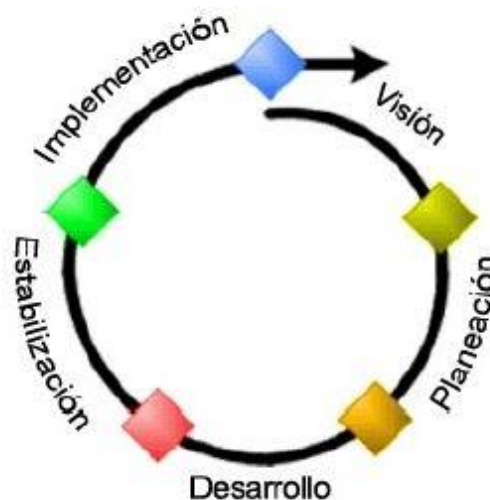


Figura 1. Fases de MSF (27)

Rational Unified Process (RUP):

Capítulo 1: Fundamentación Teórica

El proceso unificado conocido como RUP, es un modelo de software que permite el desarrollo de software a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Aunque con el inconveniente de generar mayor complejidad en los controles de administración del mismo. Sin embargo, los beneficios obtenidos recompensan el esfuerzo invertido en este aspecto. Constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requerido en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas (30). Es iterativo e incremental, divide el proyecto en mini proyectos donde los casos de usos y la arquitectura cumplen sus objetivos de manera más depurada. RUP se encarga de unificar todo el equipo de desarrollo de software, además de optimizar su comunicación. Para esto provee a cada miembro del proyecto una aproximación al desarrollo de software con una base de conocimiento de acuerdo con las necesidades específicas del proyecto. No es simplemente un proceso, sino que es un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos.

A continuación se muestra la estructura del ciclo de vida del proceso de desarrollo unificado:

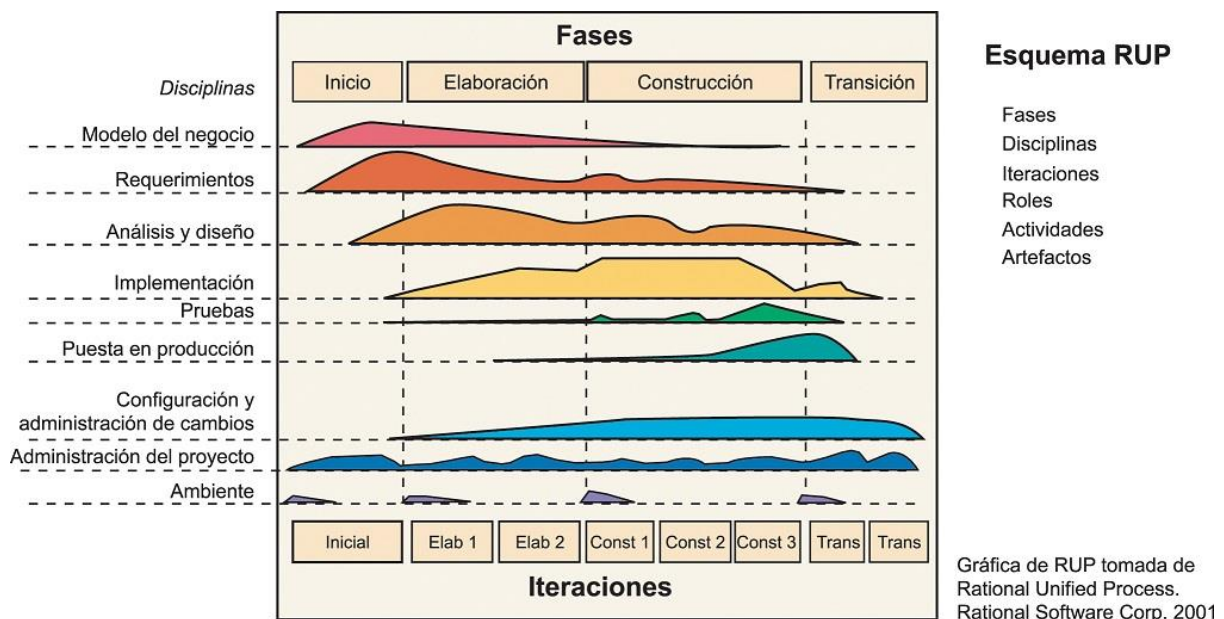


Figura 2. Fases de RUP (30)

Métrica 3:

MÉTRICA es una metodología de planificación, desarrollo y mantenimiento de sistemas de información. Ofrece a las organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software dentro del marco que permite alcanzar los siguientes objetivos:

- ✓ Proporcionar o definir sistemas de información que ayuden a conseguir los fines de la organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- ✓ Dotar a la organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.
- ✓ Mejorar la productividad de los departamentos de Sistemas y Tecnologías de la Información y las Comunicaciones, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible.
- ✓ Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.
- ✓ Facilitar la operación, mantenimiento y uso de los productos software obtenido.

Esta metodología tiene un enfoque orientado al proceso, ya que la tendencia general en los estándares se encamina en este sentido y por ello, se ha enmarcado dentro de la norma ISO 12.207, que se centra en la clasificación y definición de los procesos del ciclo de vida del software. Como punto de partida y atendiendo a dicha norma, MÉTRICA versión 3 cubre el proceso de desarrollo y el proceso de mantenimiento de sistemas de información. Esta ha sido concebida para abarcar el desarrollo completo de sistemas de información sea cual sea su complejidad y magnitud, por lo cual su estructura responde a desarrollos máximos y deberá adaptarse y dimensionarse en cada momento de acuerdo a las características particulares de cada proyecto (28).

Selección de la metodología

Luego de realizarse una profunda investigación acerca de las metodologías más usadas en el desarrollo de software en la actualidad se escogió la metodología XP por ser la que más se adapta a las

Capítulo 1: Fundamentación Teórica

necesidades del equipo de desarrollo, el cual es pequeño, también los requisitos son imprecisos y pueden cambiar. Además, XP se asocia más al tipo de proyecto que se lleva a cabo durante la construcción de la solución informática ya que es empleada para proyectos de corto plazo, con un equipo de trabajo pequeño, propone una retroalimentación continua entre el cliente y el equipo de desarrollo, es decir el cliente está disponible todo el tiempo, define un modelo de 40 horas semanales para no trabajar horas extras, entre otros aspectos mencionados con anterioridad. Para una mejor estructuración de la investigación y un mejor entendimiento, se decidió agrupar las seis fases en cuatro, sin violar el ciclo de vida de la metodología, estas fases son:

- ✓ Fase 1: Exploración
- ✓ Fase 2: Planificación
- ✓ Fase 3: Implementación
- ✓ Fase 4: Pruebas

Lenguaje de programación

PHP

PHP en su versión 5.3 significa Hypertext Pre-processor, es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es un lenguaje de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. El código fuente escrito en PHP es invisible al navegador y al cliente, esto hace que la programación sea segura y confiable (31).

HTML

HTML es el lenguaje utilizado para la creación de páginas Web. Más concretamente, HTML significa “HyperText Mark-Up Language”, que viene a ser, en nuestro idioma, “Lenguaje para el Formato de Documentos de Hipertexto”. Es decir, los documentos HTML no son documentos de texto normal, sino documentos de hipertexto ya que en el propio documento aparecen enlaces a otros documentos.

Capítulo 1: Fundamentación Teórica

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado *World Wide Web Consortium*, más conocido como W3C¹². Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global" (32).

JavaScript

JavaScript en su versión 1.2 es un lenguaje de programación interpretado, se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. A pesar de que su nombre, no guarda ninguna relación directa con el lenguaje de programación Java, tienen semánticas y propósitos diferentes.

Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Existen al menos tres formas para incluir código JavaScript en las páginas web:

- ✓ Incluir el código en la misma página web: el código JavaScript se encierra entre etiquetas `<script>` y se incluye en cualquier zona del documento. Aunque es correcto incluir el bloque de código dentro de la cabecera del documento.
- ✓ Definir el código en un archivo externo: las instrucciones JavaScript se incluyen en un archivo de extensión `*.js` que luego es enlazado a través de etiquetas dentro del código de la página.
- ✓ Es muy útil cuando se trata de reutilizar código para las demás páginas web dentro de un mismo proyecto.
- ✓ Incluir el código dentro de elementos de la página web: consiste en incluir trozos de código JavaScript dentro del código XHTML de la página. Se utiliza, en ocasiones, para definir eventos, puesto que complica el mantenimiento del código.

¹² W3C: Comunidad internacional española que desarrolla estándares que aseguran el crecimiento de la web a largo plazo, su objetivo es guiar la web hacia su máximo potencial.

Capítulo 1: Fundamentación Teórica

JavaScript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, con funciones y estructuras de datos complejas. Además, pone a disposición del programador todos los elementos que forman la página web, para que este pueda acceder a ellos y modificarlos dinámicamente (33).

CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento (34).

Las tres formas más conocidas de dar estilo a un documento son las siguientes:

- ✓ Utilizando una hoja de estilo externa que estará vinculada a un documento a través del elemento `<link>`, el cual debe ir situado en la sección `<head>`.
- ✓ Utilizando el elemento `<style>`, en el interior del documento al que se le quiere dar estilo, y que generalmente se situaría en la sección `<head>`. De esta forma los estilos serán reconocidos antes de que la página se cargue por completo.
- ✓ Utilizando estilos directamente sobre aquellos elementos que lo permiten a través del atributo `<style>` dentro de `<body>`. Pero este tipo de definición del estilo pierde las ventajas que ofrecen las hojas de estilo al mezclarse el contenido con la presentación.

Estándares de servicios web

XML

Es un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas

Capítulo 1: Fundamentación Teórica

estructurados, ofrece la posibilidad de estructurar y representar datos. Los expertos señalan varias ventajas que derivan de la utilización del XML, como por ejemplo que: es extensible (se pueden añadir nuevas etiquetas tras el diseño del documento); su analizador es estándar (no requiere de cambios para cada versión del metalenguaje); facilita el análisis y el procesamiento de los documentos XML creados por terceros (35).

WSDL

Por sus siglas (Web Services Description Language). Es una notación XML para describir un servicio web describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje (36).

SOAP

Es un protocolo estándar que permite la comunicación y la interoperabilidad entre diversas aplicaciones web desarrolladas bajo tecnologías diferentes, por sus siglas en inglés (*Simple Object Access Protocol*). Define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Algunas de sus principales ventajas son:

Entre las ventajas de SOAP se tiene que:

- ✓ Es sencillo de implementar, probar y usar
- ✓ Atraviesa firewalls y routers, pues estos piensan que es una comunicación HTTP.
- ✓ Tanto los datos como las funciones se describen en XML, lo que permite que el protocolo no solo sea más fácil de utilizar sino que también sea muy sólido.
- ✓ Es independiente del sistema operativo y procesador.
- ✓ Se puede utilizar tanto de forma anónima como con autenticación (nombre/clave).
- ✓ Permite la interoperabilidad entre múltiples entornos (37).

Entorno de desarrollo integrado

Netbeans IDE

Capítulo 1: Fundamentación Teórica

El entorno de desarrollo integrado NetBeans en su versión 7.3 disponible para Windows, MacOS, Linux y Solaris. Consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general para compilar cualquier tipo de aplicación.

Es una herramienta para que los programadores puedan escribir, depurar y ejecutar programas. Está escrito en Java, es un producto libre y gratuito sin restricciones de uso (38).

Principales características:

- ✓ Propone un esqueleto para organizar el código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS. Además, posee un sistema para examinar todos los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.
- ✓ El editor de PHP, es mucho más ágil y a la vez robusto, contiene más ayuda en línea, reconocimiento de sintaxis y todo lo que provee la última versión de PHP.

Herramienta CASE

Visual Paradigm:

Es una herramienta CASE (Ingeniería de Software Asistida por Computación). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos (28).

Características principales:

- ✓ Tiene disponibilidad en múltiples plataformas y en múltiples versiones. Esta característica es muy importante pues Visual Paradigm está disponible para varios sistemas operativos como Windows, Linux, Unix.

Capítulo 1: Fundamentación Teórica

- ✓ Proporciona una plataforma de modelado colaborativo para el trabajo en equipo, los miembros pueden ver y editar el mismo proyecto, o el mismo esquema, incluso de forma simultánea.
- ✓ Todos los cambios se almacenan en el servidor de Visual Paradigm en función de revisión.
- ✓ Brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir de código.

Marco de Trabajo

Symfony 2.0

Symfony2 es la versión más reciente de Symfony, el popular framework para desarrollar aplicaciones PHP. Se anunció por primera vez a principios de 2009 y supone un cambio radical tanto en arquitectura interna como en filosofía de trabajo respecto a sus versiones anteriores. Ha sido ideado para exprimir al límite todas las nuevas características de PHP 5.3 y por eso es uno de los frameworks PHP con mejor rendimiento. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en el proyecto (39).

Ventajas:

- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo
- ✓ Código fácil de leer.
- ✓ Fácil de extender, lo que permite su integración con bibliotecas desarrolladas por terceros.
- ✓ La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- ✓ Nos permite reutilizar código entre los proyectos.
- ✓ Separación de modelo, vista y controlador (implementación MVC2).
- ✓ Dispone de una amplia documentación.

Desventajas:

- ✓ Fuerte curva de aprendizaje, aunque la documentación disponible facilita enormemente esta tarea. Este es un factor a tener en cuenta ya que la productividad del equipo de desarrollo no será alta en los primeros meses.
- ✓ Todavía hay muchas aplicaciones y desarrolladores que programan en PHP4. Esto puede plantear un problema ya que para extraer toda la potencia del Framework, el programador debe tener un buen conocimiento del paradigma de programación orientado a objetos.
- ✓ Aumenta el peso de la aplicación, ya que la implementación del patrón MVC introduce varias capas de objetos que debe procesar el intérprete. La parte que más afecta al rendimiento es Propel, el sistema ORM utilizado por defecto en Symfony.

Servidor web

Apache 2.2

El servidor Apache se desarrolla dentro del proyecto HTTP Servidor de la Apache Software Fundación. Presenta características como, mensajes de errores altamente configurables, bases de datos de autenticación y negociado de contenido. Tiene amplia aceptación en la red convirtiéndose en el servidor HTTP más usado, y el mejor servidor de páginas web en el mundo.

La arquitectura que utiliza es cliente-servidor, el cliente hace la petición al servidor y este atiende dicha petición. El protocolo utilizado para la transferencia de hipertexto es HTTP que está basado en el envío de mensajes y establece el conjunto de normas mediante las cuales se envían las peticiones de acceso a una web y la respuesta de esa web (40).

Ventajas de Apache:

- ✓ Código abierto.
- ✓ Extensible.
- ✓ Fácil de conseguir ayuda/suporte.
- ✓ Es multiplataforma, aunque idealmente está preparado para funcionar bajo Linux.
- ✓ Muy sencillo de configurar.
- ✓ Amplias librerías de PHP y PERL a disposición de los programadores.

Capítulo 1: Fundamentación Teórica

- ✓ Posee diversos módulos que permiten incorporarle nuevas funcionalidades, estos son muy simples de cargar.
- ✓ Es capaz de utilizar lenguajes como PHP, TCL, Pitón, etc.

Sistema gestor de bases de datos (SGBD).

Un sistema de base de datos es básicamente un sistema computarizado para guardar registros; es decir, es un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones. La información en cuestión puede ser cualquier cosa que sea de importancia para el individuo u organización; en otras palabras, todo lo que sea necesario para auxiliarle en el proceso general de su administración (41).

MYSQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Este gestor de bases de datos es uno de los más usados en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación se debe a que existen infinidad de librerías y otras herramientas que permiten su uso a través de un gran número de lenguajes de programación, además de su fácil instalación y configuración (42).

