

# **GENERACIÓN DE EXÁMENES ALEATORIOS II**

**Jorge Ignacio Morales**

**Ingeniería Sistemas de Información**

**Jorge Iván Bedoya Restrepo**

**INSTITUTO TECNOLÓGICO METROPOLITANO**

**22/07/2015**

# Resumen

Este proyecto de Generación Aleatoria de Exámenes está diseñado bajo el Framework Codeigniter Versión 2.0 y su patrón de arquitectura de Software modelo–vista–controlador (MVC), el cual se encarga de separar los datos y la lógica de negocio de la interfaz de usuario, también es el encargado de gestionar los eventos y las comunicaciones de la aplicación. Para ello la estructura MVC consta de tres componentes distintos, que son modelo, vista y controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

A continuación se describe el funcionamiento de los componentes MVC:

**Vista:** Esta es principalmente la página bajo lenguaje HTML la cual se encarga del entorno de visualización.

**Controlador:** Es la parte principal del proyecto, ya que en ella se procesa toda la lógica del proyecto y por medio del código se obtienen datos dinámicamente y se genera el contenido de la vista

**Modelo:** Es en donde se interactúa con la información almacenada en una base de datos o en XML, junto con las reglas de negocio que transforman esa información.

Nuestro principal resultado es haber cumplido las expectativas y objetivos planteados por la Jefatura de sistemas del ITM, y poder dejar un software tan importante a la universidad que nos vio crecer y formar como profesionales, además con este proyecto incrementamos nuestro conocimiento de programación e investigación, ya que al momento de desarrollar esta aplicación web, hubieron algunos módulos de desarrollo que no sabíamos implementar, pero consultando e investigando sobre los diferentes temas pudimos optar por las mejores soluciones, utilizando las mejores prácticas de desarrollo.

Como conclusión tenemos que en la universidad se pueden automatizar muchos más procesos de docencia para mejorar la calidad de los estudiantes, y además poder medir la forma en que los docentes imparten su cátedra.

Con este software se podrá realizar un seguimiento detallado de las actividades que están realizando los docentes, y como las están haciendo.

*\*(MVC): Modelo -Vista - Controlador*

# Reconocimientos

Le agradezco principalmente a mi madre, ya que gracias a ella he podido llegar a donde estoy, porque con su esfuerzo me ha brindado todo el apoyo posible para que yo salga adelante y sea una persona de bien.

Agradezco a Dios por permitirme ser la persona que soy, por iluminar mi camino para tomar las mejores decisiones, colocando a personas maravillosas que han sido parte de mi aprendizaje diario, brindándome su apoyo para salir cada día adelante.

También quiero agradecer a mis familiares y amigos allegados, ya que han sido un gran soporte para no tirar la toalla y asumir los retos difíciles que me ha puesto la vida, haciéndome más fuerte y llenándome de sabiduría.

Por ultimo quiero agradecer a la Institución ya que con el apoyo de muchos profesores no me he formado solo en conocimientos sino como persona.

**Jorge Ignacio Morales**

# Tabla De Contenido

## Contenido

Resumen.....	2
Reconocimientos.....	3
Tabla De Contenido .....	4
1. Introducción .....	5
• Generalidades .....	5
• Objetivos .....	5
○ General .....	5
○ Específicos.....	6
2. Marco Teórico.....	7
• Patrones De Diseño .....	7
○ Patrones Creacionales.....	7
○ Patrones Estructurales:.....	7
○ Patrones de Comportamiento .....	7
○ Patrón de diseño Modelo-Vista-Controlador .....	7
• Frameworks de Desarrollo .....	10
○ Framework Codeigniter .....	11
• Metodologías Agiles de Desarrollo .....	13
3. Metodología .....	14
4. Resultados Y Discusión.....	19
5. Conclusiones, Recomendaciones Y Trabajo Futuro.....	20
• Conclusión General.....	20
• Conclusiones Específicas.....	20
6. Referencias .....	21

# 1. Introducción

- **Generalidades**

El propósito de este trabajo está basado en la necesidad de automatizar el proceso de evaluaciones para los estudiantes de la Jefatura de sistemas de información, donde el proceso que está definido actualmente es que los docentes generen el examen de los estudiantes manualmente, dependiendo de los temas que se hayan dictado en cada una de sus asignaturas y también adicional a esto generar los compromisos académicos de acuerdo a los temas estipulados por la decanatura.

