



**Universidad de Granma  
Facultad de Ciencias Técnicas**

**Universidad Técnica de Cotopaxi  
Unidad Académica de Ciencias de la Ingeniería y  
Aplicadas**

**Sistema de gestión de avances académicos  
y asistencia de docentes en el colegio  
técnico nacional “Dr. Camilo Gallegos  
Domínguez”**

**Trabajo de diploma para optar por el título de  
Ingenieros en Informática y Sistemas Computacionales**

**Autores: Lourdes Catalina Paredes Gavilanez  
Marco Javier Guanotuña Toasa**

**Tutores: Ing. Eric Ismael Leonard Brizuela  
Ing. Erodís Pérez Michel**

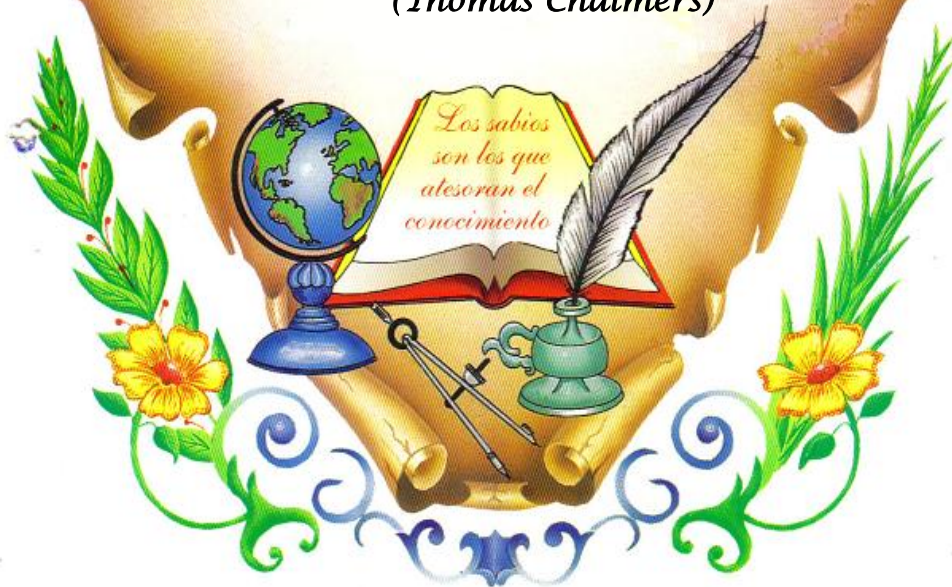
**Bayamo, Cuba, Junio 2013**



# Pensamiento

*“La dicha de la vida  
consiste en tener siempre  
algo que hacer, alguien a  
quien amar y alguna cosa  
que esperar”.*

*(Thomas Chalmers)*



## **PENSAMIENTO**

*Bienaventurado el hombre que halla la sabiduría, Y que obtiene la inteligencia; Porque su ganancia es mejor que la ganancia de la plata, Y sus frutos más que el oro fino. Más preciosa es que las piedras preciosas; Y todo lo que puedes desear, no se puede comparar a ella.*

*Prov. 3:13-15*

# Agradecimiento

*Lourdes Catalina Paredes Gavilanez*

*Primero y antes que nada, dar gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el período de estudio.*

*A mis Maestros quienes me han enseñado a ser mejor en la vida y a realizarme profesionalmente.*

*Un agradecimiento especial a mis tutores por hacer posible esta tesis.*

*A mis compañeros de clases quienes me acompañaron en esta trayectoria de aprendizaje y conocimientos.*

*En general quisiera agradecer a todas y cada una de las personas que han vivido*

*conmigo la realización de esta tesis , que no necesito nombrar porque tanto ellas como yo sabemos que desde los más profundo de mi corazón les agradezco el haberme brindado todo el apoyo, colaboración, ánimo pero sobre todo cariño, amistad y amor.*

*Marco Javier Guanotuña Toasa*

*Agradezco a Dios por darme un segundo más de vida y permitirme llegar hasta donde me encuentro a mis padres y hermanos por todo el apoyo y sus sabios consejos que me brindaron, a mis tutores Eric y Erodís por ayudarnos a desarrollar la tesis y como no a mi compañera de tesis Catalina por ser parte de la misma.*

# Dedicatoria

*Lourdes Catalina Paredes Gavilanez*

*A mi madre Lourdes Gavilanez y a mi padre Gerardo Paredes porque creyeron en mí y porque me sacaron adelante, dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mí, fue lo que me hizo ir hasta el final. Va por ustedes, por lo que valen, porque admiro su fortaleza y por lo que han hecho de mí.*

*A mis hermanos (as), sobrinos, amigos, y en especial a mi cuñado.*

*Gracias por haber fomentado en mí el deseo de superación y el anhelo de triunfo en la vida.*

*Mil palabras no bastarían para agradecerles su apoyo, su comprensión y sus consejos en los momentos difíciles.*

*A todos, espero no defraudarlos y contar siempre con su valioso apoyo, sincero e incondicional.*

*Marco Javier Guanotuña Toasa*

*Esta tesis se la dedico a Dios por darme la fuerza y el conocimiento necesario durante los años de formación profesional, a mis padres Gloria y Francisco por toda su ayuda y confianza incondicional que me han dado y por hacer realidad mis sueños, a mis hermanos Patricio, Consuelo, Luis y Cristian que sin duda alguna también me han apoyado, a mi cuñada Inés y a mis dos sobrinitos Anthony y Bryan que han despertado en mí el valor del sentimiento. En general a todos quienes confiaron y siguen confiando en mí.*

## Declaración de autoría

Declaramos que somos los únicos autores del trabajo de diploma titulado: **“Sistema de gestión de avances académicos y asistencia de docentes en el colegio técnico nacional “Dr. Camilo Gallegos Domínguez””** y autorizamos a la Universidad de Granma para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los **15** días del mes de **Junio** del **2013**.

Firma de los Autores

---

Lourdes Catalina Paredes Gavilanez

---

Marco Javier Guanotuña Toasa

## **Opinión del Diseñador**

Considero que el Trabajo de Diploma titulado: **Sistema de gestión de avances académicos y asistencia de docentes en el colegio técnico nacional “Dr. Camilo Gallegos Domínguez”** cumple con las normas de diseño establecidas por la Facultad de Ciencias Técnicas de la Universidad de Granma.

---

Ing. Camilo Trujillo Pacheco.

## Opinión del Usuario del Trabajo de Diploma

El trabajo de diploma titulado: **Sistema de gestión de avances académicos y asistencia de docentes en el colegio técnico nacional “Dr. Camilo Gallegos Domínguez”** Esta entidad considera que en correspondencia con los objetivos trazados el trabajo de diploma le satisface.

- Totalmente**
- Parcialmente en un — %**

Los resultados de este trabajo de diploma le reportan a la entidad los beneficios siguientes:

**La aplicación Web presentada en esta investigación es de vital importancia pues facilitará el registro, control y procesamiento de la información relacionada con la gestión docente y estudiantil, lo que permitirá un mejor trabajo a la hora de procesar la información referente al colegio técnico nacional “Dr. Camilo Gallegos Domínguez”, favoreciendo la obtención de registros válidos, confiables y así emitir la información correcta en el momento requerido.**

Y para que así conste se firma el presente a los **15** días del mes de **Junio** del año **2013**.

\_\_\_\_\_  
Nombre del representante de la Entidad

\_\_\_\_\_  
Cargo

\_\_\_\_\_  
Firma y Cuño

## **RESUMEN**

El Centro Educativo Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” ubicado en la ciudad de Latacunga, Provincia de Cotopaxi, país Ecuador, se lleva a cabo el control y procesamiento de la información relacionada con los avances académicos y asistencia de los docentes para el desempeño de su trabajo, con el fin de tener constancia del cumplimiento laboral de los docentes, y el estado académico de los estudiantes. Este proceso se hace complejo, debido a que debe registrar dicha información de un modo constante para así tener el control preciso de los datos obtenidos diariamente.

Atendiendo a estas necesidades, se desarrolla una aplicación para facilitar la gestión de la información y procesamiento de los datos en la secretaria del Colegio Técnico “Dr. Camilo Gallegos Domínguez” de forma rápida, confiable y segura. El proceso de desarrollo de la aplicación se realiza apoyándose en tecnologías libres, multiplataforma, utilizando PHP 5.3.3 como el lenguaje de programación, se implementa el patrón de arquitectura Modelo Vista Controlador, utilizando el framework CodeIgniter, MySQL 5.5.20 como Sistema Gestor de Bases de Datos y Netbeans7.3 como Entorno de Desarrollo Integrado.

## **Summary**

The Educational Center Technical National High School “Dr. Camilo Gallegos Domínguez” located in Cotopaxi city, Latacunga province, country Ecuador, gets to stub the control and information processing related with the academic advances and help of the teachers for the performance of its work, with the aim of having enough evidence about the labor fulfillment of the teachers, and the academic state of the students. This process becomes complex, because he should register said information of a constant mode in order to have the precise control of the data obtained daily.

Attending to these needs, develops an application to facilitate the information science and processing of the data in the secretary of the Technical High School “Dr. Camilo Gallegos Domínguez” of fast, reliable and safe way. The process of development of the application comes true basing itself on free technologies, multi-platform, using PHP 5,3,3 like the programming language, the Controlling pattern of Model Obvious architecture takes effect, using the framework CodeIgniter, MySQL 5,5,20 like Negotiating System of Data Bases and Netbeans7.3 like Entorno of Integrated Desarrollo.

## ÍNDICE GENERAL

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA</b> .....	5
1.1 Introducción .....	5
1.2 Gestión de la información en la secretaría del Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” .....	5
1.3 Sistemas informáticos para el control de avances académicos y asistencia de docentes .....	6
1.4 Herramientas para el desarrollo del producto informático .....	8
1.4.1 Aplicaciones Web .....	8
1.4.2 Lenguajes de Programación .....	10
1.4.2.1 Lenguajes del lado del Cliente .....	11
1.4.2.2 Lenguajes del lado del Servidor .....	13
1.4.3 Servidor Web .....	15
1.4.3.1 El servidor Web Apache .....	16
1.4.4 Sistemas Gestores de Bases de datos (SGDB) .....	16
Figura 1.1 Funcionamiento de los SGBD. ....	17
1.4.4.1 PostgreSQL .....	18
1.4.4.2 MySQL .....	19
1.4.5 Frameworks .....	19
1.4.5.1 Zend Framework .....	20
1.4.5.2 CodeIgniter .....	21
1.4.5.3 Symfony .....	22
1.4.6 Herramienta de desarrollo para PHP. NetBeans .....	22
1.5 Metodología de desarrollo de software .....	23
1.5.1 Rational Unified Process (RUP) .....	23
1.5.2 Programación Extrema (XP) .....	24
1.6 Fundamentación de las tecnologías a usar .....	25
1.7 Conclusiones del Capítulo .....	26
<b>Capítulo II. Descripción y Construcción de la Solución Propuesta</b> .....	27
2.1 Introducción .....	27
2.2 Personas relacionadas con el sistema .....	27
2.3 Fase de Exploración .....	28
2.3.1 Historias de Usuario .....	28

2.4 Fase de Planificación .....	29
2.4.1 Estimación de esfuerzo por Historias de Usuario .....	30
2.4.2 Plan de Duración de las Iteraciones .....	31
2.4.3 Plan de Entrega .....	32
2.5 Iteraciones .....	32
2.5.1 Tareas .....	33
2.6 Producción .....	37
2.6.1 Diseño .....	37
2.6.2 Codificación .....	38
2.6.3 Pruebas de Software .....	38
2.7 Mantenimiento .....	41
2.8 Muerte del Proyecto .....	41
2.9 Conclusiones del Capítulo .....	41
<b>CONCLUSIONES GENERALES .....</b>	<b>42</b>
<b>RECOMENDACIONES .....</b>	<b>43</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>44</b>
<b>BIBLIOGRAFÍA .....</b>	<b>48</b>
<b>ANEXOS .....</b>	<b>I</b>