PostgresSQL

PostgreSQL es un SGBD objeto-relacional, utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Desde su creación, hace más de 16 años la estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Es de código abierto, al igual que todo el software libre. Cuenta con dos ventajas claras: un código fuente optimizado que puede ser modificado y adaptado, y una baja inversión por implementación, ya que no existen costos por licencia (29).

Fundamentación de la selección

Capítulo 1: Fundamentación Teórica

Después del estudio realizado de los SGBD anteriormente mencionados se seleccionó PostgreSQL para llevar a cabo el desarrollo de la aplicación web ya que es un software de código abierto, posee una gran escalabilidad, es capaz de ajustarse al número de unidades de procesamiento y a la cantidad de memoria que posee el sistema de forma óptima. Es multiplataforma y soporta el lenguaje de programación PHP. Además, otra de las ventajas de PostgreSQL es que es libre, lo que no requiere pago de licencia, mientras que MySQL es una herramienta de carácter propietario en sus últimas versiones, lo cual imposibilita la obtención de su licencia.

1.5 Conclusiones

En este capítulo se realizó un estudio de los diferentes conceptos asociados al objeto de estudio, destacándose los sistemas de información, la gestión de la información y los sistemas de gestión de la información; se demuestra que la selección de los métodos científicos fueron los correctos, permitiendo desarrollar la teoría que sustenta esta investigación. Además, estos métodos posibilitaron realizar un estudio detallado de las herramientas, tecnologías, metodologías y lenguajes de programación existentes para el desarrollo de un sistema informático. Siendo seleccionada como metodología para guiar el desarrollo del sistema XP, como servidor web Apache y como framework de desarrollo Symfony, siendo todo esto las opciones más factibles para dar cumplimiento al objetivo general.

Capítulo 2 Exploración y Planificación

2.1 Introducción:

Luego de haber realizado un estudio del estado del arte y haber elegido la metodología y las herramientas a utilizar, se está en condiciones de plantear una propuesta de solución al sistema, basadas en las dos primeras fases de la metodología de desarrollo XP que son la exploración y la planificación, cuyo objetivo fundamental es conocer el alcance del producto final así como estimar los tiempos del Plan de iteraciones, y planes de entrega.

2.2 Propuesta de solución

El sistema propuesto tiene como objetivo el apoyo a la gestión de la información académica para contribuir en el desarrollo de la labor docente del profesor. Dentro del mismo los usuarios podrán desempeñar tres roles diferentes: Administrador, Jefe de Asignatura y Profesor. Inicialmente el acceso al sistema se realizará mediante un usuario con rol Administrador que contiene el mismo en su configuración predeterminada.

El usuario con rol Administrador es el encargado de la gestión completa del sitio y es el usuario de mayor privilegio en el sistema, además va a ser el encargado de gestionar las asignaturas así como los demás usuarios y los roles que estos van a desempeñar. Una vez creados los usuarios se le asignan los permisos para diferenciar los diferentes recursos a los que van a tener acceso (asignaturas, temas, evaluaciones, etc.), esto garantiza que un profesor de una asignatura no tenga acceso a los recursos de otra asignatura.

El acceso con rol Jefe de Asignatura le permite al usuario gestionar todo los recursos de la asignatura que le fue asignada como: temas, evaluaciones, actividades, habilidad y grupo en caso de darle clases a alguno. Finalmente el usuario con rol Profesor va a interactuar con los recursos de la asignatura que le fue asignada, el mismo va a gestionarse sus grupos, pasar asistencia y evaluar a sus estudiantes. Luego con los datos registrados podrá visualizar el comportamiento de sus grupos mediante gráficas y reportes, los cuales podrán ser exportados en formato “.doc”.

Capítulo 2: Exploración y Planificación

Cabe resaltar que el Profesor no tendrá que gestionar los grupos de forma manual, para ello el equipo de desarrollo acordó utilizar un servicio SOAP (<http://pregrado.uci.cu/v2/PregradoWS.wsdl>) que brinda la universidad para acceder a los datos de los estudiantes y crear los grupos de forma dinámica.

Finalmente con todos estos aspectos se creará un sistema para apoyar la gestión de la información académica de los estudiantes, el cual será capaz de brindar reportes en cuanto a las evaluaciones, y asistencia de los estudiantes, así como habilidades no superadas en las distintas asignaturas, además el sistema brindará al profesor un corte general evaluativo acerca de la trayectoria de los estudiantes en un período determinado. Todo lo antes mencionado será visualizado mediante gráficas de comportamiento teniendo en cuenta los indicadores docentes gestionados. Con el desarrollo del sistema propuesto se desea crear un instrumento de apoyo que automatice muchos de los procesos que se realizan manualmente y mejore los que se realizan de forma computacional.

2.3 Definición de la audiencia

La audiencia constituye uno de los elementos más importantes de esta fase de desarrollo del sistema, ya que consiste en definir al público a quien va dirigida la solución del sistema, en este caso la solución va dirigida a los profesores de la facultad 4 de la universidad.

2.4 Personas relacionadas con el sistema

Existen diferentes tipos de personas que trabajan directamente con el sistema o sea interactúan con él obteniendo resultados en los procesos desarrollados. Ellos trabajan directamente con el sistema en funcionamiento e inciden en el resultado de la automatización de los procesos que realiza esta aplicación, a estos individuos se les denomina: personas relacionadas con el sistema.

Tabla 2. Personal relacionado con el sistema

Personal relacionado con el sistema	Descripción
Administrador	Es el encargado de la gestión y configuración del sitio en general, gestiona las asignaturas y los demás usuarios.

Capítulo 2: Exploración y Planificación

Jefe de Asignatura	Es la persona encargada de gestionar los recursos de las asignaturas del sitio como: temas, evaluaciones, actividades y habilidades, su nivel de privilegio es medio.
Profesor	Es el usuario que va a interactuar directamente con las evaluaciones de los estudiantes y es quien le va a dar seguimiento a su desempeño académico, su nivel de privilegios es el más bajo en el sistema.

2.5 Historia de usuario

Las historias de usuario (HU) son la forma en que se especifican en XP los requisitos del sistema. Estas se escriben desde la perspectiva del cliente; aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas; se estima que el tiempo de desarrollo ideal para una HU varía entre una y tres semanas (43). El contenido de estas debe ser concreto y sencillo de forma tal que en cualquier momento se puedan eliminar, reemplazar por otras más específicas o generales, añadirse nuevas o ser modificadas. Para ser implementadas las HU, el cliente y los desarrolladores se reúnen para detallar las funcionalidades de cada una. En esta primera fase de desarrollo el equipo de trabajo acordó con el cliente que las HU referentes a adicionar, actualizar y eliminar se iban a tratar bajo el término de gestionar.

Según Kent Beck cada HU define una serie de aspectos que pueden variar según la necesidad, es decir no siempre están todos presentes en cada una, además se añade un campo de observaciones con información extra para mejor comprensión (24):

- ✓ **Número:** Posee el número asignado a la HU.
- ✓ **Nombre de HU:** Atributo que contiene el nombre de la HU.
- ✓ **Usuario:** El usuario del sistema que utiliza o protagoniza la HU.
- ✓ **Prioridad en el negocio:** Evidencia el nivel de prioridad de la HU en el negocio.
- ✓ **Riesgo de desarrollo:** Evidencia el nivel de riesgo en caso de no realizarse la HU.

Capítulo 2: Exploración y Planificación

- ✓ **Puntos estimados:** Este atributo no es más que una estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como 5 días hábiles trabajando 40 horas, es decir, 8 horas diarias. Por lo que cuando el valor de dicho atributo es 0.5 equivale a 2 días y medio de trabajo, lo que se traduce en 20 horas.
- ✓ **Puntos reales:** Igual que el parámetro anterior, pero en este caso será el tiempo real en el que se realizó la HU.
- ✓ **Descripción:** Posee una breve descripción de lo que realizará la HU.

A continuación se muestran las HU que representan algunas de las funcionalidades implementadas para la propuesta de solución. En los anexos (Ver Anexo I) se encuentran las HU de las restantes funcionalidades.

Tabla 3. HU Gestionar usuario

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Gestionar usuario
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Iteración Asignada: 1	Puntos Estimados: 1.5
<p>Descripción: El sistema permite crear, modificar y eliminar usuarios en la Base de Datos. En caso de que el Administrador adicione un nuevo usuario, el sistema le muestra en pantalla los datos necesarios, en caso de que quiera, modificar usuario, le aparecen los datos insertados del mismo y este efectuará los cambios pertinentes, y en caso de que quiera eliminar un usuario, este lo selecciona, eliminándose de la base de datos. En caso de adicionar los permisos administrativos solo los asigna el rol Administrador. En cada una de las opciones el sistema muestra un mensaje de confirmación.</p> <p>Del usuario se registran los siguientes datos:</p> <ul style="list-style-type: none">✓ Nombre (Obligatorio)✓ Contraseña (Obligatorio)✓ Rol (Obligatorio)✓ Activado (Opcional)	

Capítulo 2: Exploración y Planificación

Tabla 4. HU Gestionar grupos

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Gestionar grupos
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Ato
Iteración Asignada: 2	Puntos Estimados: 1.5
<p>Descripción: Los usuarios pueden crear, modificar y eliminar los grupos. En caso de que el usuario adicione un nuevo grupo, el sistema le muestra en pantalla los datos necesarios, en caso de que quiera, modificar grupo, le aparecen los datos insertados del mismo y este efectuará los cambios pertinentes, y en caso de que quiera eliminar un grupo, este lo selecciona, eliminándose de la base de datos; además una vez creado el grupo se le podrán asignar los estudiantes, eliminarlos, pasarle asistencia y evaluarlos. En cada una de las opciones el sistema muestra un mensaje de confirmación.</p> <p>Del grupo se registran los siguientes datos:</p> <ul style="list-style-type: none">✓ Nombre (Obligatorio)	

Tabla 5. HU Corte evaluativo

Historia de Usuario	
Número: 14	Nombre Historia de Usuario: Corte evaluativo
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Iteración Asignada: 3	Puntos Estimados: 1.5
<p>Descripción: Se le inserta el grupo y la asignatura, el sistema generará un reporte de las evaluaciones de los estudiantes y la asistencia de los mismos con una evaluación propuesta por el sistema (bien, regular o mal), el cual puede ser exportado a un documento en formato “.doc”.</p>	

Capítulo 2: Exploración y Planificación

Tabla 6. HU Estudiantes con bien, regular y mal

Historia de Usuario	
Número: 21	Nombre Historia de Usuario: Estudiantes con bien, regular y mal
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Medio
Iteración Asignada: 4	Puntos Estimados: 0.5
Descripción: Se le inserta el grupo y la asignatura, el sistema generará un reporte gráfico con la cantidad de estudiantes evaluados de bien, regular y mal así como el porcentaje que representan en el grupo.	

2.6 Estimación en esfuerzos por HU

Para el desarrollo del sistema propuesto se realizó una estimación del esfuerzo para cada una de las historias de usuario las cuales deben tener el detalle mínimo para que los programadores puedan realizar una estimación poco riesgosa del tiempo que le llevará su desarrollo. Primero, el equipo de desarrolladores estima el esfuerzo necesario para implementarlas y los clientes certifican los objetivos y tiempos de entrega, la estimación temporal se basa en un cálculo estimado por parte de los desarrolladores de cada una de las mismas (44).

Para la implementación del sistema se utilizarán las semanas como medida para la estimación de esfuerzos establecidas por los programadores. A continuación se muestra la tabla Estimación de esfuerzos:

Tabla 7. Estimación de esfuerzos

No	Historia de usuario	Estimación (semanas)
1	Gestionar usuario	1.5
2	Autenticar usuario	0.5
3	Gestionar asignatura	1

Capítulo 2: Exploración y Planificación

4	Gestionar tema	0.5
5	Gestionar grupos	1.5
6	Gestionar actividad docente	0.5
7	Gestionar evaluación	0.5
8	Gestionar habilidad	0.5
9	Mostrar asistencia	0.5
10	Asistencia por encima de 20 por ciento	0.5
11	Promedio de evaluaciones	1
12	Evaluaciones por criterio	0.5
13	Habilidades suspensas	0.5
14	Corte evaluativo	1.5
15	Resultado de las evaluaciones	0.5
16	Presentes por actividades	0.5
17	Presentes por tema	0.5
18	Estudiantes suspensos y aprobados por habilidad	0.5
19	Promedio de notas por evaluaciones	0.5
20	Asistencia y aprobados general	0.5
21	Estudiantes con bien, regular y mal	0.5

2.7 Plan de iteraciones

Una vez descritas las historias de usuario por parte del cliente y estimado el esfuerzo por los desarrolladores para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema. Este plan agrupa las historias de usuario por iteraciones, especificando cuáles serán desarrolladas en cada iteración del proceso de implementación. En relación con lo antes mencionado se decide implementar el sistema en tres iteraciones, las cuales se describen a continuación:

Iteración 1.

En esta iteración se implementan las historias de usuario de mayor prioridad según el cliente, la realización de las mismas proporciona una idea de cómo funcionará la aplicación aunque todavía esté en sus inicios.

Iteración 2.

En esta iteración se realizarán las historias de usuario de prioridad media relacionadas con la parte de gestión del sistema. Una vez concluida esta iteración el sistema se encontrará en un estado casi completamente funcional.

Iteración 3.

En esta iteración se les dará cumplimiento a las historias de usuario de prioridad media. Esta es la etapa de trabajo en la que se necesita de mayor esfuerzo ya que exige mucho de los desarrolladores. Todas las historias que incidan en la lógica de la programación del sistema deben quedar implementadas.

Iteración 4.

En esta iteración se les dará cumplimiento a las historias de usuario de menor prioridad. Haciendo que el sistema esté funcionalmente listo. En esta etapa los desarrolladores terminan las últimas HU relacionadas con los reportes gráficos, una vez concluidas el sistema estará listo funcionalmente según lo pactado con el cliente.

Capítulo 2: Exploración y Planificación

2.8 Plan de duración de las iteraciones

El plan de duración de las iteraciones se realiza luego de tener el estimado en días que demora implementar cada historia de usuario. El objetivo del plan de duración de iteraciones es especificar detalladamente el orden de desarrollo de las HU dentro de cada iteración y la duración de las iteraciones.

Tabla 8. Duración de las iteraciones

Iteración	Historia de usuario	Puntos estimados	Duración total
Iteración 1	Gestionar usuario	1.5	3
	Autenticar usuario	0.5	
	Gestionar asignatura	1	
Iteración 2	Gestionar tema	0.5	3.5
	Gestionar grupos	1.5	
	Gestionar actividad docente	0.5	
	Gestionar evaluación	0.5	
	Gestionar habilidad	0.5	
Iteración 3	Mostrar asistencia	0.5	4.5
	Asistencia por encima del 20 por ciento	0.5	
	Promedio de evaluaciones	1	
	Evaluaciones por criterio	0.5	

Capítulo 2: Exploración y Planificación

	Habilidades suspensas	0.5	
	Corte evaluativo	1.5	
Iteración 4	Resultado de las evaluaciones	0.5	3.5
	Presentes por actividades	0.5	
	Presentes por temas	0.5	
	Estudiantes suspensos y aprobados por habilidad	0.5	
	Promedio de notas por evaluaciones	0.5	
	Asistencia y aprobados general	0.5	
	Estudiantes con bien, regular y mal	0.5	

2.9 Plan de entregas

En el plan de entregas se establecen qué HU son agrupadas para conformar una entrega y el orden de las mismas. Este plan será el resultado de una reunión entre todos los actores del grupo de desarrollo, incluyendo el cliente permitiendo alcanzar un mayor entendimiento en la implementación del sistema.