La forma de generar los exámenes e impartir los temas a los estudiantes actualmente, permite la posibilidad implícitamente que algunos docentes omitan la evaluación de algunos temas que hayan sido impartidos por la facultad, y no habría forma de detectar cuáles fueron los temas que se dejaron de dictar y cuáles fueron los docentes que no los dictaron.

## **SEGUNDA FASE**

La segunda fase del proyecto tiene como objetivo optimizar el proceso de la generación de exámenes, implementando un módulo adicional, que permita la visualización de los exámenes previamente generados con sus respectivas respuestas; además producir compromisos académicos de cada asignatura aprovechando la información almacenada en la aplicación.

- **Objetivos**

- **General**

Crear una aplicación web, donde el Jefe de Programa de Sistemas o algún coordinador encargado, puedan generar de forma aleatoria las evaluaciones de los estudiantes en el ITM, teniendo la posibilidad de controlar la gestión que realizan los docentes al momento de dictar sus temas de trabajo.

También se podrá obtener un reporte diario que muestre las fechas de compromiso que tienen los docentes al momento de impartir los diferentes temas a sus grupos de trabajo.

## **SEGUNDA FASE**

Visualizar exámenes generados previamente con sus respectivas respuestas, mediante la consulta por un código único de examen, además producir compromisos académicos de cada asignatura incluyendo el respectivo contenido temático.

- **Específicos**

- ✓ Mejorar el proceso de evaluaciones de los estudiantes de sistemas de Información en el ITM, para que sea realizada por un software de manera aleatoria y no manualmente como lo están realizando los docentes.
- ✓ Facilitar a la Jefatura de sistemas el control de la forma en que los docentes imparten los Temas a los estudiantes de sistemas de Información.
- ✓ Identificar los roles y responsabilidades de cada una de las personas que intervendrán en la automatización del proceso de evaluaciones, para que los funcionarios involucrados asuman el papel y la responsabilidad que les corresponde en el contexto del proyecto y del ITM.

### **SEGUNDA FASE**

- ✓ Generar compromisos académicos detallando la competencia general de cada asignatura, sus logros y los contenidos temáticos de la misma
- ✓ Visualizar exámenes generados previamente por medio de la aplicación y las respuestas de los mismos.

## 2. Marco Teórico

A continuación se describe en forma breve los Patrones de Diseño, Arquitecturas de Software, Frameworks y Proceso de Desarrollo, que utilizamos para la implementación de nuestro proyecto, sobre el cual hacemos uso del patrón de diseño “**Modelo-Vista-Controlador**”, con el Frameworks Codeigniter.

### • Patrones De Diseño

La mayor parte de los desarrolladores de software, necesitan patrones de diseño; estos puedan ser modulares y reutilizables.

Los patrones de diseño no son líneas de código que puedan reutilizarse después, son descripciones de clases y objetos relacionados para resolver un problema general en un determinado contexto.

En base al propósito, o en función de lo que cada patrón realiza, tenemos la siguiente clasificación de patrones:

- **Patrones Creacionales:** Están relacionados con los procesos de creación de objetos, solucionan problemas de creación de instancias, y ayudan a encapsular y extraer dicha creación.
- **Patrones Estructurales:** Se relacionan con la composición de clases u objetos, estos separan la interfaz de la implementación, se ocupan de cómo las clases y objetos se agrupan para formar estructuras más grandes.
- **Patrones de Comportamiento:** Describen el modo en el cual clases y objetos interactúan y distribuyen sus responsabilidades, así como los algoritmos que encapsulan o muestran la comunicación entre los objetos.
- **Patrón de diseño Modelo-Vista-Controlador:** Para el diseño de las aplicaciones Web con interfaces enriquecidas que ofrezcan una gran interactividad con el usuario, se utiliza el patrón de diseño Modelo - Vista – Controlador; se parte del hecho que la lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño que mezcle los componentes de interfaz y de negocio, cuando se requiera cambiar la interfaz de usuario, tendremos que modificar necesariamente los componentes de negocio, esto trae consigo mayor trabajo y una alta probabilidad de error.

