



Universidad de las Ciencias en Informática
Facultad Regional “Mártires de Artemisa”

**Título: Sistema de Gestión para la planificación docente en la
Facultad Regional “Mártires de Artemisa”**

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

Autora: Liana Abreu Guerra

Tutora: Ing. Leidy Laura Sánchez González

Co-Tutor: Boris R. Núñez Fernández

Artemisa, junio del 2012

“Año 54 de la Revolución”

Declaración de autoría

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Liana Abreu Guerra

Autor

Ing. Leidy Laura Sánchez González

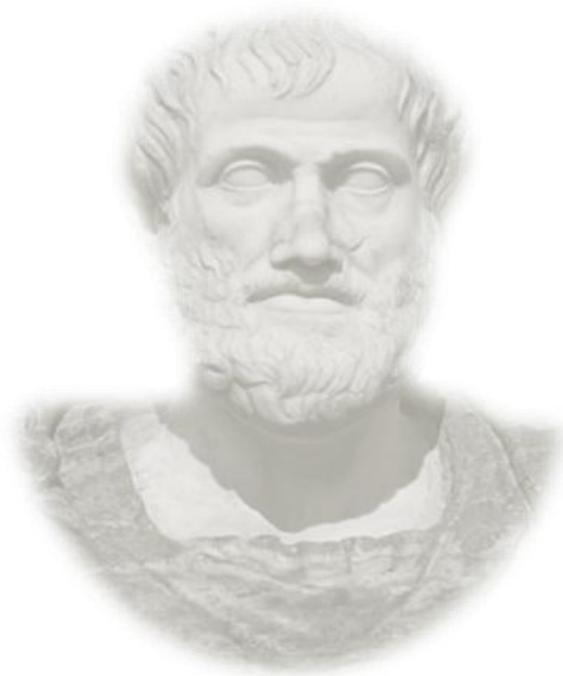
Tutor

Boris Ramón Núñez Fernández

Co - Tutor

La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.

Aristóteles



RESUMEN

Los Sistemas de Gestión han sido cruciales para el procesamiento y almacenamiento de datos en cualquier Empresa e Institución que maneje grandes cantidades de información. La siguiente investigación se realiza debido a que en la Facultad Regional “Mártires de Artemisa” se necesita gestionar las tareas que en ella se ejecutan sobre el proceso docente educativo en busca de mejorar la integridad de sus datos. Las funcionalidades de estas tareas son: Las planificaciones de horario docente y de cuidados de prueba, y llevar el control de Asistencia a los profesores tanto a clases como a los cuidados de prueba.

Para dar respuesta a la problemática se creó un Sistema de Gestión para la planificación docente comenzando por la realización del estado del arte acerca de los sistemas que gestionan la planificación docente, analizando las descripciones de las metodologías de desarrollo y adaptándose a SXP para el ciclo de vida de la propuesta de solución, la aplicación se implementó utilizando el lenguaje de programación Java mediante la herramienta NetBeans, se empleó el Visual Paradigm en la modelación de la Ingeniería de Software, el Power-Architect en el diseño de la Base de Datos, para el almacenamiento de los datos se empleó el PostgreSQL y Hibernate para el mapeo de sus tablas.

El Sistema logra una mayor rapidez y eficiencia en sus procesos, satisfaciendo así las necesidades del cliente luego de haber sido comprobado con las pruebas de aceptación al producto final para obtener los errores de funcionalidad.

Palabras Claves: Sistemas de Gestión, procesamiento de datos, almacenamiento de datos, proceso docente educativo.

INDICE DE CONTENIDO

Introducción	7
Capítulo 1: Fundamentación teórica	14
1.1 Estado del Arte	14
1.2 Conceptos que ayudan a comprender el objeto de estudio del trabajo.....	15
1.3 Metodologías que se tuvieron en cuenta.	18
1.4 Tecnologías que se tienen en consideración para la elaboración de la aplicación. 24	
1.4.1 Herramientas para el desarrollo del producto	24
1.4.2 Sistemas Gestores de Base de Datos	26
1.4.3 Herramientas para el diseño de la Base de datos y el Modelado	31
1.4.4 Herramientas para el mapeo de la base de Batos.....	33
Capítulo 2: Características, Análisis y Diseño del Sistema.	36
2.1 Propuesta de la Solución	36
2.2 Tecnologías Escogidas para el desarrollo de la aplicación	37
* Metodología	37
* Herramienta para el desarrollo del producto	38
* Gestor de Base de Datos para el almacenamiento de la información	39
* Herramientas para el diseño de la Base de datos y el Modelado	39
* Hibernate vs JDO para el mapeo de la base de datos	40
2.3 Modelo de Dominio	40
2.4 Artefactos Generados.....	¡Error! Marcador no definido.
2.4.1 Lista de Reserva del Producto (LRP)	41
2.4.2 Historia de Usuario y Tareas de la Ingeniería correspondiente.....	45
2.4.3 Plan Releases.....	52

2.5 Diseño con metáforas.....	52
2.6 Diagrama de Componentes.....	53
2.7 Patrones utilizados.....	55
Capítulo 3: Implementación y Prueba.....	56
* Funcionalidades Obtenidas.....	56
3.1 Casos de prueba.....	57
3.1.1 Métodos de Prueba.....	57
3.1.2 Casos de Prueba.....	58
* Resultados Obtenidos.....	66
* Aportes Sociales.....	66
Conclusiones Generales.....	67
Recomendaciones.....	68
Bibliografía Referenciada.....	69
Bibliografía Consultada.....	72
Glosario de Términos.....	73

INTRODUCCIÓN

Internet le ha dado una nueva y mayor dimensión a la revolución de la información y específicamente al impacto de las tecnologías de la información y la comunicación (TIC) en las personas, las organizaciones y en la sociedad como un todo. Cada vez que necesitamos o deseamos conocer algo acerca de acontecimientos, hechos, sucesos, personas, empresas, instituciones ciudades, países, etcétera, ya sea porque necesitamos reducir nuestra incertidumbre para tomar decisiones o simplemente porque queremos incrementar nuestro conocimiento acerca de algo, buscamos aquello que denominados información. (Premonogocio, 2008). La información es la fuente principal para la comunicación en todos los aspectos y se considera como un recurso que se encuentra al mismo nivel que los recursos financieros, materiales y humanos. (Palmero, 2010)

En las empresas, la información se utiliza para plasmar todo movimiento necesario en el cumplimiento de sus funciones o actividades. Esta se manipula dentro del contexto al que es equivalente, con el objetivo de llevar un control de manera eficiente en su contenido, convirtiéndose así en la raíz fundamental para el desarrollo de cualquier compañía. Existen áreas de cada institución que manejan grandes cantidades de información, por lo que se hace complejas tratarlas de manera tradicional, es decir, sin el empleo de cómputos que agilicen el proceso. Esto hace que surjan un grupo de técnicas y estrategias para la gestión de la información, ya que de esta forma el trabajo a realizar se desempeña de manera fácil y con la calidad y eficiencia requerida.

Para la gestión de la información se crean sistemas de gestiones que actualmente son muy empleados principalmente en compañías famosas debido a las satisfacciones que proporcionan, pues con su utilización se tienen seguros los datos, existe organización a la hora de guardar información y de obtenerla, rapidez en el trabajo y el mismo se actualiza automáticamente. (Institution, 2012)

INTRODUCCIÓN

Cuba a pesar de que es un país subdesarrollado y con características propias, mira hacia el futuro buscando un mayor desarrollo con el avance de las Tecnologías de la Información y las Comunicaciones (TIC) haciendo su esfuerzo por disponer de los recursos tecnológicos necesarios para emplearlos de manera tal que ayude al avance económico del país y a la sistematización de la información de sus organismos. Para lograrlo, trabaja en base al perfeccionamiento empresarial, mostrando los cambios dirigidos hacia una mayor centralización con el empleo de los sistemas de gestión de los procesos de negocio en sus instituciones u empresas. Estos sistemas se utilizan en el país para ayudar a lograr los objetivos que se persiguen en estos establecimientos con un grupo de normas que incluyen el pensamiento disciplinado, mejor organización de las ideas, el enfoque centrado en la gestión y la optimización de los procesos, fluyendo así en la comodidad que facilita la elaboración de un sistema de gestión de la información y la calidad del trabajo. (Reyes, 2012)

Una de las esferas que trabaja en base a su perfeccionamiento, es la Educación. Se han creado condiciones en busca de mejoras para esta. De aquí han nacido nuevas ideas para lograr este objetivo cambiando los planes de estudio para intensificarlos y el surgimiento de nuevas escuelas para nuevos fines. (Portal UCI)

Ejemplo de esto es el nacimiento de la Universidad de las Ciencias Informáticas (UCI), mediante la batalla de ideas y con dos objetivos: informatizar el país y desarrollar la industria del Software para contribuir al desarrollo económico del mismo. Desde sus inicios a venido evolucionando progresivamente en la automatización, sistematización y optimización de los procesos correspondientes a los problemas existentes. También en ella se lleva a cabo un grupo de transformaciones para la corrección de su funcionamiento, empleando Sistemas de Gestiones de Información que han resultado satisfactorios al cumplir las

INTRODUCCIÓN

expectativas buscadas, brindar la comodidad y calidad requerida para su eficiencia y al servir de muestra para otros que se quieran crear. (2012)

En la Facultad Regional de la UCI “Mártires de Artemisa” se realizan una serie de tareas que tributan al adecuado desempeño y funcionamiento del proceso docente educativo, tarea que se encarga de cumplir el Departamento Docente General de la Facultad. Este es un trabajo engorroso cuando se trata de forma manual, pues uno de sus principales problemas es el exceso de la información que se maneja.

Dentro de las actividades que ejecuta el Departamento están:

- Planificación del horario docente.
- Planificación de cuidados de prueba.
- Llevar el control del Registro de Asistencia.

Estas labores proceden de forma manual y se tornan complejas, debido a que la información que se utiliza para poder ejecutarlas está dispersa, conllevando entonces, a una serie de conflictos como:

- Re planificación cuando surgen errores que no se tuvieron en cuenta.
- Poca rapidez en el procesamiento de los datos.
- Pérdida de tiempo.
- Pérdida de la documentación almacenada.
- Poca rapidez ante los reportes sistemáticos.

Durante la ejecución de las tareas a realizar por el Departamento Docente General, solo se hace uso de herramientas como Microsoft Excel y Microsoft Word, trayendo consigo falta de integridad de la información que se manipula en dicho Departamento.

INTRODUCCIÓN

Por lo antes mencionado surge como **Problema Científico**:

¿Cómo lograr la integridad de la información en los procesos de planificación docente de la Facultad Regional “Mártires de Artemisa”?

El **Objeto de Estudio** sería: Procesos de Gestión de información, dando como **Campo de Acción**: Procesos de Gestión de información para la planificación docente.

Debido a lo antes expuesto se describe como **Objetivo General**: Desarrollar un Sistema de Gestión para la planificación docente del Departamento Docente General de la Facultad Regional “Mártires de Artemisa” que contribuya a mejorar la integridad de la información que se maneja en el mismo.

Para cumplir con el objetivo general se definen las siguientes **Preguntas Científicas**.

1. ¿Cuáles son los Fundamentos Teóricos de la investigación?
2. ¿Cuál es el Análisis y Diseño que responde a los requisitos del Área de Formación de la Facultad Regional “Mártires de Artemisa”?
3. ¿Cómo implementar la solución de software propuesta para el Área de Formación de la Facultad Regional “Mártires de Artemisa”?
4. ¿Cómo validar los resultados obtenidos con la solución?

A las preguntas científicas se les dará cumplimiento con la formulación de las siguientes **tareas de la investigación**.

1. Realización del marco teórico de la investigación elaborando el estado del arte acerca de los sistemas de gestión y los procesos de planificación docente.