Tabla 9. Plan de Entregas

No	Historia de usuario	Primera Iteración	Segunda Iteración	Tercera Iteración	Cuarta iteración
1	Gestionar usuario	V 1.0	Finalizado	Finalizado	Finalizado
2	Autenticar usuario	V 1.0	Finalizado	Finalizado	Finalizado

Capítulo 2: Exploración y Planificación

3	Gestionar asignatura	V 1.0	Finalizado	Finalizado	Finalizado
4	Gestionar temas	-	V 1.0	Finalizado	Finalizado
5	Gestionar grupos	-	V 1.0	Finalizado	Finalizado
6	Gestionar Actividad docente	-	V 1.0	Finalizado	Finalizado
7	Gestionar Evaluación	-	V 1.0	Finalizado	Finalizado
8	Gestionar habilidad	-	V 1.0	Finalizado	Finalizado
9	Registrar asistencia	-	V 1.0	Finalizado	Finalizado
10	Promedio de evaluaciones	-	-	V 1.0	Finalizado
11	Evaluaciones por criterios	-	-	V 1.0	Finalizado
12	Asistencia	-	-	V 1.0	Finalizado
13	Habilidades suspensas	-	-	V 1.0	Finalizado
14	Corte evaluativo	-	-	V 1.0	Finalizado
15	Resultado de las evaluaciones	-	-	-	V 1.0
16	Presentes por actividades	-	-	-	V 1.0
17	Presentes por tema	-	-	-	V 1.0
18	Estudiantes suspensos y aprobados por habilidad	-	-	-	V 1.0
19	Promedio de notas por evaluaciones	-	-	-	V 1.0
20	Asistencia y aprobados general	-	-	-	V 1.0

Capítulo 2: Exploración y Planificación

21	Estudiantes con bien, regular y mal	-	-	-	V 1.0
----	-------------------------------------	---	---	---	-------

2.10 Diseño de la base de datos

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la misma. El objetivo principal del diseño de bases de datos es generar tablas que modelan los registros en los que se guardará información. Un buen diseño de una base de datos evita redundancias en los datos almacenados y por consiguiente evita fallos de integridad, este aspecto es de vital importancia, aunque en ocasiones, por motivos de velocidad, es conveniente tener alguna información redundante siempre y cuando su uso esté justificado.

Symfony maneja la base de datos de forma independiente mediante uno de los frameworks ORM más importantes dentro de los existentes para PHP llamado Doctrine, el cual es el encargado de la comunicación con la base de datos, permitiendo un control casi total de los datos sin importar si se está hablando de MySQL, PostgreSQL, SQL server, Oracle, entre otros motores ya que la mayoría de las sentencias SQL no son generadas por el programador sino por el mismo Doctrine.

A continuación se muestra el modelo físico de la base de datos generado por la herramienta Visual Paradigm.

Capítulo 2: Exploración y Planificación

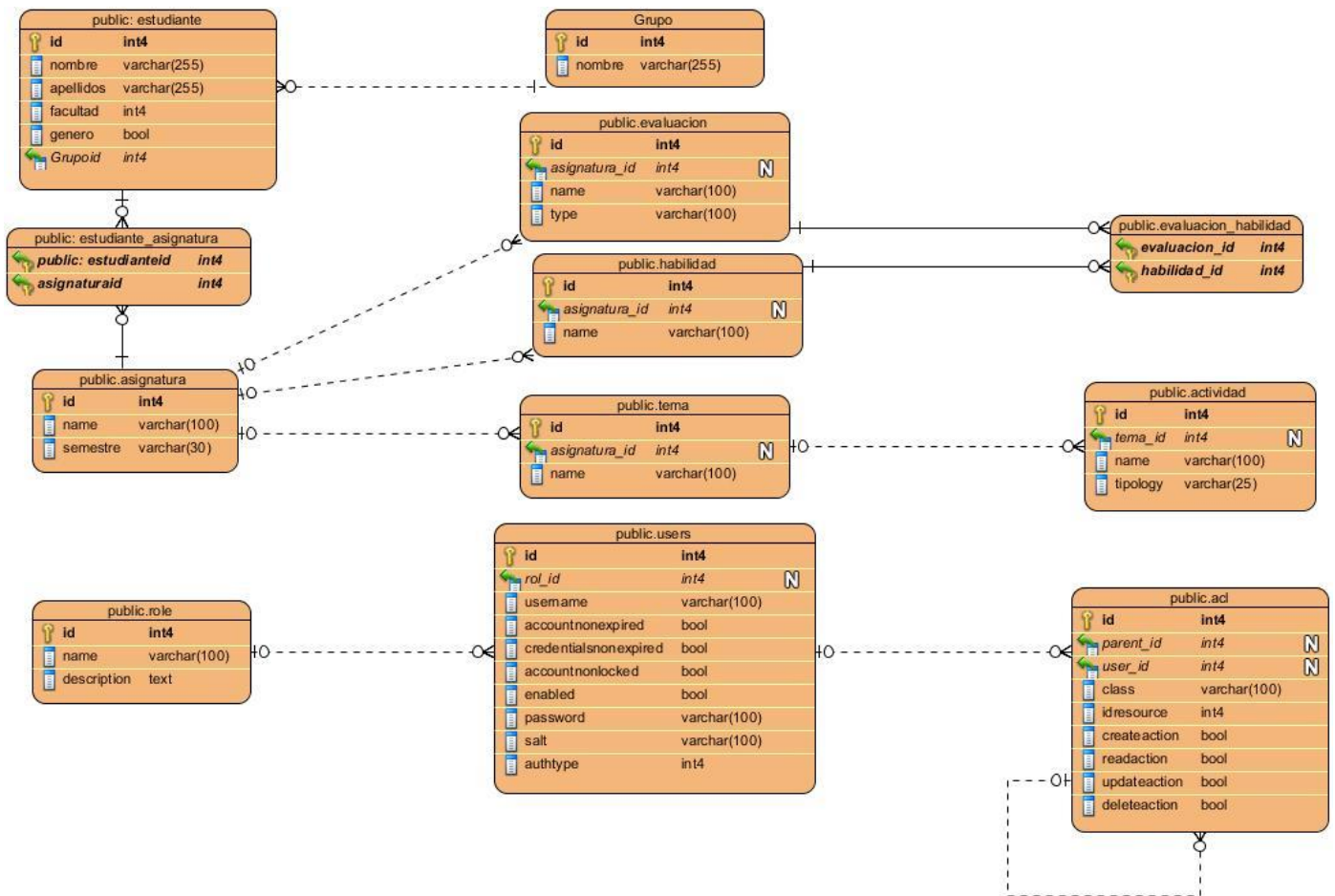


Figura 3. Modelo físico de la base de datos

2.11 Conclusiones

La metodología XP seleccionada para el desarrollo de las funcionalidades del sistema de gestión de información académica permitió realizar la propuesta del sistema y describirla, así como definir la audiencia para la cual está dirigido el mismo, destacándose como personal relacionado con el sistema al administrador, el jefe de colectivo y el profesor. Los métodos empíricos permitieron, junto a los clientes realizar el levantamiento de las Historias de Usuario y su prioridad, detectándose 21 en total, así como el orden de su implementación, en una visión futura de las funcionalidades a implementar y se instituyó el plan de entrega del proyecto, determinando un cronograma en conjunto con el cliente que especifica las entregas que deben hacerse. Una vez definidos todos estos artefactos generados por la metodología

Capítulo 2: Exploración y Planificación

seleccionada quedan sentadas las bases para el diseño e implementación del sistema el cual será validado por un proceso de pruebas posteriormente.

Capítulo 3: Implementación y pruebas del sistema

Capítulo 3 Implementación y pruebas del sistema

3.1 Introducción

En el presente capítulo se modelan los artefactos que exige la metodología ágil XP para la fase de implementación. Para el diseño de las aplicaciones, esta metodología no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC, las cuales definen relaciones y responsabilidades entre clases; se hace una descripción de las tareas de ingeniería que se obtuvieron de las historias de usuario y se muestra el diseño correspondiente a la base de datos del sistema. Además, se realizan las pruebas que es un elemento crítico para la garantía de la calidad del software y representan una revisión final de las especificaciones del diseño y de la codificación.

3.2 Estándares de codificación

Los estándares de codificación son las reglas que se van a seguir para escribir el código. XP enfatiza la comunicación de los programadores a través de dicho código, por lo cual es indispensable que se sigan ciertos estándares para la programación, los cuales mantienen el mismo legible para los miembros del equipo y así evitar que algún programador sea imprescindible para realizar cambios en alguna porción de código en un futuro. Cumpliendo de esta manera con la propiedad colectiva del código, dentro de las reglas y prácticas de la metodología (43).

A continuación se explican más detalladamente, los estándares de codificación seguidos en el desarrollo de este sistema:

- ✓ Usar una sangría de 4 espacios sin usar la tecla *tab*.
- ✓ En las declaraciones de función, la llave de apertura empieza al final de la línea de definición después de un espacio.
- ✓ Utiliza el estilo de código lowerCamel-Case en nombres de variable, función, método o argumentos.
- ✓ Los nombres de las clases comienzan con mayúscula.
- ✓ Las constantes true, false y null deben ser escritas en minúsculas.
- ✓ El número de caracteres por línea deben ser de 80 columnas aunque se permiten hasta 120.

Capítulo 3: Implementación y pruebas del sistema

- ✓ Debe haber un espacio después de cada estructura de control (if, for, foreach, while, switch, try...catch, etc.).

3.3 Tarjetas CRC

Las tarjetas CRC son técnicas de modelado creadas para ayudar a los desarrolladores de software a crear diseños de clases orientados a responsabilidades. Dichas tarjetas constan de tres secciones, nombre de la clase, responsabilidades y colaboradores. En la sección responsabilidades se listan cada una de las funciones o tareas que deben ser capaz de cumplir con un objeto de dicha clase, mientras que la sección colaboradores contiene otras clases del diseño que pueden colaborar para proveer datos o funcionalidades. Esto permite que exista un esclarecimiento continuo de la realización de cada clase a medida que esta se implementa ayudando al programador a centrarse en su trabajo. A continuación se muestran algunas de las principales clases de la aplicación con sus respectivas responsabilidades y colaboraciones.

Tabla 10. Tarjeta CRC de la clase Usuario

Clase: Usuario	
Responsabilidad	Colaboración
Adicionar usuario Modificar usuario Eliminar usuario Mostrar usuario Asignar permisos Cambiar contraseña	Asignatura

Tabla 11. Tarjeta CRC de la clase Asignatura

Clase: Asignatura	
Responsabilidad	Colaboración
Adicionar asignatura Modificar asignatura Eliminar asignatura Mostrar asignatura	

Capítulo 3: Implementación y pruebas del sistema

Tabla 12. Tarjeta CRC de la clase Actividad

Clase: Actividad	
Responsabilidad	Colaboración
Adicionar actividad Modificar actividad Eliminar actividad Mostrar actividad	Asignatura

Tabla 13. Tarjeta CRC de la clase Evaluación

Clase: Evaluación	
Responsabilidad	Colaboración
Adicionar evaluación Modificar evaluación Eliminar evaluación Mostrar evaluación Asignar habilidades	Asignatura

Tabla 14. Tarjeta CRC de la clase Grupo

Clase: Grupo	
Responsabilidad	Colaboración
Adicionar grupo Modificar grupo Eliminar grupo Mostrar grupo Eliminar estudiantes Asignar estudiantes Listar estudiantes Leer asistencia Crear asistencia Leer evaluación Construir habilidades Construir habilidades por nota Crear nota	Estudiantes Evaluación Usuario

Capítulo 3: Implementación y pruebas del sistema

Tabla 15. Tarjeta CRC de la clase Habilidad

Clase: Habilidad	
Responsabilidad	Colaboración
Adicionar habilidad Modificar habilidad Eliminar habilidad Mostrar habilidad	Asignatura Evaluación

Tabla 16. Tarjeta CRC de la clase Login

Clase: Login	
Responsabilidad	Colaboración
Login Logout	

Tabla 17. Tarjeta CRC de la clase Report

Clase: Report	
Responsabilidad	Colaboración
Asistencia Promedio de Evaluaciones Evaluaciones por debajo de criterio Asistencia por encima de 20 por ciento Asistencia cercana al 20 por ciento Estudiantes con habilidades suspensas Corte evaluativo	Grupo Asignatura Evaluación

Tabla 18. Tarjeta CRC de la clase Tema

Clase: Tema	
Responsabilidad	Colaboración
Adicionar tema Modificar tema Eliminar tema Mostrar tema	Asignatura

Capítulo 3: Implementación y pruebas del sistema

3.4 Implementación

Para llevar a cabo la correcta implementación de las HU definidas en la propuesta de solución, se deben definir por parte del equipo de desarrollo las tareas que serán llevadas a cabo. Lo anterior permite a los desarrolladores obtener un nivel de detalle más avanzado que el sugerido por las HU. A continuación se muestran algunas de las tareas de implementación realizadas por parte del equipo de desarrollo durante cada iteración, en los Anexos (Ver Anexo II) se encuentran las restantes.

Iteración 1

En esta iteración se implementan las historias de usuario de mayor prioridad según el cliente, la realización de las mismas proporciona una idea de cómo funcionará la aplicación aunque todavía esté en sus inicios.

Tabla 19. Tarea 1 de HU Gestionar usuario

Tarea de ingeniería	
Número de la tarea:1	Número de la HU:1
Nombre de la tarea: Gestión de usuario	
Tipo de tarea: Desarrollo	
Fecha de inicio: 1 de marzo del 2014	Fecha de fin: 3 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para la gestión de los usuarios como son adicionar, actualizar y eliminar.	

Tabla 20. Tarea 2 de HU Gestionar usuario

Tarea de ingeniería	
Número de la tarea:2	Número de la HU:1
Nombre de la tarea: Asignar permisos	
Tipo de tarea: Desarrollo	
Fecha de inicio: 5 de marzo del 2014	Fecha de fin: 7 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	

Capítulo 3: Implementación y pruebas del sistema

Descripción: Se les asignan los permisos a los usuarios para que puedan acceder a los recursos creados, como asignaturas creadas, temas, evaluaciones y demás.

Iteración 2

En esta iteración se realizarán las historias de usuario de prioridad media relacionadas con la parte de gestión del sistema. Una vez concluida esta iteración el sistema se encontrará en un estado casi completamente funcional.

Tabla 21. Tarea 1 de HU Gestionar grupo

Tarea de ingeniería	
Número de la tarea:1	Número de la HU:5
Nombre de la tarea: Gestión de grupo	
Tipo de tarea: Desarrollo	
Fecha de inicio: 20 de marzo del 2014	Fecha de fin: 22 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para la gestión de los grupos como son adicionar, actualizar y eliminar.	

Tabla 22. Tarea 2 de HU Gestionar grupo

Tarea de ingeniería	
Número de la tarea:2	Número de la HU:5
Nombre de la tarea: Asignar estudiantes	
Tipo de tarea: Desarrollo	
Fecha de inicio: 23 de marzo del 2014	Fecha de fin: 25 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para asignar los estudiantes a sus grupos respectivos.	

Tabla 23. Tarea 3 de HU Gestionar grupo

Tarea de ingeniería

Capítulo 3: Implementación y pruebas del sistema

Número de la tarea: 3	Número de la HU: 5
Nombre de la tarea: Eliminar estudiantes	
Tipo de tarea: Desarrollo	
Fecha de inicio: 26 de marzo del 2014	Fecha de fin: 27 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para eliminar los estudiantes de sus grupos respectivos.	