Con este patrón se trata de realizar un diseño que separe la vista del modelo, con la finalidad de mejorar la reusabilidad, de esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Los siguientes son las 3 capas del patrón MVC:

**Modelo:**

En esta capa se hace el levantamiento de todos los objetos que debe utilizar el sistema, es decir, es el proveedor de recursos. Es donde se encapsulan los datos y la lógica del negocio, esto hace que sea independiente de la base de datos.

**Vista:**

Esta es la capa de presentación la cual muestra la información del modelo al usuario, se agrupan todas las clases y archivos que tengan relación de la interfaz de usuario. Cada vista tiene asociado un componente controlador como se indica a continuación.

**Controlador:**

El controlador viene a ser el unificador de los diferentes eventos generados en la interfaz de usuario, se encarga de llamar en el modelo, quien es el que sabe lo que hay que hacer con la petición del usuario.

Una vez que el modelo ha realizado su tarea se lo comunica al controlador; el controlador invoca a la vista para que se actualice con los cambios hechos en el modelo.

En resumen, el Controlador es el que escucha los cambios en la vista y se los envía al modelo, el cual le regresa los datos a la vista, es un ciclo donde cada acción del usuario causa que se inicie un nuevo ciclo.

En la Figura 1 se muestran las responsabilidades de cada capa del patrón de diseño MVC y cómo interactúan entre sí

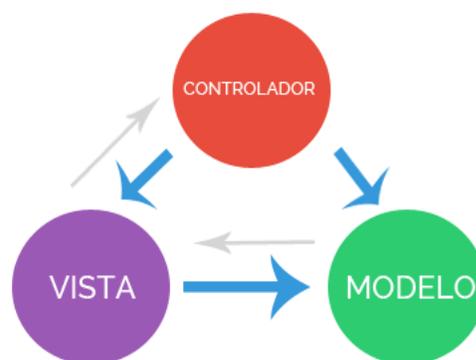


Figura 1

<http://andresoller.es/blog/tutorial-basico-codeignaiter-conocimientos-basicos-e-instalacion/>

La interacción que se muestra en la Figura 1 es la siguiente:

- I. **Solicitud:** El usuario ejecuta un navegador, pide un URL solicitando a la Aplicación.
- II. **Despacha:** La Aplicación delega a la capa Controladora la petición del usuario.
- III. **Almacena:** Consulta en la capa Modelo por medio de la capa de datos del manejador de base de datos existente.
- IV. **Devuelve:** La capa de Datos devuelve los resultados en datos puros a la capa Controladora.
- V. **Genera:** La capa Controladora genera la capa de Vista en base a los tipos de datos generados.
- VI. **Responde:** La Vista generada se envía como respuesta a la solicitud del navegador.

Se puede afirmar que en el desarrollo de aplicaciones Web, la programación utilizando el patrón de diseño MVC es una alternativa que evitará en el futuro muchos dolores de cabeza, gracias a sus características que en resumen son las siguientes:

- ✓ Patrón de diseño orientado a objetos.
- ✓ Adecuado para aplicaciones Web con alta interacción humana.
- ✓ Separa clara y consistentemente las preocupaciones en las capas indicadas anteriormente.
- ✓ Permite múltiples representaciones (vistas) de la misma información (modelo).
- ✓ Facilidad para agregar, eliminar o modificar interfaces de usuario.
- ✓ Facilita los desarrollos simultáneos con actualizaciones de interfaces, lógica del negocio o incluso introducir una nueva aplicación sin la necesidad de afectar a otro código fuente.
- ✓ Garantiza la reducción del código fundamentalmente porque los modelos pueden ser utilizados en varias vistas.
- ✓ Los desarrolladores centran su atención en un solo aspecto de la aplicación al mismo tiempo

- **Frameworks de Desarrollo**

Los Frameworks son una estructura básica o genérica de soporte sobre la cual un proyecto puede ser organizado y desarrollado en su totalidad.