INTRODUCCIÓN

2. Especificación de las herramientas y tecnologías a utilizar para el desarrollo del sistema de gestión.
3. Análisis y diseño del sistema a implementar de acuerdo a los modelos realizados y los requerimientos identificados.
4. Implementación del sistema diseñado que tribute a los requerimientos funcionales.
5. Ejecución de las pruebas de funcionalidad al Sistema de Planificación Docente de la Facultad Regional “Mártires de Artemisa”.

A continuación se muestran los **Métodos Científicos** que se utilizarán.

Métodos empíricos:

Entrevista: Se realizaron varias entrevistas a la planificadora docente de la Facultad Regional “Mártires de Artemisa”, así como a la vicedecana de formación para ultimar algunos detalles, con el objetivo de fomentar la problemática para seleccionar los principales requisitos que debe cumplir el sistema.

Observación: Se tuvo la oportunidad de compartir experiencia con la planificadora docente realizando las tareas que le compete como: planificación del horario, del cuidado de prueba y de llevar el control del registro de asistencia, con los fines de tener una visión y acercamiento al problema real.

Revisión de documentos: Se revisarán documentos de la ingeniería del software, relacionados con las herramientas y tecnologías existentes y otros que pudieran aportar información sobre el tema a tratar.

Métodos teóricos:

Histórico-Lógico: Se utiliza en la investigación realizada sobre los antecedentes de los sistemas de gestión, posibilitando el análisis de la trayectoria de estos sistemas para una mejor comprensión de los mismos.

INTRODUCCIÓN

Modelación: Se efectuará mediante el análisis y diseño que responde a los requisitos del Área de Formación con el empleo de las diferentes herramientas que se seleccionen así como la implementación del producto final, para obtener el resultado que se espera.

Analítico-Sintético: Mediante este método se determinaron los principales fundamentos que caracterizan a los sistemas de gestión y las herramientas más utilizadas en el proceso de desarrollo de los mismos, partiendo de los aspectos relacionados fundamentalmente con la gestión de la información y la planificación.

Métodos estadísticos: Se utilizó la estadística descriptiva a través del análisis porcentual de los datos agrupados de la aplicación de las técnicas e instrumentos empíricos de investigación.

Resumen de los capítulos

El documento estará compuesto por 3 capítulos, de los cuales se expondrá a continuación una breve reseña, donde se especifica qué contendrá cada uno de ellos.

Capítulo 1- Fundamentación Teórica:

En este capítulo se detallarán los temas importantes para el desarrollo del trabajo. Aquí se especifican algunos conceptos que se tendrán en cuenta para la realización del mismo como: la definición de Gestión, Sistemas de Información, Planificación Docente, entre otras. Además se analizarán las metodologías de desarrollo de software y las tecnologías que serán utilizadas, así como las posibles herramientas a utilizar en la construcción del sistema, especificando sus principales características, ventajas y lenguajes de programación utilizados.

Capítulo 2- Características, Análisis y Diseño del Sistema

INTRODUCCIÓN

En este capítulo se define el negocio detalladamente, fundamentado en cómo funcionan los procesos de planificación del horario, del cuidado de pruebas y el registro de asistencia que se lleva a cabo diariamente a los profesores de forma manual en el Departamento Docente General de la Facultad Regional “Mártires de Artemisa” mediante la propuesta de la solución. Además se puntualizan los requisitos del futuro componente, se realiza el diseño del modelo de dominio y diagrama de paquetes, así como la creación de todos los artefactos que se generan durante el análisis y el diseño del sistema y la descripción también de los patrones utilizados para el mismo.

Capítulo 3- Implementación y Prueba

En este capítulo se elabora la solución propuesta evidenciándose las funcionalidades del sistema de gestión para la planificación docente en la Facultad Regional “Mártires de Artemisa”, se realiza una breve descripción de los métodos de prueba que existen y explicando el porqué del seleccionado, finalizando así el capítulo con los diseños de casos de pruebas que validen la implementación realizada.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se abordará sobre los conceptos que se tienen en cuenta para el desarrollo del sistema a realizar. Además se describe el estado del arte acerca los Procesos de Gestión de información y planificación docente en todos los niveles, es decir, a nivel mundial, en Cuba y en la UCI. Se especifican las características de un grupo de metodologías para seleccionar la que se va emplear para el desarrollo del producto y el por qué se escogió la misma, así como también las herramientas que se pueden utilizar dando apoyo a la solución del problema y el lenguaje de programación escogido para la implementación del mismo.

1.1 ESTADO DEL ARTE

Existen varios sistemas de gestión para la planificación docente, pero se diferencian en los procesos de planificación que realizan cada uno de ellos aunque ejecuten algunas tareas de igual manera, ejemplo de estas se tiene:

Universidad Autónoma de Guadalajara (UAG): Es la primera Universidad de América Latina en utilizar el Sistema Integral de Información que ofrece la empresa de talla internacional a las mejores universidades del mundo como Harvard, Stanford, Yale y Oxford, entre otras. Esta alianza le permite a la UAG incorporar prácticas de clase mundial en su gestión operativa que mejoran el proceso educativo y experiencia de aprendizaje entre los alumnos y los profesores. (UAG, 2011)

Sistema de Gestión de Alumnado, Planificación Docente y Campus Virtual Integrado (UAL (Universidad de Almería en España)): Aporta funcionalidades que completan y cubren con totalidad la planificación docente universitaria y entre las que se encuentran la gestión de actividades docentes, asignación docente,

definición de horarios, gestión de espacios, disponibilidad y situación de los recursos y relación y cómputo de la capacidad y carga docente. (2011)

Dentro de la investigación realizada sobre los Sistemas de Gestión de Información para la planificación Docente en Cuba, solo se encontró información dentro del ámbito de la Universidad de la Ciencias Informáticas que si cuenta con un **Sistema de Gestión para la Planificación Docente en la Facultad 5 (UCI)**: Esta lleva a cabo el control de la Planificación Docente de la Facultad donde el mismo desarrolla las tareas de planificación de guardia docente y el cuidado de exámenes. (Álvarez, 2009)

El Sistema a desarrollar en la Facultad Regional de Artemisa, no es similar a ninguna de las antes mencionadas aunque consiste en gestionar los procesos de planificación docente también. Las funcionalidades que brindan los Sistemas existentes son parecidas a las que se requieren, pero estos no responden los intereses del Departamento Docente General de la Facultad aunque se quiera adaptar, debido a que son privativos, se desarrollaron para ser visitados por cualquier cantidad de usuario y las características de los procesos de planificación docente no son iguales a las dicho Departamento, es decir, el Sistema a desarrollar tendrá otras características físicas y lógicas con nuevas funcionalidades y podrá implantarse en cualquier institución que lo necesite. Además con su desarrollo se lograría un mejor aprovechamiento de las tecnologías existentes en la Facultad y una mejor manipulación de la documentación que se genera en dicho Departamento.

1.2 CONCEPTOS QUE AYUDAN A COMPRENDER EL OBJETO DE ESTUDIO DEL TRABAJO.

* **Gestión:** Son guías para orientar la acción, previsión, visualización y empleo de los recursos y esfuerzos a los fines que se desean alcanzar, la secuencia de

actividades que habrán de realizarse para lograr objetivos y el tiempo requerido para efectuar cada una de sus partes y todos aquellos eventos involucrados en su consecución. Existen distintos tipos de gestión, dentro de ellos se encuentra: (Díaz, 2008)

- **La gestión de proyectos:** Es la disciplina que se encarga de organizar y de administrar los recursos de manera tal que se pueda concretar todo el trabajo requerido por un proyecto dentro del tiempo y del presupuesto definido.
- **La gestión tecnológica:** Es el proceso de adopción y ejecución de decisiones sobre las políticas, estrategias, planes y acciones relacionadas con la creación, difusión y uso de la tecnología.
- **La gestión social:** Es un proceso completo de acciones y toma de decisiones, que incluye desde el abordaje, estudio y comprensión de un problema, hasta el diseño y la puesta en práctica de propuestas. (Díaz, 2008)

* Sistema de Información:

- Es un conjunto organizado de elementos, que pueden ser personas, datos, actividades o recursos materiales en general. Estos elementos interactúan entre sí para procesar información y distribuirla de manera adecuada en función de los objetivos de una organización. Los sistemas de información no sólo almacenan información, también deben proporcionar servicios usando la información que almacenan, además de que los mismos están concebidos para capturar, actualizar, integrar, consultar y analizar información pertinente a la organización a la cual pertenecen. (Definicion.de, 2012)

Funcionamiento de los sistemas de información:

- ^ Gestionan los datos que almacenan.
- ^ Gestionan los usuarios que utilizan la información, que además pueden agregarla.
- ^ Poseen una interfaz en correspondencia con la información que contienen.
- ^ Integran no sólo datos e información, sino también programas y otros sistemas de información. (Sarduy, 2006)

- Es el conjunto de los elementos y procesos que intervienen dinámicamente en la explotación de información cognitiva concebida en el marco de un grupo social concreto y para áreas determinadas, cuyo propósito es facilitarles el acceso al conocimiento y apoyarlos en la toma correcta de decisiones. (Livia M. Reyes Ramírez, 2007)

Se hace elección del primer concepto referenciado de los Sistemas de Gestión, debido a que el mismo se adecúa a las características del producto.

* **Gestión de la Información:** Es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de una Institución. (Palmero Berberena, 2010)

* **Sistema de Gestión de Información:** Los sistemas de gestión de Información constituyen hoy una alternativa de imprescindible presencia en cada organización al permitir operar casi todos los activos tangibles e intangibles de la institución y llegar a convertirse en la herramienta integral de gerencia más cotizada y necesaria para alcanzar con éxito los resultados propuestos por la organización. (Contreras, Yimian de Lyz: Rivero Soleydi)

Por lo tanto un Sistema de Gestión de Información puede considerarse como un programa o aplicación que se utilice para almacenar y consultar cierta información específica que maneje la organización que disponga y trabaje con el mismo.

* **Planificación Docente:**

La planificación docente se desarrolla en una serie de pasos. La primera etapa es el **diagnóstico**, donde se vinculan las necesidades educativas, las condiciones de aprendizaje y los factores externos que afectan al proceso educativo. El paso siguiente es el **análisis** de la naturaleza del problema, que supone la comprensión integral de la complejidad de la realidad educativa y continúa con el **diseño** y la **evaluación** de las opciones de acción.

De acuerdo a la dimensión técnica, la planeación educativa debe considerar el uso de la tecnología en la pedagogía, mientras que en cuanto a su dimensión política, debe atender a los marcos normativos existentes. (Definicion.de, 2012)

1.3 METODOLOGÍAS QUE SE TUVIERON EN CUENTA.

* **Metodologías de desarrollo:** Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo es garantizar la eficacia (p.ej. cumplir los requisitos iniciales) y la eficiencia (p.ej. minimizar las pérdidas de tiempo) en el proceso de generación de software. (2009)

* **Metodología RUP:** El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

Características Metodología RUP (2009)

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software

¿Cuándo se puede seleccionar la metodología? (2008)

- ⤴ Exista comunicación en el equipo.
- ⤴ Complejidad de desarrollo de acuerdo al tamaño del proyecto.
- ⤴ Configuración y control de cambios.

Ventajas de la metodología

- Mayor documentación
- Verificar la cantidad de software
- Configuración y control de cambio
- Es modelado, guiado por casos de uso
- Centrado en la arquitectura, guiado por riesgos

Desventajas de la metodología

- Los cambios son en una fase.
- Se emplea en proyectos grandes

* **Metodología Scrum:** Es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa (ROI). Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación. (SOFTENG, 2010)

En Scrum, un proyecto se ejecuta en bloques temporales (iteraciones) de un mes natural (pueden ser de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea susceptible de ser entregado con el mínimo esfuerzo cuando el cliente lo solicite.

¿Cuándo utilizar esta metodología?

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En esta metodología se realizan entregas parciales del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, los requisitos son cambiantes o poco definidos y donde la innovación, la competitividad y la productividad son fundamentales.

