



**ESPE**  
ESCUELA POLITECNICA DEL EJERCITO  
CAMINO A LA EXCELENCIA

**DPTO. DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**ANÁLISIS, DESARROLLO E IMPLANTACIÓN DE UN SISTEMA  
ORIENTADO A LA WEB PARA GESTIÓN ACADÉMICA.**

**CASO PRÁCTICO: JOHN OSTEEN CHRISTIAN ACADEMY**

**PLAN DE TESIS**

**AUTORES:**

**SR. EDISON AYALA**

**SR. GABRIEL ENRIQUEZ**

**SANGOLQUÍ, OCTUBRE DEL 2011**

## **1.- TEMA.**

Análisis, desarrollo e implantación de un sistema orientado a la web para gestión académica, como una herramienta de apoyo para cada uno de los procesos del colegio John Osteen Christian Academy.

## **2.- ANTECEDENTES**

En la actualidad el Colegio John Osteen Christian Academy, no posee un sistema que mejore la automatización de sus procesos académicos, lo que ocasiona gran pérdida de tiempo en la realización de actividades repetitivas y constantes en cada periodo escolar.

El sistema orientado a la web para el colegio John Osteen Christian Academy, permitirá a cada una de las personas relacionadas al mismo, tener acceso a información destacada de la institución, así como una interfaz amigable, la cual permita verificar cada una de las actividades académicas de cada estudiante, sean estas, registro de faltas, registro de calificaciones, inscripciones, matriculación. Con el único fin de poder mejorar, ayudar y llevar un seguimiento de cada una de las labores relacionadas a cada estudiante y personal docente.

Razón por la cual es necesario desarrollar un sistema orientado a la web para poder comunicar de manera más eficiente y eficaz el desempeño de cada uno de los estudiantes, así como el rendimiento de los mismos.

## **3.- PLANTEAMIENTO**

Se propone el desarrollo de un sistema orientado a la web para gestión estudiantil que permita realizar el ingreso de información de los alumnos nuevos al colegio, además de llevar un registro de su historial académico, mantener una base de datos con los profesores que son parte de la unidad educativa, así también permitir la admisión, matrículas, ingreso de las notas obtenidas por los alumnos en cada uno de los periodos educativos del colegio, con las cuales se generarán los reportes que el colegio requiere para un mejor desempeño, funcionamiento y servicio. Al mismo tiempo cabe recalcar que se implementará un portal web para el alojamiento de información destacada de la institución, la cual permita dar a conocer sus servicios al público en general.

Se implementará un aula virtual (Moodle), la cual proveerá un espacio educativo en el portal web, mismo que ayude a los docentes y a los alumnos a mejorar cada una de sus actividades académicas y de investigación.

Por último se implementará un correo institucional que facilite a la institución la distribución de la información a cada uno de sus funcionarios y estudiantes.

El sistema orientado a la web contará con varios perfiles, donde cada perfil tendrá los privilegios adecuados para las acciones que necesitan realizar.

El equipo de trabajo estará bajo la responsabilidad de presentar toda la documentación respectiva del desarrollo del proyecto durante el período de cimentación del mismo.

#### **4.- JUSTIFICACIÓN**

A continuación se mencionan cada una de las razones con las que se justifica la creación del sistema orientado a la web para gestión académica del colegio John Osteen Christian Academy:

- El personal administrativo, personal docente y estudiantil podrán acceder a la información correspondiente de cada uno, por internet, sin la necesidad de acercarse a la institución para la consulta o adquisición de la misma.
- Se propone un portal web para la publicación de información de cada uno de los servicios que proporciona la institución, así como la localización de la institución con el único objetivo de darse a conocer a nivel nacional e internacional.
- Se propone un sistema orientado a la web el cual automatizará la gestión académica del colegio, ya que en la actualidad cada uno de estos procesos se realizan de forma manual, provocando pérdidas de tiempo en cada uno de estos procesos repetitivos correspondientes a cada año escolar.
- Para una fácil comunicación entre institución y personal relacionada con la misma, se implementa un sistema de correo electrónico, el mismo que permitirá optimizar el tiempo en la comunicación de cada una de las actividades.
- Para contribuir con cada una de las actividades académicas, se implementa un aula virtual (Moodle), la cual ayudará a cada uno de los estudiantes y docentes, en el desarrollo de cada una de sus actividades académicas, así como un espacio de interacción para debates e inquietudes por parte de los estudiantes y docentes.
- Se proveerá al personal administrativo de la institución que realice reportes de acuerdo a las necesidades. Y no por archivo como se realiza actualmente.