En otras palabras, se puede considerar como una aplicación genérica incompleta y configurable a la que se le pueden añadir las piezas faltantes para construir una aplicación completa.

En general los Frameworks son un conjunto de buenas prácticas de desarrollo representados en una serie de librerías, bajo un único esquema de colaboración que resuelven un problema en particular el cual se puede tomar como base para resolver nuevos problemas con las mismas características y así crear aplicaciones de manera rápida.

En el momento en que decidimos utilizar un Frameworks para construir una aplicación, adoptamos la arquitectura del Frameworks, es decir, heredará la estructura general, su partición en clases y objetos, las responsabilidades clave, cómo colaboran las clases y objetos y el hilo de control, esto permite al diseñador o programador de la aplicación concentrarse en las funcionalidades propias del sistema que se está desarrollando.

Los Frameworks hacen hincapié en la reutilización del diseño frente a la reutilización del código, sin embargo, incluye normalmente subclases concretas listas para trabajar, lo cual permite construir aplicaciones rápidamente y que tengan estructuras parecidas por lo que son más fáciles de mantener y resultan más consistentes para los usuarios.

Encontramos Frameworks que parten del paradigma MVC que como se indicó anteriormente separa en la aplicación la gestión de los datos, las operaciones, la presentación y Frameworks que pueden llegar a tal detalle de definir los nombres de los archivos su estructura y los estándares de programación.

Los principales objetivos que persigue un Frameworks son los siguientes:

- Acelerar el proceso de desarrollo.
- Reutilizar código existente.
- Promover buenas prácticas de desarrollo como es el uso de patrones.
- Desarrollo estructurado.
- Disminuir el esfuerzo en el desarrollo.
- Aprovechar funcionalidades ya implementadas.
- Centrarse directamente en la solución del problema.
- Trabajar directamente con metodologías de desarrollo Ágiles

En la actualidad encontramos una infinidad de Frameworks, en la Tabla 1 se detallan las más comunes que se utilizan con lenguajes de programación PHP y

Java.

PHP	JAVA
Yii	Struts
CodeIgniter	Hibernate
CakePHP	JavaServerFaces
Zend	Spring
Symfony	Apache Wicket
PHPDevShell	Google Web Toolkit

*Tabla 1*

Existen muchos Frameworks como los indicados en la Tabla 1, que tienen características similares a Codeigniter, sin embargo, se ha seleccionado Codeigniter por cuatro motivos fundamentales:

- I. Trabaja con PHP como lenguaje de programación.
- II. Utiliza el patrón de diseño MVC.
- III. La aplicación será completamente Web.
- IV. Se encuentra bajo una licencia Open Source.

- **Framework Codeigniter**

Codeigniter es un Frameworks orientado a personas que construyen sus aplicaciones web utilizando PHP.

Permite desarrollar proyectos mucho más rápido de lo que se podría hacer si se lo escribiese desde cero, para esto provee muchas librerías para tareas repetitivas y necesarias así como una interface simple y estructura lógica para acceder a esas librerías.

Codeigniter permite enfocarse principalmente en el proyecto minimizando la cantidad de código necesaria para una tarea dada.

Codeigniter se encuentra bajo una licencia open Source Apache/BSD-style, es compatible con PHP4 y PHP5; está basado en el patrón de diseño Modelo-Vista-Controlador que permite separar el diseño con el código, agilizar las funciones a la base de datos y crear librerías independientes para ser reutilizadas en cualquier instancia de la aplicación. Al utilizarse librerías en una aplicación se logra agilizar y optimizar la programación a nivel Web.

Codeigniter viene con un conjunto de librerías que permiten realizar las tareas de desarrollo Web típicamente necesarias como: Acceder a una base

de datos, enviar un email, validar datos de un formulario, mantener sesiones, manipular imágenes, trabajar con datos XML-RPC, etc.