Ventajas de la metodología:

- Entrega de un producto funcional al finalizar cada iteración.
- Posibilidad de ajustar la funcionalidad en base a la necesidad de negocio del cliente.
- Visualización del proyecto día a día.

- Gestión regular de las expectativas del cliente y basada en resultados tangibles.
- Resultados anticipados (time to market).
- Productividad y calidad.
- Mitigación sistemática de los riesgos del proyecto.

Desventajas de la metodología

- ⤴ Dificultad de aplicación en grandes proyectos.
- ⤴ Se requiere de un “agile champion”, experto en la metodología que monitorice su cumplimiento.
- ⤴ Parte del paradigma consiste en usar el método “tal cual”, evitando adaptarlo a la empresa (supuestamente esto saca a la luz problemas en la metodología de desarrollo existente). Claro, si por otro lado arrancamos con el método ya adaptado a la empresa, es posible que no estemos aprovechando las ventajas de la metodología ágil.
- ⤴ Plantea un problema si el desarrollo está restringido por una fecha de entrega y un precio de entrega cerrados por contrato.
- ⤴ Presupone que los requerimientos cambian, pero no de forma que el cliente acepte un diseño funcional/técnico.
- ⤴ Presupone que el equipo está muy formado y motivado.
- ⤴ Presupone que el cliente está muy involucrado en el desarrollo, participa de forma activa y continua, y revisa frecuentemente el avance de la funcionalidad conforme salen a la luz los sprints. Esto sin embargo no parece producirse en la mayoría de los proyectos puesto a que el cliente

participa, pero no hasta el punto de dedicar tiempo y recursos para revisar pequeños avances en el desarrollo.

- ⤴ Presupone que el cliente no exige ni necesita toda la documentación que manejan actualmente las empresas y que las diversas normativas internacionales requieren.

* **Metodología Programación Extrema XP:** (Javier, 2012)

Es una metodología de desarrollo de software eficiente, de bajo riesgo y flexible . De todas las metodologías ágiles, ésta es la que ha recibido más atención desde que fue dada a conocer en 1999 a través del libro “Extreme programming Explained” de Kent Beck. XP está formada por valores, principios y prácticas. 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.

¿Cuándo se puede seleccionar la metodología? (2008)

- ⤴ Los requisitos cambian (clientes indecisos).
- ⤴ Proyectos con alto grado de riesgos.
- ⤴ Grupos pequeños de programadores entre 2 y 12.

Ventajas de la metodología

- Comunicación
- Simplicidad

- Realimentación
- Coraje(satisfacción de los programadores)
- Disminuye traza de errores
- Alta calidad en mínimo tiempo

Desventajas de la metodología

- Dificultad para determinar el costo del proyecto
- Se utiliza principalmente en proyectos pequeños.

* **Metodología SXP:** Esta metodología de desarrollo está compuesta por la unión de las metodologías Scrum (Metodología para la gestión de trabajo) y XP (Metodología para el proceso de desarrollo de software). La misma consta de varias fases para su desarrollo.

Fases de la metodología SXP: (Metodología Ágil de desarrollo de Software SXP, 2011)

- ✓ **Planificación- Definición:** Propósito: Establecer la visión, fijar expectativas, y asegurar financiamiento.
- ✓ **Desarrollo:** Propósito: Implementar un sistema listo para entrega. (El tiempo puede decrementarse a la medida que se está refinando el producto).
- ✓ **Entrega:** Propósito: Puesta en Operación
- ✓ **Mantenimiento:** Propósito: Soporte del Software

Ventajas de SXP

- × El trabajo es ágil y mantiene al cliente dentro del equipo de desarrollo.
- × Integración continua del sistema.

- × Reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación.
- × Todos los integrantes del equipo de trabajo conocen algo sobre todas las partes y muy bien aquellas en las que trabajan.
- × Todo el código se escribe en parejas.
- × Se realiza el trabajo de 1 persona en casi la mitad del tiempo y mejor.

1.4 TECNOLOGÍAS QUE SE TIENEN EN CONSIDERACIÓN PARA LA ELABORACIÓN DE LA APLICACIÓN.

1.4.1 HERRAMIENTAS PARA EL DESARROLLO DEL PRODUCTO

- **Java:** Es un lenguaje de programación orientado a objetos y la primera plataforma informática creada por Sun Microsystems en 1995. (Computación Aplicada al Desarrollo SA de CV)

Las características y ventajas principales que nos ofrece Java son:
(Características de java, 2009)

- 1.- Lenguaje Simple
- 2.- Orientado a Objeto
- 3.- Distribuido
- 4.- Robusto
- 5.- Seguro
- 6.- Indiferente a la arquitectura
- 7.- Portable
- 8.- Interpretado y compilado a la vez
- 9.- Multihebra o Multihilos
- 10.- Dinámico
- 11.- Produce Applets

12.- Alto rendimiento

- **ExtJS:** Es una librería Javascript que permite construir aplicaciones complejas en internet además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX, DHTML, tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales. (2012)

Esta librería incluye:

- Componentes UI del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias OpenSource (GPL) y comerciales

Navegadores Compatibles

ExtJS soporta todos los navegadores web más importantes, incluyendo:

- ⤴ Firefox 1.5+ (PC, Mac)
- ⤴ Safari 3+
- ⤴ Chrome 3+
- ⤴ Opera 9+ (PC, Mac)
- ⤴ Internet Explorer 6+

Librerías Importantes

Existe una condición primaria para el uso de esta herramienta y es la correcta ubicación dentro del código HTML de las siguientes librerías, imprescindible para su correcto funcionamiento:

- ⤴ **ext-all.css:** contiene todos los estilos para visualizar correctamente los componentes.
- ⤴ **ext-base.js:** contiene solo lo que necesita ext para su correcto

funcionamiento.

- ✦ **ext-all.js:** contiene todos los componentes y características disponibles en ExtJS.

Ventajas

- ✦ Permite crear aplicaciones complejas utilizando componentes predefinidos.
- ✦ Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores (Firefox, IE, Safari, Opera etc.).
- ✦ Relación entre Cliente-Servidor balanceado: Se distribuye la carga de procesamiento entre, permitiendo que el servidor pueda atender más clientes al mismo tiempo.
- ✦ Eficiencia de la red: Disminuye el tráfico en la red pues las aplicaciones cuentan con la posibilidad de elegir que datos desea transmitir al servidor y viceversa (Criterio este que puede variar con el uso de aplicaciones de pre-carga).
- ✦ Comunicación asíncrona: En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.

Desventajas

- ✦ Necesidad de una plataforma: Pues dependemos del paquete ExtJS para obtener los resultados deseados.
- ✦ Para algunos él no contar con una licencia LGPL.

1.4.2 SISTEMAS GESTORES DE BASE DE DATOS

CAPÍTULO 1

Un sistema gestor de base de datos (SGBD) es un tipo de software muy específico, dedicado a servir de interfaz entre la Base de Datos, el usuario y las aplicaciones que la utilizan. Permite introducir, organizar y recuperar la información de las bases de datos; es decir, administrarlas. Brinda los siguientes servicios:

- Construcción y definición de base de datos.
- Trabajo con los datos almacenados efectuando consultas, inserciones y actualizaciones utilizando lenguaje de manipulación de datos.
- Mecanismo de seguridad a través de nivel de acceso para usuarios y grupos de usuarios para acceder a los datos de la base de datos.
- Mecanismos de respaldo y recuperación para en caso de fallos en el sistema no se pierda la información.
- Utilización de mecanismos para evitar que los datos sean perjudicados por cambios no autorizados.

Existen muchos SGBD que son usados para la creación y administración de sus bases de datos pero se seleccionó como el más adecuado teniendo en cuenta las ventajas y desventajas de cada uno de los existentes al PostgreSQL. (McGraw-Hill, 2008)

- **MySQL**

Es un sistema de gestión de bases de datos relacional, fue creado por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece y la posibilidad de integrar este gestor en un software propietario, ya que de otra

manera se vulneraría la licencia GPL. El principal objetivo de MySQL es la velocidad y robustez. Entre sus principales características se destacan las siguientes: soporta gran cantidad de tipos de datos para las columnas, gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos, aprovecha la potencia de sistemas multiproceso gracias a su implementación multihilo. (Cruz-Chávez, 2009)

Ventajas

1. Para la elaboración de bases de datos no se requiere de amplios requerimientos de hardware, por lo que se puede ejecutar en una máquina que sea de escasos recursos.
2. Los diferentes sistemas se pueden conectar usando socket TCP/IP en cualquier plataforma.
3. Es un sistema fácil de instalar y configurar.
4. Cuando se utilizan aplicaciones Web, en Drupal y la conexión es programada en el lenguaje PHP brinda un mejor rendimiento.

Desventajas:

5. Puede provocar problemas de integridad y la información se puede dañar en entornos donde se gestione mucha información simultáneamente, altas concurrencias en las modificaciones de datos.
6. No es viable para su uso con grandes bases de datos a las que se acceda continuamente ya que no implementa una buena escalabilidad.
7. Carece de soporte para transacciones, rollback y subconsultas.

- **PostgreSQL:** Es uno de los Sistemas de Gestión de Bases de Datos de código abierto más avanzado del mundo. Brinda avanzadas características en materia de seguridad, soporte de aplicaciones, seguimiento y control, rendimiento y almacenamiento de datos especializado, las mismas han tenido gran aceptación entre los usuarios.

PostgreSQL 9.0 ofrece cambios importantes que perfeccionarán muchos aspectos de diseño y rendimiento de aplicaciones de base de datos, como: soporte Windows 64-bit, triggers condicionales y por columna, actualización in-situ desde las versiones 8.3 y 8.4, restricciones de unicidad postergables, y paso de mensajes de alto rendimiento. (PostgreSQL vs. MySQL)

Una de las características que comparte con otros motores de bases de datos es el hecho de ser multiplataforma y de tener varias herramientas para administrar.

Ventajas

1. Posee manejo y control de transacciones para asegurar la consistencia de los datos.
2. Soporta los tipos de datos, cláusulas, funciones y comandos de tipo estándarSQL92/SQL99 y extendidos propios de PostgreSQL. Los tipos de datos internos han sido mejorados incluyendo nuevos tipos.
3. Para una fácil conectividad al servidor PostgreSQL, se implementa un ODBC (Open Database Connectivity) llamado psqIODBC, que es un API de interfaz entre clientes y servidores de bases de datos PostgreSQL. Además se tiene el proveedor de acceso a datos Npgsql que es utilizado en aplicaciones que usan la plataforma .Net para acceder y manipular bases de datos PostgreSQL.

4. PostgreSQL puede operar sobre distintas plataformas incluyendo Linux, Unix, MacOSX, Solaris y últimamente Windows.
5. Posee una gran capacidad de almacenamiento.
6. Posee ciertas características de orientación a objetos, como la herencia entre tablas.
7. Tiene una arquitectura Cliente – Servidor.
8. Tiene una buena escalabilidad ya que es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, soportando una mayor cantidad de peticiones simultáneas a la base de datos de manera correcta; se dice que “PostgreSQL ha llegado a soportar el doble de carga de lo que soporta MySQL” (1996-2012)

Desventajas

1. Consume más recursos que MySQL, por lo que se necesitan mayores características de hardware para ejecutarlo.
2. Aproximadamente es 2 veces más lento que MySQL (aunque en la práctica no se nota esta diferencia). (1996-2012)

* **PgAdmin 3:** Es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados. Está escrita en C++ y utiliza la librería gráfica multiplataforma wxWidgets lo que admite ser usado en diferentes sistemas operativos como Windows, Linux, FreeBSD, Solaris, Mac OS X. Se diseñó para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL, hasta desarrollar bases de datos complejas. Una característica interesante de pgAdmin3 es que, cuando se realizan modificaciones en los objetos,

la misma escribe la/s sentencia/s SQL correspondiente/s, lo que la convierte en un herramienta didáctica. (2008)

Algunas características de este manejador de datos son:

- × Cuenta con una sintaxis SQL editor, servidor en el editor de código.
- × La conexión es utilizando TCP/IP o Unix Domain Sockets y no requiere de drives para comunicarse con el servidor.
- × Muy rápido para la visualización y entrada de datos.
- × Procedimientos almacenados.
- × Transacciones. (Ubuntu, 2008)

1.4.3 HERRAMIENTAS PARA EL DISEÑO DE LA BASE DE DATOS Y EL MODELADO

- **Power-Architect:** Es una herramienta de modelado de datos que fue creada por los diseñadores de almacenamiento de datos. Le brinda la posibilidad a los usuarios de la herramienta ingeniería inversa de bases de datos existentes, realizar perfiles de datos en bases de datos de origen y generar automáticamente los metadatos de ETL. (www.tuinformaticafacil.com, 2010)

Algunas Características de SQL Power-Architect.