## ÍNDICE DE FIGURAS

Figura 1.1 Funcionamiento de los SGBD. ....	17
---	----

## ÍNDICE DE TABLAS

Tabla 2.1 Personas Relacionadas con el Sistema. ....	27
Tabla 2.2 Historia de Usuario Autenticar Usuario. ....	28
Tabla 2.3 Historia de Usuario Gestionar Cuenta de Usuario.....	28
Tabla 2.4 Historia de Usuario Gestionar Estudiante. ....	29
Tabla 2.5 Estimación de esfuerzos por Historia de Usuario.....	30
Tabla 2.6 Plan de Duración de las Iteraciones.....	31
Tabla 2.7 Plan de Duración de la Entrega. ....	32
Tabla 2.8 Tareas establecidas en cada Iteración.....	33
Tabla 2.9 Tarea # 1 de la Historia Usuario Autenticar Usuario.....	36
Tabla 2.10 Tarea # 1 de la Historia de Usuario Gestionar Cuenta de Usuario. .....	36
Tabla 2.11 Tarea # 2 de la Historia de Usuario Gestionar Cuenta de Usuario. .....	36
Tabla 2.12 Prueba # 1 de la Historia de Usuario Autenticar Usuario. ....	39
Tabla 2.13 Prueba # 1 de la Historia de Usuario Gestionar Cuenta de Usuario. .....	39
Tabla 2.14 Prueba # 1 de la Historia de Usuario Gestionar Estudiante.....	40

## ÍNDICE DE ANEXOS

<b>Anexo # 1 Historias de Usuario</b> .....	I
Tabla 1.1 Historia de Usuario Gestionar Docente. ....	I
Tabla 1.2 Historia de Usuario Gestionar Departamento. ....	I
Tabla 1.3 Historia de Usuario Gestionar Asignatura. ....	I
<b>Anexo # 2 Tareas para realizar las Historias de Usuario</b> ¡Error! Marcador no definido.	
Tabla 2.1 Tarea # 1 de la Historia de Usuario Gestionar Estudiante.....	II
Tabla 2.2 Tarea # 2 de la Historia de Usuario Gestionar Estudiante.....	II
Tabla 2.3 Tarea # 3 de la Historia de Usuario Gestionar Estudiante.....	II
<b>Anexo # 3 Imágenes del sistema</b> .....	III
Figura 3.1 Pantalla de inicio al sistema.....	III
Figura 3.2 Pantalla de para la gestión de usuarios del sistema.....	III
<b>Anexo # 4 Diseño de la Bases de Datos</b> .....	V
<b>Anexo # 5 Pruebas de Aceptación del sistema</b> .....	VI
Tabla 5.1 Prueba # 1 de la Historia de Usuario Gestionar Docente. ....	VI

## INTRODUCCIÓN

Hoy en día la tecnología es indispensable para facilitar la realización de las tareas del hombre. A partir del surgimiento de los ordenadores estos se han mostrado como un dispositivo útil y necesario en la vida cotidiana.

Las tecnologías han evolucionado exponencialmente en los tiempos modernos siendo de vital importancia para el procesamiento de la información que utiliza el hombre. Poco a poco la humanidad ha ido encaminándose hacia la informática la cual ha obtenido una aceptación apreciable en todos sus niveles y con ello han surgido nuevas necesidades e inconvenientes en la sociedad.

Cada vez se desarrollan aplicaciones más completas, ligeras y estables, que permitan ser usadas sin ninguna traba a cualquier hora y en cualquier lugar; por las ventajas que ofrecen las redes informáticas se inició el desarrollo de aplicaciones de asistencia y solución de problemas dentro de las que se hallan las aplicaciones Web (Ignacio, 2008).

Las aplicaciones Web son utilizadas por varios organismos o empresas, por las facilidades que estas brindan como que no necesitan ser instaladas para su uso, ya que se usan tecnologías Web y sobre todo no son difíciles de utilizar. Estas aplicaciones brindan una mayor y más fácil propagación de la información en el proceso de gestión de información (Masadelante.com, 2009).

Aprovechando los adelantos tecnológicos, el uso de las redes informáticas y las mejorías de las aplicaciones Web, Ecuador está llevando a cabo la informatización a todos los niveles de la sociedad para su perfeccionamiento.

Teniendo en cuenta que el Centro Educativo Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” ubicado en Latacunga desarrolla una serie de operaciones técnicas para brindar un mejor servicio, entre las que se demanda llevar el control de los avances académicos y asistencia de los docentes para garantizar un control interno, continuo y fiable. Debido al constante crecimiento de su población estudiantil y por ende del personal administrativo y docente, se dificulta un rápido procesamiento de dicho control de manera manual. Además debe estar acorde al avance tecnológico que exige ser una institución de excelencia.

El crecimiento antes mencionado sumado al control manual resulta una dificultad para manejar información clasificada, actualizada y ordenada de las labores que realizan los profesores, en el área académica. Es por esto que se necesita informatizar el proceso de Control de Avance Académico que actualmente se lo realiza manualmente en los leccionarios.

Luego de analizar la situación existente en el Centro Educativo Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”, se define el siguiente **problema científico**: ¿Cómo favorecer el procesamiento de la información referente a los avances académicos generados por la secretaría del Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”?

Este problema se evidencia en el proceso de control de la información generada por la secretaría del Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” que constituye al **objeto de la investigación**.

Se delimita el **campo de acción**, sistema de gestión de avances académicos y asistencia de los docentes en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” Latacunga.

**Objetivo general**: Desarrollar un sistema informático de gestión de avances académicos y asistencia de los docentes en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” Latacunga.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

1. Analizar la bibliografía científica para el desarrollo del estado del arte, usando las herramientas generales y especializadas de Internet.
2. Implementar un sistema informático para la gestión de avances académicos y asistencia de los docentes en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.

**Idea a defender**: El despliegue de un sistema informático de gestión permitirá llevar un control centralizado de los avances académicos y asistencia de los docentes, con el cumplimiento de la jornada laboral.

Para el desarrollo de la aplicación se deben tener en cuenta las siguientes **tareas de investigación**:

1. Analizar la situación existente del objeto de estudio para identificar mejoras en el proceso de gestión de los avances académicos y asistencia de los docentes en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.
2. Análisis de las tecnologías, metodologías y herramientas a utilizar en el desarrollo del sistema deseado.
3. Diseño e implementación de los artefactos de la metodología a usar.
4. Implementación del sistema informático como solución al problema planteado según el diseño realizado.

Para dar cumplimiento a las tareas anteriores se utilizaron los siguientes métodos y técnicas:

#### **Métodos teóricos**

**Análisis histórico-lógico:** A través de este método se estudia la labor realizada por la secretaria de la institución y las aplicaciones anteriores que se utilizaron para en la realización de aplicaciones Web similares a la de la presente investigación, tanto el origen como evolución de los diferentes lenguajes de programación, framework y sistemas gestores de bases de datos.

**Análisis y Síntesis:** Para la recopilación y el procesamiento de la información y arribar a las conclusiones de la investigación, la obtención de conocimiento y resumir la información a procesar.

**La modelación sistémica:** permitió realizar un estudio del trabajo del especialista jurídico que desarrolla esta tarea en la entidad y su incidencia directa en la solución propuesta.

#### **Métodos Empíricos**

**Observación:** Se utilizó para obtener un resultado visual haciendo exámenes reales a la aplicación para determinar que cumpla con los objetivos, detectar y corregir errores.

## **Técnica**

**Entrevista:** se realizan entrevistas a directivos, con el objetivo de conocer los aspectos fundamentales que deben tomarse para la gestión de la información relacionada con el control interno y obtener información sobre cómo es el flujo de información, cómo la procesan y lo que necesitan.

El documento estará compuesto por dos capítulos que hacen referencia a:

**Capítulo 1:** Consta de la fundamentación teórica que sustenta la investigación, analizándose las herramientas y tecnologías Web más utilizadas actualmente en Cuba y el mundo, con sus características fundamentales.

**Capítulo 2:** Se enfoca en el entorno de la aplicación propuesta, utilizando como base la ingeniería de software para hacer una descripción del diseño e implementación de la aplicación.

## **CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

En el presente capítulo se hace un análisis del objeto de estudio y el campo de acción de la investigación enmarcado en el proceso a informatizar; así como un estudio bibliográfico actualizado sobre las principales tendencias y tecnologías actuales utilizadas como base para el desarrollo de la investigación.

### **1.2 Gestión de la información en la secretaría del Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”**

El objetivo principal de la gestión de la información es organizar y poner en uso los recursos de información de la organización (tanto de origen interno como externo) para permitirle operar, aprender a adaptarse a los cambios ambiente. Los actores principales en la gestión de la información son los mismos profesionales de la información, en unión estrecha con sus usuarios. Los procesos principales de la gestión de información son: la identificación, la adquisición de las fuentes informativas, su organización y almacenamiento, el desarrollo de productos y servicios, su distribución y uso (Mulet, 2002).

De forma general, la gestión de la información es la adquisición de la información necesaria, al costo adecuado para ser distribuida oportunamente y a través del medio apropiado, a la persona que la utilizará en el desarrollo de sus funciones, de manera que contribuya al logro de los objetivos y metas de la organización y sus individuos.

El proceso de control de avances académicos y asistencia de docentes no es más que el conjunto de procesos mediante los cuales se puede llevar un registro bien organizado diariamente, de los que apoyan la práctica pedagógica y permiten construir y modelar el perfil deseable del estudiante. Desde la gestión se crean condiciones para prever, organizar, coordinar, controlar, planificar y monitorear los procesos docentes generados durante la ejecución de proyectos y acciones (Inciarte, Marcano y Reyes, 2006).

Los constantes cambios en los que está sumergido el mundo actual y como se conoce hoy en día, imponen asegurar la calidad y la eficiencia de la docencia secundaria, resultando el control de avances académicos y asistencia de docentes, de vital importancia para lograr esto.

## **Control de asistencia y evaluación**

El control de la asistencia y la evaluación es una de las actividades más importantes del proceso docente, porque de ahí depende la evaluación final de cada uno de los estudiantes. Esta forma parte de las responsabilidades de los profesores de cada asignatura y el control se realiza a través de un registro impreso entregado a cada profesor por parte de la secretaria docente de la facultad al comenzar un nuevo año.

El registro es un documento auditable y debe especificar: curso académico, año, especialidad, tipo de curso (matutino, diurno o nocturno), año escolar, grupo, matrícula, el número de lista de cada estudiante así como sus nombres y apellidos, día y tipo de clase. Al impartirse un turno de clase, se marca la asistencia del estudiante y en caso de realizarse alguna evaluación, se plasma también la nota obtenida por el mismo. Al final del año el profesor realiza un resumen evaluativo y entrega el registro a la secretaria docente para ser archivado.

## **Evaluación del Docente**

La evaluación al docente es una actividad importante del proceso profesional, porque de ahí depende la formación de los estudiantes. Esta forma parte de las responsabilidades de los profesores de cada asignatura y el control se realiza a través de un registro impreso entregado a cada decano por parte de la secretaria docente de la facultad al comenzar un nuevo año.

El registro es un documento específico: facultad, año, cargo, experiencia, el número de lista de cada docente así como sus nombres y apellidos, día y tipo de clase. Al impartirse un turno de clase, se marca la asistencia del docente, se plasma también la nota obtenida por el mismo. Al final del año el rector realiza un resumen evaluativo y entrega el registro a la secretaria docente para ser archivado.

### **1.3 Sistemas informáticos para el control de avances académicos y asistencia de docentes**

En los Colegios de la nueva generación, las aplicaciones encargadas de divulgar y gestionar las actividades del proceso docente y estudiantil han alcanzado un gran desarrollo, por su vital importancia para lograr una educación secundaria con

mayor calidad. Existen diversos sistemas informáticos para suplir las necesidades de tales instituciones y esta gran variedad se debe a las diferentes políticas adoptadas por cada una de ellas, aspecto a tener en cuenta para diseñar e implementar un sistema informático con este objetivo. Algunos de estos sistemas, consolidados en el ámbito internacional son Agora y SIU-Guaraní.