Tabla 24. Tarea 4 de HU Gestionar grupo

Tarea de ingeniería	
Número de la tarea: 4	Número de la HU: 5,9
Nombre de la tarea: Registrar asistencia	
Tipo de tarea:	
Fecha de inicio: 29 de marzo del 2014	Fecha de fin: 30 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para registrar la asistencia de los estudiantes.	

Tabla 25. Tarea 5 de HU Gestionar grupo

Tarea de ingeniería	
Número de la tarea: 5	Número de la HU: 5
Nombre de la tarea: Dar evaluación	
Tipo de tarea: Desarrollo	
Fecha de inicio: 1 de abril del 2014	Fecha de fin: 3 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para dar una evaluación a un estudiante en una actividad determinada.	

Capítulo 3: Implementación y pruebas del sistema

Iteración 3

En esta iteración se les dará cumplimiento a las historias de usuario de prioridad media. Esta es la etapa de trabajo en la que se necesita de mayor esfuerzo ya que exige mucho de los desarrolladores. Todas las historias que incidan en la lógica de la programación del sistema deben quedar implementadas.

Tabla 26. Tarea 1 de HU Corte evaluativo

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 14
Nombre de la tarea: Obtener listado para dar corte evaluativo	
Tipo de tarea: Desarrollo	
Fecha de inicio: 3 de mayo del 2014	Fecha de fin: 6 de mayo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Permite al sistema obtener un listado con las evaluaciones de los estudiantes del grupo así como el porcentaje de inasistencia y una evaluación cualitativa de bien, regular o mal propuesta por el sistema.	

Tabla 27. Tarea 2 de HU Corte evaluativo

Tarea de ingeniería	
Número de la tarea: 2	Número de la HU: 14
Nombre de la tarea: Generar reporte	
Tipo de tarea: Desarrollo	
Fecha de inicio: 5 de mayo del 2014	Fecha de fin: 6 de mayo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en formato .doc con el recorrido del estudiante para darle una evaluación cualitativa en el corte evaluativo.	

Capítulo 3: Implementación y pruebas del sistema

Iteración 4

En esta iteración se les dará cumplimiento a las historias de usuario de menor prioridad. Haciendo que el sistema esté funcionalmente listo. En esta etapa los desarrolladores terminan las últimas HU relacionadas con los reportes gráficos, una vez concluidas el sistema estará listo funcionalmente según lo pactado con el cliente.

Tabla 28. Tarea 1 de HU Obtener porciento de los cortes

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 21
Nombre de la tarea: Obtener porciento de los cortes	
Tipo de tarea: Desarrollo	
Fecha de inicio: 27 de mayo del 2014	Fecha de fin: 29 de mayo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en forma de gráficas con el porciento de los estudiantes evaluados de bien, regular y mal así como la cantidad de cada uno.	

3.5 Diagrama de despliegue

Los diagramas de despliegue muestran nodos, conexiones, componentes y objetos. Los nodos representan objetos físicos con recursos computacionales como procesadores y periféricos; pueden mostrarse como una clase o una instancia, por lo que su nombre sigue la misma sintaxis establecida para clases y objetos. Las conexiones son asociaciones de comunicación entre los nodos, y se etiquetan con un estereotipo que identifica el protocolo de comunicación o la red utilizada. En el siguiente diagrama se muestra como el usuario interactúa con el sistema, de modo que el acceso al mismo se realizará mediante un navegador haciendo uso del protocolo HTTP, una vez que el usuario se encuentre en el sistema podrá realizar peticiones al servidor de base de datos que va a estar en la misma pc que el servidor web para optimizar recursos, por último el usuario podrá acceder a un servidor de base de datos de la universidad mediante el protocolo SOAP el cual permite la comunicación y la interoperabilidad entre diversas aplicaciones web desarrolladas bajo tecnologías diferentes por medio de intercambio de datos XML. El estándar que describe esta comunicación será WSDL el cual describe la forma de comunicación, es decir,

Capítulo 3: Implementación y pruebas del sistema

los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo, el archivo usado para el sistema en este caso fue: `pregrado.uci.cu/v2/PregradoWS.wsdl`.

A continuación se representan los nodos utilizados con sus respectivas conexiones.

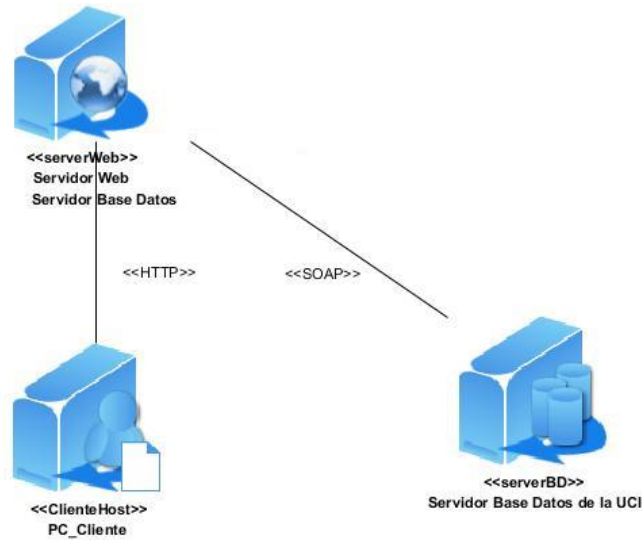


Figura 4. Diagrama de despliegue

3.6 Patrón arquitectónico

Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. No es una arquitectura como tal, sino un concepto que captura elementos esenciales de una arquitectura de software. El patrón de desarrollo Modelo-Vista-Controlador (MVC), separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres capas conceptuales (45).

Capítulo 3: Implementación y pruebas del sistema

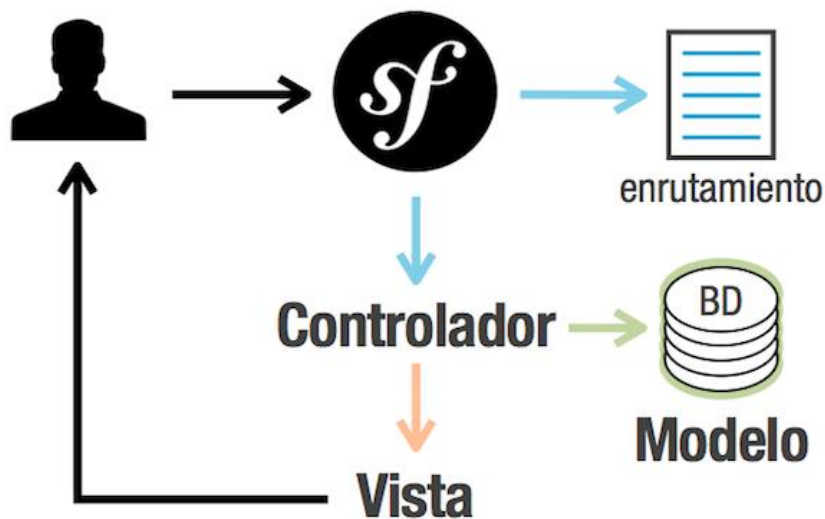


Figura 5. Patrón MVC (39)

- ✓ **Modelo:** representa la información con la que trabaja la aplicación, o sea, su lógica de negocio. Gestiona los datos solicitados por el controlador. Esto se evidencia cuando se necesita mostrar la fuente de datos.
- ✓ **Vista:** convierte el modelo en una página web que facilita al usuario interactuar con ella. Contiene los elementos asociados a la presentación de los datos y la información visual de los elementos que conformarán el reporte, encontrándose el código asociado a las interfaces, que se encargan de visualizar la información que el controlador devuelve. Para el desarrollo del sistema no se tuvo en cuenta el motor de plantillas twig que brinda symfony, para las vistas se usó JavaScript mediante la librería Extjs.
- ✓ **Controlador:** es el encargado de procesar las interacciones del usuario y ejecuta los cambios adecuados en el modelo o en la vista. La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista), lo que permite un mantenimiento más sencillo de las aplicaciones. El controlador es el encargado de aislar al modelo y a la vista de los detalles del protocolo usado para las peticiones (HTTP, consola de comandos, email). El controlador se encarga de la transferencia de información entre la vista y el modelo. Esto se aprecia cuando se envía desde la vista la configuración de cualquier entidad al controlador, este lo captura y se lo manda al modelo para que

Capítulo 3: Implementación y pruebas del sistema

lo almacene. Cabe resaltar que en esta capa se implementa el consumo de un servicio web mediante el archivo <http://pregrado.uci.cu/v2/PregradoWS.wsdl>.

3.7 Patrones de diseño

Un patrón de diseño es un conjunto de reglas que describen cómo afrontar tareas y solucionar problemas que surgen durante el proceso de desarrollo de software. En otras palabras, proporcionan una solución adaptable y escalable a problemas de desarrollo de software que están sujetos a contextos similares. Entre las familias de patrones de diseño más conocidas están los Patrones Generales de Asignación de Responsabilidad (GRASP por sus siglas en inglés), los cuales permiten la asignación de responsabilidades (46).

A continuación se mencionan los que se usaron en el diseño de la aplicación:

Creador: El sistema cuenta con una clase `AbstractController.php` que contiene las acciones que hacen al sistema funcional. En esta clase las acciones se encargan de crear los objetos de las clases que representan las entidades, evidenciando de este modo que la clase `AbstractController.php` es el creador de las entidades.

Experto: Se evidencia este patrón puesto que Doctrine es la librería externa que utiliza Symfony para realizar su capa de abstracción al modelo de datos, encapsulando toda la lógica de los datos y generando las clases con funcionalidades comunes de las entidades. Por tanto, cada clase creada por Doctrine a partir de una entidad es experta en manejar su información.

Bajo acoplamiento: Se garantiza en la arquitectura del marco de trabajo la separación de las funcionalidades en clases con baja dependencia entre ellas y definición de interfaces, lo que representa una ventaja a la hora de realizar modificaciones en algún componente sin repercusiones en el resto de la plataforma.

Alta cohesión: Una de las características principales de Symfony es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo las clases con sufijo `Controller` contienen varias funcionalidades estrechamente relacionadas entre ellas, teniendo un sentido común y un propósito único, siendo las encargadas de controlar las acciones de cada entidad a la que pertenecen y por lo tanto corresponden a la capa del Controlador dentro de la

Capítulo 3: Implementación y pruebas del sistema

arquitectura Modelo-Vista-Controlador. Este hace posible que el software sea flexible a cambios sustanciales con efecto mínimo.

Controlador: Todas las peticiones web son manejadas por un solo controlador frontal (app.php), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

3.8 Pruebas

La metodología XP anima frecuentemente en su fase de prueba a estar probando tanto como sea posible el software obtenido. Para ello en cada una de las iteraciones se lleva a cabo los casos de pruebas definidos a los que se debe someter la aplicación. Estas pruebas deben ser verificadas en cada entrega que se realice del producto en la planificación establecida.

Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todo el sistema debe pasar las pruebas unitarias antes de ser liberado o publicado. Por otra parte que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas deben ser guardados junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo (43).

Symfony2 ha optado por utilizar la librería PHPUnit, que prácticamente se ha convertido en un estándar en el mundo PHP. De esta forma, los tests unitarios de Symfony2 combinan la potencia de PHPUnit con las utilidades y facilidades proporcionadas por Symfony2 (39). Las pruebas unitarias aplicadas fueron llevadas a cabo por los programadores luego de concluir cada iteración, las mismas tienen como principal finalidad verificar el código de forma automática, se aplicaron a algunas de las funcionalidades del sistema comprobando el correcto funcionamiento de cada una de estas y obteniendo resultados satisfactorios.

A continuación se muestra un ejemplo del código de cómo fueron implementadas estas pruebas unitarias.

Capítulo 3: Implementación y pruebas del sistema

```
class AsistenciaTest extends \PHPUnit_Framework_TestCase {

    private $arregloEvaluaciones;

    public function setUp() {

        for ($index = 0; $index < 4; $index++) {...}
        for ($index = 4; $index < 8; $index++) {...}
        for ($index = 8; $index < 12; $index++) {...}
        for ($index = 12; $index < 16; $index++) {...}
        for ($index = 16; $index < 20; $index++) {...}
    }

    public function TestValidarCorte() {
        $asist = new Asistencia();
        $this->assertEquals("R", $asist->calculateCorte($this->arregloEvaluaciones, 15));
    }

    public function TestValidarCorte2() {
        $arr = null;
        for ($index = 0; $index < 15; $index++) {
            $arr[$index] = $this->arregloEvaluaciones[$index];
        }
        $asist = new Asistencia();
        $this->assertEquals("B", $asist->calculateCorte($arr, 8));
    }
}
}
```

Figura 6. Implementación de los métodos de la clase AsistenciaTest()

Al ejecutar las pruebas se observa en la consola los resultados de las mismas.

```
Time: 142 ms, Memory: 6.00Mb
<[30;42mOK (6 tests, 6 assertions)<[0m
C:\wamp\www>php phunit.phar -c app\app app\src\UCI\AppBundle\Tests
PHPUnit 4.1.0 by Sebastian Bergmann.
Configuration read from C:\wamp\www\app\app\phpunit.xml.dist
.<[41;37mF<[0m
Time: 137 ms, Memory: 6.25Mb
There was 1 failure:
1) UCI\AppBundle\Tests\Controller\AsignaturaTest::testValidarNombreak
Error validando el nombre, no debe ser nulo
Failed asserting that 0 matches expected 2.
C:\wamp\www\app\src\UCI\AppBundle\Tests
<[37;41m                                     <[0m
<[37;41mFAILURES!                                     <[0m
<[37;41mTests: 6, Assertions: 5, Failures: 1.<[0m
C:\wamp\www>
```

Figura 7. Resultados al ejecutar las prueba unitarias

Capítulo 3: Implementación y pruebas del sistema

Pruebas de aceptación

Como técnica para garantizar que los requerimientos hayan sido cumplidos y que la aplicación es realmente lo que el cliente necesita, además de asegurar su correcto funcionamiento son realizadas las pruebas de aceptación. Las cuales son creadas a partir de las historias de usuario y desde la perspectiva del cliente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento.

Estas pruebas funcionan como una caja negra, pues cada una de ellas representa una salida esperada del sistema, donde es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones acerca de las mismas. Su objetivo final consiste en lograr que los requerimientos sean cumplidos y que el sistema sea aceptable, una vez que todas las historias de usuario hayan pasado sus pruebas de aceptación se considera entonces terminada la aplicación.