- Permite ejecutar operaciones sobre bases de datos a través del API JDBC.
- Brinda la posibilidad de conectarse a varias bases de datos al mismo tiempo.
- Compara modelos de datos y estructuras de bases de datos e identifica las discrepancias.

- Drag-and-drop de las tablas origen y las columnas en el área de trabajo.
- Ingeniería directa/inversa para PostgreSQL, Oracle, MS SQL Server y muchas más.
- Todos los proyectos se guardan en formato XML.

- **Visual Paradigm:** Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (Beatriz, 2009).

- **Rational Rose:** Es una herramienta de modelado visual que posibilita que los diseñadores de bases de datos, analistas, arquitectos, desarrolladores y todos los demás miembros del equipo de desarrollo trabajen juntos, capturando y compartiendo los requerimientos de negocio y dándoles seguimiento a medida que cambian a través del proceso. Proporciona una realización de la metodología ER usando la notación **UML** para unificar a los diseñadores de bases de datos con el equipo de desarrollo de software. Con **UML**, el diseñador de bases de datos puede capturar información tal como restricciones, triggers e índices directamente en el diagrama en lugar de tener que representarlos con propiedades ocultas por fuera. **Rational Rose Data Modeler** da la libertad de pasar del objeto al modelo de datos y sacar ventaja de los tipos de transformación básicos como las relaciones muchos-

a-muchos. Esta herramienta brinda una forma intuitiva de visualizar la arquitectura de las bases de datos y la forma en que se vincula con la aplicación. (GSI, 2007)

1.4.4 HERRAMIENTAS PARA EL MAPEO DE LA BASE DE DATOS

- **Hibernate:** Es una herramienta para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. También es una herramienta ORM completa que ha conseguido en un tiempo record una excelente reputación en la comunidad de desarrollo posicionándose claramente como el producto OpenSource líder en este campo gracias a sus prestaciones, buena documentación y estabilidad.

Hibernate parte de una filosofía de mapear objetos Java "normales", también conocidos en la comunidad como "POJOs" (Plain Old Java Objects), no contempla la posibilidad de automatizar directamente la persistencia de Entity Beans tipo BMP (es decir, generar automáticamente este tipo de objetos), aunque aún así es posible combinar Hibernate con este tipo de beans utilizando los conocidos patrones para la delegación de persistencia en POJOs. (Hibernate)

- **JDO:** ObjectDB para Java/JDO es un acuerdo, con todo la base de datos de objeto de gran alcance (ODBMS) para Java, obediente con el estándar de los objetos de datos (JDO) de Java desarrollado por Sun. Fue escrito enteramente en Java. (Programming_4Us, 2012)

ObjectDB para Java/JDO se diseña para manejar eficientemente las bases de datos de varios tamaños, extendidas de algunos KBs a los centenares de GBs.

Entre sus características están la pequeña huella, el alto rendimiento, el desarrollo rápido, el despliegue fácil y la arquitectura flexible (1-Tier, 2-Tier o N-Grada).

Limitaciones:

- La restricción de las clases de la entidad del máximo 10 y de un millón entidades se opone por archivo de base de datos.

* **Netbeans:** Netbeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento. (affiliates, 2012)

Hoy en día hay disponibles dos productos: El Netbeans IDE y el Netbeans Plataforma.

El Netbeans IDE es un entorno de desarrollo - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE Netbeans. El IDE Netbeans es un producto libre y gratuito sin restricciones de uso.

El Netbeans Platform es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

Ambos productos son de código abierto y gratuito para el uso tanto comercial como no comercial. Posee soporte multiplataforma por lo que puede ser ejecutado en sistemas operativos como Microsoft Windows, GNU/Linux, Macintosh OS X y Solaris El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL).

CAPÍTULO 1

* **Linux (Distribución Ubuntu):** Es una distribución de Linux con base en Debian, esta distribución es una de las más populares y está diseñada para el usuario medio, cuenta con una gran facilidad de uso y de instalación. Esta distribución está basada en la filosofía del software libre y de código abierto, de tal manera que cualquier persona es capaz de mejorarlo (siempre y cuando forme parte de la comunidad). (Rodríguez, 2012)

CONCLUSIONES DEL CAPÍTULO:

En busca de dar seguimiento y tener una pequeña noción de la fundamentación teórica para dar solución al problema a resolver, con la comprensión de este capítulo se logrará dar el aporte necesario para la continuidad de los siguientes capítulos, donde se desglosarán una serie de tareas que ayudarán a completar el trabajo final, con el éxito y la calidad que se requiere.

CAPÍTULO 2: CARACTERÍSTICAS, ANÁLISIS Y DISEÑO DEL SISTEMA.

INTRODUCCIÓN

Para poder continuar con el despliegue detallado de todas las tareas a realizar para lograr el producto final y a raíz de la fundamentación teórica del capítulo anterior, en este se identificarán las características que tendrá el sistema, donde se realiza la descripción de la propuesta para dar solución a la problemática, se hace un levantamiento de los requisitos funcionales y no funcionales del producto a partir de las necesidades del cliente, se describen también los patrones que se utilizaron para el diseño y la arquitectura del mismo además de realizar la descripción de los artefactos que son generados durante la fase de **Planificación – Definición y Desarrollo**.

2.1 PROPUESTA DE LA SOLUCIÓN

La Facultad Regional de Artemisa necesita automatizar todo su sistema y por ende las actividades docentes que se realizan en la misma.

El sistema a desarrollar para el Departamento Docente General de la Facultad debe permitir autenticarse solo a los usuarios planificadora o vicedecana que son los que podrán trabajar en él, ya que no existirán otros usuarios que no sean esos.

* Dentro de las funcionalidades principales que debe tener el sistema después que se autentica la vicedecana se encuentran:

- Gestionar un Usuario
- Modificar el Día Afectado de los Departamentos Docentes de la Facultad

* Dentro de las funcionalidades principales que debe tener el sistema después que se autentica la planificadora se encuentran:

- Planificar el Horario Docente.

- Planificar el Cuidado de Pruebas.
- Llevar el control de Registro de Asistencia a los profesores.

La interfaz principal es la que se encarga de que el usuario se autentique para poder realizar las tareas a las cuales tenga acceso.

La aplicación será robusta, la interfaz externa contará con colores amigables, el sistema será fiable, fácil de manipular y cómodo para trabajar con él.

2.2 TECNOLOGÍAS ESCOGIDAS PARA EL DESARROLLO DE LA APLICACIÓN

*** METODOLOGÍA**

Hoy en día se trabaja en base a la obtención de productos con buena calidad, en el menor tiempo posible y con la realización de la documentación necesaria. Debido a esto es que debemos escoger cuidadosamente que tipo de metodología emplear a la hora de comenzar a trabajar en un proyecto, puesto a que una vez que comencemos a trabajar con una metodología no se debe de cambiar, pues traería consigo conflictos que no se pueden mitigar. En este caso, el proyecto es pequeño, puede existir un cambio rápido de requisitos o requisitos imprecisos, un alto riesgo técnico, se orienta a una entrega rápida de resultados, una alta flexibilidad y consistencia y a tener la calidad necesaria para que pueda ser utilizado por el cliente a raíz de los requisitos analizados. SXP es una metodología ágil que tiene como base las metodologías SCRUM y XP que permiten actualizar los procesos de desarrollo de software para el mejoramiento de su producción, consta de 4 fases, donde cada una desglosada en flujos de trabajo y actividades que generan artefactos.

Se decide utilizar la metodología SXP debido a que con su empleo se evita generar ciertas cantidades de documentación, es una metodología ágil y por sus características es la más adecuada para proyectos de poca duración como este. Esta metodología ayuda a fortalecer el trabajo en equipo, enfocados a una misma dirección y permite seguir de forma clara el avance de las tareas a realizar, a partir

de la inserción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la producción, aumentando así, el nivel de interés del equipo.

*** HERRAMIENTA PARA EL DESARROLLO DEL PRODUCTO**

Java y ExtJS son 2 herramientas muy potentes para la realización de aplicaciones, con características propias y diferentes, pero con fines comunes. Después de haber analizado la problemática se tomó la decisión de realizar la implementación del sistema con Java, un lenguaje de programación muy fácil de aprender y de trabajar, el cual se integrará con Hibernate para la conexión con la Base de Datos. ¿Por qué tomar esta decisión?

*** Aplicación Escritorio**

- Se ejecutará como un programa al dar doble clic sobre el mismo.
- Se necesitaría estudiar el lenguaje de programación a utilizar y la librería para el mapeo de las tablas de la Base de Datos donde se almacenará la información.
- Solo será empleado por los usuarios Planificadora y Vicedecana de Formación.
- No necesariamente hay que tener conexión en la PC que se emplee.

*** Aplicación Web**

- Se necesita un navegador instalado.
- Se necesitaría estudiar los lenguajes de programación, PHP, Javascript, Java (en caso de que se use un marco de desarrollo que lo utilice como jWebSocket) y las librerías para el desarrollo tanto para el mapeo de las

tablas como para emplear métodos predefinidos que ayuden al desarrollo de la aplicación.

- Se emplea cuando pueden acceder un grupo de usuarios al Sistema.
- Se necesitaría tener conexión 24 horas para cuando un usuario quiera acceder a él.

Entonces por la comparación entre una aplicación escritorio a obtener con una aplicación web para resolver el problema queda expuesto que sería más ventajoso el desarrollo de una aplicación de escritorio.

*** GESTOR DE BASE DE DATOS PARA EL ALMACENAMIENTO DE LA INFORMACIÓN**

Después de haber comparado las características de los Gestores de Base Datos PostgreSQL y MySQL, se empleará el PostgreSQL que proporciona y establece más seguridad, persistencia y funcionalidades que se requieren para el almacenamiento de los datos en la aplicación a realizar, siendo entonces el más adecuado para el desarrollo de la misma.

*** HERRAMIENTAS PARA EL DISEÑO DE LA BASE DE DATOS Y EL MODELADO**

El diseño de la Base de Datos se realizó con el Power-Architect debido a que se tenía experiencia con su empleo, es cómodo trabajar con él y responde a los intereses del diseñador, mientras que el Visual Paradigm a pesar de que puede resolver el problema también se seleccionó solo para obtener los diseños de los diagramas como el de entidad – relación y otros que se necesitan mientras que avanza el ciclo de vida del proyecto, como los diagramas de paquetes, de despliegue y de componente ya que el Rational Rose es una herramienta propietaria y no puede emplearse en el Sistema Operativo Linux.