**Agora:** La institución académica mexicana ha implementado un producto de software estándar de gran calidad para la gestión de Centros Docentes y Academias de todo tipo. Fue desarrollado para cualquier tipo de centro docente: academias, colegios, escuelas de negocios, centros de impartición de masters, formación ocupacional y continua, centros universitarios, centros de idiomas, oposiciones, informática, música, capacitación, etc. Es una plataforma que cumple las siguientes funciones: control de alumnos actuales, históricos y potenciales, control independiente de alumnos (quien recibe docencia) y clientes (quien la abona), introducción de notas y gestión de actas y control automatizado de la asistencia (Posted, 2008).

**SIU-guaraní:** La Facultad de Bellas Artes actualmente se encuentra implementando este sistema que permitirá agilizar el tratamiento de la información y la gestión académica. Se trata de un sistema informático que administra la gestión académica de los alumnos, desde que ingresan hasta que obtienen el diploma, de manera segura, íntegra y consistente. El sistema provee servicios para alumnos, docentes, personal administrativo y autoridades del Sistema Universitario Nacional. Permite mejorar el tratamiento de la información y agilizar los mecanismos de gestión académica. El sistema también asegura la privacidad de los datos, al tiempo que brinda una gran flexibilidad en el manejo de la información.

Entre las principales prestaciones de la plataforma se destacan: La gestión de matrícula, inscripción por Web a cursos y exámenes, gestión de exámenes finales, gestión de equivalencias, rapidez en confección de actas oficiales, carga de notas en actas de exámenes y cursadas, consulta de inscripciones, consulta de plan de estudios y datos académicos, consulta de materias cursadas y aprobadas, solicitud de certificados y actualización de datos censales (Eduardo, 2009).

En Ecuador, también se ha desarrollado un sistema informático con el mismo objetivo que los dos anteriores, se trata de un sitio Web para la supervisión de avances académicos y asistencia de los docentes de la UTEQ.

Con este sitio Web se puede obtener un control de cada uno de los docentes en el cumplimiento académico de una manera fácil y rápida. Ha sido desarrollado usando Microsoft Visual Studio.Net 2005 y Bases de datos SQL Server 2005. Entre sus principales funcionalidades se pueden citar: registrar los avances académicos, observar los temas ya registrados en clases anteriores, supervisar el cumplimiento de los programas académicos establecidos para cada una de las unidades de aprendizaje por parte del docente, obtener reportes y gráficos estadísticos de cada uno de los avances académicos de los periodos actuales y anteriores, asentar faltas a docentes y subir un programa académico (Tubay, 2010).

#### **1.4 Herramientas para el desarrollo del producto informático**

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción, teniéndose en cuenta la tendencia actual y las novedades de cada una de ellas.

##### **1.4.1 Aplicaciones Web**

Una Aplicación Web es un conjunto de páginas Web enlazadas que visualizan diferentes partes de la información que se quiere mostrar a través de ella. Constituye una de las mejores herramientas para divulgar, gestionar y compartir la información por lo que trae consigo un aumento de la eficiencia en cuanto a la manipulación de gran cantidad de información. Permiten modificar la lógica del negocio a través de un navegador (Hernán Ruiz, 2006; Cinca, 2008).

La *World Wide Web* (WWW) permite visualizar en la pantalla del usuario páginas con información alojadas en computadoras remotas (llamadas genéricamente sitios). En la actualidad, existen dos tipos de sitios Web: los que se comportan como magazines, donde solo se puede leer la información que en él esté escrita; y los que se comportan a manera de software, donde se ejecutan una series de tareas específicas. Estos últimos son las llamadas aplicaciones Web (Sanz, 2002; Pressman, 2008).

La arquitectura de un sitio Web es simple. Contiene como componentes principales: el Servidor Web, una Red y un Navegador o cliente. La arquitectura de una aplicación Web además incluye la aplicación en el Servidor, que es la que permite al sistema manejar lógica de negocio y tener un estado (Bellman, 2007).

Con el uso de las aplicaciones Web en cualquier entidad se logra, entre otros beneficios, una alta eficiencia que se alcanza en el área que haga uso de los mismos. Debido a una serie de características que ellos poseen, se mejoran los mecanismos de intercambio de información en la entidad, disminuyendo en un alto grado cualquier dificultad que exista para difundir la misma (Villa, 2006).

Entre las principales comodidades que una aplicación Web pueden traer se encuentran las siguientes:

- **Compatibilidad multiplataforma.** Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales (Colombo, Demetrescu, I. y Laura, 2003).
- **Actualización.** Las aplicaciones basadas en Web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones) (Oliver, 2000).
- **Inmediatez de acceso.** Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y listos para trabajar sin importar cuál es su configuración o su hardware.
- **Facilidad de prueba.** Finalmente no habrá más obstáculos para permitir pruebas sencillas y efectivas de herramientas y aplicaciones antes de cargar su tarjeta de crédito. Actualmente, especialmente cuando se habla de software costoso, hay todavía una gran cantidad de funcionalidades y pequeños detalles que no pueden ser totalmente probados descubiertos antes de comprometer dinero en alguna compra total.
- **Menos requerimientos de memoria.** Las aplicaciones basadas en Web tienen muchas más razonables demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del

proveedor, a esas aplicaciones basadas en Web usa en muchos casos la memoria de las computadoras que ellos corren, dejando más espacio para correr múltiples aplicaciones del mismo tiempo sin incurrir en frustrantes deterioros en el rendimiento.

- **Menos Bugs.** Las aplicaciones basadas en Web, deberían ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos. Esta es la razón por la cual las aplicaciones basadas en Web deberían tener mucho menos bugs que el software de escritorio descargable tradicional (Zelkovitz, Shaw y Gannon, 2009).
- **Precio.** Las aplicaciones basadas en Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables si no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.
- **Múltiples usuarios concurrentes.** Las aplicaciones basadas en Web puedan realmente ser utilizada por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. Las compañías de conferencia Web y colaboración online están involucradas algunas transformaciones claves y los usuarios necesitan explorar que significa realmente trabajar efectivamente y co-editar documentos juntos (Brusilovsky, Gilles, Frasson y VanLehn, 2007).

Usando la Web, se tiene acceso a millones de páginas de información, la exploración se realiza por medio de un software especial denominado browser o explorador. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero (Prentzas, Hatzilygeroudis y Garofalakis, 2002).

#### 1.4.2 Lenguajes de Programación

En la programación de la Web se habla de dos tipos de lenguajes: los lenguajes del *lado servidor* que son aquellos lenguajes que se caracterizan por desarrollar la

lógica de negocio dentro del servidor. Además de ser los encargados del acceso a la base de datos, tratamiento de la información, etc. Por otro lado, los lenguajes de *lado cliente*, que son los encargados de aportar dinamismo a la aplicación en los navegadores (Furgeta, 2002).

Entre los lenguajes del lado del servidor se pueden encontrar entre los más sobresalientes al Active Server Pages (ASP), Personal Home Pages (PHP), Java Server Pages (JSP). Del lado del cliente se encuentra el HTML, Java y JavaScript los cuales son simplemente incluidos en el código HTML (Aldana, 2009).

#### **1.4.2.1 Lenguajes del lado del Cliente**

##### **Hypertext Markup Language (HTML)**

HTML es el lenguaje de la Web, estas siglas significan Hiper Text Markup Language (Lenguaje de Marcas de Hipertextos), en donde hipertexto hace referencia a la capacidad del lenguaje para la utilización de los conocidos hipervínculos. Se basa en el uso de etiquetas o marcas, gracias a ellas es posible definir el formato del texto, los distintos elementos que conforman la página, sus propiedades y disposición. Este lenguaje es interpretado por los navegadores, procesado pasados y convertidos en una página Web (Málaga, 2008).

Una página Web o documento HTML no es otra cosa que un archivo de texto que se puede crear con cualquier editor; desde el bloc de notas Windows hasta cualquiera de DOS o LINUX.

Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomó dos herramientas preexistentes: El concepto de Hipertexto (conocido también como link o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda (Hernán Ruiz, 2006).

##### **Java Script**

JavaScript es un lenguaje de scripts desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. Se utiliza embebido en el código HTML,

entre las tags <script> y </script>. Sus características más importantes son: [JS-a] [JS-b].

Es un lenguaje interpretado basado en guiones, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias Java Script contenidas en una página HTML y ejecutarlas adecuadamente (Colombo, et al., 2003).

Java Script es un lenguaje orientado a objetos. El modelo de objetos de Java Script está reducido y simplificado, pero incluye los elementos necesarios para que los Scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

Cuando un usuario selecciona un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante JavaScript se pueden desarrollar Scripts que ejecuten acciones en respuesta a estos eventos. En este caso actúa como un lenguaje orientado a eventos. Es además un lenguaje de alto nivel multiplataforma que admite programación estructurada y no necesita ningún kit o entorno de desarrollo.

## **Java**

Java es un lenguaje de programación clásico en cuanto a que requiere un proceso de compilación. El código compilado puede ser integrado en la página Web para ser ejecutado por el cliente.

Su nacimiento data de enero de 1996, tras el lanzamiento por la empresa creadora Sun Microsystem del JDK 1.0 (Java Development Kit). Es un entorno que puede obtenerse de forma gratuita e incorpora las herramientas básicas para la creación de una aplicación en Java (JDK, 2009).

Con Java se pueden crear dos tipos de programas:

**Applets:** programas que se integran en las páginas Web y que, residiendo en el servidor, son ejecutados por el cliente. La ejecución necesita de la interpretación del código compilado por el software cliente (Askallorg, 2007).

**Aplicaciones:** Programas autónomos que se pueden ejecutar en cualquier equipo. En este último caso puede optarse por generar código compilado similar al de los applets y que para su ejecución necesita de un intérprete o código

compilado ejecutable directamente como en cualquier otro lenguaje de programación.

En el caso de applets, el código fuente no se aloja directamente en el documento HTML, si no al que se lo añade es un código binario resultado de la compilación, el denominado JBC (Java Byte Code). Esto permite proteger el código fuente, aunque hasta cierto punto, ya que las particularidades de este código compilado hacen que sea factible el proceso inverso, es decir, la descompilación, recuperar el código fuente a partir del compilado. En la propia Internet puede encontrarse programas capaces de hacerlo. La razón de todo esto está en el hecho de que para conseguir la portabilidad de los programas el código compilado es un código que se encuentra entre el código fuente y un código objeto fuertemente dependiente de una plataforma. Es por ello que se suele decir de Java que es un lenguaje que combina la flexibilidad de los lenguajes interpretados y el poder de los compilados (JBC, 2009).

#### **1.4.2.2 Lenguajes del lado del Servidor**

##### **PHP**

PHP es el acrónimo de *Hipertext Preprocesor*. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores (Gracia, 2006).

Una vez que ya se conoce el concepto de lenguaje de programación de scripts del lado del servidor se puede hablar de PHP. PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el popular ASP de Microsoft, pero con algunas ventajas como su gratuidad, independencia de plataforma, rapidez y seguridad. Cualquiera puede descargar a través de la página principal de PHP, [www.php.net](http://www.php.net) y de manera gratuita, un módulo que hace

que nuestro servidor Web comprenda los scripts realizados en este lenguaje. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor Web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

PHP, en el caso de estar montado sobre un servidor Linux u Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP (Eloi, 2009).

Respecto a la seguridad, es importante el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son de sobra conocidos como más veloces y seguros que el sistema operativo donde se ejecuta las ASP, Windows NT o 2000. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores (Cobo, Gómez, Pérez y Rocha, 2010).

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes como: MySQL, mSQL, Oracle, Informix, y ODB. Incluye funciones para el envío de correo electrónico, subida de archivos, crear dinámicamente en el servidor, imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

### **Características de PHP**

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones Web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

- Capacidad de conexión con la mayoría de los motores de bases de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador.
- Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

### **1.4.3 Servidor Web**

Un servidor Web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor Web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página Web o información de todo tipo de acuerdo a los comandos solicitados (Vázquez, 2008).

### **1.4.3.1 El servidor Web Apache**

Apache es el servidor Web por excelencia, su configurabilidad, robustez y estabilidad hacen que millones de servidores reiteren la confianza en sus servicios. Fueron Brian Behlendorf y Cliff Skolnick quienes a través de una simple lista de correo coordinaron el trabajo y lograron establecer un espacio compartido de libre acceso para los desarrolladores.