A continuación se muestran las pruebas de aceptación de las principales funcionalidades implementadas para la propuesta de solución. En los anexos (Ver Anexo III) se encuentran las restantes pruebas de aceptación:

Tabla 29. Prueba 1 de HU Gestionar usuario

Caso de Prueba de Aceptación	
Código: HU1_p1	Historia de Usuario (No.1): Gestionar usuario
Nombre: Adicionar usuario	
Descripción: Se desea probar que el sistema adicione un usuario correctamente.	
Condiciones de ejecución: Se accede a la opción de adicionar usuario. No pueden existir campos vacíos. El usuario debe estar autenticado como administrador.	
Entrada: Juego de datos válidos: <ul style="list-style-type: none">✓ Nombre (Obligatorio)✓ Contraseña (Obligatorio)✓ Rol (Obligatorio)✓ Activado (Opcional)	
Resultado esperado: El sistema debe adicionar correctamente el usuario y mostrar un mensaje de confirmación: "Usuario adicionado satisfactoriamente".	
Evaluación de la prueba: Satisfactoria	

Capítulo 3: Implementación y pruebas del sistema

Tabla 30. Prueba 2 de HU Gestionar usuario

Caso de Prueba de Aceptación	
Código: HU1_p2	Historia de Usuario (No.1): Gestionar usuario
Nombre: Actualizar usuario	
Descripción: Se desea probar que el sistema actualice los datos de un usuario correctamente.	
Condiciones de ejecución: Se accede a la opción de actualizar usuario. El usuario debe estar autenticado como administrador.	
Entrada: Juego de datos válidos, se modifican los campos que se desean actualizar: <ul style="list-style-type: none">✓ Nombre (Opcional)✓ Contraseña (Opcional)✓ Rol (Opcional)✓ Activado (Opcional)	
Resultado esperado: El sistema debe actualizar correctamente el usuario y mostrar un mensaje de confirmación: "Usuario actualizado satisfactoriamente".	
Evaluación de la prueba: Satisfactoria	

Tabla 31. Prueba 3 de HU Gestionar usuario

Caso de Prueba de Aceptación	
Código: HU1_p3	Historia de Usuario (No.1): Gestionar usuario
Nombre: Eliminar usuario	
Descripción: Se desea probar que el sistema elimine un usuario satisfactoriamente.	
Condiciones de ejecución: Se accede a la opción de eliminar usuario. El usuario debe estar autenticado como administrador.	
Entrada: Seleccionar el usuario a eliminar, verificar, y confirmar la eliminación.	
Resultado esperado: El sistema debe eliminar satisfactoriamente el usuario y mostrar un mensaje de confirmación: "Usuario eliminado satisfactoriamente".	
Evaluación de la prueba: Satisfactoria	

Tabla 32. Prueba 4 de HU Gestionar usuario

Caso de Prueba de Aceptación	
Código: HU1_p4	Historia de Usuario (No.1): Gestionar usuario

Capítulo 3: Implementación y pruebas del sistema

Nombre: Insertar permisos
Descripción: El sistema debe permitir asignar los permisos seleccionados al usuario de los diferentes recursos creados. El usuario debe estar autenticado como administrador.
Condiciones de ejecución: Se accede a la opción de asignar permisos. Se marcan las casillas disponibles según los recursos creados como: asignatura, tema, evaluación, actividad y habilidad.
Entrada: Se seleccionan los recursos a los cuales tendrá acceso el usuario: <ul style="list-style-type: none"> ✓ Asignatura ✓ Tema ✓ Evaluación ✓ Actividad ✓ Habilidad
Resultado esperado: El sistema debe asignar los permisos correctamente y mostrar un mensaje de confirmación: "Acción realizada satisfactoriamente".
Evaluación de la prueba: Satisfactoria

Tabla 33. Prueba 1 de HU Gestionar grupo

Caso de Prueba de Aceptación	
Código: HU5_p1	Historia de Usuario (No.5): Gestionar grupo
Nombre: Adicionar grupo	
Descripción: Se desea probar que el sistema adicione un grupo correctamente.	
Condiciones de ejecución: No pueden existir campos vacíos.	
Entrada: Datos del grupo: <ul style="list-style-type: none"> ✓ Nombre (Obligatorio) 	
Resultado esperado: El sistema debe adicionar correctamente el grupo y mostrar un mensaje de confirmación: "Grupo adicionado satisfactoriamente".	
Evaluación de la prueba: Satisfactoria	

Tabla 34. Prueba 2 de HU Gestionar grupo

Caso de Prueba de Aceptación	
Código: HU5_p2	Historia de Usuario (No.5): Gestionar grupo

Capítulo 3: Implementación y pruebas del sistema

Nombre: Eliminar grupo
Descripción: Se desea probar que el sistema elimine un grupo satisfactoriamente.
Condiciones de ejecución: Se accede a la opción de eliminar grupo.
Entrada: Seleccionar el grupo a eliminar, verificar, y confirmar la eliminación.
Resultado esperado: El sistema debe eliminar satisfactoriamente el grupo y mostrar un mensaje de confirmación: "Grupo eliminado satisfactoriamente".
Evaluación de la prueba: Satisfactoria

Tabla 35. Prueba 3 de HU Gestionar grupo

Caso de Prueba de Aceptación	
Código: HU5_p3	Historia de Usuario (No.5): Gestionar grupo
Nombre: Adicionar estudiantes al grupo	
Descripción: Se desea probar que el sistema adicione estudiantes al grupo correctamente	
Condiciones de ejecución: La operación se realiza mediante un servicio soap con el cual se cuenta en la universidad, por tanto la estación de trabajo debe tener red en ese momento.	
Entrada: Se selecciona los estudiantes que van a conformar el grupo y se arrastran hacia la ventana del nuevo grupo.	
Resultado esperado: Los estudiantes son asignados correctamente al nuevo grupo y se muestra un mensaje de confirmación: "Estudiante adicionado satisfactoriamente".	
Evaluación de la prueba: Satisfactoria	

Tabla 36. Prueba 4 de HU Gestionar grupo

Caso de Prueba de Aceptación	
Código: HU5_p4	Historia de Usuario (No.5): Gestionar grupo
Nombre: Eliminar estudiantes al grupo	
Descripción: Se desea probar que el sistema elimine estudiantes del grupo correctamente	
Condiciones de ejecución: Se selecciona el grupo y luego se marcan los estudiantes que se desea eliminar.	
Entrada: Ir a la opción eliminar estudiantes, seleccionar los mismos y eliminarlos del grupo.	

Capítulo 3: Implementación y pruebas del sistema

Resultado esperado: Los estudiantes son eliminados correctamente del grupo así como todas las evaluaciones relacionadas al mismo y se muestra un mensaje de confirmación: “Estudiante eliminado satisfactoriamente”.
Evaluación de la prueba: Satisfactoria

Tabla 37. Prueba 5 de HU Gestionar grupo

Caso de Prueba de Aceptación	
Código: HU5_p5	Historia de Usuario (No.5): Gestionar grupo
Nombre: Registrar la asistencia de los estudiantes de un grupo	
Descripción: Se desea probar que el sistema registre la asistencia de los estudiantes correctamente.	
Condiciones de ejecución: Al registrar la asistencia no deben quedarse campos vacíos como la asignatura, el tema o la actividad.	
Entrada: Se selecciona el grupo, el tema, la actividad y fecha y luego se le activa la asistencia al estudiante.	
Resultado esperado: La asistencia se registra correctamente y se muestra un mensaje de confirmación: “Asistencia registrada satisfactoriamente”.	
Evaluación de la prueba: Satisfactoria	

Tabla 38. Prueba 6 de HU Gestionar grupo

Caso de Prueba de Aceptación	
Código: HU5_p6	Historia de Usuario (No.5): Gestionar grupo
Nombre: Dar evaluación a los estudiantes de un grupo	
Descripción: Se desea probar que el sistema permita evaluar un grupo correctamente.	
Condiciones de ejecución: No pueden haber campos vacíos como la asignatura o el nombre de la evaluación.	
Entrada: Se selecciona el grupo, luego la asignatura y el nombre de la evaluación. Se le asigna una nota a cada estudiante.	
Resultado esperado: La evaluación se registra correctamente y se muestra un mensaje de confirmación: “Estudiante evaluado satisfactoriamente”.	
Evaluación de la prueba: Satisfactoria	

Tabla 39. Prueba 1 de HU Corte evaluativo

Capítulo 3: Implementación y pruebas del sistema

Caso de Prueba de Aceptación	
Código: HU14_p1	Historia de Usuario (No.14): Corte evaluativo
Nombre: Listar corte evaluativo.	
Descripción: Se desea probar que el sistema permita emitir un corte evaluativo a partir de las evaluaciones de los estudiantes y el porcentaje de inasistencia en un período determinado.	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura.	
Entrada: Seleccionar grupo y asignatura.	
Resultado esperado: El sistema debe mostrar un resumen de las evaluaciones de los estudiantes y el porcentaje de inasistencia en la asignatura seleccionada así como la evaluación propuesta por el mismo (bien, regular o mal).	
Evaluación de la prueba: Satisfactoria	

Tabla 40. Prueba 2 de HU Corte evaluativo

Caso de Prueba de Aceptación	
Código: HU14_p2	Historia de Usuario (No.14): Corte evaluativo
Nombre: Generar reporte del corte evaluativo	
Descripción: Se desea probar que el sistema genere el reporte con el corte evaluativo de los estudiantes.	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura y luego acceder a la opción generar reporte.	
Entrada: Seleccionar grupo y asignatura.	
Resultado esperado: El sistema debe exportar el listado a formato .doc para luego poder imprimirlo.	
Evaluación de la prueba: Satisfactoria	

Tabla 41. Prueba 1 de HU Estudiantes con bien, regular y mal

Caso de Prueba de Aceptación	
Código: HU21_p1	Historia de Usuario (No.21): Estudiantes con bien, regular y mal

Capítulo 3: Implementación y pruebas del sistema

Nombre: Generar reporte gráfico con la evaluación del corte.
Descripción: Se desea probar que el sistema genere un reporte gráfico con la cantidad de estudiantes evaluados de bien, regular o mal, así como el porcentaje que representan respecto al total de estudiantes.
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura y luego acceder a la opción generar reporte.
Entrada: Seleccionar grupo y asignatura.
Resultado esperado: El sistema debe mostrar reporte gráfico con la cantidad de estudiantes evaluados de bien, regular y mal así como el porcentaje que representan en el grupo.
Evaluación de la prueba: Satisfactoria

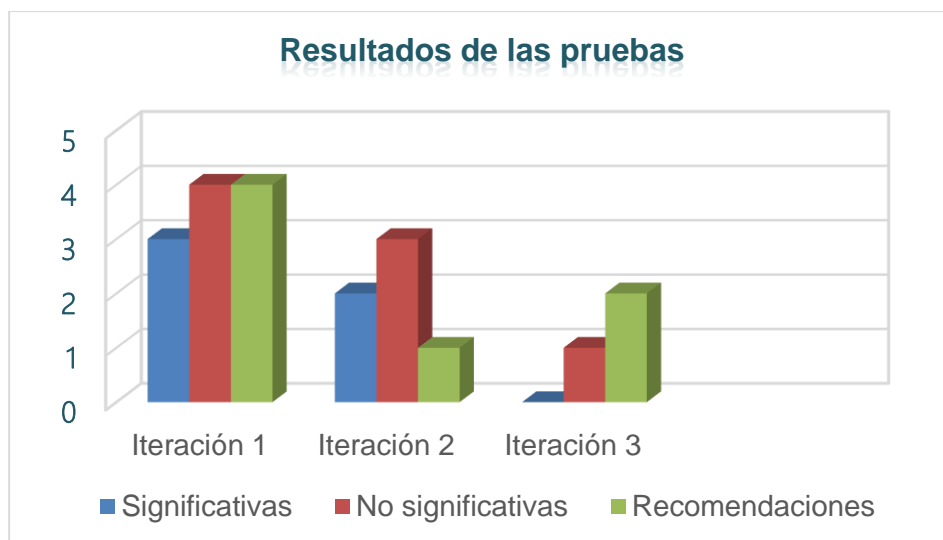


Figura 8. Resultado de las Pruebas

Al concluir las pruebas se detectaron algunas no conformidades que fueron corregidas luego de cada entrega al cliente. Dentro de las no significativas se detectaron errores ortográficos como: omisiones de tildes, cambio de mayúscula por minúscula y aceptar valores numéricos donde se esperaban letras, por otra parte dentro de las significativas se detectaron errores de validación a los cuales posteriormente se les dio el tratamiento adecuado.

Capítulo 3: Implementación y pruebas del sistema

En resumen con la realización de dichas pruebas se verificó el funcionamiento completo de la aplicación con la intención de satisfacer todas las peticiones del cliente descritas antes en las historias de usuario.

3.9 Conclusiones

El diseño propuesto en el capítulo anterior permitió desarrollar las funcionalidades descritas a partir de las herramientas, lenguajes, metodologías y tecnologías seleccionadas en el capítulo 1, solucionando de esta manera las insuficiencias descritas en la situación problemática planteada en la introducción de esta investigación.

En el transcurso de este capítulo se detallaron las tareas CRC brindando claridad en aspectos como las principales funcionalidades que presentan las clases así como la relación existente entre ellas. Mediante el uso los patrones de diseño y las tareas de ingeniería se garantizó una mejor estructuración a la hora de implementar las HU detectadas. Las pruebas realizadas al sistema a lo largo de todo el ciclo de desarrollo del mismo garantizaron su calidad y buen funcionamiento, dando por concluida la propuesta de solución de la aplicación a implementar simultáneamente con el fin del capítulo actual.

Conclusiones generales

Después de desarrollar el presente trabajo y analizar los resultados obtenidos, las conclusiones esenciales a las que se arriban son las siguientes:

- ✓ Los métodos científicos utilizados permitieron desarrollar los conceptos y teorías que sustentan la investigación para el desarrollo de la solución.
- ✓ La selección de la metodología de desarrollo, lenguajes de programación, tecnologías y herramientas posibilitaron al equipo de desarrollo cumplir con el objetivo general de la investigación.
- ✓ La implementación del sistema posibilitó obtener un producto funcional capaz de satisfacer las necesidades del cliente.
- ✓ La validación de la solución demostró el correcto funcionamiento del sistema.

Recomendaciones

Como parte del proceso investigativo llevado a cabo se recomienda:

- ✓ A la Universidad de las Ciencias Informáticas generalizar el sistema de apoyo a la gestión de la información académica que se desarrolló para la facultad 4, al resto de sus facultades con el objetivo de verificar su viabilidad e incorporar las correcciones pertinentes en caso de que sea necesario.
- ✓ A los profesores y estudiantes de la facultad 4 continuar el desarrollo del sistema con el objetivo de incorporar nuevas funcionalidades, una vez que los profesores interactúen y se familiaricen con el mismo.

Referencias Bibliográficas

1. **Medina Quintero, José Melchor.** Evaluación del impacto de los sistemas de información en el desempeño individual del usuario . 2005.
2. **Woodman, L.** Information Management in Large Organizations. Information Management from Estrategies to Action. 1985.
3. **Luan, J.** Data Mining and Knowledgw Management in Higher Education - Applications. Annual Forum for the Association for Institucional Research. Toronto : s.n., 2002.
4. **Wilger , A K y Massy, W F.** Improving Productivity: What Faculty Think About It - And It`s Effect on Quality Change:The Magazine of Higher Learning.
5. **Rodríguez, A y Haedo, Y.** Análisis Multivariado del Proceso de Aprendizaje en el Primer Año de Ingeniería. 14 Coonvención Científica de Ingeniería y Arquitectura. La Habana, Cuba : s.n.
6. **Lucas, H.C.** Sistemas de Información. 1897.
7. **Andreu, R., Ricart J. E. Y Valor.** Estrategia y Sistemas de Información. 1991.
8. **Trasobares, Alejandro Hernández.** Hernández. Los sistemas de información: evolución y desarrollo. Proyecto social: Revista de relaciones laborales. 2003. ISSN 1133-3189.
9. **Rodríguez, A. y Márquez, A.** Técnicas de organización y análisis de sistemas: organización de los servicios informáticos. Madrid: McGraw-Hill : s.n., 1993.
10. **Pablo, I.** El reto informático: la gestión de la información en la empresa. Madrid: Pirámide : s.n., 1989.
11. **Ponjuán, G.** Gestión de la información en las organizaciones: principios, conceptos y aplicaciones. Santiago de Chile. 1998.
12. —. Gestión de la Información: dimensiones e implementación para el éxito organizacional. Rosario, Ediciones Nuevo Paradigma. 2004.