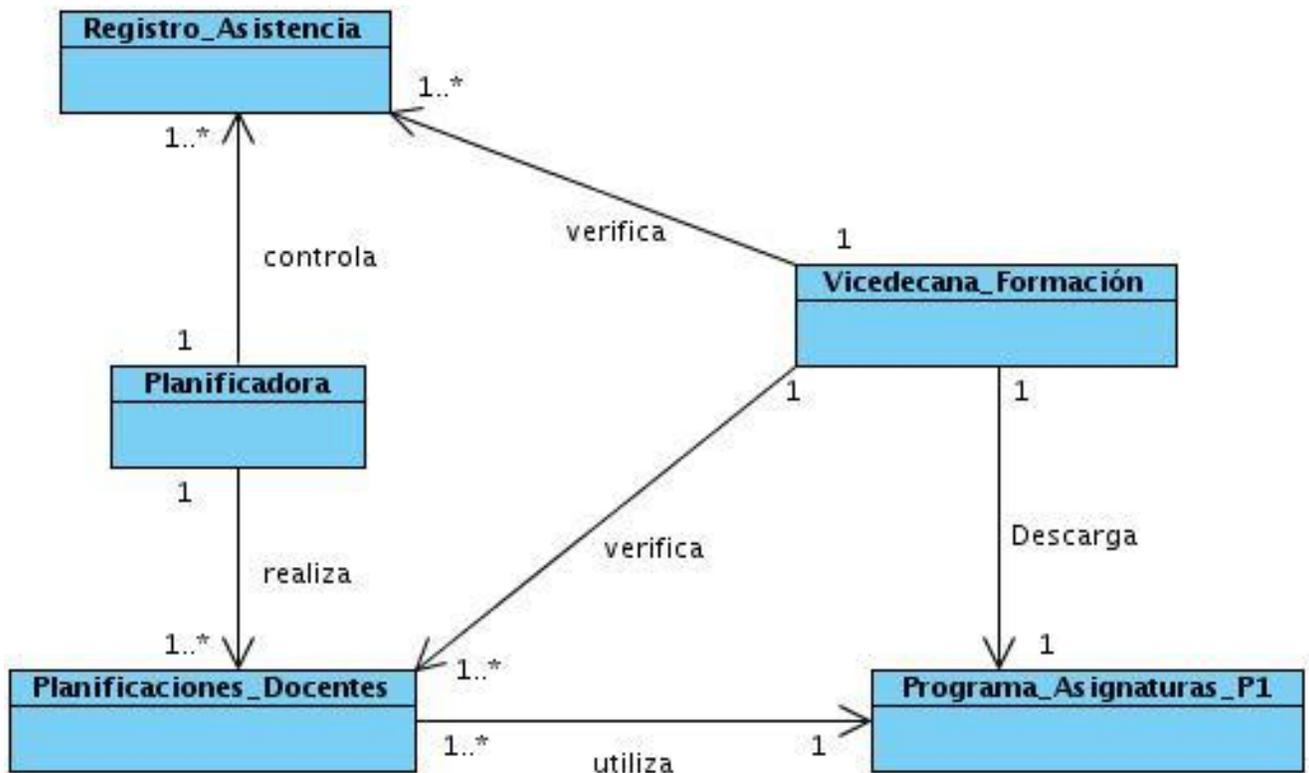
* HIBERNATE VS JDO PARA EL MAPEO DE LA BASE DE DATOS

Para gestionar la información que se almacenará en la Base de Datos del Sistema directamente con Java, se hará uso de Hibernate como librería para el mapeo de objetos relacional, además de ser una tecnología potente y con una gran documentación disponible para su empleo, lo que lo hace diferenciarse aun más de JDO el cual no tiene muchos seguidores tampoco.

2.3 MODELO DE DOMINIO

El modelo de dominio consiste en representar una detallada descripción del negocio mediante un diagrama. El mismo se realiza cuando no están bien definidos los requisitos funcionales del sistema.

A continuación se muestra el modelo de dominio para el sistema que se propone.



2.4 ARTEFACTOS GENERADOS

2.4.1 LISTA DE RESERVA DEL PRODUCTO (LRP)

Una de las actividades más importantes en la metodología SXP es la Lista de Reserva del Producto (**LRP**), la misma consta de una lista de prioridad de los Requisitos Funcionales y no Funcionales del sistema que se va a desarrollar, el ítem de prioridad, la descripción de cada requisito, la estimación en días y por quién fue estimado. La prioridad se puede categorizar en Muy Alta, Alta, Medio y Baja.

Esta Lista puede modificarse mientras avanza el desarrollo del producto. Al final del ciclo de vida del producto, la Lista de Reserva del Producto debe haber asegurado que el mismo sea útil, tenga la calidad requerida a raíz de su funcionamiento, sea fácil de manipular y satisfaga la necesidad del cliente.

Prioridad	Ítem *	Descripción	Estimación	Estimado por
Muy Alta				
	1	Autenticar Usuario	1 día	Analista
	2	Insertar Usuario	2 días	Analista
	3	Eliminar Usuario	1 día	Analista
	4	Modificar Usuario	3 días	Analista
	5	Buscar Usuario	1 día	Analista
	6	Insertar Profesor	3 días	Analista
	7	Eliminar Profesor	2 días	Analista
	8	Modificar Profesor	2 días	Analista
	9	Mostrar los Profesores Disponibles en la Facultad.	2 días	Analista
	10	Mostrar los Profesores no Disponibles de la Facultad.	1 día	Analista
	11	Modificar el día afectado de cada Departamento Docente por su reunión y no pueden impartir clases sus profesores	2 días	Analista
	12	Mostrar el Listado de los miembros de cada Departamento Docente con los días que no pueden impartir clases los mismos	2 días	Analista
	13	Insertar Profesor en el Listado de los profesores de Superación en el Sistema	2 días	Analista
	14	Eliminar Profesor en el Listado de los profesores de Superación en el Sistema	1 día	Analista
	15	Modificar los datos del Profesor en Superación	2 días	Analista
	16	Mostrar el listado de los profesores de la Facultad que realizan la Superación	1 día	Analista
	17	Mostrar la planilla del Programa de las Asignaturas	2 días	Analista
	18	Insertar planificación de horario	3 días	Analista

	19	Eliminar planificación de horario	2 días	Analista
	20	Modificar planificación de horario	3 días	Analista
	21	Exportar la planificación del horario docente	4 días	Analista
	22	Mostrar planificaciones de horario docente	2 días	Analista
	23	Insertar Registro de Asistencia	2 días	Analista
	24	Modificar Registro de Asistencia	2 días	Analista
	25	Exportar Registro de Asistencia	2 días	Analista
	26	Mostrar Registro de Asistencia	1 día	Analista
	27	Insertar planificación del cuidado de prueba	3 días	Analista
	28	Modificar planificación del cuidado de prueba	2 días	Analista
	29	Exportar la planificación del cuidado de prueba	2 días	Analista
	30	Mostrar planificación de cuidado de prueba	2 días	Analista
Requisitos no Funcionales (RNF)				
	31	En el sistema o aplicación solo trabajarán la Vicedecana de Formación y la Planificadora Docente (Usabilidad).		
	32	Tendrá disponible un mapa como ayuda para su utilización. (Usabilidad)		
	33	Debe estar bien definida la seguridad. Se ha establecido un sistema de roles para usuarios diferentes como la planificadora y la Vicedecana de Formación. Cada uno de ellos tendrá acceso solo a ejecutar las acciones que le (Confidencialidad)		
	34	El sistema puede quedar fuera de servicio en caso de fallos por 1 o 2 horas solamente para su reparación.(Disponibilidad)		
	35	En caso de que falle el sistema, no se pierden los datos almacenados		

		ya que no le ocasionará problemas al Gestor de Base de Datos.(Integridad)		
	36	Las excepciones que se produzcan con la utilización del sistema se mostrarán a través de mensajes o serán tratadas mediante una llamada a un gestor de excepciones que informará al usuario el error a través de la interfaz.(Soporte)		
	37	El sistema debe responder las peticiones del usuario en un tiempo razonable, a pesar de que tiene que hacer cálculos complejos y va a interactuar con la base de datos.(Soporte)		
	38	Solo tendrá acceso a las modificaciones o mantenimiento del sistema el informático del Departamento.(Soporte)		
	39	El sistema será fácil para el mantenimiento, de configuración sencilla y factible para los clientes. (Soporte)		
	40	Se necesita una PC con más de 512 mg de RAM.(Hardware)		
	41	Se necesita como Gestor de Base de Datos el PostgreSQL la versión 8.4.(Software)		

2.4.2 HISTORIA DE USUARIO Y TAREAS DE LA INGENIERÍA CORRESPONDIENTE

La Historia de Usuario (HU) es otra de las actividades que se realiza dentro de la metodología SXP. La misma depende de la Lista de Reserva del Producto y en ella se describen detalladamente las operaciones que debe realizar la aplicación y un grupo de observaciones o condiciones para que este se cumpla, estas descripciones deben ser claras y concisas debido a que el objetivo de la Historia de Usuario es que el cliente entienda cada una de las funcionalidades del sistema además de que también se utiliza como guía para la construcción del producto.

Nombre HU 1: Autenticar Usuario

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Autenticar Usuario
Modificación de Historia de Usuario Número: 1	
Usuario: Liana Abreu Guerra	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1 día
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: El objetivo de la historia de usuario es permitir que un usuario se autentique para darle los permisos o acceso solo a las funcionalidades que este puede ejercer.	
Observaciones: Si el usuario no se autentica, tampoco podrá realizar ninguna operación deshabilitada.	
Prototipo de interfaz: Ver Imagen 1	

Tarea de Ingeniería

Número Tarea: 1.1	Número Historia de Usuario: HU_1
Nombre Tarea: Autenticar Usuario	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 16/01/2012	Fecha Fin: 16/01/2012
Programador Responsable: Liana Abreu Guerra	
<p>Descripción: El objetivo la tarea es permitir que un usuario se autentique para darle los permisos o acceso solo a las funcionalidades que este puede ejercer.</p>	

Nombre HU_11: Gestionar planificación de horario docente

Historia de Usuario	
Número: HU_11	Nombre Historia de Usuario: Gestionar planificación de horario docente
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Liana Abreu Guerra	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 8 días
Riesgo en Desarrollo: Alta	Puntos Reales: 8
<p>Descripción: En la siguiente Historia de Usuario es donde se gestionan los datos que lleva el modelo de planificación del horario docente y dentro de sus operaciones está:</p> <p>Insertar planificación de horario: Se insertan los datos que conforman una planificación de horario a un grupo determinado.</p>	

<p>Eliminar planificación de horario: Se elimina la planificación realizada a un grupo determinado</p>
<p>Modificar planificación de horario: Se modifican los datos de la planificación realizada a un grupo determinado.</p>
<p>Observaciones: Para que esta HU se realice, antes debe de haberse autenticado el usuario con rol planificadora.</p> <p>Para eliminar y modificar alguna planificación de horario, esta tiene que existir en la Base de datos.</p>
<p>Prototipo de interfaz: Ver Imagen 2 y Imagen 3</p>

Tarea de Ingeniería	
Número Tarea: 11.1	Número Historia de Usuario: HU_11
Nombre Tarea: Insertar planificación de horario para un grupo determinado	
Tipo de Tarea : Desarrollo	Puntos Estimados: 3 días
Fecha Inicio: 22/02/2012	Fecha Fin: 24/02/2012
Programador Responsable: Liana Abreu Guerra	
Descripción: Se insertan los datos que conforman una planificación de horario a un grupo determinado	

Tarea de Ingeniería	
Número Tarea: 11.2	Número Historia de Usuario: HU_11
Nombre Tarea: Eliminar planificación de horario para un grupo determinado	

CAPÍTULO 2

Tipo de Tarea : Desarrollo	Puntos Estimados: 2 días
Fecha Inicio: 25/02/2012	Fecha Fin: 27/02/2012
Programador Responsable: Liana Abreu Guerra	
Descripción: Se elimina la planificación realizada a un grupo determinado	

Tarea de Ingeniería	
Número Tarea: 11.3	Número Historia de Usuario: HU_11
Nombre Tarea: Modificar planificación de horario a un grupo determinado.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 3 días
Fecha Inicio: 28/02/2012	Fecha Fin: 01/03/2012
Programador Responsable: Liana Abreu Guerra	
Descripción: Se modifican los datos de la planificación realizada a un grupo determinado.	