Tomando en cuenta estas consideraciones, la presente solución se basa en una herramienta tecnológica de información, denominada sistema orientado a la web para gestión académica del colegio John Osteen Christian Academy, la cual funcionara mediante una plataforma de libre distribución, para obtener un producto que permita optimizar el tiempo de todos los procesos relacionados con la institución de la manera más fácil, ágil y oportuna.

## **5.- OBJETIVOS**

### **5.1.- Objetivo General**

- Analizar, diseñar e implementar un sistema orientado a la web para gestión académica, utilizando frameworks y herramientas open source, que permita a cada una de las personas relacionadas al colegio John Osteen Christian Academy, la administración del alumnado y personal docente, así como, mejorar la comunicación entre cada uno de los mismos.

### **5.2.- Objetivos Específicos**

- Analizar los requerimientos necesarios para el desarrollo del sistema, utilizando el estándar IEEE 830, de forma que cubra las diferentes consideraciones metodológicas y técnicas.
- Implementar un sistema orientado a la web, utilizando un sistema de administración de contenidos (Joomla), el cual dé a conocer información importante de la institución la cual contenga una interfaz amigable para el usuario.
- Facilitar la comunicación interna entre funcionarios y estudiantes con la implementación del correo electrónico institucional.
- Facilitar la administración de admisiones, matrículas, notas y asistencias con la ayuda de reportes académicos de cada uno de los estudiantes, así como de personal docente, utilizando la herramienta Jasper Report.

## **6.- ALCANCE**

El alcance del presente proyecto denominado análisis, desarrollo e implementación de un sistema orientado a la Web para gestión académica del

colegio John Osteen Christian Academy, abarca la utilización de las últimas tecnologías de libre distribución disponibles.

El sistema orientado a la Web de manera general contemplará los siguientes módulos:

- La gestión de admisión de nuevos alumnos.
- La gestión de matrícula de estudiantes.
- La gestión académica.
- La gestión de recurso humano (personal docente).
- Reportes académicos
- La gestión de perfiles, donde cada perfil tendrá los privilegios adecuados para las acciones que necesitan realizar.

Adicional, como una ayuda para la educación y basándose en los avances tecnológicos, se adaptará al sistema de gestión académica orientado a la web un aula virtual (Moodle) de libre distribución, la cual facilite un espacio educativo en el que tanto los docentes como los alumnos puedan desarrollar sus actividades académicas y de investigación.

Por último se implementara un correo electrónico institucional, para mejorar la comunicación entre personal administrativo, docente y alumnado.

Se manejará la metodología XP debido: a que es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo, a que el equipo de desarrollo necesario para aplicar XP es pequeño y al desarrollo modular del sistema. Teniendo en cuenta que se realizara reuniones constantes para refinar el sistema, basándose en la retroalimentación que se realiza conjuntamente con el cliente y en las pruebas unitarias, con el fin de tener una mejor implementación de la metodología.

## **7. MARCO TEORICO**

### **7.1.- Extreme Programming (XP)**

Es una de las Metodologías Agiles de desarrollo de software más exitosas en la actualidad. XP considera que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar

a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

### **7.1.1.- Características fundamentales**

Las características fundamentales del método son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera - el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Cuanto más simple es el sistema, menos tendrá que comunicar sobre éste, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

### **7.2.- IEEE 830 – Especificación de Requisitos de Software (ERS)**

La IEEE 830 es la Especificación de Requerimientos del Software (ERS), cuyo propósito principal es el proporcionar la ayuda pertinente para la elaboración del documento muy útil de Especificación de Requerimientos de Software. Esta Especificación es esencialmente una guía para la redacción del documento de especificación de requerimientos del software.