El servidor Web Apache es un software libre de código abierto para plataformas Unix (BSD, GNU/Linux), Windows, Macintosh y otras. Tiene la opción de emitir mensajes de error altamente configurables y permite el trabajo con bases de datos de autenticación y negociado de contenido, entre otras características. Aun así, ha sido bastante criticado por la falta de una interfaz gráfica que le permita a sus usuarios una cómoda configuración.

A pesar de dicha desobediencia, este servidor cuenta con una amplia aceptación en la red desde 1996 y es el más utilizado de su tipo. Se considera flexible, rápido y eficiente, goza de una actualización continua y ha sido adaptado a los nuevos protocolos (HTTP-1.1). Puede adaptarse además a diferentes entornos y necesidades a través de sus diferentes módulos de apoyo y con la API de programación de módulos. Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y los correspondientes parches para la solución de los mismos.

### **1.4.4 Sistemas Gestores de Bases de datos (SGDB)**

Un Sistema de Gestión de Bases de Datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información de una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.

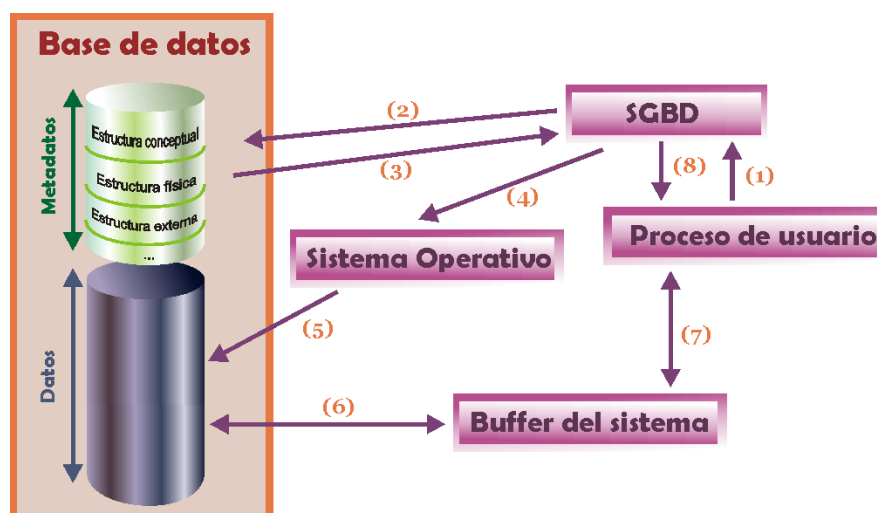


Figura 1.1 Funcionamiento de los SGBD.

Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.

Los SGBD también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y recuperar la información si el sistema se corrompe (Stonebraker, 2007).

El SGBD se encarga de mantener la validez y la consistencia de los datos almacenados mediante restricciones o reglas que no pueden violarse, lo que garantiza una mejor integridad de los mismos. Estos aportan una mayor seguridad, ya que si no se cuenta con las medidas de seguridad pertinentes, la integridad de los datos puede ser vulnerable.

Sin embargo, los SGBD permiten mantener la seguridad mediante el establecimiento de claves, para identificar al personal autorizado a utilizar la base de datos. Mejoran la accesibilidad a los datos proporcionando lenguajes de consultas o generadores de informes, que permiten al usuario establecer cualquier tipo de consulta sobre los datos, sin necesidad de un programador que escriba una nueva aplicación para realizar tal tarea.

Muchos SGBD proporcionan un entorno de cuarta generación, consistente en un conjunto de herramientas que simplifican en gran medida el desarrollo de las aplicaciones que acceden a las bases de datos. Gracias a estas herramientas, el programador puede ofrecer una mayor productividad en un menor tiempo. La mayoría de los SGBD gestionan el acceso concurrente a las bases de datos y garantizan que no ocurran problemas, mejoran el mantenimiento gracias a la

independencia de los datos y facilitan los servicios de copias de seguridad y de recuperación ante fallos.

A pesar de las muchas ventajas que ofrecen los SGBD, también tienen muchas desventajas, por ejemplo, son un conjunto de programas muy complejos con una gran funcionalidad. Son muy extensos y requieren gran cantidad de espacio en disco y en memoria para trabajar de forma eficiente. Los SGBD están escritos para ser generales y útiles en muchas aplicaciones, lo que generalmente conlleva a una pérdida de rapidez en el funcionamiento de estas últimas (Angel, 2010).

Entre los SGBD más usados se pueden mencionar el Oracle, MySQL, Microsoft SQL Server, PostgreSQL, entre otros.

#### **1.4.4.1 PostgreSQL**

PostgreSQL es un Sistema Gestor de Bases de Datos, (ORDBMS) objeto-relacionales libre, desarrollado originariamente en el Departamento de Ciencias de Computación de la Universidad de California en Berkeley.

PostgreSQL es un servidor de bases de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD. Como muchos otros proyectos de código abierto (*open source*), el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo.

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. Cuenta con herramientas gráficas como PgAdmin, phpPgAdmin, las cuales hacen que la administración de la bases de datos sea sencilla. Debido a su licencia libre PostgreSQL puede ser utilizado, modificado y distribuido por todo el mundo de forma gratuita para cualquier fin, ya sea privado, comercial o académico (Gamboa R. W., 2009).

Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir de forma gratuita para cualquier fin, ya sea privado, comercial o académico (Armando, 2008).

#### **1.4.4.2 MySQL**

La empresa que desarrolla MySQL es MySQL AB, de origen sueco. Según definición de sus autores: es un servidor de bases de datos muy rápido, robusto, fácil de usar, multitarea y multiusuario. Tiene enfoque relacional y soporta AnsiSQL.

Soporta clientes en C, C++, Eiffel, Java, Perl, PHP, Python y Tcl. Trabaja en diferentes plataformas además; de soportar múltiples idiomas, completo y optimizado uso del SQL y contiene un MyODBC. Está disponible en diferentes plataformas, es gratis, se puede disponer del código fuente.

Es muy utilizado en aplicaciones Web porque existe baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que lo hace ideal para este tipo de software. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. A través de constante desarrollo, MySQL ha logrado ofrecer un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hace de este un gestor altamente conveniente para acceder a bases de datos en Internet (Cabrera, 2008).

#### **1.4.5 Frameworks**

Un framework es un diseño reutilizable del todo o la parte de un sistema/software descrito por varias jerarquías de herencia de clases (generalmente algunas abstractas) y las distintas colaboraciones que se establecen entre las instancias de estas.

El uso de un framework minimiza el tiempo de desarrollo de un sistema, ya que proporciona sub-sistemas que funcionan, lo que reduce su codificación y acelera su puesta en marcha. El framework reduce los riesgos de desarrollo del software, proporcionando una base fiable suficientemente probada y una arquitectura consistente entre aplicaciones.

Al usar frameworks, todas las aplicaciones generadas comparten una arquitectura común, facilitándose su comprensión, mantenimiento y soporte. Cualquier programador al trabajar con un framework, no deberá invertir gran parte de su tiempo en buscar las clases necesarias, interconectarlas, ni descubrir los métodos que estas contienen. Los frameworks ocultan toda esa complejidad, dando un alto

nivel de abstracción y facilitando el desarrollo del software, permitiendo pasar más tiempo identificando requerimientos para el mismo que tratando con los detalles de bajo nivel (Crespo y Pablo, 2000).

#### 1.4.5.1 Zend Framework

Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones Web y servicios Web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible framework de aplicaciones Web al combinarse. ZF ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de bases de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes, como Zend\_Auth y Zend\_Acl, proveen autenticación de usuarios y autorización diferentes a las tiendas de certificados comunes. También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los **Web services** más populares. Cualesquiera que sean las necesidades de su solicitud, usted tiene todas las posibilidades de encontrar un componente de Zend Framework que se pueda utilizar para reducir drásticamente el tiempo de desarrollo, con una base completamente sólida.

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y Strikelron se han asociado con Zend para proporcionar interfaces de servicios Web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework (Zuares, 2009).

#### 1.4.5.2 CodeIgniter

CodeIgniter es un conjunto de herramientas para construir aplicaciones Web usando PHP. Su objetivo es permitirle al programador el desarrollo de proyectos de forma rápida, mediante un conjunto completo de librerías para tareas comúnmente necesarias, así como una interfaz simple y una estructura lógica para acceder a estas bibliotecas. Utiliza el acercamiento al MVC permitiendo una buena separación entre lógica y presentación, lo que ayuda a programar de manera más ordenada. Contiene también diversas herramientas que le proporcionan a las aplicaciones un mayor grado de versatilidad y seguridad (Pedro, 2009).

Es un framework muy fácil de instalar. No se necesita más que una cuenta de FTP para subirlo al servidor y su configuración se realiza con apenas la edición de un archivo, donde se escribe el acceso a la bases de datos. Durante su configuración no se necesita el acceso a herramientas como la línea de comandos, que no suele estar disponible en todos los alojamientos. También se debe destacar su versatilidad, pues es capaz de trabajar en la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo se tiene un acceso por FTP para enviar los archivos al servidor y donde no se tiene acceso a su configuración (Cambiaso, 2009).

CodeIgniter es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. Es bastante menos rígido que otros frameworks y aun cuando define una manera específica de trabajo, en muchos de los casos se pueden seguir o no sus reglas de codificación, lo que demuestra su flexibilidad.

El núcleo de CodeIgniter es bastante ligero, permitiendo que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando vayan a utilizarse realmente. Como se puede apreciar, este framework tiene muchas ventajas, sobre todo en comparación con otros productos similares (Arias, 2012).

### 1.4.5.3 Symfony

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones Web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web (Potencier, 2008).

Symfony está desarrollado completamente en PHP 5.3. Ha sido probado en numerosos proyectos reales y se utiliza en sitios Web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows (Labs, 2012).

### 1.4.6 Herramienta de desarrollo para PHP. NetBeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE1 es un producto libre y gratuito sin restricciones de uso (Leonard, 2011).

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. *Sun Microsystems* fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (*manifest file*) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados

independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software (Arteaga, 2011).

## 1.5 Metodología de desarrollo de software

Durante el ciclo de vida del software se deben completar una serie de tareas para obtener un producto. A menudo, se dice que los distintos componentes de software deben pasar por distintas fases o etapas durante el ciclo de vida.

Pues bien, cada una de esas tareas puede ser abordadas y resueltas de múltiples maneras, con distintas herramientas y técnicas. Es necesario saber cuándo se puede dar por concluida una tarea, quién debe realizarla, qué tareas preceden o anteceden a una dada, y qué documentación se utiliza para llevarla a cabo (Pressman, 2007).

Se habla de detalles organizativos, de un "estilo" de hacer las cosas. Pero al ir un poco más allá de un simple estilo, al formalizar ese "estilo" y añadir algo de rigurosidad y normas, se obtiene una metodología.

Las Metodologías de Desarrollo de Software se pueden dividir en dos grupos: metodologías ágiles y las metodologías tradicionales (no ágiles) (Jacobson, Boock y Rumbaugh, 2005).

### 1.5.1 Rational Unified Process (RUP)

Actualmente no existe una metodología de desarrollo de software que sea global, es decir que encierre características que puedan aplicarse a cualquier tipo de proyecto. Las características de cada proyecto conjuntamente con su equipo de desarrollo, recursos, y requisitos exigen que se escoja una que se adapte en la mayor medida posible a estas características. Su ciclo de vida se caracteriza por:

- **Está dirigido por Casos de Usos:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más

importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

- **Iterativo e Incremental:** Es el modelo utilizado por RUP para el desarrollo de un proyecto de software. Este modelo plantea la implementación del proyecto a realizar en iteraciones, con lo cual se pueden definir objetivos por cumplir en cada iteración y así poder ir completando todo el proyecto iteración por iteración, con lo cual se tienen varias ventajas, entre ellas se puede mencionar la de tener pequeños avances del proyectos que son entregables al cliente el cual puede probar mientras se está desarrollando otra iteración del proyecto, con lo cual el proyecto va creciendo hasta completarlo en su totalidad (Pressman, 2008b).