Nombre HU 14: Gestionar Registro de Asistencia

Historia de Usuario	
Número: HU_14	Nombre Historia de Usuario: Gestionar Registro de Asistencia
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Liana Abreu Guerra	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 4 días
Riesgo en Desarrollo: Medio	Puntos Reales: 4

<p>Descripción: La siguiente historia de usuario consiste en gestionar el registro de asistencia, teniendo como operaciones para efectuar:</p>
<p>Insertar Registro de Asistencia: Se insertan los datos que conforman el registro de asistencia.</p>
<p>Modificar Registro de Asistencia: Se modifican los datos que conforman el registro de asistencia en caso de algún error.</p>
<p>Observaciones: Para realizar alguna de estas operaciones antes debe de autenticarse el usuario con rol planificadora.</p> <p>Todos los datos para elaborar el registro de asistencia los introduce el usuario.</p> <p>Para modificar un registro de asistencia, el mismo debe de existir en la base de datos.</p>
<p>Prototipo de interfaz: Ver Imagen 4 y Imagen 5</p>

Tarea de Ingeniería	
Número Tarea: 14.1	Número Historia de Usuario: HU_14
Nombre Tarea: Insertar Registro de Asistencia	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 días
Fecha Inicio: 09/03/2012	Fecha Fin: 10/03/2012
Programador Responsable: Liana Abreu Guerra	
Descripción: Se insertan los datos que conforman el registro de asistencia.	

Tarea de Ingeniería	
Número Tarea: 14.2	Número Historia de Usuario: HU_14
Nombre Tarea: Modificar Registro de Asistencia	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 días
Fecha Inicio: 12/03/2012	Fecha Fin: 13/03/2012
Programador Responsable: Liana Abreu Guerra	
Descripción: Se modifican los datos que conforman el registro de asistencia en caso de algún error.	

Nombre HU 17: Gestionar planificación de cuidados de pruebas

Historia de Usuario	
Número: HU_17	Nombre Historia de Usuario: Gestionar planificación de cuidados de pruebas
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Liana Abreu Guerra	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta	Puntos Estimados: 5 días
Riesgo en Desarrollo: Medio	Puntos Reales: 5
Descripción: La historia de usuario consiste en gestionar los datos sobre el año docente que tiene prueba y sus operaciones son:	
Insertar planificación de cuidado de prueba: Inserta los datos necesarios que conforman el modelo de planificación de cuidado de prueba.	
Modificar planificación del cuidado de prueba: Permite modificar los datos	

insertados por error o en caso de algún cambio inesperado, por ejemplo, cuando aplazan o cancelan alguna prueba por cualquier motivo o cambia el plan de estudio de alguna asignatura que no lleve prueba.

Observaciones: Antes debe de haberse autenticado el usuario con rol planificadora.

Los datos los inserta el usuario por parámetros

Prototipo de interfaz: Ver [Imagen 6](#)

Tarea de Ingeniería	
Número Tarea: 17.1	Número Historia de Usuario: HU_17
Nombre Tarea: Insertar planificación de cuidado de prueba	
Tipo de Tarea : Desarrollo	Puntos Estimados: 3 días
Fecha Inicio: 17/03/2012	Fecha Fin: 20/03/2012
Programador Responsable: Liana Abreu Guerra	
Descripción: Esta tarea permite insertar los datos necesarios que conforman el modelo de planificación de cuidado de prueba.	

Tarea de Ingeniería	
Número Tarea: 17.2	Número Historia de Usuario: HU_17
Nombre Tarea: Modificar planificación de cuidado de prueba	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 días

Fecha Inicio: 23/03/2012	Fecha Fin: 24/03/2012
Programador Responsable: Liana Abreu Guerra	
Descripción: Esta tarea permite modificar los datos insertados por error o en caso de algún cambio inesperado, por ejemplo: Cuando aplazan o cancelan alguna prueba por cualquier motivo o cambia el plan de estudio de alguna asignatura que no lleve prueba.	

2.4.3 PLAN RELEASES

En el Plan de Release se definen las iteraciones para la entrega del producto. En cada iteración se encuentran una serie de Historias de Usuario (HU) definidas para implementarse y donde cada iteración está organizada teniendo en cuenta la prioridad de cada una de las HU.

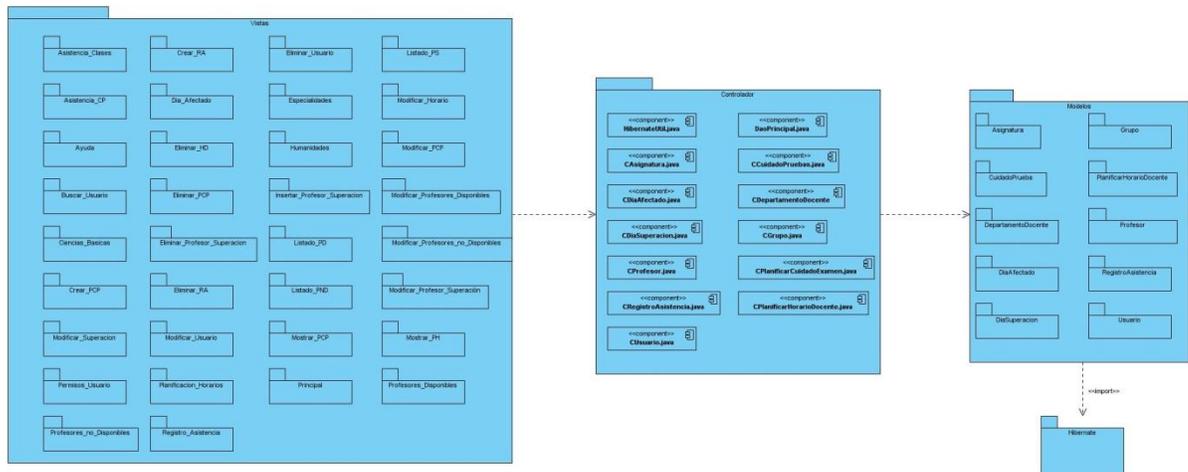
La planificación en el Plan Release quedó de la siguiente manera:

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
2	En esta iteración se van a desarrollar las Historia de Usuario con prioridad muy alta	HU_1, HU_2 , HU_3, HU_4, HU_5, HU_6, HU_7, HU_8, HU_9, HU_10, HU_11, HU_12, HU_13, HU_14, HU_15 ,HU_16, HU_17, HU_18, HU_19	12 semanas

2.5 DISEÑO CON METÁFORAS.

El diseño de metáfora no es más que la descripción de cómo debería funcionar el sistema. Este se genera a partir del artefacto Modelo de Diseño que está compuesto por el diagrama de paquetes del sistema.

A continuación se muestra el diagrama de paquetes del sistema propuesto:



Descripción:

Modelos: En este paquete se registran todos los modelos o tablas de la base de datos a emplear generados a través de la librería Hibernate.

Vistas: En este paquete se registran las interfaces externa de la aplicación con las cuales interactúa el usuario.

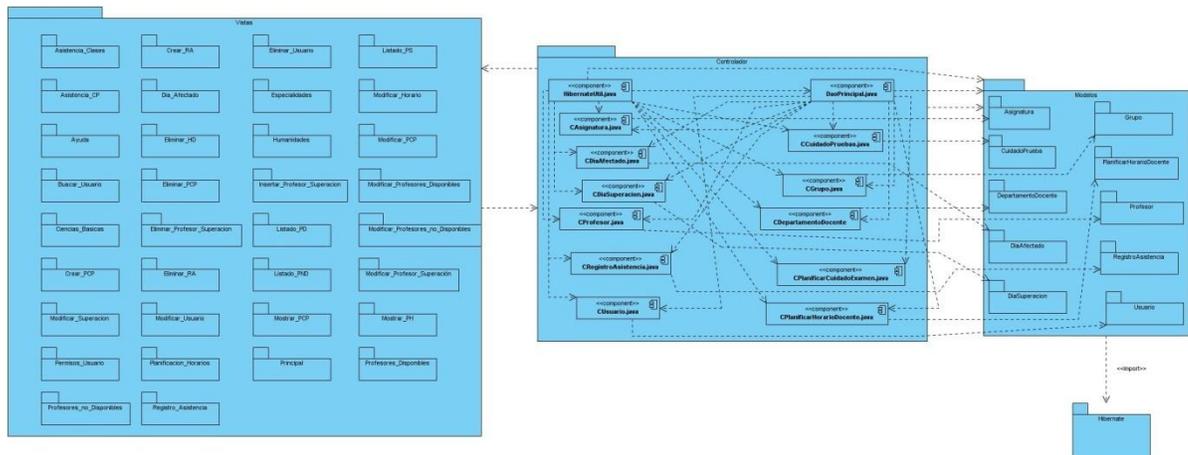
Controlador: Este paquete está integrado por las clases controladoras de los modelos que se generan de la Base de Datos y que se encargarán del manejo de los datos que almacenaremos en la misma.

Hibernate: Librería utilizada para el mapeo de los modelos.

2.6 DIAGRAMA DE COMPONENTES

El diagrama de componentes describe los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de la aplicación los cuales son simples archivos, paquetes y bibliotecas cargadas dinámicamente.

A continuación se presenta el diagrama de componentes para el sistema que se propone.



En este diagrama se muestran las relaciones de cada uno de los componentes que contiene cada paquete.

En el paquete Controlador las clases se relacionan con las vistas en las que sea llamado una funcionalidad de la misma, ya que ellas son las clases controladoras y almacenan una serie de funciones para el desarrollo de la aplicación.

Ejemplo: La clase controladora Cusuario.java se relaciona con las vistas Modificar_Usuario.java y Buscar_Usuario.java.

Las clases controladoras se relacionan entre sí porque el componente daoPrincipal.java es el que contiene los métodos generales que son llamados en las clases controladoras pertenecientes a cada modelo.

Las clases controladoras se relacionan con los modelos puestos a que las mismas contienen funcionalidades de los modelos que le correspondan a cada una.

Ejemplo:

CCuidadoPrueba.java se relaciona con el modelo CuidadoPrueba.java

CDíaAfectado. Java se relaciona con el modelo DiaAfectado.java

y así sucesivamente.

Los modelos son importados mediante Hibernate con el componente hibernate.cfg.xml.

Ejemplo: daoPrincipal.java con el componente Asignatura.java.

2.7 PATRONES UTILIZADOS

- Patrón de Diseño

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos. El principio de solución de este patrón se basa en que una instancia de un objeto tiene que ser creada por el objeto que tiene la información para ello.

- Patrón arquitectónico

Modelo Vista-Controlador (MVC): Este permite que en el modelo se administre del comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador). En la vista maneja la visualización de la información y en el controlador controla el flujo entre la vista y el modelo (los datos).

CONCLUSIONES DEL CAPÍTULO

Al finalizar este capítulo se realizaron un grupo de tareas que ayudaron a la organización y desarrollo de la implementación del producto, haciendo más fácil el trabajo del programador.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

INTRODUCCIÓN

En este capítulo se elabora la solución propuesta del Sistema de Gestión para la Planificación Docente en la Facultad Regional “Mártires de Artemisa” evidenciando sus funcionalidades y concluyendo el mismo con los diseños de casos de pruebas que validen la implementación realizada después de haber dado a conocer el método de prueba seleccionado para la realización de dichas pruebas.

El Sistema de Gestión para la Planificación Docente en la Facultad Regional “Mártires de Artemisa”, es un producto el cual permite almacenar, modificar, eliminar y acceder a la información que se genera en dicho Departamento y que contiene un grupo de funcionalidades para manipular la información generada.