13. **Paez, U.** ¿Qué es la gestión de Información? La Gestión de Información. Estudio del tema en los Congresos INFO. Tesis de Licenciatura. Cuba, Departamento de Bibliotecología y Ciencia de la Información, Universidad de La Habana. s.l. : INFOLAC, 2005.
14. **Barroso, J.** Propuesta de pautas para el diseño de un Sistema de Gestión de Información. Tesis Doctoral. Cuba, Departamento de Bibliotecología y Ciencia de la Información. Universidad de La Habana : s.n., 2009.
15. The British Standards Institution. bsi. [En línea] 2013. [Citado el: 13 de enero de 2014.] <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion>.
16. **Vargas, Belamiro Caro.** Guía pedagógica del seminario: Gestión Académica. [En línea] [Citado el: 20 de Mayo de 2014.] <http://api.ning.com/files/zniINCG9rk6jPjqia8NjAnxQokk0swCI0TF7HP2U9q5cF4rjEPpTnr3V9ZEKZNKiiTAqhNzVttGaDbHBYoReXyOi0GyWQPT/GUIASEMINARIOgestinacadmica.pdf>.
17. **CENACAD.** CENACAD. [En línea] CISE, 2006. [Citado el: 10 de enero de 2014.] <http://www.cenacad.espol.edu.ec/>.
18. **SIGA.** [En línea] Universidad Técnica Federico Santa María. [Citado el: 10 de enero de 2014.] <https://www.siga.usm.cl>.
19. **SIU.** [En línea] Universidad de San Juan (Argentina). [Citado el: 10 de enero de 2014.] http://www.facso.unsj.edu.ar/gestion_academica.php.
20. **XAUGE.** SGU. [En línea] Universidad de la Ciencias Informáticas (UCI). [Citado el: 10 de enero de 2014.] <https://gestionuniversitaria.uci.cu/base/comun/portada>.
21. **Valencia, D.D.S.I.Y.C.D.** Proceso de desarrollo de Software. [En línea] Universidad Politécnica de Valencia. [Citado el: 9 de enero de 2014.] <http://www.upv.es/entidades/DSIC/index.html>.
22. **Piattini, P.** Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. 1996.

Referencias bibliográficas

23. **Calderón, A.** Metodologías Ágiles. Universidad Nacional de Trujillo. Facultad de Ciencias Físicas y Matemáticas (Escuela de Informática) : s.n., 2007.
24. **Beck, Kent.** Extreme Programming Explained Primera Edición . Addison-Wesley Pub Co : s.n., 1999.
25. **Jeffries, R., Anderson, A., Hendrickson, C.** Extreme Programming Installed.
26. **Beck, K.** Extreme Programming Explained, Addison-Wesley The XP Series. 2000.
27. **Letelier, Patricio y Penadés, Ma.Carmen.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] www.cyta.com.ar/ta0502/v5n2a1.htm.
28. **Pressman, Roger S.** Ingeniería de Software, un enfoque práctico. Quinta edición. s.l. : McGraw-Hill Companies, 2002. ISBN: 8448132149.
29. Microsoft Solution Framework (MSF). [En línea] Universidad de Oriente (Monagas, Venezuela). [Citado el: 15 de enero de 2014.] http://wiki.monagas.udo.edu.ve/index.php/Microsoft_Solution_Framework_%28MSF%29.
30. IBM Rational Unified Process (RUP). [En línea] [Citado el: 9 de enero de 2014.] <http://www-01.ibm.com/software/rational/rup/00000>.
31. PROGRAMACION EN CASTELLANO, ¿Por qué elegir PHP? [En línea] 13 de Febrero de 2014. http://www.programacion.com/articulo/por_que_elegir_php_143.00000.
32. Lenguaje Html. [En línea] [Citado el: 15 de Mayo de 2014.] <http://www.buenastareas.com/ensayos/Lenguaje-Html/2757842.html>.
33. **Eguiluz, J.** ¿Qué es JavaScript? [En línea] 2012. [Citado el: 13 de enero de 2014.] <http://www.librosweb.es/javascript/index.html>.
34. Guía breve de CSS. [En línea] [Citado el: 12 de enero de 2014.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.
35. Definición de XML. [En línea] [Citado el: 28 de mayo de 2014.] <http://definicion.de/xml/> .

Referencias bibliográficas

36. IBM Knowledge Center. [En línea] [Citado el: 28 de mayo de 2014.] http://www-01.ibm.com/support/knowledgecenter/SSKM8N_7.0.0/com.ibm.etools.mft.doc/ac34640_.htm?lang=es00000.
37. Protocolo simple de acceso a objetos (SOAP). Desarrolloweb.com. [En línea] [Citado el: 28 de mayo de 2014.] <http://www.desarrolloweb.com/articulos/1557.php?manual=54> .
38. Oracle Corporation and/or its affiliates. Netbeans. [En línea] [Citado el: 13 de enero de 2014.] http://netbeans.org/community/releases/68/index_es.html.
39. **Eguiluz, J.** Desarrollo web ágil con Symfony2. 2013.
40. Apache. [En línea] [Citado el: 13 de enero de 2014.] <http://www.apache.org/> .
41. **Date, C.J.** Introducción a los sistemas de bases de datos. Séptima edición . 2001. ISBN: 968-444-419-2.
42. **Pecos, Daniel.** PostGreSQL vs. MySQL. [En línea] [Citado el: 13 de enero de 2014.] http://danielpecos.com/docs/mysql_postgres/x15.html.
43. **Joskowicz, José.** Reglas y Prácticas en eXtremeProgramming. España : s.n., 2008.
44. **Carvajal Riola, José Carlos.** Metodologías ágiles: Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial. 2008.
45. Patron_Modelo_Vista_Controlador. [En línea] [Citado el: 14 de enero de 2014.] <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador>.
46. **Mariñán Pérez, Martín.** Patrones de Diseño. 2009.
47. Info@tltas. [En línea] Inder, 2008. [Citado el: 12 de enero de 2014.] <http://www.inder.cu/FichaAtleta/home>.
48. **Martínez López, F.J.** Evolución y paradigma de los Sistemas de Información: Huelva. Universidad de Huelva : s.n., 1995.

Anexo I

Historias de usuario

Tabla 42. HU Autenticar usuario

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Autenticar usuario
Usuario: Administrador, Jefe de Colectivo, Profesor	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración Asignada: 1	Puntos Estimados: 0.5
<p>Descripción: El usuario podrá autenticarse para acceder a algunas opciones del sistema según los permisos otorgados.</p> <p>Datos para autenticarse:</p> <ul style="list-style-type: none"> ✓ Nombre del Usuario (Obligatorio). ✓ Contraseña (Obligatorio). <p>Observaciones: Para que un usuario pueda autenticarse en el sistema debe primero ser registrado por el administrador.</p>	

Tabla 43. HU Gestionar asignatura

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Gestionar asignatura
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Iteración Asignada: 1	Puntos Estimados: 1
<p>Descripción: Permite crear, modificar y eliminar las asignaturas en la Base de Datos. En caso de agregar los permisos solo lo asigna el rol que posea permisos de Administrador.</p> <p>De una asignatura se registran los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Nombre (Obligatorio) 	

- ✓ Semestre (Obligatorio)
- ✓ Usuario editor (Obligatorio)

Tabla 44. HU Gestionar tema

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Gestionar tema
Usuario: Administrador, Jefe de Colectivo	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Iteración Asignada: 2	Puntos Estimados: 0.5
<p>Descripción: Permite crear, modificar y eliminar los temas de las asignaturas en la Base de Datos. En caso de agregar los permisos solo lo asigna el usuario que posea rol de Administrador.</p> <p>De un tema se registran los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Nombre (Obligatorio) ✓ Asignatura (Obligatorio) 	

Tabla 45. HU Gestionar Actividad docente

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Gestionar Actividad docente
Usuario: Administrador, Jefe de Colectivo	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Iteración Asignada: 2	Puntos Estimados: 0.5
<p>Descripción: Permite crear, modificar y eliminar las actividades docentes de las asignaturas.</p> <p>De una actividad docente se registran los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Nombre (Obligatorio) ✓ Asignatura (Obligatorio) ✓ Tema (Obligatorio) ✓ Tipología (Obligatorio) 	

Tabla 46. HU Gestionar evaluación

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Gestionar evaluación
Usuario: Administrador, Jefe de Colectivo	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Iteración Asignada: 2	Puntos Estimados: 0.5
<p>Descripción: El sistema le permitirá al usuario crear, modificar y eliminar las evaluaciones de las actividades docentes de los estudiantes.</p> <p>De una evaluación docente se registran los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Nombre (Obligatorio) ✓ Asignatura (Obligatorio) ✓ Tipo (Obligatorio) 	

Tabla 47. HU Gestionar habilidad

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Gestionar habilidad
Usuario: Administrador, Jefe de Colectivo	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Iteración Asignada: 2	Puntos Estimados: 0.5
<p>Descripción: Permite crear, modificar y eliminar las evaluaciones de las habilidades docentes de las evaluaciones.</p> <p>De una habilidad docente se registran los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Nombre (Obligatorio) ✓ Asignatura (Obligatorio) 	

Tabla 48. HU Mostrar asistencia

Historia de Usuario	
Número: 9	Nombre Historia de Usuario: Mostrar asistencia
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio

Iteración Asignada: 3	Puntos Estimados: 0.5
Descripción: Permite mostrar la asistencia de los estudiantes de un grupo en determinada asignatura. Puede ser exportado a un documento en formato “.doc”.	

Tabla 49. Asistencia por encima del 20 por ciento

Historia de Usuario	
Número: 10	Nombre Historia de Usuario: Asistencia por encima del 20 por ciento
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Iteración Asignada: 3	Puntos Estimados: 0.5
Descripción: Permite mostrar los estudiantes de un grupo que tengan el porcentaje de inasistencia por encima del 20 por ciento. Puede ser exportado a un documento en formato “.doc”.	

Tabla 50. HU Promedio de evaluaciones

Historia de Usuario	
Número: 11	Nombre Historia de Usuario: Promedio de evaluaciones
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Iteración Asignada: 3	Puntos Estimados: 1
Descripción: Permite al profesor visualizar un registro de todas las evaluaciones del grupo, y promedio de la brigada. Puede ser exportado a un documento en formato “.doc”.	

Tabla 51. HU Evaluaciones por criterio

Historia de Usuario	
Número: 12	Nombre Historia de Usuario: Evaluaciones por criterio
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio

Iteración Asignada: 3	Puntos Estimados: 0.5
Descripción: Permite al profesor visualizar los estudiantes con notas por debajo de un criterio establecido (3,4 ó 5). Puede ser exportado a un documento en formato “.doc”.	

Tabla 52. HU Habilidades suspensas

Historia de Usuario	
Número: 13	Nombre Historia de Usuario: Habilidades suspensas
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Medio
Iteración Asignada: 3	Puntos Estimados: 0.5
Descripción: Muestra un listado de los estudiantes con habilidades suspensas en las evaluaciones de la asignatura. Puede ser exportado a un documento en formato “.doc”.	

Tabla 53. HU Resultado de las evaluaciones

Historia de Usuario	
Número: 15	Nombre Historia de Usuario: Resultado de las evaluaciones
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Medio
Iteración Asignada: 4	Puntos Estimados: 0.5
Descripción: Genera un reporte gráfico de las evaluaciones de los estudiantes en una asignatura determinada.	

Tabla 54. HU Estudiantes presentes por actividades

Historia de Usuario	
Número: 16	Nombre Historia de Usuario: Estudiantes presentes por actividades
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Medio
Iteración Asignada: 4	Puntos Estimados: 0.5

Descripción: Genera un reporte gráfico con el porcentaje de asistencia de los estudiantes por cada una de las actividades realizadas. Se le pasa el grupo y la asignatura.

Tabla 55. HU Estudiantes presentes por tema

Historia de Usuario	
Número: 17	Nombre Historia de Usuario: Estudiantes presentes por tema
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Medio
Iteración Asignada: 4	Puntos Estimados: 0.5
Descripción: Genera un reporte gráfico con el porcentaje de asistencia de los estudiantes por cada uno de las temas de la asignatura. Se le pasa el grupo y la asignatura.	

Tabla 56. HU Estudiantes suspensos y aprobados por habilidad

Historia de Usuario	
Número: 18	Nombre Historia de Usuario: Estudiantes suspensos y aprobados por habilidad
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Medio
Iteración Asignada: 4	Puntos Estimados: 0.5
Descripción: Genera un reporte gráfico con la cantidad de estudiantes suspensos en cada una de las habilidades. Se le pasa el grupo y la asignatura.	

Tabla 57. HU Promedio de notas por evaluaciones

Historia de Usuario	
Número: 19	Nombre Historia de Usuario: Promedio de notas por evaluaciones
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Medio
Iteración Asignada: 4	Puntos Estimados: 0.5
Descripción: Genera un reporte gráfico con el comportamiento de los estudiantes en las evaluaciones realizadas. Se	

le pasa el grupo y la asignatura.

Tabla 58. HU Asistencia y aprobados general

Historia de Usuario	
Número: 20	Nombre Historia de Usuario: Asistencia y aprobados general
Usuario: Administrador, Jefe de Colectivo y Profesor	
Prioridad en Negocio: Baja	Riesgo en Desarrollo: Medio
Iteración Asignada: 4	Puntos Estimados: 0.5
Descripción: Genera un reporte gráfico con el comportamiento de los estudiantes en cuanto a las evaluaciones realizadas y a la asistencia. Se le pasa el grupo y la asignatura.	

Anexo II

Tareas de Implementación

Iteración 1

Tabla 59. Tarea 1 de HU Autenticar usuario

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 2
Nombre de la tarea: Acceso al sistema	
Tipo de tarea: Desarrollo	
Fecha de inicio: 8 de marzo del 2014	Fecha de fin: 9 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se realiza la validación para que el usuario pueda acceder al sistema.	

Tabla 60. Tarea 2 de HU Autenticar usuario

Tarea de ingeniería	
Número de la tarea: 2	Número de la HU: 2
Nombre de la tarea: Salir del sistema	
Tipo de tarea: Desarrollo	
Fecha de inicio: 9 de marzo del 2014	Fecha de fin: 9 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se define e implementa la acción para que el usuario pueda salir del sistema.	

Tabla 61. Tarea 3 de HU Autenticar usuario

Tarea de ingeniería	
Número de la tarea: 3	Número de la HU: 2
Nombre de la tarea: Cambiar contraseña	
Tipo de tarea: Desarrollo	
Fecha de inicio: 9 de marzo del 2014	Fecha de fin: 10 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se define e implementa la acción para que el usuario pueda cambiar su contraseña.	

Tabla 62. Tarea 1 de HU Gestionar asignatura

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 3
Nombre de la tarea: Gestión de asignatura	
Tipo de tarea: Desarrollo	
Fecha de inicio: 11 de marzo del 2014	Fecha de fin: 16 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	

Descripción: Se definen e implementan las acciones que se realizarán para la gestión de las asignaturas como son adicionar, actualizar y eliminar.

Iteración 2

En esta iteración se realizarán las historias de usuario de prioridad media relacionadas con la parte de gestión del sistema. Una vez concluida esta iteración el sistema se encontrará en un estado casi completamente funcional.

Tabla 63. Tarea 1 de HU Gestionar tema

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU:4
Nombre de la tarea: Gestión de tema	
Tipo de tarea: Desarrollo	
Fecha de inicio: 17 de marzo del 2014	Fecha de fin: 19 de marzo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para la gestión de los temas como son adicionar, actualizar y eliminar.	

Tabla 64. Tarea 1 de HU Gestionar actividad docente

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 6
Nombre de la tarea: Gestión de actividad docente	
Tipo de tarea: Desarrollo	
Fecha de inicio: 4 de abril del 2014	Fecha de fin: 6 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para la gestión de las actividades docentes como son adicionar, actualizar y eliminar.	

Tabla 65. Tarea 1 de HU Gestionar evaluación

Tarea de ingeniería

Número de la tarea: 1	Número de la HU: 7
Nombre de la tarea: Gestión de evaluación	
Tipo de tarea: Desarrollo	
Fecha de inicio: 7 de abril del 2014	Fecha de fin: 9 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para la gestión de las evaluaciones como son adicionar, actualizar y eliminar.	