### 1.5.2 Programación Extrema (XP)

Es una metodología ágil para el desarrollo de software y consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo. La Programación Extrema es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo, preocupándose en todo momento del aprendizaje de los desarrolladores y estableciendo un buen clima de trabajo. Este tipo de método se basa en una realimentación continuada entre el cliente y el equipo de desarrollo con una comunicación fluida entre todos los participantes, también busca simplificar las soluciones implementadas y coraje para los múltiples cambios. Este tipo de programación es la adecuada para los proyectos con requisitos imprecisos, muy cambiantes y con un riesgo técnico excesivo (Solís, 2003).

La metodología de XP se basa en:

- **Pruebas Unitarias:** Se basa en las pruebas realizadas a los principales procesos, de tal manera que se adelanta en algo hacia el futuro, se puede hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.
- **Re fabricación:** Se desarrolla en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento (Calero, 2010).

## 1.6 Fundamentación de las tecnologías a usar

La selección de las herramientas y tecnologías a utilizar para el desarrollo de esta aplicación se basó fundamentalmente en la utilidad de las mismas, en la plataforma disponible y en las exigencias del cliente en cuestión. A continuación se describe más detalladamente la colección utilizada y los criterios que le dieron lugar.

- El framework CodeIgniter, ya que implementa buenas técnicas y patrones de diseño para la creación de aplicaciones Web.
- El lenguaje HTML, por ser de su tipo el que predomina para la creación y publicación de páginas Web y al que se le pueden añadir numerosas funcionalidades mediante las CCS y el código JavaScript, obteniéndose páginas Web rápidas y sencillas.
- PHP 5.3.3 como lenguaje de lado servidor, ya que fue diseñado específicamente para el desarrollo de páginas dinámicas, permite incluir el código HTML con facilidad e incluye un sinnúmero de bibliotecas para el soporte de aplicaciones Web sin costo alguno (e-mail, generación de ficheros PDF, etc.).
- MySQL 5.5.20 como gestor de bases de datos, por ser multiplataforma, libre, robusto y rápido, entre otras funcionalidades.
- Apache 2.2.21 como servidor Web, pues ofrece un amplio soporte para PHP, es una tecnología de código abierto, multiplataforma, extensible y se considera uno de los más utilizados actualmente.
- 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, y se define especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes.

### **1.7 Conclusiones del Capítulo**

Con el desarrollo de la presente investigación se han identificado fallas en el proceso de control de avances académicos y asistencia de docentes en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”, las que pueden solucionarse con la creación y utilización del sistema informático propuesto.

Al concluir este capítulo se ha podido apreciar todo lo referente a la base teórica que fundamenta la presente investigación. Se tiene una valoración sobre el porqué de las elecciones referentes al lenguaje de programación a usar, el sistema gestor de bases de datos, la herramienta de desarrollo, el servidor web y la metodología de ingeniería del software.

## **Capítulo II. Descripción y Construcción de la Solución Propuesta**

### **2.1 Introducción**

Tomándose como base la metodología seleccionada en el capítulo anterior, se estructuran los temas de este capítulo por las disciplinas que la metodología XP define.

Se realiza una descripción de las características principales del sistema a desarrollar, de acuerdo al problema por el cual fue concebido. Se especifican las funcionalidades que se desean informatizar en la búsqueda de satisfacer las necesidades de los clientes. Además, se describe la construcción de la propuesta con el desarrollo de las historias de usuario a través de iteraciones, se definen tareas y principios de diseño para su implementación. Se realizan pruebas de aceptación para garantizar que los requerimientos se cumplan y que el sistema sea aceptable.

### **2.2 Personas relacionadas con el sistema**

Como se puede apreciar en la tabla 2.1 se define como persona relacionada con el sistema toda aquella que de una manera u otra interactúa con este, y obtiene un resultado de uno o varios procesos que se ejecutan en el mismo.

Tabla 2.1 Personas relacionadas con el sistema.

<b>Personas</b>	<b>Justificación</b>
Director	Representa al Director del Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”, es la persona encargada de registrar la evaluación de los docentes.
Secretaria	Representa la o el secretaria(o) de la secretaría docente del Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”, es la persona encargada de registrar los docentes y su asistencia, además los estudiantes y el horario de clase.
Docente	Representa a los profesores que imparten clases en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”, son las personas encargadas de registrar toda la información referente los avances académicos y asistencia de los estudiantes.
Usuario	Representa a todas las personas que tienen privilegio para acceder al sistema, pudiendo visualizar la información que se maneja en el este.

## 2.3 Fase de Exploración

En esta fase se define el alcance del proyecto, además el equipo de trabajo se familiariza con las herramientas y tecnologías que se utilizaran, realizándose las historias de usuarios. La duración de esta etapa puede variar en dependencia de la familiarización que tengan los desarrolladores con las herramientas.

### 2.3.1 Historias de Usuario

Las historias de usuario (HU) son el medio mediante el cual se logra una especificación de los requisitos que conformarán el sistema. Éstas son generadas por el cliente contando con alguna ayuda del desarrollador, en caso de ser necesario. El nivel de detalle de las HU debe de ser el mínimo posible que permita hacerse una idea de cuánto costará realizar la implementación del sistema.

En la tabla 2.2, 2.3 y 2.4 se describen las historias de usuario autenticar usuario, gestionar cuenta de usuario y gestionar estudiante respectivamente. Las restantes historias de usuario pueden ser consultadas en el Anexo 1.

Tabla 2.2 Historia de Usuario Autenticar Usuario.

<b>Historia de Usuario</b>	
<b>Número:</b> 1	<b>Nombre:</b> Autenticar Usuario
<b>Usuario:</b> Usuario	<b>Iteración asignada:</b> 1
<b>Prioridad del negocio:</b> Alta	<b>Puntos estimados:</b> 0.1
<b>Riesgo de desarrollo:</b> Media	
<b>Programador Responsable:</b> Marco Javier Guanotuña Toasa y Lourdes Catalina Paredes Gavilanez.	
<b>Descripción:</b> Permitir registrarse en el sistema y así acceder a las opciones que este brinda según su rol mientras que el registro sea correcto.	
<b>Observaciones:</b> Se definirán cuatro roles: Director, Secretaria(o), Docente y usuario.	

Tabla 2.3 Historia de Usuario Gestionar Cuenta de Usuario.

<b>Historia de Usuario</b>	
<b>Número:</b> 2	<b>Nombre:</b> Gestionar Cuenta de Usuario
<b>Usuario:</b> Secretaria(o)	<b>Iteración asignada:</b> 1
<b>Prioridad del negocio:</b> Alta	<b>Puntos estimados:</b> 0.2
<b>Riesgo de desarrollo:</b> Media	

<b>Programador Responsable:</b> Marco Javier Guanotuña Toasa y Lourdes Catalina Paredes Gavilanez.
<b>Descripción:</b> Se podrá insertar, modificar y eliminar las cuentas de usuario en el sistema.
<b>Observaciones:</b> Los usuarios del sistema solo podrán ser personas que trabajan en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.

Tabla 2.4 Historia de Usuario Gestionar Estudiante.

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre:</b> Gestionar Estudiante
<b>Usuario:</b> Secretaria(o)	<b>Iteración asignada:</b> 1
<b>Prioridad del negocio:</b> Alta	<b>Puntos estimados:</b> 0.3
<b>Riesgo de desarrollo:</b> Alto	
<b>Programador Responsable:</b> Marco Javier Guanotuña Toasa y Lourdes Catalina Paredes Gavilanez.	
<b>Descripción:</b> Se podrá insertar, modificar y eliminar los estudiantes del Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.	
<b>Observaciones:</b> Solo se permite la inserción de estudiantes que cursen estudios en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” y la eliminación de aquellos que se les ha dado baja previamente.	

## 2.4 Fase de Planificación

XP plantea la planificación como un permanente diálogo entre la parte empresarial y técnica del proyecto, en la que los primeros decidirán el alcance, ¿qué es lo realmente necesario del proyecto?, la prioridad qué debe ser hecho en primer lugar la composición de las versiones qué debería incluir cada una de ellas y la fecha de las mismas. En cuanto a los técnicos, son los responsables de estimar la duración requerida para implementar las funcionalidades deseadas por el cliente, de informar sobre las consecuencias de determinadas decisiones, de organizar la cultura de trabajo y, finalmente, de realizar la planificación detallada dentro de cada versión. XP no es sólo un método centrado en el código que lo es sino que sobre todo es un método de gestión de proyectos software (Escribano, 2007).

### 2.4.1 Estimación de esfuerzo por Historias de Usuario

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. A continuación se muestra en la tabla 2.5 un resumen de las Historias de Usuario donde se exponen los principales resultados según estos aspectos.

Tabla 2.5 Estimación de esfuerzos por Historia de Usuario.

No	Historia de Usuario	Prioridad	Riesgo	Esfuerzo	Iteración
1	Autenticar Usuario	Alta	Medio	0.1	1
2	Gestionar Cuenta de Usuario	Alta	Medio	0.2	1
3	Gestionar Estudiante	Alta	Alto	0.3	1
4	Gestionar Docente	Alta	Alto	0.3	1
5	Gestionar Departamento	Media	Medio	0.2	1
6	Gestionar Asignatura	Alta	Medio	0.2	1
7	Gestionar Temas por Asignatura	Medio	Medio	0.2	1
8	Gestionar Horario	Alta	Alto	0.2	1
9	Gestionar Evaluación de Estudiante	Alta	Alto	0.2	1
10	Gestionar Asistencia de Estudiante	Alta	Alto	0.2	1
11	Gestionar Asistencia de Docente	Alta	Alto	0.2	2
12	Gestionar Evaluación de Docente	Alta	Alto	0.2	2
13	Gestionar Curso	Media	Bajo	0.2	2
14	Gestionar Grupo	Bajo	Bajo	0.1	2
15	Visualizar Usuarios	Media	Bajo	0.2	2
16	Visualizar Estudiantes	Alta	Alta	0.3	2
17	Visualizar Docentes	Alta	Alta	0.3	2
18	Visualizar Departamento	Media	Medio	0.2	2
19	Visualizar Asignatura	Alta	Medio	0.2	2
20	Visualizar Temas por Asignatura	Media	Medio	0.2	2

21	Visualizar Horario	Alta	Alto	0.3	3
22	Visualizar Evaluación de Estudiantes	Alta	Alto	0.3	3
23	Visualizar Asistencia de Estudiantes	Alta	Alto	0.3	3
24	Visualizar Asistencia de Docente	Alta	Alto	0.3	3
25	Visualizar Evaluación de Docente	Alta	Alto	0.3	3
26	Visualizar Curso	Media	Bajo	0.3	3
27	Visualizar Grupo	Bajo	Bajo	0.2	3

#### 2.4.2 Plan de Duración de las Iteraciones

Una vez identificadas las historias de usuario del sistema y estimado el esfuerzo dedicado a la realización de cada una de estas se procede a la planificación de la etapa de implementación del proyecto.

Para ello XP define la elaboración del Plan de Entrega, compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio).

De acuerdo a lo mencionado anteriormente se decidió realizar dicha planificación en tres iteraciones, las cuales se detallan en la tabla 2.6.

Tabla 2.6 Plan de Duración de las Iteraciones.

Iteración	Orden de las historias de usuario	Duración de la iteración
<b>Primera Iteración</b>	Autenticar Usuario Gestionar Cuenta de Usuario Gestionar Estudiante Gestionar Docente Gestionar Departamento Gestionar Asignatura Gestionar Temas por Asignatura Gestionar Horario Gestionar Evaluación de Estudiante Gestionar Asistencia de Estudiante	3 Semanas
<b>Segunda</b>	Gestionar Asistencia de Docente	3 Semanas

<b>Iteración</b>	Gestionar Evaluación de Docente Gestionar Curso Gestionar Grupo Visualizar Usuarios Visualizar Estudiantes Visualizar Docentes Visualizar Departamento Visualizar Asignatura Visualizar Temas por Asignatura	
<b>Tercera Iteración</b>	Visualizar Horario Visualizar Evaluación de Estudiantes Visualizar Asistencia de Estudiantes Visualizar Asistencia de Docente Visualizar Evaluación de Docente Visualizar Curso Visualizar Grupo	3 Semanas

### 2.4.3 Plan de Entrega

Después de determinar qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas se realiza el cronograma de entregas que establece la fecha acordada con el cliente para la liberación de las diferentes versiones. En la siguiente tabla se muestra el Plan de duración de entregas en el cual se especifican un aproximado de las fechas para cada iteración.