La Especificación (IEEE, 1990) dice que el ERS 4 *“Documento que define, de forma completa, precisa y verificable, los requisitos, el diseño, el comportamiento u otras características de un sistema o componente de un sistema”*.

La ERS sirve para:

- Un cliente describa claramente las necesidades que tiene, es decir lo que quiere del software.
- Un proveedor entienda claramente lo que el usuario/ cliente quieren del software.
- Se establezca la base para el contrato del desarrollo o de compra/venta.
- Se reduzca el esfuerzo de análisis, diseño y desarrollo; de manera que se evite realizar de nuevo el trabajo ya hecho.
- Se tenga una base o una referencia para realizar las pruebas del software solicitado o para realizar la validación del mismo.
- Para realizar nuevas versiones, que mejoren el sistema ya establecido.

### **7.3 Lenguaje Unificado de Modelado UML**

UML es un lenguaje estándar de modelado de sistemas, que proporciona los “planos” de la arquitectura del sistema, puede utilizarse para visualizar, especificar, construir, y documentar procesos de negocios, funciones del sistema, lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

UML es un lenguaje que ayuda en la interpretación de grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que permiten la comunicación durante el desarrollo ya que al ser estándar, los modelos pueden ser interpretados por personas que no participaron en su diseño (e incluso por herramientas) sin ninguna ambigüedad. En este contexto, UML sirve para especificar, modelos concretos, no ambiguos y completos.

#### **7.3.1.- Diagramas del UML**

Un diagrama es la representación gráfica de un conjunto de elementos. Los diagramas se dibujan para visualizar el sistema desde diferentes perspectivas, de forma que un diagrama es una proyección de un sistema, Hay varios tipos de diagramas los cuales se muestran en diferentes aspectos de las entidades representadas:

**7.3.1.1.- Diagramas de Estructura:** Son los principales elementos que deben existir en el sistema modelado.

- Diagrama de clases.
- Diagrama de componentes.
- Diagrama de objetos.
- Diagrama de estructura compuesta (UML 2.0).
- Diagrama de despliegue.
- Diagrama de paquetes.

**7.3.1.2.- Diagramas de Comportamiento:** Su principal función es observar lo que sucede en el sistema modelado.

- Diagrama de actividades.
- Diagrama de casos de uso.
- Diagrama de estados.

**7.3.1.3.- Diagramas de Interacción:** Se los llama también diagramas de comportamiento, su función es observar el flujo de control y de datos entre los elementos del sistema modelado.

- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de tiempos (UML 2.0).
- Diagrama de vista de interacción (UML 2.0)

#### **7.4.- Introducción a la Aplicación en Capas**

Esta técnica de programación permite crear aplicaciones a nivel empresarial gracias a la fácil administración que implica la separación de los componentes de la aplicación en capas, debido a que se separa la lógica de negocios de la lógica de diseño, de esta manera se separa el código fuente según su funcionalidad dentro de la aplicación.

El diseño de una aplicación en capas, permite a cada capa mostrar servicios que otras aplicaciones o capas pueden consumir.

Una aplicación se divide en varias capas, lo cual proporciona una separación adecuada de la funcionalidad, además permite tener mayor escalabilidad dando como resultado aplicaciones robustas.

Una de las mayores ventajas de realizar una aplicación en capas es el alto grado de encapsulamiento que maneja por lo tanto permite la reutilización de sus componentes para el desarrollo de nuevas aplicaciones, que brinda un soporte y mantenimiento más sencillo.

### **7.5- Aplicaciones Web**

Una aplicación Web es una interfaz o conjuntos de páginas Web que interactúan con el usuario final, de esta manera le permite el acceso a la información solicitada y se toma los datos propios del modelo de negocio, de esta manera cualquier persona puede interactuar con ella desde Internet por medio de un Navegador Web (browser).