Las características principales de Codeigniter son las siguientes:

- ✓ Extremadamente Liviano
- ✓ Clases de base de datos llenas de características con soporte para varias plataformas.
- ✓ Soporte de Active Record para Base de Datos
- ✓ Formulario y Validación de Datos
- ✓ Seguridad y Filtro XSS
- ✓ Manejo de Sesión
- ✓ Clase de Envío de Email. Soporta Archivos Adjuntos, email detexto/HTML, múltiples protocolos (sendmail, SMTP, and Mail) y más.
- ✓ Librería de Manipulación de Imagen (cortar, redimensionar, rotar, etc.). Soporta GD, ImageMagick, y NetP
- ✓ Clase de Carga (upload) de Archivo
- ✓ Clase de FTP
- ✓ Localización
- ✓ Encriptación de Datos
- ✓ Puntos de referencia
- ✓ Historial de Errores
- ✓ Perfilando la Aplicación
- ✓ Scaffolding
- ✓ Clase de Calendario
- ✓ Clase de Agente del Usuario
- ✓ Clase de Codificación Zip
- ✓ Clase de Motor de Plantillas
- ✓ Clase de Trackback
- ✓ Librería XML
- ✓ RPC
- ✓ Clase de Prueba de Unidad
- ✓ URLs amigables a motores de búsqueda
- ✓ Ruteo de URI Flexible
- ✓ Soporte para Ganchos, Extensiones de Clase y Plugins
- ✓ Larga librería de funciones "asistentes"

En el desarrollo del presente proyecto se explota la mayoría de estas Características, principalmente las siguientes:

- ✓ Uso del patrón de diseño Modelo-Vista-Controlador.
- ✓ Active Record para Base de Datos
- ✓ Formulario y Validación de Datos
- ✓ Manejo de Sesión
- ✓ Encriptación de Datos
- ✓ Clase de Codificación Zip

- ✓ URL amigables a motores de búsqueda
- ✓ Seguridad y Filtro XSS

- **Metodologías Ágiles de Desarrollo**

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes del proyecto, en nuestro caso la Jefatura de sistemas, simplicidad en las soluciones implementadas y tolerancia para enfrentar los cambios.

XP se define especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes como lo teníamos en este proyecto, donde existe un alto riesgo técnico.

Como caso de estudio en este proyecto se trabajará con el Framework para PHP **Codeigniter**, con el cual se aplicará el patrón de diseño MVC para la construcción de una aplicación Web, con la combinación de estas herramientas y teniendo definido la forma de trabajo y como se implementara, manejaremos la metodología de desarrollo ágil XP la cual se acopla a nuestras necesidades.

### 3. Metodología

El proyecto surgió de una necesidad identificada por el Jefe de Programa de sistemas de información de la Institución (Leonel Velásquez), con el fin de llevar un mayor control de los temas evaluados en cada una de las asignaturas.

En vista de esto, tuvimos una primera reunión en la cual hablamos sobre lo que se quería alcanzar con el proyecto, resaltando que el objetivo principal era generar un examen con preguntas aleatorias de los diferentes temas de las asignaturas.

Ver Figura2.