*** FUNCIONALIDADES OBTENIDAS**

- Gestionar usuario
 - * Insertar, Eliminar y Modificar Usuario
- Gestionar Planificaciones
 - * Planificación de Horarios
 - Insertar, Eliminar, Modificar y Mostrar Horarios Docentes
 - * Planificación de Cuidados de Prueba
 - Insertar, Eliminar, Modificar y Mostrar Cuidados de Prueba
- Control del Registro de Asistencia.
 - * Registros de Asistencia a Clases y Registros de Cuidados de Prueba
 - Insertar, Eliminar, Modificar y Buscar Registro de Asistencia

3.1 CASOS DE PRUEBA

Las pruebas de aceptación son una actividad en la cual un sistema o componente es ejecutado bajo los requerimientos especificados, es decir, son básicamente pruebas funcionales sobre el sistema completo. Los resultados son observados y registrados, de manera que encontrar un error que no había sido detectado con anterioridad es un objetivo fundamental y por tanto es importante resaltar que las pruebas no pueden asegurar la ausencia de defectos, solo demuestran que existen defectos en el software.

Estas pruebas las realiza el cliente, las mismas no se realizan durante el desarrollo, sino sobre el producto terminado e integrado, o pudiera ser una versión del producto o una iteración funcional pactada previamente con el cliente.

Actualmente existen varios tipos de métodos para llevar a cabo la realización de las pruebas, dos de los más utilizados se especifican a continuación:

3.1.1 MÉTODOS DE PRUEBA.

Dos de los métodos de prueba más conocidos que se le realizan a un sistema son: los métodos de caja blanca y los de caja negra, permitiendo que se tracen pruebas en la menor cantidad de tiempo posible.

La prueba de caja blanca del software: Se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
2. Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
3. Ejecuten todos los bucles en sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de caja blanca como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

La prueba de caja negra: Es también conocida por varios nombres como pruebas funcionales, pruebas de caja opaca, pruebas de entrada/salida y pruebas inducidas por los datos. Estas pruebas son las que no tienen en cuenta el código, solo necesita saber cuáles pueden ser las posibles entradas sin necesidad de entender cómo se deben obtener las salidas; donde se trata de encontrar errores en la interfaz mientras se está usando, el cómo luce, se maneja, etc., es decir, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

El método de prueba que se le realizó a la aplicación fue la de caja negra, realizando los casos de prueba de aceptación, ya que lo que se quiere es que la aplicación resuelva el problema sin importar el código del mismo.

3.1.2 CASOS DE PRUEBA

Caso de Prueba para la HU_1: Autenticar Usuario

Caso de Prueba de Aceptación	
Código Caso de Prueba: SGPDFRMA-HU_01-1	Nombre Historia de Usuario: Autenticar Usuario
Nombre de la persona que realiza la prueba: Liana Abreu Guerra	
Descripción de la Prueba: Esta prueba consiste en la autenticación de un usuario para poder acceder a las funcionalidades que solo él tiene permisos.	
Condiciones de Ejecución: Es necesario definir los usuarios que tendrán acceso a la aplicación.	

<p>Entrada / Pasos de ejecución: Se entra por parámetro el Nick del usuario en la aplicación y la contraseña correspondiente al mismo.</p> <p>Una vez realizada la entrada dar clic en el botón Aceptar.</p>
<p>Resultado Esperado: Se espera que al autenticarse el usuario, se activen las operaciones a las que el mismo tiene acceso.</p>
<p>Evaluación de la Prueba: Satisfactoria</p>
<p>Entrada / Pasos de ejecución: En la interfaz principal se autentica el usuario que cumpla con el rol de vicedecana. Luego en el menú principal se selecciona el submenú Archivo, después en Gestionar Usuario, seleccionar Modificar Usuario y aparecerá un nuevo formulario donde se pasa por parámetros el usuario a modificar y se seleccionan los campos que se modificarán de ese usuario, seguidamente los datos por los que se sustituirán los anteriores y luego dar clic en el botón Actualizar.</p>
<p>Resultado Esperado: Se espera que los datos del usuario a modificar se actualicen correctamente luego de ser modificados.</p>
<p>Evaluación de la Prueba: Satisfactoria</p>

Caso de Prueba para la HU_11: Gestionar planificación de horario docente.

Caso de Prueba de Aceptación	
<p>Código Caso de Prueba: SGPDFRMA-HU_11-1</p>	<p>Nombre Historia de Usuario: Gestionar planificación del horario docente.</p>
<p>Nombre de la persona que realiza la prueba: Liana Abreu Guerra</p>	
<p>Descripción de la Prueba: Esta prueba consiste en insertar los datos en el modelo de planificación de horario para conformar el mismo.</p>	
<p>Condiciones de Ejecución: Debe de haberse autenticado el usuario con rol planificadora y para poder insertar los datos en el modelo de planificación, antes debe</p>	

de haberse entrado por parámetros el grupo, la semana y la fecha inicio y fecha fin de la semana a planificar como campos obligatorios.

Entrada / Pasos de ejecución: En la interfaz principal se autentica el usuario que cumpla con el rol de planificadora. Luego se selecciona el menú principal Operaciones, después el submenú Insertar y luego Horario Docente. Al dar clic en este aparece un nuevo formulario el cual tiene una serie de campos a llenar para planificar el horario docente, luego se insertan los datos obligatorios como el grupo, la semana y la fecha de inicio y fecha fin de la semana correspondiente a la planificación, luego los demás datos que conforman el horario y una vez que termine de llenarse el modelo, dar clic en el botón Insertar.

Resultado Esperado: Se espera con esta prueba que cuando se autentique el usuario con rol planificadora esta sea la única con los permisos para desarrollar la funcionalidad y que cuando se entre por parámetros los campos obligatorios y se comience a llenar el modelo de la planificación con los datos respectivos, estos se guarden en la Base de Datos luego de haber dado clic en el botón Insertar.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba: SGPDFRMA-HU_11-2	Nombre Historia de Usuario: Gestionar planificación del horario docente.
Nombre de la persona que realiza la prueba: Liana Abreu Guerra	
Descripción de la Prueba: Esta prueba consiste en eliminar la planificación realizada a un grupo determinado.	
Condiciones de Ejecución: Para poder eliminar una planificación de horario docente tienen que existir en la base de datos y antes debe de autenticarse el usuario con rol planificadora.	

Entrada / Pasos de ejecución: En la interfaz principal se autentica el usuario que cumpla con el rol de planificadora. Luego se selecciona el menú principal Operaciones, después Insertar y en Horario Docente dar clic, acción que al ejecutarse hace que aparezca un nuevo formulario, el cual tiene una serie de campos donde en la tabla aparecerán los datos de la planificación de horario a eliminar luego de que se inserte primeramente el grupo y la semana a eliminar, una vez insertados estos datos, dar clic en el botón Eliminar.

Resultado Esperado: Se espera con esta prueba que cuando se dé clic en el botón Eliminar, la planificación perteneciente al grupo y la semana insertados por parámetros no exista en la Base de Datos.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba: SGPDFRMA-HU_11-3	Nombre Historia de Usuario: Gestionar planificación de horario docente.
Nombre de la persona que realiza la prueba: Liana Abreu Guerra	
Descripción de la Prueba: Esta prueba consiste en modificar los datos insertados erróneamente o por necesidad de hacer cambios en alguna de las planificaciones realizadas con anterioridad cuando hay algún cambio a última hora que no estaba previsto.	
Condiciones de Ejecución: Debe de haberse autenticado el usuario con rol planificadora y solo se podrán modificar los datos de una planificación existente o cuando ya existen datos insertados en la planificación.	
Entrada / Pasos de ejecución: En la interfaz principal se autentica el usuario que cumpla con el rol de planificadora. Luego se selecciona el menú principal Operaciones, después el submenú Insertar y luego Horario Docente. Al dar clic en este aparece un nuevo formulario el cual tiene una serie de campos a llenar para	

planificar el horario docente así como la tabla para ir introduciendo los datos de la planificación, luego de que esta tabla tenga algún dato y requieran ser modificados, se selecciona el día a modificar, los nuevos valores del horario docente para ese día y se da clic en el botón Modificar.

Resultado Esperado: Se espera con esta prueba que cuando se autentique el usuario con rol planificadora esta sea la única con los permisos para desarrollar la funcionalidad y que cuando se quiera modificar algún campo, este se actualice en la Base de Datos, luego de haber seleccionado los datos nuevos y dar clic en el botón Modificar.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba para la HU_14: Gestionar Registro de Asistencia

Caso de Prueba de Aceptación	
Código Caso de Prueba: SGPDFRMA-HU_14-1	Nombre Historia de Usuario: Gestionar Registro de Asistencia
Nombre de la persona que realiza la prueba: Liana Abreu Guerra	
Descripción de la Prueba: Esta prueba consiste en insertar los datos en el modelo del Registro de Asistencia para elaborar el mismo.	
Condiciones de Ejecución: Para que esta pueda realizarse, el usuario con rol planificadora debe de haberse autenticado en el Sistema.	
Entrada / Pasos de ejecución: En la interfaz principal se autentica el usuario que cumpla con el rol de planificadora. Luego se selecciona el menú principal Operaciones, después Insertar y en Registro de Asistencia dar clic, acción que al ejecutarse hace que aparezca un nuevo formulario, el cual tiene una serie de campos a llenar que conformarán el Registro de Asistencia, se seleccionan y se insertan los datos correspondientes y se da clic en el botón Aceptar.	

<p>Resultado Esperado: Se espera con esta prueba que cuando se autentique el usuario con rol planificadora esta sea la única con los permisos para desarrollar la funcionalidad y que cuando se inserten los datos en el modelo del Registro de Asistencia, los mismos se guarden en la Base de Datos al dar clic en el botón Aceptar.</p>
<p>Evaluación de la Prueba: Satisfactoria</p>

Caso de Prueba de Aceptación	
<p>Código Caso de Prueba: SGPDFRMA-HU_14-2</p>	<p>Nombre Historia de Usuario: Gestionar Registro de Asistencia</p>
<p>Nombre de la persona que realiza la prueba: Liana Abreu Guerra</p>	
<p>Descripción de la Prueba: Esta prueba consiste en modificar los datos del modelo del Registro de Asistencia en caso de algún cambio o error.</p>	
<p>Condiciones de Ejecución: Para poder modificar los datos de un registro de Asistencia se debe de haber insertado por parámetros la fecha del registro deseado y antes debe de haberse autenticado el usuario con rol planificadora.</p>	
<p>Entrada / Pasos de ejecución: En la interfaz principal se autentica el usuario que cumpla con el rol de planificadora. Luego se selecciona el menú principal Operaciones, después Modificar y en Registro de Asistencia dar clic, acción que al ejecutarse hace que aparezca un nuevo formulario, el cual tiene una serie de campos donde en la tabla serán mostrados los datos del registro a modificar luego de haber insertado por parámetros la fecha de dicho registro y dar clic en el botón Mostrar. Una vez que el registro se muestre; se seleccionan los nuevos valores y se actualiza el registro en la base de datos al dar clic en el botón Aceptar.</p>	
<p>Resultado Esperado: Se espera con esta prueba que cuando se autentique el usuario con rol planificadora esta sea la única con los permisos para desarrollar la funcionalidad y que al introducir la fecha del registro a modificar y modificar los datos deseados, estos se actualicen correctamente en la Base de Datos cuando se dé clic</p>	

en el botón Aceptar.

Evaluación de la Prueba: Satisfactoria

Caso de Prueba para la HU_17: Gestionar planificación de cuidado de prueba

Caso de Prueba de Aceptación	
Código Caso de Prueba: SGPDFRMA-HU_17-1	Nombre Historia de Usuario: Gestionar planificación de cuidado de prueba
Nombre de la persona que realiza la prueba: Liana Abreu Guerra	
Descripción de la Prueba: Esta prueba consiste en insertar los datos necesarios en el modelo de planificación de cuidado de prueba	
Condiciones de Ejecución: Para que esta pueda realizarse, la planificadora debe de haberse autenticado en el Sistema y no debe existir la planificación en la Base de Datos.	
Entrada / Pasos de ejecución: En la interfaz principal se autentica el usuario que cumpla con el rol de planificadora. Luego se selecciona el menú principal Operaciones, después Insertar y en Cuidado de Prueba dar clic, acción que al ejecutarse hace que aparezca un nuevo formulario, el cual tiene una serie de campos donde en la tabla se mostrarán los datos acabados de insertar al dar clic en el botón Mostrar, y los datos se almacenarán en la base de datos luego de seleccionar los datos correspondientes al modelo y dar clic en el botón Guardar.	
Resultado Esperado: Se espera con esta prueba que cuando se autentique el usuario con rol planificadora esta sea la única con los permisos para desarrollar la funcionalidad y que cuando se inserten los datos en el modelo de planificación de cuidado de pruebas, los mismos se guarden automáticamente en la Base de Datos al dar clic en el botón Guardar y se muestren en la tabla para verificar su inserción al dar clic en el botón Mostrar.	