En otras palabras aplicación web es aquella aplicación en la que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

De una manera más técnica la aplicación web es una aplicación software que se codifica en un lenguaje soportado por los navegadores web, en la que se confía la ejecución al navegador.

#### **7.5.1.- Características de Aplicaciones Web.**

- El usuario puede tener facilidad de acceso a las aplicaciones web mediante un navegador web.
- El usuario puede ingresar a la aplicación web desde cualquier parte del mundo, donde tenga un acceso a Internet.
- Existe una sola aplicación web instalada en el servidor, pero pueden existir miles de clientes que acceden al mismo tiempo a dicha aplicación; por lo tanto se puede actualizar y mantener una única aplicación y todos sus clientes verán los resultados inmediatamente.
- Utilizan tecnologías como ASP.NET, JAVA, AJAX, FLASH, JAVASCRIPT, HTML, JSP, JSF que brindan una gran interfaz de usuario a la aplicación web.

### **7.6.- MYSQL**

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo permite soportar una gran carga de forma muy eficiente. MySQL se crea por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

## **8.- METODOLOGÍA DE DESARROLLO**

### **8.1.- Extreme Programming (XP)**

Los proyectos de desarrollo de software de la actualidad se desenvuelven en un entorno muy cambiante donde se exige reducir los tiempos de desarrollo, manteniendo una alta calidad en su resultado final sin involucrar un aumento considerable en la complejidad de las fases en las que el proyecto es realizado.

La elección de una metodología adecuada que se ajuste al equipo de desarrollo permitirá una flexibilidad para la obtención de los objetivos definidos al inicio del proyecto y además proveerá satisfacción al cliente más allá de sus necesidades o expectativas.

Ya que las metodologías ágiles están orientadas para proyectos pequeños que requieran un tiempo corto para desarrollo, se las consideran como una solución a la medida que no abandona las prácticas tradicionales que aseguran la calidad del producto final que se entrega al usuario.

Debido a que el sistema del presente proyecto requiere entregas a corto plazo, cumpliendo con los requerimientos planteados, la metodología XP sugiere iteraciones que pueden aplicarse plenamente, llegando en cada una de ellas a satisfacer las necesidades existentes en la institución.

El equipo de desarrollo necesario para aplicar XP es pequeño, apostando por el desarrollo modular del sistema y a reuniones constantes para refinar el sistema, basándose en la retroalimentación que se realiza conjuntamente con el cliente y en las pruebas unitarias.

Esta metodología plantea la reducción de tiempos de desarrollo en base a simplificar el diseño, disminuir el número de procesos involucrados; todo esto para agilizar los cambios que puedan darse en el transcurso del proyecto.

A continuación se presenta una pequeña comparación entre XP (Programación Extrema), HDM (Método de Diseño Hipermedia), EORM (Metodología de Relaciones de Objetos Mejorada), OOHDM (Metodología de Diseño de Hipermedia Orientada a Objetos), SOHDM (Metodología de Diseño Hipermedia orientada a objetos y basada en escenarios), y WSDM (Método de Diseño de Sitios Web).