Tabla 66. Tarea 2 de HU Gestionar evaluación

Tarea de ingeniería	
Número de la tarea: 2	Número de la HU: 7
Nombre de la tarea: Asignación de habilidades	
Tipo de tarea: Desarrollo	
Fecha de inicio: 9 de abril del 2014	Fecha de fin: 10 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para asignar las habilidades a las evaluaciones de los estudiantes.	

Tabla 67. Tarea 1 de HU Gestionar evaluación

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 8
Nombre de la tarea: Gestión de habilidades	
Tipo de tarea: Desarrollo	
Fecha de inicio: 11 de abril del 2014	Fecha de fin: 13 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Se definen e implementan las acciones que se realizarán para la gestión de las habilidades como son adicionar, actualizar y eliminar.	

Iteración 3

En esta iteración se les dará cumplimiento a las historias de usuario de prioridad media. Esta es la etapa de trabajo en la que se necesita de mayor esfuerzo ya que exige mucho de los desarrolladores. Todas las historias que incidan en la lógica de la programación del sistema deben quedar implementadas.

Tabla 68. Tarea 1 de HU Mostrar asistencia

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 9
Nombre de la tarea: Visualizar asistencia	
Tipo de tarea: Desarrollo	
Fecha de inicio: 23 de abril del 2014	Fecha de fin: 24 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Permite al sistema visualizar la asistencia de un grupo en determinada asignatura, así como sus respectivos porcentos (ausente y presente).	

Tabla 69. Tarea 2 de HU Mostrar asistencia

Tarea de ingeniería	
Número de la tarea: 2	Número de la HU: 9
Nombre de la tarea: Generar reporte de asistencia	
Tipo de tarea: Desarrollo	
Fecha de inicio: 26 de abril del 2014	Fecha de fin: 27 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en formato .doc con la asistencia un grupo en determinada asignatura, así como sus respectivos porcentos (ausente y presente).	

Tabla 70. Tarea 1 de HU Asistencia por encima de 20 porciento

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 10
Nombre de la tarea: Asistencia por encima de 20 porciento	
Tipo de tarea: Desarrollo	
Fecha de inicio: 25 de abril del 2014	Fecha de fin: 27 de abril del 2014

Programador responsable: Ernesto Mastrapa Serrano
Descripción: Permite al sistema obtener los estudiantes con la asistencia por encima del 20 por ciento.

Tabla 71. Tarea 2 de HU Asistencia por encima de 20 por ciento

Tarea de ingeniería	
Número de la tarea: 2	Número de la HU: 10
Nombre de la tarea: Generar reporte de asistencia por encima de 20 por ciento	
Tipo de tarea: Desarrollo	
Fecha de inicio: 26 de abril del 2014	Fecha de fin: 27 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en formato .doc con la asistencia de los estudiantes por encima del 20 por ciento.	

Tabla 72. Tarea 1 de HU Promedio de evaluaciones

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 10
Nombre de la tarea: Promedio de evaluaciones	
Tipo de tarea: Desarrollo	
Fecha de inicio: 14 de abril del 2014	Fecha de fin: 16 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite visualizar una tabla con todas las evaluaciones del grupo y su promedio respectivo	

Tabla 73. Tarea 2 de HU Promedio de evaluaciones

Tarea de ingeniería	
Número de la tarea: 2	Número de la HU: 10
Nombre de la tarea: Generar reporte	
Tipo de tarea: Desarrollo	

Fecha de inicio: 17 de abril del 2014	Fecha de fin: 18 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en formato .doc con el promedio de evaluaciones de los estudiantes.	

Tabla 74. Tarea 1 de HU Evaluaciones por criterio

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 11
Nombre de la tarea: Obtener evaluaciones por criterio	
Tipo de tarea: Desarrollo	
Fecha de inicio: 19 de abril del 2014	Fecha de fin: 20 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener los estudiantes con evaluaciones por debajo de un criterio especificado (3, 4 ó 5).	

Tabla 75. Tarea 2 de HU Evaluaciones por criterio

Tarea de ingeniería	
Número de la tarea: 2	Número de la HU: 11
Nombre de la tarea: Generar reporte	
Tipo de tarea: Desarrollo	
Fecha de inicio: 21 de abril del 2014	Fecha de fin: 22 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en formato .doc con el promedio de evaluaciones de los estudiantes por debajo del criterio especificado.	

Tabla 76. Tarea 1 de HU Habilidades suspensas

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 13
Nombre de la tarea: Obtener estudiantes con habilidades suspensas	

Tipo de tarea: Desarrollo	
Fecha de inicio: 28 de abril del 2014	Fecha de fin: 30 de abril del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: Permite al sistema visualizar un listado de los estudiantes que tienen habilidades suspensas en las evaluaciones que se le han realizado.	

Tabla 77. Tarea 2 de HU Habilidades suspensas

Tarea de ingeniería	
Número de la tarea: 2	Número de la HU: 13
Nombre de la tarea: Generar reporte	
Tipo de tarea: Desarrollo	
Fecha de inicio: 1 de mayo del 2014	Fecha de fin: 2 de mayo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en formato .doc con los estudiantes que tienen habilidades suspensas en las evaluaciones que se le han realizado	

Iteración 4

En esta iteración se les dará cumplimiento a las historias de usuario de menor prioridad. Haciendo que el sistema esté funcionalmente listo. En esta etapa los desarrolladores terminan las últimas HU relacionadas con los reportes gráficos, una vez concluidas el sistema estará listo funcionalmente según lo pactado con el cliente.

Tabla 78. Tarea 1 de HU Obtener resultado de las evaluaciones

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 15
Nombre de la tarea: Obtener resultado de las evaluaciones	
Tipo de tarea: Desarrollo	
Fecha de inicio: 10 de mayo del 2014	Fecha de fin: 12 de mayo del 2014

Programador responsable: Ernesto Mastrapa Serrano
Descripción: El sistema permite obtener un reporte en forma de gráficas los resultados obtenidos por los estudiantes de un grupo en las evaluaciones hechas hasta el momento.

Tabla 79. Tarea 1 de HU Obtener presentes por actividades

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 16
Nombre de la tarea: Obtener presentes por actividades	
Tipo de tarea: Desarrollo	
Fecha de inicio: 13 de mayo del 2014	Fecha de fin: 15 de mayo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en forma de gráficas con el porcentaje de estudiantes presentes por cada una de las actividades realizadas hasta el momento.	

Tabla 80. Tarea 1 de HU Obtener presentes por tema

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 17
Nombre de la tarea: Obtener presentes por tema	
Tipo de tarea: Desarrollo	
Fecha de inicio: 16 de mayo del 2014	Fecha de fin: 18 de mayo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en forma de gráficas con el porcentaje de estudiantes presentes por cada uno de los temas de una asignatura.	

Tabla 81. Tarea 1 de HU Obtener suspensos y aprobados por habilidad

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 18
Nombre de la tarea: Obtener suspensos y aprobados por habilidad	
Tipo de tarea: Desarrollo	

Fecha de inicio: 19 de mayo del 2014	Fecha de fin: 21 de mayo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en forma de gráficas con la cantidad de estudiantes aprobados y suspensos por habilidades evaluadas.	

Tabla 82. Tarea 1 de HU Obtener promedio de notas por evaluación

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 19
Nombre de la tarea: Obtener promedio de notas por evaluación	
Tipo de tarea: Desarrollo	
Fecha de inicio: 21 de mayo del 2014	Fecha de fin: 23 de mayo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en forma de gráficas con el promedio de las evaluaciones en general de todo el grupo en una asignatura	

Tabla 83. Tarea 1 de HU Obtener la asistencia y el promedio de aprobados

Tarea de ingeniería	
Número de la tarea: 1	Número de la HU: 20
Nombre de la tarea: Obtener la asistencia y el promedio de aprobados	
Tipo de tarea: Desarrollo	
Fecha de inicio: 24 de mayo del 2014	Fecha de fin: 26 de mayo del 2014
Programador responsable: Ernesto Mastrapa Serrano	
Descripción: El sistema permite obtener un reporte en forma de gráficas con el promedio de todas las evaluaciones mayores que dos y la asistencia de un grupo en una asignatura determinada.	

Anexo III

Pruebas de aceptación

Tabla 84. Prueba 1 de HU Autenticar usuario

Caso de Prueba de Aceptación	
Código: HU2_p1	Historia de Usuario (No.2): Autenticar usuario
Nombre: Entrada al sistema con datos válidos	
Descripción: Se desea probar que el sistema de acceso a un usuario.	
Condiciones de ejecución: Juego de datos válidos de autenticación de usuario.	
Entrada: El sistema comprueba que los datos estén correctos para dar acceso al usuario. <ul style="list-style-type: none"> ✓ Usuario (Obligatorio) ✓ Contraseña (Obligatorio) 	
Resultado esperado: Entrada a sistema del usuario esperado con el rol que le corresponde.	
Evaluación de la prueba: Satisfactoria	

Tabla 85. Prueba 2 de HU Autenticar usuario

Caso de Prueba de Aceptación	
Código: HU2_p2	Historia de Usuario (No.2): Autenticar usuario
Nombre: Entrada al sistema con datos erróneos	
Descripción: Se desea probar que el sistema no de acceso a un usuario con datos erróneos.	
Condiciones de ejecución: Juego de datos incorrectos o incompletos. <ul style="list-style-type: none"> ✓ Usuario (Opcional) ✓ Contraseña (Opcional) 	
Entrada: Datos incorrectos. El sistema comprueba que los datos estén correctos para dar acceso al usuario.	
Resultado esperado: El sistema muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 86. Prueba 3 de HU Autenticar usuario

Caso de Prueba de Aceptación	
Código: HU2_p3	Historia de Usuario (No.2): Autenticar usuario
Nombre: Salir del sistema	
Descripción: Se desea probar que el sistema permita cerrar la sesión (deslogearse).	

Condiciones de ejecución: Se accede a la opción de salir del sistema.
Entrada: Deslogearse.
Resultado esperado: El sistema finaliza la sesión y redirige al usuario a la página de autenticación.
Evaluación de la prueba: Satisfactoria

Tabla 87. Prueba 4 de HU Autenticar usuario

Caso de Prueba de Aceptación	
Código: HU2_p4	Historia de Usuario (No.2): Autenticar usuario
Nombre: Cambiar contraseña	
Descripción: Se desea probar que el sistema permita cambiar la contraseña del usuario logueado.	
Condiciones de ejecución: Se accede a la opción de cambiar contraseña.	
Entrada: Hacer clic en el botón de cambiar contraseña.	
Resultado esperado: El sistema le permite al usuario cambiar la contraseña.	
Evaluación de la prueba: Satisfactoria	

Tabla 88. Prueba 1 de HU Gestionar asignatura

Caso de Prueba de Aceptación	
Código: HU3_p1	Historia de Usuario (No.3): Gestionar asignatura
Nombre: Adicionar asignatura	
Descripción: Se desea probar que el sistema adicione una asignatura correctamente.	
Condiciones de ejecución: No pueden existir campos vacíos.	
Entrada: Juego de datos válidos, es decir sin casillas en blanco.	
Resultado esperado: El sistema debe adicionar correctamente la asignatura.	
Evaluación de la prueba: Satisfactoria	

Tabla 89. Prueba 2 de HU Gestionar asignatura

Caso de Prueba de Aceptación	
Código: HU3_p2	Historia de Usuario (No.3): Gestionar asignatura
Nombre: Actualizar asignatura	
Descripción: Se desea probar que el sistema actualice los datos de una asignatura correctamente.	
Condiciones de ejecución: Modificar los datos ya existentes de la asignatura	
Entrada: Juego de datos válidos, es decir sin casillas en blanco.	
Resultado esperado: El sistema debe actualizar correctamente la asignatura.	
Evaluación de la prueba: Satisfactoria	

Tabla 90. Prueba 3 de HU Gestionar asignatura

Caso de Prueba de Aceptación	
Código: HU3_p3	Historia de Usuario (No.3): Gestionar asignatura
Nombre: Eliminar asignatura	
Descripción: Se desea probar que el sistema elimine una asignatura satisfactoriamente.	
Condiciones de ejecución: Se accede a la opción de eliminar asignatura.	
Entrada: Seleccionar la asignatura a eliminar, verificar, y confirmar la eliminación.	
Resultado esperado: El sistema debe eliminar satisfactoriamente la asignatura.	
Evaluación de la prueba: Satisfactoria	

Tabla 91. Prueba 1 de HU Gestionar tema

Caso de Prueba de Aceptación	
Código: HU4_p1	Historia de Usuario (No.4): Gestionar tema
Nombre: Adicionar tema	
Descripción: Se desea probar que el sistema adicione un tema correctamente.	
Condiciones de ejecución: No pueden existir campos vacíos.	
Entrada: Juego de datos válidos, es decir sin casillas en blanco.	
Resultado esperado: El sistema debe adicionar correctamente la asignatura.	

Evaluación de la prueba: Satisfactoria

Tabla 92. Prueba 2 de HU Gestionar tema

Caso de Prueba de Aceptación	
Código: HU4_p2	Historia de Usuario (No.4): Gestionar tema
Nombre: Actualizar tema	
Descripción: Se desea probar que el sistema actualice los datos de un tema correctamente.	
Condiciones de ejecución: Modificar los datos ya existentes del tema	
Entrada: Juego de datos válidos, es decir sin casillas en blanco.	
Resultado esperado: El sistema debe actualizar correctamente el tema.	
Evaluación de la prueba: Satisfactoria	

Tabla 93. Prueba 3 de HU Gestionar tema

Caso de Prueba de Aceptación	
Código: HU4_p3	Historia de Usuario (No.4): Gestionar tema
Nombre: Eliminar tema	
Descripción: Se desea probar que el sistema elimine un tema satisfactoriamente.	
Condiciones de ejecución: Se accede a la opción de eliminar tema.	
Entrada: Seleccionar el tema a eliminar, verificar, y confirmar la eliminación.	
Resultado esperado: El sistema debe eliminar satisfactoriamente el tema.	
Evaluación de la prueba: Satisfactoria	

Tabla 94. Prueba 1 de HU Gestionar actividad docente

Caso de Prueba de Aceptación	
Código: HU6_p1	Historia de Usuario (No.6): Gestionar actividad docente
Nombre: Adicionar Actividad docente	
Descripción: Se desea probar que el sistema adicione una actividad docente correctamente.	
Condiciones de ejecución: No pueden existir campos vacíos.	