Evaluación de la Prueba: Satisfactoria

Caso de Prueba de Aceptación	
Código Caso de Prueba: SGPDFRMA-HU_17-2	Nombre Historia de Usuario: Gestionar planificación de cuidado de prueba
Nombre de la persona que realiza la prueba: Liana Abreu Guerra	
Descripción de la Prueba: Esta prueba consiste en modificar los datos insertados por error o en caso de algún cambio inesperado, por ejemplo: Cuando aplazan o cancelan alguna prueba por cualquier motivo o cambia el plan de estudio de alguna asignatura que no lleve prueba.	
Condiciones de Ejecución: Para modificar los datos de una planificación, la misma debe de existir en la Base de Datos y antes el usuario con rol planificadora debe de haberse autenticado en el Sistema.	
Entrada / Pasos de ejecución: En la interfaz principal se autentica el usuario que cumpla con el rol de planificadora. Luego se selecciona el menú principal Operaciones, después el submenú Modificar, luego Cuidado de Prueba, al dar clic en este aparece un nuevo formulario el cual contiene una tabla que muestra la planificación a modificar luego de haber insertado la fecha o la semana de planificación, se inserta el id de la planificación a modificar, se seleccionan o insertan los nuevos valores que tendrá la planificación, se da clic en el botón Guardar y este se actualiza automáticamente en la base de datos.	
Resultado Esperado: Se espera con esta prueba que cuando se autentique el usuario con rol planificadora esta sea la única con los permisos para desarrollar la funcionalidad y que cuando se quiera modificar algún campo, este se actualice en la Base de Datos al dar clic en el botón Guardar.	
Evaluación de la Prueba: Satisfactoria	

*** RESULTADOS OBTENIDOS**

Se obtuvo la versión 1.0 del Sistema de Gestión para la Planificación Docente en la Facultad Regional “Mártires de Artemisa”, el cual permite que se realicen los procesos de planificación docente en menor tiempo y posibilita la integridad de la información que se genera.

*** APORTES SOCIALES**

- * Ahorro de tiempo y rapidez en la realización del trabajo.
- * Rapidez en los reportes sistemáticos.
- * Seguridad y organización de la información que se maneja.

CONCLUSIONES DEL CAPÍTULO

Para realizar las pruebas al Sistema de Gestión para la planificación Docente en la Facultad Regional “Mártires de Artemisa” se diseñaron y ejecutaron los casos de prueba para chequear los errores de funcionalidad en la aplicación donde se mostraron los resultados obtenidos en dichas pruebas, resaltando las funcionalidades de la aplicación y obteniendo la primera versión del Sistema de Gestión para la Planificación Docente en la Facultad Regional “Mártires de Artemisa”.

CONCLUSIONES GENERALES

- Se fundamentó teóricamente la investigación.
- Se identificaron los requerimientos funcionales y no funcionales.
- Se especificaron las herramientas y tecnologías a utilizar para el desarrollo de la solución.
- Se realizó el análisis y diseño de la propuesta de solución.
- Se implementó la solución diseñada dándole respuesta a los requerimientos identificados.
- Se le realizaron las pruebas de aceptación al producto obtenido.

RECOMENDACIONES

- ✓ Emplear el Sistema en otras instituciones educacionales, principalmente con características similares a los de la Facultad Regional “Mártires de Artemisa”.

BIBLIOGRAFÍA REFERENCIADA

Walter Hidalgo Córdova. 2009. Diseño de Sistemas de Información. *Diseño de Sistemas de Información*. [Online] 02 14, 2009. [Cited: 02 15, 2012.]
<http://es.scribd.com/doc/59660069/Rational-Rose-Caracteristicas>.

2009. [Online] diciembre 3, 2009.
<http://kezadalp.wordpress.com/2009/12/03/metodologias-de-desarrollo-de-software/>.

2008. [Online] 2008. metodologias/principales-caractersticas.html .

2009. [Online] 2009. principales-caractersticas.html .

2008. [Online] 2008. http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.

2008. [Online] 2008. cuadro.bmp.html.

2009. [Online] 2009. http://danielpecos.com/docs/mysql_postgres/x15.html.

affiliates, Oracle Corporation and/or its. 2012. NetBeans. [Online] 2012.
http://netbeans.org/index_es.html.

Alvarez, Dailé: Martínez Elizeth. 2009. *Sistema de Gestión para la Planificación Docente en la Facultad 5*. Ciudad Habana : s.n., 2009.

Beatriz, Lopez Isis Imelda: Rivera Amalia. 2009. *Sistema de Control Presupuestal*. 2009.

2009. Características de java. [Online] 2009. [caracteristicas de java.html](http://caracteristicas-de-java.html).

Computación Aplicada al Desarrollo SA de CV. Java. [Online] <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>www.cad.com.mx/historia_del_lenguaje_java.htm.

Contreras, Yimian de Lyz: Rivero Soleydi. GestioPolis.com. *GestioPolis.com*. [Online] [Cited: 1 19, 2012.] <http://www.gestiopolis.com/administracion-estrategia/sistemas-de-gestion-de-informacion-en-estudio-de-medio-ambiente.htm>.

2012. definicion.de. *definicion.de*. [Online] 2012. [Cited: 1 19, 2012.]
<http://definicion.de/informacion/>.

2012. Definicion.de. [Online] 2012. <http://definicion.de/sistema-de-informacion/>.

Díaz, Maria Isabel: Mota, Jonaidi: Tovar, Johana. 2008. [Online] octubre 2008.
<http://johanatov.blogspot.es/>.

Hibernate.

- Institution, The British Standards. 2012.** BSI. [Online] 2012.
<http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/> .
- Ivan Thompson. 2008.** PromonegocioS.net. *PromonegocioS.net*. [Online] 10 2008. [Cited: 1 19, 2012.] <http://www.promonegocios.net/mercadotecnia/definicion-informacion.html>.
- Javier, Enrique:. 2012.** [Online] 2012.
<http://www.buenastareas.com/ensayos/Trabajo/987038.html>.
- 2009.** Lenguajes de programación . [Online] 2009. <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.
- Livia M. Reyes Ramírez . 2007.** [Online] 1 4, 2007. [Cited: 1 19, 2012.]
http://bvs.sld.cu/revistas/aci/vol15_1_07/aci06107.htm.
- McGraw-Hill, Editorial. 2008.** mailxmail.com. [Online] octubre 21, 2008.
<http://www.mailxmail.com/curso-sistemas-bases-datos/sgbd-sistemas-gestores-bases-datos-introduccion>.
- Palmero Berberena, Raúl. 2010.** GestioPolis. *GestioPolis*. [Online] Julio 2, 2010. [Cited: 12 20, 2011.] <http://www.gestiopolis.com/administracion-estrategia-2/sistemas-integrados-gestion-informacion.htm>.
- Palmero, Raul. 2010.** [Online] julio 2, 2010. <http://www.gestiopolis.com/administracion-estrategia-2/sistemas-integrados-gestion-informacion.htm> .
- 2012.** Portal UCI. *Portal UCI*. [Online] 02 10, 2012. [Cited: 03 12, 2012.]
http://www.ecured.cu/index.php/Universidad_de_las_Ciencias_Inform%C3%A1ticas.
- PosgreSQL vs. MySQL. [Online] <http://www.postgresql.org/about/press/presskit90/es/>.
- Premonegocio. 2008.** [Online] octubre 2008.
<http://www.promonegocios.net/mercadotecnia/definicion-informacion.html>.
- Rodriguez, Ricardo:. 2012.** MaestroHack. [Online] 2012.
http://maestrohack.ucoz.com/publ/linux/ubuntu_10_10_es_la_ultima_version_estable_de_la_distribucion/9-1-0-53.
- Sarduy, Yanetsys: Urra, Pedro. 2006.** [Online] abril 14, 2006.
http://bvs.sld.cu/revistas/aci/vol14_4_06/aci11406.htm.
- 2011.** Sistema de Gestión de Alumnado, Planificación Docente y Campus Virtual Integrado. [Online] 02 25, 2011. [Cited: 03 15, 2012.]
http://www.ual.es/Universidad/epropias/privado/planes_estudio/agora_modulos.htm.
- UAG. 2011.** [Online] 2011.
http://www.uag.mx/index.php?option=com_content&view=article&id=664&Itemid=395&lang=mx.

- Vergara, Gonzalo. 2009.** MejoraTuGestion . [Online] 2009.
<http://mejoratugestion.com/mejora-tu-gestion/que-es-un-sistema-de-gestion/>.
- www.tuinformaticafacil.com. 2010.** [Online] 2010.
<http://www.tuinformaticafacil.com/herramientas-desarrollo/sql-power-architect-herramienta-de-modelado-de-datos>.
- 2008** Apoyo a la cultura Libre"Applying UML and Patterns" Craig Larman. Ed Prentice Hall.http://migueljaque.com/index.php/tecnicas/tecnicasmodnegocio/37-modelado_negocio/46
- 2011** El Rincon del Vago<http://html.rincondelvago.com/metodos-de-prueba-caja-de-pandora.html>
- 2007** GSInnova<http://www.rational.com.ar/herramientas/rosedatamodeler.html>
- 2003** IngenieroSoftwareMicrosoft<http://www.ingenierosoftware.com/equipos/estabilizar-aplicacion.php>
- 2012** Integra-Vision<http://veloz-informacioncontable.com/blog-3/la-importancia-de-los-sistemas-de-gestion.html>
- Metodología Ágil de desarrollo de Software SXP 2011* Artemisa2011
- 2011** MSDNMicrosoft<http://msdn.microsoft.com/es-es/library/bb972240.aspx>
- 2012** Programming 4 Us<http://mscerts.programming4.us/es/762408.aspx>
- 2010** SOFTENG<http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>
- 2012** Definicion.de<http://definicion.de/planeacion-educativa/>

BIBLIOGRAFÍA CONSULTADA

2008 Apoyo a la cultura Libre "Applying UML and Patterns" Craig Larman. Ed Prentice Hall.
http://migueljaque.com/index.php/tecnicas/tecnicasmodnegocio/37-modelado_negocio/46

2011 El Rincon del Vago <http://html.rincondelvago.com/metodos-de-prueba-caja-de-pandora.html>

2003 IngenieroSoftwareMicrosoft <http://www.ingenierosoftware.com/equipos/estabilizar-aplicacion.php>

2011MSDNMicrosoft <http://msdn.microsoft.com/es-es/library/bb972240.aspx>

GLOSARIO DE TÉRMINOS

Consistencia: Propiedad que asegura que una transacción no romperá con la Integridad de una base de datos, pues respeta todas las reglas y directrices de ésta.

Sprint: Iteración

Robustez: Capacidad de los sistemas software de reaccionar adecuadamente ante condiciones excepcionales.

Usabilidad: Se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.

Flexibilidad: Relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.

SQL: Siglas de Structured Query Language (lenguaje estructurado de consulta). Es un lenguaje de alto nivel, no procedural, normalizado, que permite la consulta y actualización de los datos de bases de datos relacionales.

XML: Siglas de Extensible Markup Language (lenguaje de marcas generalizado). Se trata de un lenguaje utilizado para estructurar la información en cualquier documento que contenga texto.