	Procesos	Técnica de modelado	Notación	Herramienta de soporte
<b>XP</b>	<ol style="list-style-type: none"> <li>1. Planificación</li> <li>2. Gestión</li> <li>3. Diseño</li> <li>4. Desarrollo</li> <li>5. Pruebas</li> </ol>	UML según se requiera	Flujogramas Diagramas de clase Diagramas de interacción Bocetos de IU, etc.	No especificada
<b>HDM</b>	<ol style="list-style-type: none"> <li>1. Desarrollo a largo plazo</li> <li>2. Desarrollo a corto plazo</li> </ol>	Entidad-Relación	Diagrama E-R	No especificada
<b>EORM</b>	<ol style="list-style-type: none"> <li>1. Clases del entorno de desarrollo</li> <li>2. Composición del entorno de desarrollo</li> <li>3. Entorno de desarrollo de UI</li> </ol>	Orientando a Objetos	<ol style="list-style-type: none"> <li>1. Diagrama de clases</li> <li>2. Diseño GU</li> </ol>	ONTOS Studio
<b>OOHDM</b>	<ol style="list-style-type: none"> <li>1. Diseño conceptual</li> <li>2. Diseño navegacional</li> <li>3. Diseño abstracto de la UI</li> <li>4. Implementación</li> </ol>	Orientando a Objetos	<ol style="list-style-type: none"> <li>1. Diagrama de clases</li> <li>2. Diagrama navegacional, clase + contexto</li> <li>3. Diagrama de configuración de ADV + Diagrama ADV (Abstract Data View)</li> </ol>	OOHDM-Web
<b>SOHDM</b>	<ol style="list-style-type: none"> <li>1. Análisis del dominio</li> <li>2. Modelo en OO</li> <li>3. Diseño de la vista</li> <li>4. Diseño navegacional</li> <li>5. Diseño implementación</li> <li>6. Construcción</li> </ol>	Escenarios, Vistas Orientando a Objetos	<ol style="list-style-type: none"> <li>1. Diagramas de escenarios de actividad</li> <li>2. Diagrama de estructura de clase</li> <li>3. Vista OO</li> <li>4. Esquema de enlace navegacional</li> <li>5. Esquema de páginas</li> </ol>	No especificada
<b>WSDM</b>	<ol style="list-style-type: none"> <li>1. Modelado del usuario</li> <li>2. Diseño conceptual               <ol style="list-style-type: none"> <li>2.1. Modelo objetos</li> <li>2.2. Diseño navegacional</li> </ol> </li> <li>3. Diseño implementación</li> <li>4. Implementación</li> </ol>	Entidad-Relación Orientando a Objetos	<ol style="list-style-type: none"> <li>1. Diagrama de E-R o clase</li> <li>2. Capas de navegación</li> </ol>	No especificada

## **8.2 SERVIDOR DE APLICACIONES WEB**

Como servidor de aplicaciones web se decidió trabajar con Apache, debido a las facilidades que proporciona esta herramienta, además se la ha utilizado para numerosos proyectos universitarios.

Otra de las razones por las cuales se optó por la utilización de Apache, la más importante a mencionar, es la compatibilidad que existe entre las diversas plataformas existentes, de esta manera se podrá realizar la implantación ya sea en un sistema operativo Windows o Unix.

Además al ser Apache uno de los servidores de aplicaciones más utilizados, existe gran cantidad de soporte y ayuda, que facilita la implantación y el manejo de esta herramienta.

## **8.3.- GESTOR DE BASE DE DATOS**

Como sistema gestor de base de datos (DBMS), se escogió MySQL, por la experiencia de haber trabajado en varios proyectos con esta herramienta, teniendo así un mayor dominio en su uso y mejor rendimiento de la misma.

## **8.4.- LENGUAJE DE PROGRAMACIÓN**

La gran compatibilidad e integración existente con el servidor de aplicaciones y con el gestor de base de datos es una de las causas por las cuales se ha seleccionado a JAVA como el lenguaje de programación a utilizarse.

Cabe recalcar que el lenguaje es bastante expansible y posee un gran número de librerías y módulos que mejoran su funcionalidad, permitiendo tener aplicaciones web mucho más dinámicas y permitiendo la integración con JavaScript, para mejorar considerablemente la operatividad de las aplicaciones.

## **8.5.- ENTORNO DE DESARROLLO INTEGRADO (IDE)**

Como IDE de desarrollo se decidió por utilizar Netbeans, este editor nos permitirá desarrollar la aplicación web, proporcionando varias ayudas tanto para la generación del código requerido, como para la administración en si del proyecto y la depuración del código generado.

## **9.- FACTIBILIDAD**

### **9.1.- Factibilidad Operativa**

La administradora del Colegio John Osteen, como auspiciante de éste proyecto de tesis, facilitará toda la información que se requiera, así como los equipos necesarios y colaborará con todos los aspectos necesarios para la satisfactoria culminación del mismo.

Además garantiza el compromiso de ofrecer colaboración total con los desarrolladores, y cumplimiento en fechas y tiempos estimados para las entrevistas.