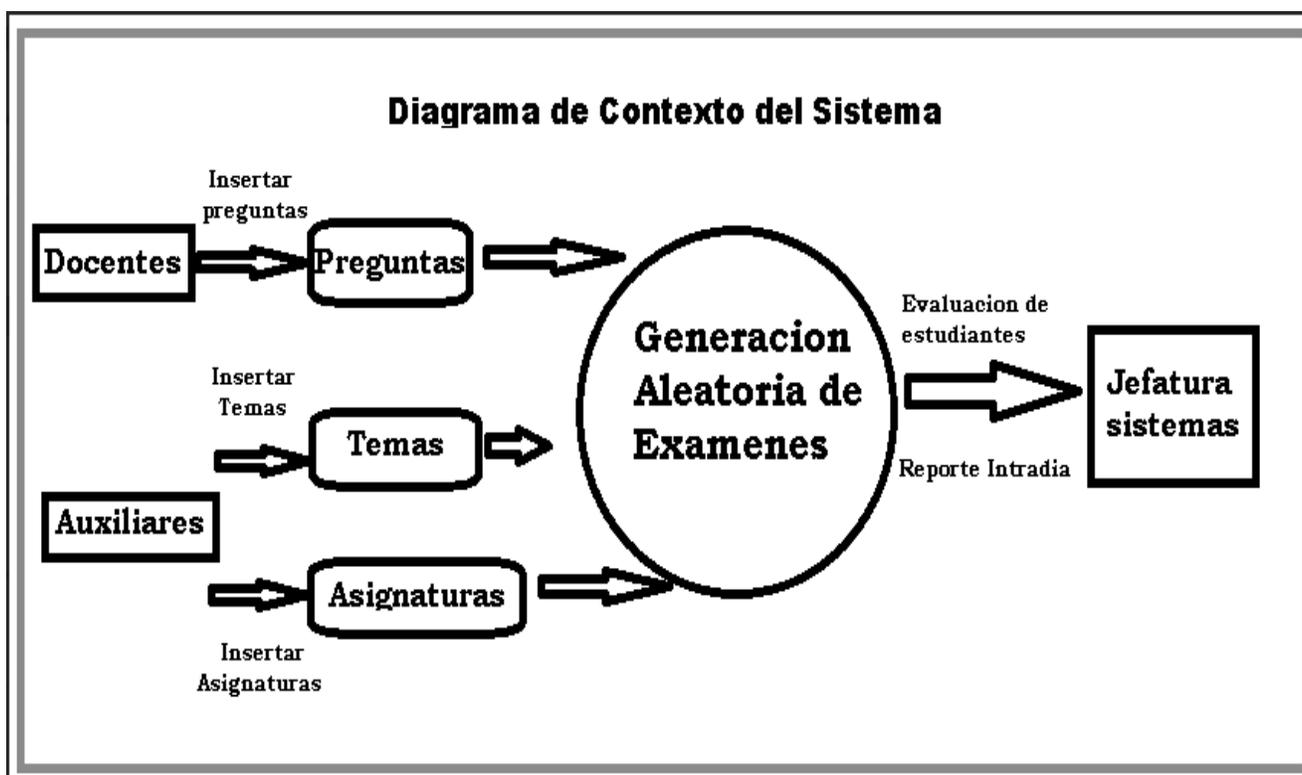


Figura 2

En compañía del profesor Jorge Iván Bedoya, en la siguiente reunión analizamos de qué manera podíamos implementar la estructura de la base de datos según lo requerido por Leonel; de acuerdo a lo evaluado comenzamos a implementar la estructura para su aprobación, la cual fue aprobada satisfactoriamente (ver Figura 3)