Tabla 2.7 Plan de Duración de la Entrega.

<b>Iteración</b>	<b>Iteración 1</b>	<b>Iteración 2</b>	<b>Iteración 3</b>
Entrega	Final 1ra Iteración 3ra semana de Abril del 2013	Final 2da Iteración 2da semana de Mayo del 2013	Final 3ra Iteración 5ta semana de Mayo del 2013

### 2.5 Iteraciones

Según la metodología XP, la implementación debe realizarse de forma iterativa, permitiendo así que al final de cada iteración surja un producto funcional que debe ser probado y mostrado al cliente, permitiendo de esta forma lograr una constante retroalimentación desarrolladores-cliente, este producto será previamente

probado para incrementar la visión de los desarrolladores y el cliente de posibles cambios y soluciones.

### 2.5.1 Tareas

En esta fase XP plantea la implementación de cada una de las historias de usuarios. Es por esto que al principio de éstas, se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan se crean tareas para ayudar a organizar la implementación exitosa de las HU. En cada iteración XP propone la realización de varias tareas asignadas cada una a un programador específico. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración. En la tabla 2.8 se puede apreciar las tareas generadas por cada historia de usuario.

Tabla 2.8 Tareas establecidas en cada Iteración.

Iteraciones	Historias de Usuario	Tarea	Puntos Estimados
1ra Iteración	Autenticar Usuario	Diseño y programación de la interfaz de autenticación.	0.1
	Gestionar Cuenta de Usuario	Diseño de interfaz de HU y programación de insertar cuenta de usuario.	0.1
		Programación de eliminar y modificar cuenta de usuario.	0.1
	Gestionar Estudiante	Diseño de interfaz de HU y programación de insertar estudiante.	0.1
		Programación de eliminar estudiante.	0.1
		Programación de modificar estudiante.	0.1
	Gestionar Docente	Diseño de interfaz de HU y programación de insertar trabajador.	0.1
		Programación de eliminar docente.	0.1
		Programación de modificar docente.	0.1
	Gestionar Departamento	Diseño de interfaz de HU y programación de insertar departamento.	0.1
		Programación de eliminar y modificar departamento.	0.1
	Gestionar Asignatura	Diseño de interfaz de HU y programación de insertar asignatura.	0.1

		Programación de eliminar y modificar asignatura.	0.1
	Gestionar Temas por Asignatura	Diseño de interfaz de HU y programación de Insertar Temas por Asignatura.	0.1
		Programación de eliminar y modificar Temas por Asignatura.	0.1
	Gestionar Horario	Diseño de interfaz de HU y programación de Insertar Horario.	0.1
		Programación de eliminar y modificar Horario.	0.1
	Gestionar Evaluación de Estudiante	Diseño de interfaz de HU y programación de Insertar Evaluación de Estudiante	0.1
		Programación de eliminar y modificar Evaluación de Estudiante.	0.1
	Gestionar Asistencia de Estudiante	Diseño de interfaz de HU y programación de Insertar Asistencia de Estudiante.	0.1
		Programación de eliminar y modificar Asistencia de Estudiante.	0.1
<b>2da iteración</b>	Gestionar Asistencia de Docente	Diseño de interfaz de HU y programación de Insertar Asistencia de Docente.	0.1
		Programación de eliminar y modificar Asistencia de Docente.	0.1
	Gestionar Evaluación de Docente	Diseño de interfaz de HU y programación de Insertar Evaluación de Docente.	0.1
		Programación de eliminar y modificar Evaluación de Docente.	0.1
	Gestionar Curso	Diseño de interfaz de HU y programación de Insertar Curso.	0.1
		Programación de eliminar y modificar Curso.	0.1
	Gestionar Grupo	Diseño de interfaz de HU y programación de Insertar, Modificar y Eliminar Grupo.	0.1
	Visualizar Usuarios	Diseño de interfaz de HU, consulta a la Bases de datos y programación de Visualizar Usuario.	0.2
	Visualizar Estudiantes	Diseño de interfaz de HU y consulta a la Bases de datos.	0.2

		Programación de Visualizar Estudiantes.	0.1
	Visualizar Docentes	Diseño de interfaz de HU y consulta a la Bases de datos.	0.2
		Programación de Visualizar Docente.	0.1
	Visualizar Departamento	Diseño de interfaz de HU, consulta a la Bases de datos y programación de Visualizar Departamento.	0.2
	Visualizar Asignatura	Diseño de interfaz de HU, consulta a la Bases de datos y programación de Visualizar Asignatura.	0.2
	Visualizar Temas por Asignatura	Diseño de interfaz de HU, consulta a la Bases de datos y programación de Visualizar Temas por Asignatura	0.2
<b>3ra iteración</b>	Visualizar Horario	Diseño de interfaz de HU y consulta a la Bases de datos.	0.2
		Programación de Visualizar Horario.	0.1
	Visualizar Evaluación de Estudiantes	Diseño de interfaz de HU y consulta a la Bases de datos.	0.2
		Programación de Visualizar Evaluación de Estudiantes.	0.1
	Visualizar Asistencia de Estudiantes	Diseño de interfaz de HU y consulta a la Bases de datos.	0.2
		Programación de Visualizar Asistencia de Estudiantes.	0.1
	Visualizar Evaluación de Docente	Diseño de interfaz de HU y consulta a la Bases de datos.	0.2
		Programación de Visualizar Evaluación de Docente.	0.1
	Visualizar Curso	Diseño de interfaz de HU y consulta a la Bases de datos.	0.2
		Programación de Visualizar Curso.	0.1
	Visualizar Grupo	Diseño de interfaz de HU, consulta a la Bases de datos y programación de Visualizar Grupo.	0.2

En las tablas 2.9, 2.10 y 2.11 se pueden apreciar las tareas correspondientes a las HU Autenticar Usuario y Gestionar Cuenta de Usuario correspondientes a la

primera iteración, el resto de las tareas a desarrollar pueden ser consultadas en el Anexo 2.

Tabla 2.9. Tarea # 1 de la Historia Usuario Autenticar Usuario.

Tarea	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Diseño y programación de la interfaz de autenticación.	
<b>Tipo de Tarea:</b> Diseño - Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 01 de Abril del 2013	<b>Fecha Fin:</b> 01 de Abril del 2013
<b>Programador Responsable:</b> Marco Javier Guanotuña Toasa y Lourdes Catalina Paredes Gavilanez	
<b>Descripción:</b> Se diseñará una página para que el usuario se pueda autenticar en el sistema, se programará un método para autenticar usuarios.	

Tabla 2.10. Tarea # 1 de la Historia de Usuario Gestionar Cuenta de Usuario.

Tarea	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Diseño de interfaz de HU y programación de insertar cuenta de usuario.	
<b>Tipo de Tarea:</b> Diseño-Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 02 de Abril del 2013	<b>Fecha Fin:</b> 02 de Abril del 2013
<b>Programador Responsable:</b> Marco Javier Guanotuña Toasa y Lourdes Catalina Paredes Gavilanez	
<b>Descripción:</b> Se diseñará una página donde el secretario pueda gestionar cuentas de usuario, se programará un método que permita insertar cuentas de usuario.	

Tabla 2.11. Tarea # 2 de la Historia de Usuario Gestionar Cuenta de Usuario.

Tarea	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Programación de eliminar y modificar cuenta de usuario.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 03 de Abril del 2013	<b>Fecha Fin:</b> 03 de Abril del 2013
<b>Programador Responsable:</b> Marco Javier Guanotuña Toasa y Lourdes Catalina Paredes Gavilanez.	
<b>Descripción:</b> Se programarán métodos que permitan modificar y eliminar cuentas de usuario.	

## **2.6 Producción**

En esta fase se realizan las tareas planificadas por iteración. Para ello se diseña, se codifica y se prueba. Se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

### **2.6.1 Diseño**

Este sistema fue diseñado para facilitar la gestión de la información generada por el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” de forma dinámica y agradable al usuario. Para lograrlo se emplearon algunos principios de diseño visual en las páginas Web que la conforman.

El diseño visual define la apariencia del sistema y es de gran importancia para lograr que el usuario se sienta satisfecho con la información que obtiene y con la forma en que lo hace por eso la aplicación presenta un diseño simple y sencillo, sin muchas complicaciones, orientado al entorno de trabajo del cliente para que se sienta identificado con la aplicación.

Se eligieron los colores marrón claro y oscuro, negro, azul y blanco pues se considera que estos colores contribuyen a construir una interfaz agradable a la vista del usuario. Se utilizó el marrón claro para el fondo de las páginas, algunas letras, y para el fondo de las tablas, el negro y azul para las letras garantizando una lectura favorable de los textos. Se usa la letra Verdana para los textos de las páginas. Este tipo de letra permite una lectura rápida y cómoda. Es mínimo el uso de imágenes y animaciones para evitar largos tiempos de espera a la hora de cargar la página y visualizarlas.

Para la construcción del sistema se tomaron en cuenta algunos de los estándares de implementación propuestos: un header o banner, donde se muestra la información general de sistema como logo de la institución, el nombre del sistema e imágenes y textos que muestren de manera general el contenido de la aplicación, un menú superior donde se encuentran los diferentes vínculos de acceso a las secciones del sistema, la sección del contenido donde se muestra la información que se desea buscar y finalmente un footer o pie de página donde se muestra el derecho de autor. Una muestra de las pantallas es el listado de asignaturas, que puede ser consultada en el Anexo 3.

El diseño de la bases de datos fue realizado con la herramienta Embarcadero ErEstudio 8.0, el mismo está compuesto por 23 tablas, la cuales están normalizadas, cumpliendo con las normas establecidas para el diseño de bases de datos. El modelo de datos de la aplicación puede ser visto en el Anexo 4.

### **2.6.2 Codificación**

En la implementación del sistema se utiliza el lenguaje de programación Web PHP (Personal Home Page), el cual es un lenguaje del lado del servidor y es diseñado originalmente para la creación de aplicaciones Web dinámicas. Se emplean clases en el código fuente porque según las características del sistema se considera que es necesario utilizar la programación orientada a objetos (POO). Además se hace evidente el uso de la reutilización de código pues las funcionalidades del sistema presentan características en común.

### **2.6.3 Pruebas de Software**

Uno de los pilares fundamentales de XP es el proceso de prueba, la cual constituyen el último bastión desde el que se puede evaluar la calidad de forma pragmática y descubrir los errores. Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente.

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

#### **Pruebas de Aceptación**

Las pruebas de aceptación son pruebas de caja negra que se realizan a partir de las historias de usuarios. Durante las iteraciones las historias de usuarios escogidas serán traducidas a prueba de aceptación. En ella se especifican, la perspectiva del cliente, y los escenarios para probar que la historia de usuario ha sido implementada correctamente. Una historia de usuario puede tener todas las pruebas de aceptación que desee para asegurar su funcionamiento. El objetivo específico de esta prueba es garantizar que los requerimientos han sido cumplidos y que el sistema ha sido aceptable.

En la tabla 2.12, 2.13 y 2.14 se muestran algunas de las pruebas de aceptación realizadas al sistema correspondiente a la primera iteración, el resto de las pruebas de aceptación pueden ser consultadas en el Anexo 5.

Tabla 2.12 Prueba # 1 de la Historia de Usuario Autenticar Usuario.

<b>Prueba de Aceptación</b>	
<b>Número Prueba:</b> 1	<b>Número Historia de Usuario:</b> 1
<b>Nombre de la Prueba:</b> Introducción correcta del usuario y la contraseña.	
<b>Descripción:</b> El usuario, al iniciar la aplicación verá una ventana de acceso a la aplicación, en la que se le solicitará el nombre de usuario y la contraseña. El usuario debe introducir estos campos, cuando no existe el usuario este no tendrá acceso a la aplicación.	
<b>Condiciones de ejecución:</b> Ninguna.	
<b>Entrada:</b> <ul style="list-style-type: none"> <li>▪ El usuario ejecuta la aplicación.</li> <li>▪ Aparece un cuadro de texto en el que se solicita el usuario y contraseña.</li> <li>▪ El usuario introduce los datos y presiona el botón “aceptar”.</li> <li>▪ El sistema verifica ambos campos en la bases de datos y comprueba si existe tal usuario.</li> <li>▪ El sistema muestra un mensaje de error en caso que no exista un usuario con esos datos y aparece el mismo formulario para que inserte los datos correctamente, si los datos son correctos el usuario queda autenticado pudiendo acceder a las opciones que el sistema brinda según su nivel de acceso.</li> </ul>	
<b>Resultado esperado:</b> Sólo los usuarios dados de alta en el sistema tienen permiso de acceso a la aplicación.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 2.13 Prueba # 1 de la Historia de Usuario Gestionar Cuenta de Usuario.