### **9.2.- Factibilidad Técnica**

Los desarrolladores tienen los conocimientos necesarios y suficientes acerca de: aplicaciones Web, herramientas de desarrollo a utilizar y el motor de bases de datos, para realizar el presente proyecto de tesis.

### **9.3.- Requerimientos de hardware.**

Los requerimientos mínimos de hardware son:

- Procesador Core i5 o superior.
- Memoria Ram de 4Gb o superior.
- Disco Duro de 500 Gb o superior.
- Monitor de 15" o superior.
- Lector de CD-ROM

### **9.4.- Requerimientos de software.**

Los requerimientos mínimos de Software son:

- Sistema Operativo Windows o Linux.
- Navegador Web: Internet Explorer o Mozilla.
- Java runtime environment, Java development kit.

### **9.5.- Factibilidad Económica**

Los desarrolladores del presente proyecto tendrán sus propios equipos, el mismo que garantiza que los desarrolladores no tengan ningún tipo de inconveniente y puedan culminar satisfactoriamente el proyecto de tesis.

Para la instalación del proyecto se da a conocer el presupuesto.

#### **9.5.1.- Software libre.**

Detalle	Cantidad
NetBeans	1
MySQL	1
Joomla	1
Jasper Report	1
Moodle	1

#### 9.5.2.- Hardware.

Detalle	Cantidad	Precio unitario	Costo total
Equipo	2	800	1600,00
SUBTOTAL			\$ 1600,00

#### 9.5.3.- Otros.

Detalle	Cantidad	Precio unitario	Costo total
Internet	Anual	50	600,00
Hosting	Anual	99,00	99,00
SUBTOTAL			\$ 699,00

#### 9.5.4.- Personal.

Detalle	Cantidad	Precio unitario	Costo total
Programador	2	800	1.600,00
SUBTOTAL			\$ 1.600,00
Total general			\$ 3899,00

#### 9.5.5.- Herramientas de desarrollo.

Software	Utilización
NetBeans	Desarrollo en Java
MySQL	Base de Datos
Joomla	Gestión de Contenidos
Jasper Report	Reportes
Moodle	Ambiente Educativo Virtual

## 10.- TECNOLOGÍA

Para el desarrollo de la aplicación web se tomara muy en cuenta la utilización de tecnologías de punta, las cuales nos ayudan a optimizar recursos, desarrollar aplicaciones de calidad, las mismas que prestan servicios de una manera eficaz y eficiente, permitiendo un óptimo desempeño de la aplicación y por ende no presentar complicación a cada uno de los usuarios.

Las nuevas tecnologías permiten al estudiante una mayor integración al proceso de aprendizaje, ya que se crea un nuevo sistema de enseñanza que además de potenciar al alumnado, también le permite estar involucrado al docente de una forma más activa y rápida.

Se va a utilizar la plataforma Java EE ya que es un estándar para el desarrollo de aplicaciones portables, robustas, escalables y seguras. La programación de JAVA permite el desarrollo de programas de rendimiento seguro y alto en las plataformas múltiples.

Como sistema de gestión de contenidos utilizaremos Joomla que es una aplicación de código abierto bajo licencia GPL. Entre sus principales características están:

- Organización de un sitio Web de una manera sencilla.
- Publicar contenidos en la Web.
- Escalabilidad e implementación de nuevas funcionalidades.
- Diseño y aspecto estético del sitio.
- Administración de Usuarios.

El portal Web tendrá la arquitectura de software MVC (Modelo Vista Controlador) ya que separa los datos de una aplicación, la interfaz de usuario y la lógica del control en tres componentes distintos. En aplicaciones web es la más frecuente y la más apropiada.