## Modelo Relacional

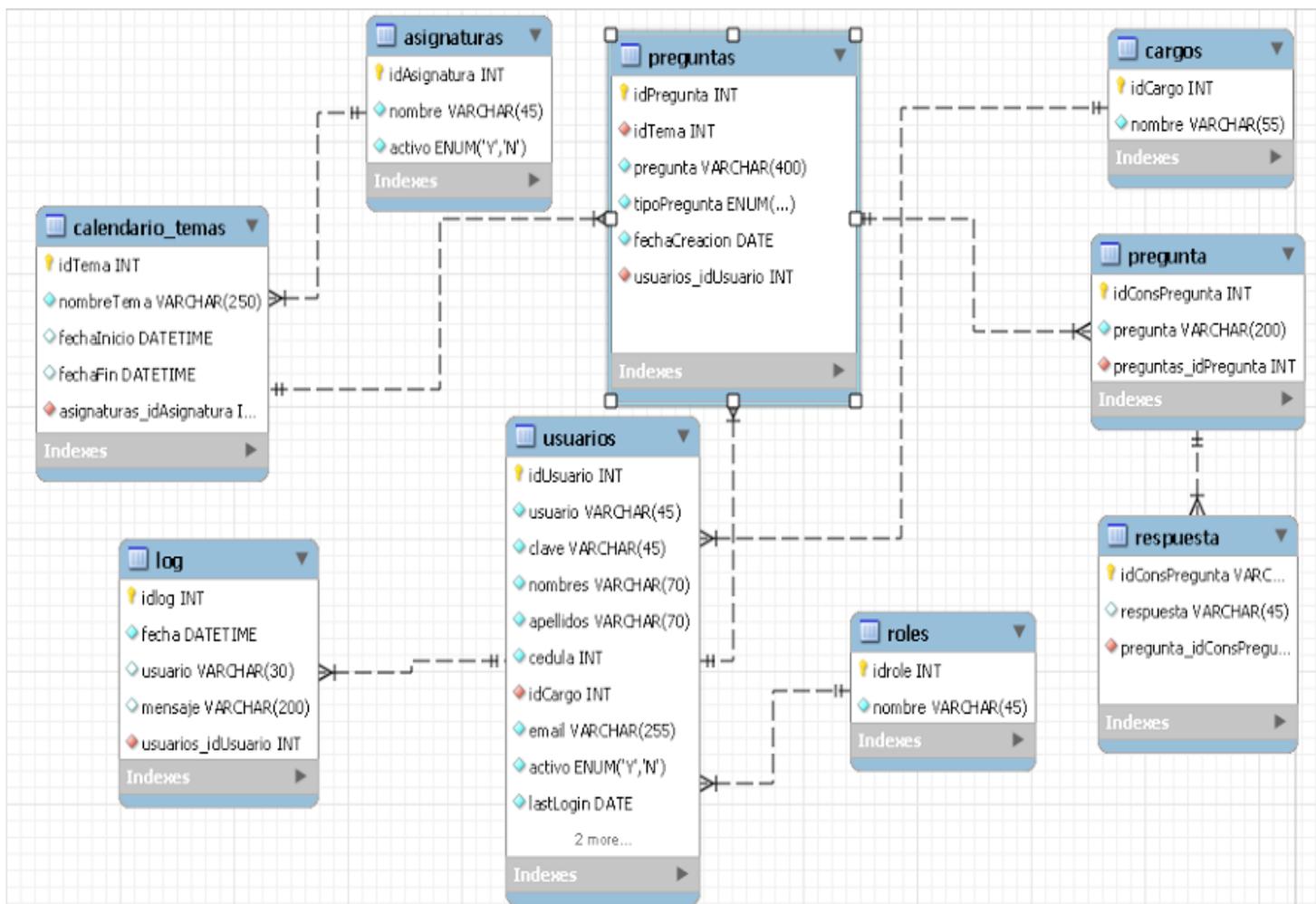


Figura 3

Posterior a la aprobación de la estructura de base de datos, comenzamos a analizar de qué forma abarcar el desarrollo, logrando identificar algunos módulos maestros que sería la base para el desarrollo del proyecto, ver *Figura 4*; en vista de esta necesidad consultamos diferentes Framework para desarrollar en PHP. Luego de varias pruebas y evaluar la funcionalidad de algunos de ellos, el que más se acoplaba para el desarrollo que vamos a implementar, era el Framework Codeigniter, el cual está diseñado bajo el patrón Modelo, Vista, Controlador, esto nos facilitaría el manejo de las librerías y estructuras de trabajo.