<b>Prueba de Aceptación</b>	
<b>Número Prueba:</b> 1	<b>Número Historia de Usuario:</b> 2
<b>Nombre de la Prueba:</b> Introducción de una nueva cuenta de usuario.	
<b>Descripción:</b> La (el) secretaria(o), al seleccionar la opción del menú “Gestionar Cuenta de Usuario” •/. “Insertar Cuenta de Usuario” se le brinda la opción de insertar una nueva cuenta de usuario. La (el) secretaria(o), debe introducir los campos obligatorios y cuando los mismos sean llenados insatisfactoriamente, se mostrará un mensaje indicándose el error.	
<b>Condiciones de ejecución:</b>	

El usuario a insertarse no debe estar actualizado en la BD.
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>▪ La (el) secretaria(o) selecciona la opción del menú “Gestionar Cuenta de Usuario”. “Insertar Cuenta de Usuario”</li> <li>▪ Aparece un formulario en el que se solicita nombre, usuario, contraseña, verificar contraseña y privilegio.</li> <li>▪ La (el) secretaria(o) introduce los campos y presiona el botón “guardar”.</li> <li>▪ El sistema verifica que los campos cumplan con las validaciones de cada uno de ellos y comprueba que la cuenta está correctamente escrita.</li> <li>▪ Si los datos del nuevo usuario son correctos el sistema muestra un mensaje indicando que los datos se insertaron correctamente, de lo contrario el sistema muestra un mensaje de error y regresa al mismo formulario.</li> </ul>
<p><b>Resultado esperado:</b> Sólo puede ser insertada una nueva cuenta de usuario cuando todos los campos requeridos sean llenados correctamente.</p>
<p><b>Evaluación de la prueba:</b> Prueba satisfactoria.</p>

Tabla 2.14 Prueba # 1 de la Historia de Usuario Gestionar Estudiante.

<b>Prueba de Aceptación</b>	
<b>Número Prueba:</b> 1	<b>Número Historia de Usuario:</b> 3
<b>Nombre de la Prueba:</b> Introducción de un nuevo estudiante incorrectamente.	
<b>Descripción:</b> La (el) secretaria(o), al seleccionar la opción del menú “Gestionar Estudiante”. “Insertar Estudiante” se le brinda la opción de insertar un nuevo estudiante. El secretario debe introducir los campos obligatorios y cuando se cumple que los mismos han sido llenados insatisfactoriamente, se mostrará un mensaje indicándose el error.	
<b>Condiciones de ejecución:</b> Los datos deben ser llenados incorrectamente.	
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>▪ La (el) secretaria(o) selecciona la opción del menú “Gestionar Estudiante” •/.</li> <li>▪ Aparece un formulario en el que se solicita nombres, apellidos, carnet de identidad, edad, dirección particular, carrera, año escolar...</li> <li>▪ El secretario introduce los campos y presiona el botón “guardar”.</li> <li>▪ El sistema verifica que los campos cumplan con las validaciones de cada uno de ellos y comprueba que estén incorrectamente escritos.</li> <li>▪ El sistema muestra un mensaje de error y regresa al mismo formulario.</li> </ul>	
<b>Resultado esperado:</b> Sólo puede ser insertado un nuevo estudiante cuando todos los campos requeridos sean llenados correctamente.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

## **2.7 Mantenimiento**

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

Como parte del proceso de perfeccionamiento de las funcionalidades del sistema y con el objetivo de satisfacer las solicitudes e inconformidades del cliente una vez que fueron entregadas cada iteración se realizaron mantenimientos al software, en algunas ocasiones fueron mantenimientos correctivos y en otras adaptativos.

## **2.8 Muerte del Proyecto**

Se produjo cuando el cliente no tuvo más historias de usuario para ser incluidas en el sistema pues quedó satisfecho en todos los aspectos incluyendo rendimiento y confiabilidad del sistema. A partir de aquí se generó la documentación final del sistema y no se realizaron más cambios en la arquitectura.

## **2.9 Conclusiones del Capítulo**

El análisis del funcionamiento del sistema de gestión de información docente para el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” permitió definir las características necesarias para la creación de una aplicación Web acorde a las peticiones del cliente, todo esto a través de la metodología de ingeniería del software XP, detallándose los resultados de cada una de sus fases.

Con la construcción de la aplicación web se introduce una nueva vía para gestionar la información relacionada con los avances académicos y asistencia docente en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” de Cotopaxi, Ecuador que brindará al cliente conformidad y seguridad ante las funcionalidades del sistema.

## **CONCLUSIONES GENERALES**

Con el desarrollo de la presente investigación, se da cumplimiento a los objetivos propuestos. Los resultados más relevantes se mencionan a continuación:

- Se obtiene una aplicación web para la gestión de avances académicos y asistencia de los docentes en el centro educativo Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.
- Permite el manejo de la información desde cualquier parte en la que exista conexión, garantizando una mejor confiabilidad, accesibilidad y seguridad de la información.

## RECOMENDACIONES

Tomando como base la investigación realizada y la experiencia acumulada durante la realización de este trabajo, se proponen las siguientes recomendaciones:

- Definir las políticas de seguridad informática para la implantación del sistema e incluir la adecuada preparación del personal que lo utilizará en su trabajo diario.
- Extender el uso de la aplicación a otras instituciones del Ministerio de Educación.

---

**REFERENCIAS BIBLIOGRÁFICAS**

- Aldana, C.R. (2009). *Apache. El servidor Web más reconocido*.
- Angel, G. (2010). Sistema Gestor de Base de Datos. Recuperado el 13 de Abril, 2013, de <http://www.slideshare.net/hcumbicusr/sistemas-gestores-de-base-de-datos-sgbd-13333545>
- Arias, C. (2012). Características de CodeIgniter. Recuperado el 20 de Abril, 2013, de <http://zonalinux.com.ar/codeigniter-un-poderoso-framework-open-source>
- Armando, C. (2008). MySQL vs. PostgreSQL. Recuperado el 15 de Abril, 2013, de <http://www.bisente.com/documentos/mysql-postgres.html>
- Arteaga, D. (2011). NetBeans. Recuperado el 20 de Abril, 2013, de <http://www.ecured.cu/index.php/NetBeans>
- Askallorg, S.A. (2007). *Programando en Java*. Standfor: Addison-Wesley Company.
- Bellman, R.E. (2007). *Web Programming* (Vol. 2). Princenton Princenton University Press.
- Brusilovsky, P., Gilles, G., Frasson, C. y VanLehn, K. (2007). *Course sequencing for Static Courses? Applying ITS Techniques in Large-Scale Web-Based Education* (Springer Verlag ed.): Proceedings of Intelligent Tutoring Systems.
- Cabrera, A. (2008). MySQL vs. PostgreSQL. Recuperado el 12/04/2013, de <http://www.bisente.com/documentos/mysql-postgres.html>
- Calero, S.M. (2010). Programación Extrema (XP). Recuperado el 21 de Abril, 2013, de <http://www.willydev.net/descargas/prev/ExplicaXp.pdf>
- Cambiaso, P. (2009). CodeIgniter un poderoso framework open source - Desarrollo Zona Linux. Recuperado el 18/05/2013 de <http://zonalinux.com.ar/codeigniter-un-poderoso-framework-open-source>
- Cinca, T.C. (2008). Características principales de la Web. Web Estática y Web Dinámica. Aplicaciones para la gestión., 12(4), 9-14,
- Cobo, Á., Gómez, P., Pérez, D. y Rocha, R. (2010). PHP y MySQL Tecnologías para el desarrollo de aplicaciones web. *Ciencias de la información*, 13,
- Colombo, B., Demetrescu, C., I., F. y Laura, L. (2003). A Java-based System for Building Animated Presentations over the Web. *Elsevier Science of Computer Programming*, 901-946,

- Crespo, P. y Pablo, G. (2000). Frameworks. Recuperado el 15 de Abril 2013, de <http://es.wikipedia.org/wiki/Framework>
- Eduardo, G. (2009). sistema SIU-Guaraní. Recuperado el 15 de Marzo, 2013, de <http://dcv.fba.unlp.edu.ar/?p=189>
- Eloi (2009). Tipos de datos en PHP. Recuperado, 15/09/2011, de <http://www.programacionweb.net/articulos/articulo/?num=644>
- Escribano, G.F. (2007). Introducción a Extreme Programming XP. A gentle introduction. Recuperado el 03/03, 2012, de <http://www.extremeprogramming.org>
- Furgeta, A. (2002). *Introduction to Systems Engineering*. New York: MIT Press.
- Gamboa R. W., N.L.P. (2009). *Modelación de un sistema informático para la gestión y control de las actividades asociadas al proceso del despliegue en Venezuela del proyecto PDVAL*. Tesis de Investigativo, Universidad de la Ciencias Informáticas (UCI), La Habana.
- Gracia, J. (2006). *Munual de PHP. WebEstilo*: Adisson-Wesley.
- Hernán Ruiz, M. (2006). *Programación Web avanzada: Soluciones rápidas y efectivas para desarrolladores de sitios*. La Habana: Editorial Félix Varela.
- Ignacio, G. (2008). Desarrollo Web: ¿Qué es cada tecnología? . Recuperado el 09 de Abril, 2013, de <http://WWW.desarrolloWeb.com/manuales/15/>
- Inciarte, A., Marcano, N. y Reyes, M.E. (2006). Gestión académico administrativa en la educación básica. *Venezolana de Jerencia*, 11(34),
- Jacobson, I., Boock, G. y Rumbaugh, J. (2005). *El Proceso Unificado de Desarrollo de Software* (Vol. 1). La Habana: Félix Varela.
- JBC (2009). *Manual de JBC (Java Byte Code)*.
- JDK (2009). Microsystem del JDK 1.0 (Java Development Kit).
- Labs, S. (2012). Symfony. Recuperado el 20 de Abril, 2013, de <http://www.symfony.com>
- Leonard, E.I. (2011). *Sistema para la Gestión de Información Relacionada con el grupo de Operación y Mantenimiento Técnico del Centro Telefónico de ETECSA Holguín*. Tesis de Ingeniero Informático, Universidad de Granma, Bayamo.
- Málaga, D. (2008). *Manual de HTML y JavaScript*.

- Masadelante.com (2009). Qué es un servidor web (Web Servers)? - Definición de servidor web. Recuperado el 08 de Abril, 2013, de <http://www.masadelante.com/faqs/servidor-web>
- Mulet, N. (2002). *Estudio del Flujo de Información para la optimización de procesos en el departamento de Capacitación de ETECSA*. Tesis de Ingeniero Informático, Universidad de La Habana.
- Oliver, K. (2000). Methods for developing constructivist learning on the web. *Educational Technology Research & Development*, 6, 1-18,
- Pedro, C. (2009). CodeIgniter un poderoso framework open source - Desarrollo Zona Linux. Recuperado el 19 de Abril, 2013, de <http://www.codeigniter.com>
- Posted, K. (2008). Agora: Software de gestión docente  
Recuperado el 15 de Marzo, 2013, de [http://www.kherian.com/agora\\_presentacion.asp](http://www.kherian.com/agora_presentacion.asp)
- Potencier, F.Z. (2008). *Symfony la Guía Definitiva*.
- Prentzas, J., Hatzilygeroudis, I. y Garofalakis, J. (2002). A web-Based Intelligent Tutoring System Using Hybrid Rules. *Intelligent Tutoring Systems- ITS 2002*, 119-128.
- Pressman, R. (2007). *Ingeniería del Software. Un enfoque práctico* (Tercera ed. Vol. I).
- Pressman, R. (2008a). *Software Engineering. A practitioner's Approach* (Fourth Edition ed. Vol. Tomo I).
- Pressman, R. (2008b). *SOFTWARE ENGINEERING. A practitioner's Approach*. (Fourth ed. Vol. I).
- Sanz, A. (2002). Introducción a UML. Recuperado el 20/05/2013, de <http://www.yoprogramo.com/docs/umlintro.doc>.
- Solís, M.C. (2003). *Una explicación de la programación extrema(XP)*. Madrid-España.
- Stonebraker, M. (2007). Sistema Gestor de Base de Datos. Recuperado el 11 de Abril, 2013, de [http://www.ecured.cu/index.php/Sistema\\_Gestor\\_de\\_Base\\_de\\_Datos](http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos)
- Tubay, J.L. (2010). *Desarrollo de una aplicación web para el control de avances académicos y asistencia de docentes*. Universidad Técnica Estatal de Quevedo, Quevedo.