## **11.- CONTENIDO**

### **CAPÍTULO 1.- INTRODUCCIÓN.**

1. Introducción.
  - 1.1. Antecedentes
  - 1.2. Objetivos
    - 1.2.1. Objetivo General
    - 1.2.2. Objetivos Específicos
  - 1.3. Justificación e Importancia
  - 1.4. Alcance
  - 1.5. Propuesta Arquitectura
  - 1.6. Gestión del Proyecto
    - 1.6.1. Fases e Iteraciones
    - 1.6.2. Calendario del Proyecto

### **CAPÍTULO 2.- MARCO TEÓRICO.**

2. Metodología.
  - 2.1. Generalidades sobre colegio John Osteen
  - 2.2. XP (eXtreme Programing)
  - 2.3. UML
  - 2.4. Herramientas
    - 2.4.1. Desarrollo
      - 2.4.1.1. Joomla
      - 2.4.1.2. NetBeans
      - 2.4.1.3. MySQL
    - 2.4.2. Herramientas de Diseño
      - 2.4.2.1. StarUML

### **CAPÍTULO 3.- ANÁLISIS Y DISEÑO.**

- 3.1 Especificación de los requerimientos del portal web.
  - 3.1.1 Propósito.
  - 3.1.2 Ámbito del portal web.
- 3.2. Documentos de requisitos del sistema
  - 3.2.1 Diseño conceptual.
    - 3.2.1.1 Diagrama de Casos de Uso.
    - 3.2.1.6 Diagrama de Clases.
- 3.3 Análisis de Navegación

- 3.3.1 Documento de análisis de navegación
- 3.4 Diseño de navegación.
  - 3.4.1 Documento de diseño de navegación
- 3.5 Diseño de interfaz.

## CAPÍTULO 4.- DESARROLLO Y PRUEBAS.

### 4.1 Capítulo : Construcción y pruebas

- 4.2 Fase Construcción
  - 4.2.1 Desarrollo del portal web
  - 4.2.2 Manual de Instalación
  - 4.2.3 Pruebas de la aplicación
  
- 4.3 Pruebas de funcionamiento.
  - 4.3.1 Pruebas Unitarias del Dominio
  - 4.3.2 Pruebas de Caja Blanca y Negra
  - 4.3.3 Pruebas de Configuración y compatibilidad
- 4.4 Manual de Instalación inicial y de usuario final
- 4.5 Documentación diseño final y manual de usuario final