Entrada: Juego de datos válidos, es decir sin casillas en blanco.
Resultado esperado: El sistema debe adicionar correctamente la actividad docente.
Evaluación de la prueba: Satisfactoria

Tabla 95. Prueba 2 de HU Gestionar actividad docente

Caso de Prueba de Aceptación	
Código: HU6_p2	Historia de Usuario (No.6): Gestionar actividad docente
Nombre: Actualizar actividad docente	
Descripción: Se desea probar que el sistema actualice los datos de una actividad docente correctamente.	
Condiciones de ejecución: Modificar los datos ya existentes de la actividad docente	
Entrada: Juego de datos válidos, es decir sin casillas en blanco.	
Resultado esperado: El sistema debe actualizar correctamente la actividad docente.	
Evaluación de la prueba: Satisfactoria	

Tabla 96. Prueba 3 de HU Gestionar actividad docente

Caso de Prueba de Aceptación	
Código: HU6_p3	Historia de Usuario (No.6): Gestionar actividad docente
Nombre: Eliminar actividad docente	
Descripción: Se desea probar que el sistema elimine una actividad docente satisfactoriamente.	
Condiciones de ejecución: Se accede a la opción de eliminar actividad docente.	
Entrada: Seleccionar la actividad docente a eliminar, verificar, y confirmar la eliminación.	
Resultado esperado: El sistema debe eliminar satisfactoriamente la actividad docente.	
Evaluación de la prueba: Satisfactoria	

Tabla 97. Prueba 1 de HU Gestionar evaluación

Caso de Prueba de Aceptación	
Código: HU7_p1	Historia de Usuario (No.7): Gestionar evaluación

Nombre: Adicionar evaluación
Descripción: Se desea probar que el sistema adicione una evaluación correctamente.
Condiciones de ejecución: No pueden existir campos vacíos.
Entrada: Juego de datos válidos, es decir sin casillas en blanco.
Resultado esperado: El sistema debe adicionar correctamente la evaluación.
Evaluación de la prueba: Satisfactoria

Tabla 98. Prueba 2 de HU Gestionar evaluación

Caso de Prueba de Aceptación	
Código: HU7_p2	Historia de Usuario (No.7): Gestionar evaluación
Nombre: Actualizar evaluación	
Descripción: Se desea probar que el sistema actualice los datos de una evaluación correctamente.	
Condiciones de ejecución: Modificar los datos ya existentes de la evaluación	
Entrada: Juego de datos válidos, es decir sin casillas en blanco.	
Resultado esperado: El sistema debe actualizar correctamente la evaluación.	
Evaluación de la prueba: Satisfactoria	

Tabla 99. Prueba 3 de HU Gestionar evaluación

Caso de Prueba de Aceptación	
Código: HU7_p3	Historia de Usuario (No.7): Gestionar evaluación
Nombre: Eliminar evaluación	
Descripción: Se desea probar que el sistema elimine una evaluación satisfactoriamente.	
Condiciones de ejecución: Se accede a la opción de eliminar evaluación.	
Entrada: Seleccionar la evaluación a eliminar, verificar, y confirmar la eliminación.	
Resultado esperado: El sistema debe eliminar satisfactoriamente la evaluación.	
Evaluación de la prueba: Satisfactoria	

Tabla 100. Prueba 4 de HU Gestionar evaluación

Caso de Prueba de Aceptación	
Código: HU7_p4	Historia de Usuario (No.7): Gestionar evaluación
Nombre: Asignar habilidades a evaluación	
Descripción: Se desea probar que el sistema permita asignar las habilidades a una evaluación satisfactoriamente.	
Condiciones de ejecución: Se accede a la opción de asignar habilidades, no se deben dejar campos vacíos.	
Entrada: Seleccionar la evaluación y luego asignarle las habilidades disponibles.	
Resultado esperado: El sistema debe asignar las habilidades satisfactoriamente a la evaluación seleccionada.	
Evaluación de la prueba: Satisfactoria	

Tabla 101. Prueba 1 de HU Gestionar habilidad

Caso de Prueba de Aceptación	
Código: HU8_p1	Historia de Usuario (No.8): Gestionar habilidad
Nombre: Adicionar habilidad	
Descripción: Se desea probar que el sistema adicione una habilidad correctamente.	
Condiciones de ejecución: No pueden existir campos vacíos.	
Entrada: Juego de datos válidos, es decir sin casillas en blanco.	
Resultado esperado: El sistema debe adicionar correctamente la habilidad.	
Evaluación de la prueba: Satisfactoria	

Tabla 102. Prueba 2 de HU Gestionar habilidad

Caso de Prueba de Aceptación	
Código: HU8_p2	Historia de Usuario (No.8): Gestionar habilidad
Nombre: Actualizar habilidad	
Descripción: Se desea probar que el sistema actualice los datos de una habilidad correctamente.	

Condiciones de ejecución: Modificar los datos ya existentes de la habilidad
Entrada: Juego de datos válidos, es decir sin casillas en blanco.
Resultado esperado: El sistema debe actualizar correctamente la habilidad.
Evaluación de la prueba: Satisfactoria

Tabla 103. Prueba 3 de HU Gestionar habilidad

Caso de Prueba de Aceptación	
Código: HU8_p3	Historia de Usuario (No.8): Gestionar habilidad
Nombre: Eliminar habilidad	
Descripción: Se desea probar que el sistema elimine una habilidad satisfactoriamente.	
Condiciones de ejecución: Se accede a la opción de eliminar habilidad.	
Entrada: Seleccionar la habilidad a eliminar, verificar, y confirmar la eliminación.	
Resultado esperado: El sistema debe eliminar satisfactoriamente la habilidad.	
Evaluación de la prueba: Satisfactoria	

Tabla 104. Prueba 1 de HU Mostrar Asistencia

Caso de Prueba de Aceptación	
Código: HU9_p1	Historia de Usuario (No.9): Mostrar Asistencia
Nombre: Mostrar asistencia	
Descripción: Se desea probar que el sistema permita visualizar la asistencia de un grupo en una asignatura determinada.	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura.	
Entrada: Seleccionar grupo y asignatura.	
Resultado esperado: El sistema debe mostrar un listado con la asistencia del grupo a todas las actividades hasta la fecha	
Evaluación de la prueba: Satisfactoria	

Tabla 105. Prueba 2 de HU Mostrar asistencia

Caso de Prueba de Aceptación	
Código: HU9_p2	Historia de Usuario (No.9): Mostrar asistencia
Nombre: Generar reporte asistencia	
Descripción: Se desea probar que el sistema genere el reporte de la asistencia de un grupo en determinada asignatura	
Condiciones de ejecución: Se debe escoger el grupo, la asignatura y luego acceder a la opción generar reporte.	
Entrada: Se selecciona el grupo, la asignatura, luego se genera el reporte de la asistencia.	
Resultado esperado: El sistema debe exportar el listado a formato .doc para luego poder imprimirlo.	
Evaluación de la prueba: Satisfactoria	

Tabla 106. Prueba 1 de HU Asistencia por encima de 20 porciento

Caso de Prueba de Aceptación	
Código: HU10_p1	Historia de Usuario (No.12): Asistencia por encima de 20 porciento
Nombre: Asistencia por encima del 20 porciento	
Descripción: Se desea probar que el sistema permita visualizar los estudiantes de un grupo con la asistencia por encima del 20 porciento en una asignatura	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura.	
Entrada: Seleccionar grupo y asignatura.	
Resultado esperado: El sistema debe mostrar un listado con la asistencia de los estudiantes con un porciento por encima del 20.	
Evaluación de la prueba: Satisfactoria	

Tabla 107. Prueba 2 de HU Asistencia por encima de 20 porciento

Caso de Prueba de Aceptación	
Código: HU10_p2	Historia de Usuario (No.12): Asistencia por encima de 20 porciento
Nombre: Generar reporte de asistencia por encima del 20 porciento	

Descripción: Se desea probar que el sistema genere el reporte de la asistencia de un grupo en determinada asignatura que esté por encima del 20 por ciento.
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura.
Entrada: Seleccionar grupo y asignatura.
Resultado esperado: El sistema debe exportar el listado a formato .doc para luego poder ser imprimido.
Evaluación de la prueba: Satisfactoria

Tabla 108. Prueba 1 de HU Promedio de evaluaciones

Caso de Prueba de Aceptación	
Código: HU11_p1	Historia de Usuario (No.10): Promedio de evaluaciones
Nombre: Listar promedio de evaluaciones	
Descripción: Se desea probar que el sistema liste las evaluaciones de un grupo con el promedio de cada estudiante	
Condiciones de ejecución: Se debe escoger el grupo y la asignatura de la cual se quiere obtener dicho listado.	
Entrada: Se selecciona el grupo y la asignatura.	
Resultado esperado: El sistema debe mostrar todas las evaluaciones de los estudiantes y su promedio respectivamente.	
Evaluación de la prueba: Satisfactoria	

Tabla 109. Prueba 2 de HU Promedio de evaluaciones

Caso de Prueba de Aceptación	
Código: HU11_p2	Historia de Usuario (No.10): Promedio de evaluaciones
Nombre: Generar reporte de evaluaciones	
Descripción: Se desea probar que el sistema genere el reporte de la evaluaciones en formato .doc para luego sea imprimido por el profesor.	
Condiciones de ejecución: Se debe escoger el grupo y la asignatura de la cual se quiere obtener dicho listado y luego acceder a la opción generar reporte.	

Entrada: Se selecciona el grupo y la asignatura, luego se genera el reporte.
Resultado esperado: El sistema debe exportar el listado a formato .doc para luego poder ser imprimido.
Evaluación de la prueba: Satisfactoria

Tabla 110. Prueba 1 de HU Evaluaciones por criterio

Caso de Prueba de Aceptación	
Código: HU12_p1	Historia de Usuario (No.11): Evaluaciones por criterio
Nombre: Listar evaluaciones por criterio	
Descripción: Se desea probar que el sistema liste las evaluaciones de un grupo de acuerdo a un criterio(3,4 ó 5)	
Condiciones de ejecución: Se debe escoger el grupo, la asignatura y el criterio para así obtener las evaluaciones por debajo de ese criterio	
Entrada: Se selecciona el grupo y la asignatura, luego se escoge el criterio.	
Resultado esperado: El sistema debe mostrar todas las evaluaciones de los estudiantes por debajo del criterio especificado.	
Evaluación de la prueba: Satisfactoria	

Tabla 111. Prueba 2 de HU Evaluaciones por criterio

Caso de Prueba de Aceptación	
Código: HU12_p2	Historia de Usuario (No.11): Evaluaciones por criterio
Nombre: Generar reporte de evaluaciones por criterio	
Descripción: Se desea probar que el sistema genere el reporte de la evaluaciones por criterio en formato .doc para luego sea imprimido por el profesor.	
Condiciones de ejecución: Se debe escoger el grupo, la asignatura y el criterio de los cuales se quiere obtener dicho listado y luego acceder a la opción generar reporte.	
Entrada: Se selecciona el grupo, la asignatura y el criterio, luego se genera el reporte.	
Resultado esperado: El sistema debe exportar el listado a formato .doc para luego poder ser imprimido.	
Evaluación de la prueba: Satisfactoria	

Tabla 112. Prueba 1 de HU Habilidades suspensas

Caso de Prueba de Aceptación	
Código: HU13_p1	Historia de Usuario (No.13): Habilidades suspensas
Nombre: Listar estudiantes con habilidades suspensas	
Descripción: Se desea probar que el sistema permita visualizar los estudiantes de un grupo que tengan habilidades suspensas en las evaluaciones de una asignatura	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura.	
Entrada: Seleccionar grupo y asignatura.	
Resultado esperado: El sistema debe mostrar un listado con los estudiantes que tengan habilidades suspensas en las evaluaciones de una asignatura.	
Evaluación de la prueba: Satisfactoria	

Tabla 113. Prueba 2 de HU Habilidades suspensas

Caso de Prueba de Aceptación	
Código: HU13_p2	Historia de Usuario (No.13): Habilidades suspensas
Nombre: Generar reporte de habilidades suspensas	
Descripción: Se desea probar que el sistema genere el reporte de los estudiantes que tengan habilidades suspensas en las evaluaciones de una asignatura.	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura.	
Entrada: Seleccionar grupo y asignatura.	
Resultado esperado: El sistema debe exportar el listado a formato .doc para luego poder ser imprimido.	
Evaluación de la prueba: Satisfactoria	

Tabla 114. Prueba 1 de HU Resultado de las evaluaciones

Caso de Prueba de Aceptación	
Código: HU15_p1	Historia de Usuario (No.15): Resultado de las evaluaciones

Nombre: Generar reporte gráfico resultado de las evaluaciones
Descripción: Se desea probar que el sistema genere un reporte gráfico con el resultado de las evaluaciones de los estudiantes de un grupo.
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura y luego acceder a la opción generar reporte.
Entrada: Seleccionar grupo y asignatura.
Resultado esperado: El sistema debe generar el reporte gráfico donde se muestre el comportamiento de los estudiantes en las evaluaciones realizadas hasta el momento.
Evaluación de la prueba: Satisfactoria

Tabla 115. Prueba 1 de HU Presentes por actividades

Caso de Prueba de Aceptación	
Código: HU16_p1	Historia de Usuario (No.16): Presentes por actividades
Nombre: Generar reporte gráfico de estudiantes presentes por actividades	
Descripción: Se desea probar que el sistema genere un reporte gráfico con la asistencia de manera general de los estudiantes de un grupo en cada una de las actividades de una asignatura.	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura y luego acceder a la opción generar.	
Entrada: Seleccionar grupo y asignatura.	
Resultado esperado: El sistema debe generar el reporte gráfico donde se muestre la asistencia por actividades de una asignatura.	
Evaluación de la prueba: Satisfactoria	

Tabla 116. Prueba 1 de HU Presentes por tema

Caso de Prueba de Aceptación	
Código: HU17_p1	Historia de Usuario (No.17): Presentes por tema
Nombre: Generar reporte gráfico de estudiantes presentes por tema.	
Descripción: Se desea probar que el sistema genere un reporte gráfico con la asistencia de manera general de los estudiantes de un grupo en cada uno de los temas de una asignatura.	

Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura y luego acceder a la opción generar.
Entrada: Seleccionar grupo y asignatura.
Resultado esperado: El sistema debe generar el reporte gráfico donde se muestre la asistencia por temas de una asignatura.
Evaluación de la prueba: Satisfactoria

Tabla 117. Prueba 1 de HU Estudiantes suspensos y aprobados por habilidad.

Caso de Prueba de Aceptación	
Código: HU18_p1	Historia de Usuario (No.18): Estudiantes suspensos y aprobados por habilidad.
Nombre: Generar reporte gráfico de estudiantes suspensos y aprobados por habilidad.	
Descripción: Se desea probar que el sistema genere un reporte gráfico con la cantidad de estudiantes suspensos y aprobados por habilidad.	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura y luego acceder a la opción generar.	
Entrada: Seleccionar grupo y asignatura.	
Resultado esperado: El sistema debe generar el reporte gráfico donde se muestren los estudiantes aprobados y suspensos por cada una de las habilidades que se miden en una asignatura.	
Evaluación de la prueba: Satisfactoria	

Tabla 118. Prueba 1 de HU Promedio de notas por evaluaciones.

Caso de Prueba de Aceptación	
Código: HU19_p1	Historia de Usuario (No.19): Promedio de notas por evaluaciones.
Nombre: Generar reporte gráfico con el promedio general de los estudiantes de un grupo en cada una de las evaluaciones (nota promedio por evaluación).	
Descripción: Se desea probar que el sistema genere un reporte gráfico con el promedio general de los estudiantes de un grupo en cada una de las evaluaciones.	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura y luego acceder a	

la opción generar.
Entrada: Seleccionar grupo y asignatura.
Resultado esperado: El sistema debe generar el reporte gráfico donde se muestre el promedio general de los estudiantes de un grupo en cada una de las evaluaciones
Evaluación de la prueba: Satisfactoria

Tabla 119. Prueba 1 de HU Asistencia y aprobados general.

Caso de Prueba de Aceptación	
Código: HU20_p1	Historia de Usuario (No.20): Asistencia y aprobados general
Nombre: Generar reporte gráfico con la asistencia y los aprobados de un grupo.	
Descripción: Se desea probar que el sistema genere un reporte gráfico con los indicadores de asistencia y aprobados por evaluaciones de un grupo.	
Condiciones de ejecución: Se debe seleccionar el grupo y la asignatura y luego acceder a la opción generar.	
Entrada: Seleccionar grupo y asignatura.	
Resultado esperado: El sistema debe generar el reporte gráfico donde se muestren los indicadores de asistencia y aprobados por evaluaciones de un grupo.	
Evaluación de la prueba: Satisfactoria	