- Vázquez, I. (2008). Clientes delgados, el advenimiento de la computación en red. Recuperado el 24/04/2013, de [http://www.mordecki.com/Otros\\_Temas/clientesdelgados/clientesdelgados.shtml](http://www.mordecki.com/Otros_Temas/clientesdelgados/clientesdelgados.shtml)
- Villa, L. (2006). La importancia creciente de la accesibilidad de sitios web. Recuperado el 20/04/2013, de <http://www.alzado.org/articulo.php>
- Zelkowitz, M.V., Shaw, A.C. y Gannon, J.D. (2009). *Principles of Software Engineering and Design* (MIT Press ed. Vol. 2).
- Zuarez, T. (2009). Programmer's Reference Guide, Zend Framework. Recuperado el 18 de Abril, 2013, de <http://framework.zend.com/manual/en/index.html>

## BIBLIOGRAFÍA

A. Navasa, M.A. Pérez, M. Sánchez. Ed. M. Sánchez. 1999. Aplicación de UML al desarrollo de sistemas orientados a objetos. 1999. 84-605-9632-X.

Alexander, Christopher, Ishikawa, Sara y Silverstein, Murray. 1977. A Pattern Language. New York: Oxford University Press, 1977. 0195019199.

Clements, Paul, y otros. 2002. Documenting Software Architectures: Views and Beyond. s.l.: Addison Wesley, 2002.0-201-70372-6.

Cockburn, Alistair. 2000. Writing Effective Use Cases. s.l.: Addison-Wesley, 2000. 0201702258.

Coplien, James. 1996. Software Patterns. s.l.: SIGS, 1996. 978-1884842504.

Delgado, Andrea, y otros. 2008. Metodologías de desarrollo para Service Oriented Architectures con Rational Unified Process. Monte Video, Uruguay: s.n., 2008. 4. Bass, Len, Clements, Paul y Kazman, Rick. 2003. Software Architecture in Practice, Second Edition. s.l.: Pearson Education, Inc., 2003. ISBN: 0-321-154959.

Billy Reinoso, Carlos y Kicillof, Nicolás. 2004. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. Buenos Aires: s.n., 2004

Grady Booch, Robert C, Martin y James Newkirk. Oriented Analysis and Design with Applications. s.l. : Addison Wesley Longman, 1998.

José H. Canós, Patricio Letelier y M<sup>a</sup> Carmen Penadés. Metodologías Ágiles en el Desarrollo de Software.

Stephens, Matt y Rosenberg, Doug. Extreme Programming Refactored: The Case Against XP. Apress, 2003.

Jorge Ferrer Zarzuela. Metodologías Ágiles.

Joskowicz. Reglas y Prácticas en eXtreme Programming. 2008.

PostgreSQL. [En Línea] <http://www.postgresql.org>

PostgreSQL. Características fundamentales [En Línea].

[http://www.netpecos.org/docs/mysql\\_postgres/x15.html](http://www.netpecos.org/docs/mysql_postgres/x15.html)

PostgreSQL Comparativas. [En Línea].

[http://www.netpecos.org/docs/mysql\\_postgres/x108.html](http://www.netpecos.org/docs/mysql_postgres/x108.html)

Miguel, Angel Alvarez. desarrolloweb.com. [Online] [Cited: Marzo 6, 2011.]

<http://www.desarrolloweb.com/articulos/que-es-html.html>

maestrosdelweb.com. [Online] [Cited: Marzo 6, 2011.]

<http://www.maestrosdelweb.com/editorial/introcss/>.

Javier, Garcia Gallego Felipe. css.infames.org. [Online] [Cited: Marzo 7, 2011.]

<http://css.infames.org/ventajas.html>.

## ANEXOS

## Anexo # 1 Historias de Usuario

Tabla 1.1 Historia de Usuario Gestionar Docente.

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre:</b> Gestionar Docente
<b>Usuario:</b> Secretaria(o)	<b>Iteración asignada:</b> 1
<b>Prioridad del negocio:</b> Alta	<b>Puntos estimados:</b> 0.3
<b>Riesgo de desarrollo:</b> Alto	
<b>Programador Responsable:</b> Marco Javier Guanotuña Toasa y Lourdes Catalina Paredes Gavilanez	
<b>Descripción:</b> Se podrá insertar, modificar y eliminar los docentes del Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.	
<b>Observaciones:</b> Solo se permite la inserción de docentes que trabajen en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez” y la eliminación de aquellos que ya no brinden su servicio en la institución.	

Tabla 1.2 Historia de Usuario Gestionar Departamento.

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre:</b> Gestionar Departamento
<b>Usuario:</b> Secretaria(o)	<b>Iteración asignada:</b> 1
<b>Prioridad del negocio:</b> Media	<b>Puntos estimados:</b> 0.2
<b>Riesgo de desarrollo:</b> Medio	
<b>Programador Responsable:</b> Marco Javier Guanotuña Toasa y Lourdes Catalina Paredes Gavilanez	
<b>Descripción:</b> Se podrá insertar, modificar y eliminar el(los) departamento(s) Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.	
<b>Observaciones:</b> Solo se permite la inserción de departamento que pertenezca al Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.	

Tabla 1.3 Historia de Usuario Gestionar Asignatura.

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre:</b> Gestionar Asignatura
<b>Usuario:</b> Docente	<b>Iteración asignada:</b> 1
<b>Prioridad del negocio:</b> Alto	<b>Puntos estimados:</b> 0.2
<b>Riesgo de desarrollo:</b> Medio	
<b>Programador Responsable:</b> Marco Javier Guanotuña Toasa y Lourdes Catalina Paredes Gavilanez	
<b>Descripción:</b> Se podrá insertar, modificar y eliminar el (las) asignatura(s) que	

se imparten en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.

**Observaciones:** Solo se permite la inserción de asignatura que se imparte en el Colegio Técnico Nacional “Dr. Camilo Gallegos Domínguez”.

## Anexo # 2 Tareas para realizar las Historias de Usuario

Tabla 2.1 Tarea # 1 de la Historia de Usuario Gestionar Estudiante.

Tarea	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Diseño de interfaz de HU y programación de insertar estudiante.	
<b>Tipo de Tarea:</b> Diseño-Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 04 de Abril del 2013	<b>Fecha Fin:</b> 04 de Abril del 2013
<b>Programador Responsable:</b> Marco Guanotuña y Lourdes Catalina Paredes	
<b>Descripción:</b> Se diseñará una página donde la (el) secretaria(o) pueda insertar estudiante, se programará un método que permita insertar estudiante.	

Tabla 2.2 Tarea # 2 de la Historia de Usuario Gestionar Estudiante.

Tarea	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Programación de eliminar estudiante.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 05 de Abril del 2013	<b>Fecha Fin:</b> 05 de Abril del 2013
<b>Programador Responsable:</b> Marco Guanotuña y Lourdes Catalina Paredes	
<b>Descripción:</b> Se programará un método que permita a la (el) secretaria(o) eliminar un estudiante.	

Tabla 2.3 Tarea # 3 de la Historia de Usuario Gestionar Estudiante.

Tarea	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Programación de modificar estudiante.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 06 de Abril del 2013	<b>Fecha Fin:</b> 06 de Abril del 2013
<b>Programador Responsable:</b> Marco Guanotuña y Lourdes Catalina Paredes	
<b>Descripción:</b> Se programará un método que permita a la (el) secretaria(o) modificar un estudiante.	

## Anexo # 3 Imágenes del sistema

Figura 3.1 Pantalla de inicio al sistema.

**SIGECAD**  
Sistema de gestión de avances académicos y asistencia de docentes

**Colegio Técnico Nacional "Dr. Camilo Gallegos Domínguez"**

Inicio Codificadores Docentes Estudiantes Reportes Usuarios Ayuda administrator

### Usuarios

A continuación está una lista de los usuarios.

Nombre	Apellidos	Correo	Grupos	Estado	Acción
Admin	istrator	admin@admin.com	<a href="#">admin</a>	<a href="#">Activo</a>	<a href="#">Edit</a>
Erodis	Pérez Michel	eperezm@udg.co.cu	<a href="#">director</a>	<a href="#">Activo</a>	<a href="#">Edit</a>
Martha	Jay	mjay@udg.co.cu	<a href="#">secretaria</a>	<a href="#">Activo</a>	<a href="#">Edit</a>
Raubel	Sánchez	raubel@udg.ec	<a href="#">estudiante</a>	<a href="#">Activo</a>	<a href="#">Edit</a>
juan	beliz	juan@udg.cu	<a href="#">docente</a>	<a href="#">Activo</a>	<a href="#">Edit</a>
Lourdes	Paredes	lourdes@yahoo.es	<a href="#">estudiante</a>	<a href="#">Inactivo</a>	<a href="#">Edit</a>
Victor	Jay	vjay@udg.co.cu	<a href="#">estudiante</a>	<a href="#">Activo</a>	<a href="#">Edit</a>

[Crear un nuevo usuario](#) | [Crear un nuevo grupo](#)

Creado por Marco y Caty

Figura 3.2 Pantalla de para la gestión de usuarios del sistema.

**SIGECAD**  
Sistema de gestión de avances académicos y asistencia de docentes

**Colegio Técnico Nacional "Dr. Camilo Gallegos Domínguez"**

Inicio Codificadores Docentes Estudiantes Reportes Ayuda eperezm

Exportar Imprimir

Mostrar 25 entradas Buscar:

Estudiante	Promedio
Juan Perico	5.44 pts
Marco Guanotuña	10 pts

Buscar Estudiante Buscar Promedio

Mostrando 1 a 2 de 2 registros Primero Anterior 1 Siguiente Último

Creado por Marco y Caty  
Página mostrada en 0.3001 segundos

Figura 3.3 Pantalla del reporte escalafón del sistema.



## Anexo # 5 Pruebas de Aceptación del sistema

Tabla 5.1 Prueba # 1 de la Historia de Usuario Gestionar Docente.

Prueba de Aceptación	
<b>Número Prueba:</b> 1	<b>Número Historia de Usuario:</b> 4
<b>Nombre de la Prueba:</b> Verificar inserción incorrecta de docente.	
<b>Descripción:</b> El (la) secretario(a), al seleccionar la opción del menú “Gestionar Docente” -/. “Insertar Docente” se le brinda la opción de insertar un nuevo docente. El (la) secretario(a) debe introducir los campos obligatorios y cuando se cumple que los mismos han sido llenados insatisfactoriamente, se mostrará un mensaje indicándose el error.	
<b>Condiciones de ejecución:</b> Ninguna.	
<b>Entrada:</b> <ul style="list-style-type: none"> <li>• El (la) secretario(a) selecciona la opción del menú “Gestionar Docente” -/. “Insertar Docente”.</li> <li>• Aparece un formulario en el que se solicita nombre, contraseña.</li> <li>• El (la) secretario(a) introduce los campos y presiona el botón “Insertar”.</li> <li>• El sistema verifica que los campos cumplan con las validaciones de cada uno de ellos y comprueba que estén incorrectamente escritos.</li> <li>• El sistema muestra un mensaje de error y regresa al mismo formulario.</li> </ul>	
<b>Resultado esperado:</b> Sólo puede ser insertado un nuevo docente cuando todos los campos requeridos sean llenados correctamente.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	