## CAPÍTULO 5.- CONCLUSIONES Y RECOMENDACIONES.

- 5.1 Conclusiones
- 5.2 Recomendaciones

## BIBLIOGRAFÍA

## ANEXOS

## 12.- CRONOGRAMA

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	➔	<b>Gestion Academica John Osteen</b>	<b>113 días</b>	<b>lun 10/10/11</b>	<b>mié 14/03/12</b>	
2	➔	<b>Presentacion</b>	<b>6 días</b>	<b>lun 10/10/11</b>	<b>sáb 15/10/11</b>	
3	➔	Elaboracion Plan de Tesis	3 días	lun 10/10/11	mié 12/10/11	
4	➔	Correccion de Plan de Tesis	2 días	jue 13/10/11	vie 14/10/11	3
5	➔	Entrega de Plan de Tesis	1 día	sáb 15/10/11	sáb 15/10/11	4
6	➔	<b>Analisis</b>	<b>17 días</b>	<b>lun 17/10/11</b>	<b>vie 04/11/11</b>	
7	➔	Recoleccion de requerimientos	5 días	lun 17/10/11	vie 21/10/11	5
8	➔	Analisis e interpretacion de requerimientos	4 días	sáb 22/10/11	mié 26/10/11	7
9	➔	Desarrollo de documento de especificaciones de requerimientos	5 días	jue 27/10/11	mar 01/11/11	8
10	➔	Validacion de SRS por Cliente	2 días	mié 02/11/11	jue 03/11/11	9
11	➔	Entrega Final de SRS	1 día	vie 04/11/11	vie 04/11/11	10
12	➔	<b>Diseño</b>	<b>30 días</b>	<b>sáb 05/11/11</b>	<b>vie 09/12/11</b>	
13	➔	Elaboracion de Diagramas UML	15 días	sáb 05/11/11	mar 22/11/11	11
14	➔	Validacion de Diagramas UML	10 días	mié 23/11/11	sáb 03/12/11	13
15	➔	Correccion de Diagramas UML	5 días	lun 05/12/11	vie 09/12/11	14
16	➔	<b>Implementacion</b>	<b>57 días</b>	<b>mié 23/11/11</b>	<b>vie 27/01/12</b>	
17	➔	Desarrollo de Pagina Web	15 días	mié 23/11/11	vie 09/12/11	13
18	➔	Implementacion de WebMail	5 días	lun 05/12/11	vie 09/12/11	14
19	➔	Desarrollo Clase Persistencia	2 días	sáb 10/12/11	lun 12/12/11	18
20	➔	Desarrollo de Sistema Web	30 días	mar 13/12/11	lun 16/01/12	19
21	➔	Validacion del sistema	5 días	mar 17/01/12	sáb 21/01/12	20
22	➔	Correccion de Validaciones	5 días	lun 23/01/12	vie 27/01/12	21
23	➔	<b>Pruebas</b>	<b>28 días</b>	<b>sáb 28/01/12</b>	<b>mié 29/02/12</b>	
24	➔	Pruebas Unitarias	10 días	sáb 28/01/12	mié 08/02/12	22
25	➔	Pruebas de Aceptacion	5 días	jue 09/02/12	mar 14/02/12	24
26	➔	Analisis de resultados de las Pruebas	3 días	mié 15/02/12	vie 17/02/12	25
27	➔	Correccion de Errores	10 días	sáb 18/02/12	mié 29/02/12	26
28	➔	<b>Implantacion</b>	<b>9 días</b>	<b>jue 01/03/12</b>	<b>sáb 10/03/12</b>	
29	➔	Manual de Usuario	3 días	jue 01/03/12	sáb 03/03/12	27
30	➔	Manual Tecnico	3 días	jue 01/03/12	sáb 03/03/12	27
31	➔	Instalacion del Sistema	3 días	lun 05/03/12	mié 07/03/12	30
32	➔	Capacitacion Usuarios	3 días	jue 08/03/12	sáb 10/03/12	31

## 13.- BIBLIOGRAFÍA

### **XP (Extreme Programming)**

- [http://en.wikipedia.org/wiki/Extreme\\_Programming](http://en.wikipedia.org/wiki/Extreme_Programming)
- [http://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_distribuida](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_distribuida)

### **Aplicación Web:**

- [http://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)
- [http://es.wikipedia.org/wiki/Servicio\\_web](http://es.wikipedia.org/wiki/Servicio_web)
- IEEE Std 610.12-1990 Standard Glossary of Software Engineering Terminology
- [http://iteso.mx/~juanjo/materiales/IEEE\\_Std1233\\_1998\\_esp\\_desarrollo\\_de\\_especificacion\\_de\\_reque.pdf](http://iteso.mx/~juanjo/materiales/IEEE_Std1233_1998_esp_desarrollo_de_especificacion_de_reque.pdf)
- <http://oness.sourceforge.net/proyecto/html/ch03s02.html>
- [www.lsi.us.es/docencia/get.php?id=2085](http://www.lsi.us.es/docencia/get.php?id=2085)

### **UML:**

- <http://www.ambysoft.com/unifiedprocess/agileUP.html>
- <http://kasyles.blogspot.com/2008/10/procesos-unificados-y-aup.html>
- <http://www.ambysoft.com/unifiedprocess/agileUP.html>
- <https://forja.rediris.es/docman/view.php/227/369/Proceso%20Unificado.pdf>
- [http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado)

### **Java EE:**

- [http://es.wikipedia.org/wiki/Java\\_EE](http://es.wikipedia.org/wiki/Java_EE)
- [http://www.uhu.es/josel\\_alvarez/NvasTecnProg/recursos/tTema1.pdf](http://www.uhu.es/josel_alvarez/NvasTecnProg/recursos/tTema1.pdf)

### **MySQL:**

- <http://www.mysql.com/>
- <http://es.wikipedia.org/wiki/MySQL>

### **NetBeans:**

- <http://www.netbeans.org/>

### **Joomla:**

- <http://www.joomlaspanish.org>