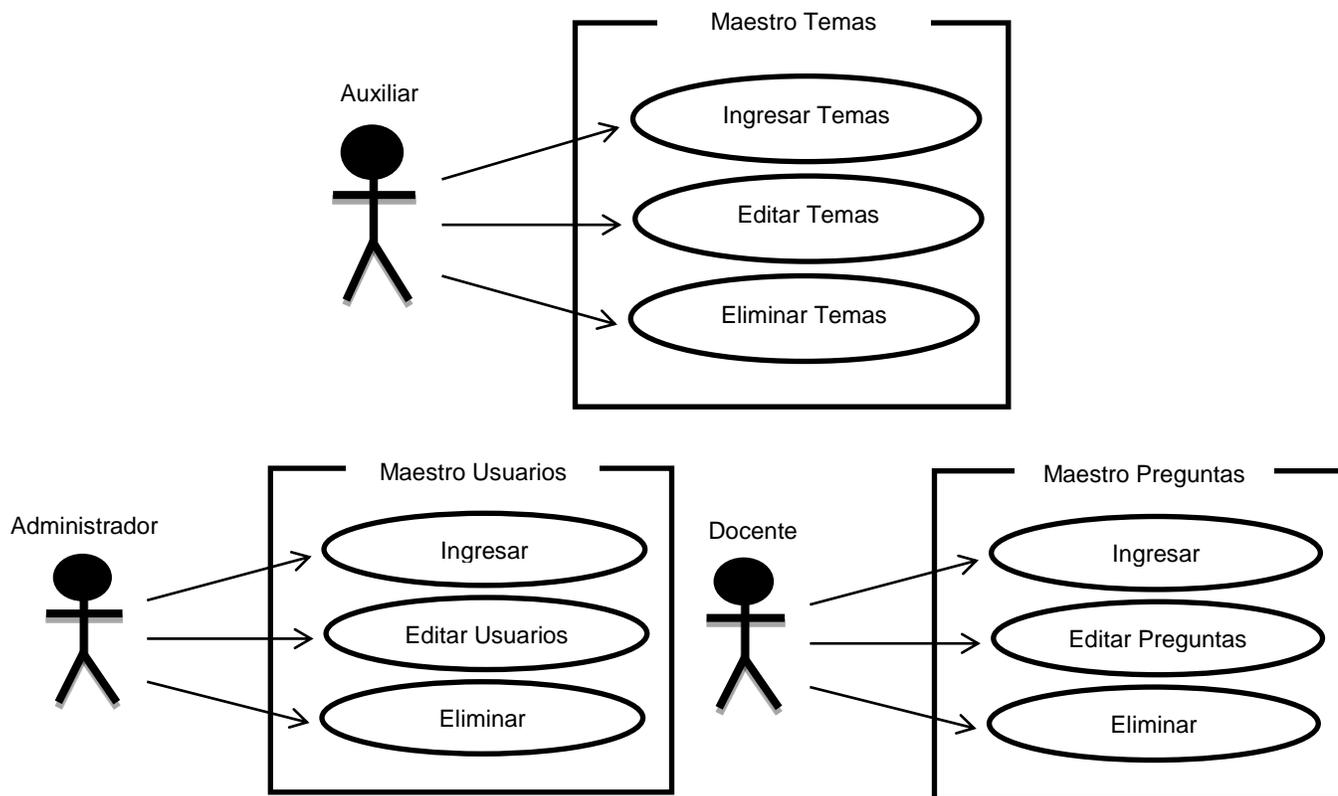


Figura 4

Luego de tener identificado el patrón y modelo de trabajo, en vista de las necesidades planteadas por parte de la facultad de sistemas optamos por utilizar la **Metodología de Programación XP (Xtrem Programming)**, la cual está diseñada para desarrollo ágil y se acopla a nuestro proyecto final, ya que tiene como características comunicación, Retroalimentación, Simplicidad, Adaptación a Cambios y desarrollo de nuevas requerimientos sobre la marcha del desarrollo, entre otras.

Dicha metodología se ven reflejadas en el presente proyecto, ya que hemos avanzado en cada uno de las etapas del proyecto como es planteada por **XP**.

Teniendo en cuenta la metodología y después de haber tomado los requerimientos y aprobaciones necesarias, comenzamos a desarrollar el proyecto planteado.

Luego de finalizar los módulos maestros, tuvimos una tercera reunión con el jefe de programa y Asesor, para mostrar y evaluar los avances del proyecto, en donde se evidencio un gran adelanto del desarrollo y se identificó la necesidad un nuevo requerimiento "Reporte Día a Día" (Figura 5); este tendría como finalidad tener un reporte donde muestre las fechas de compromiso de los docentes en cada Tema.



Figura 5

Después de obtener los nuevos requerimientos planteados por la Jefatura procedimos con el desarrollo del proceso principal “Generación de Exámenes” (Figura 6), y el nuevo reporte solicitado.

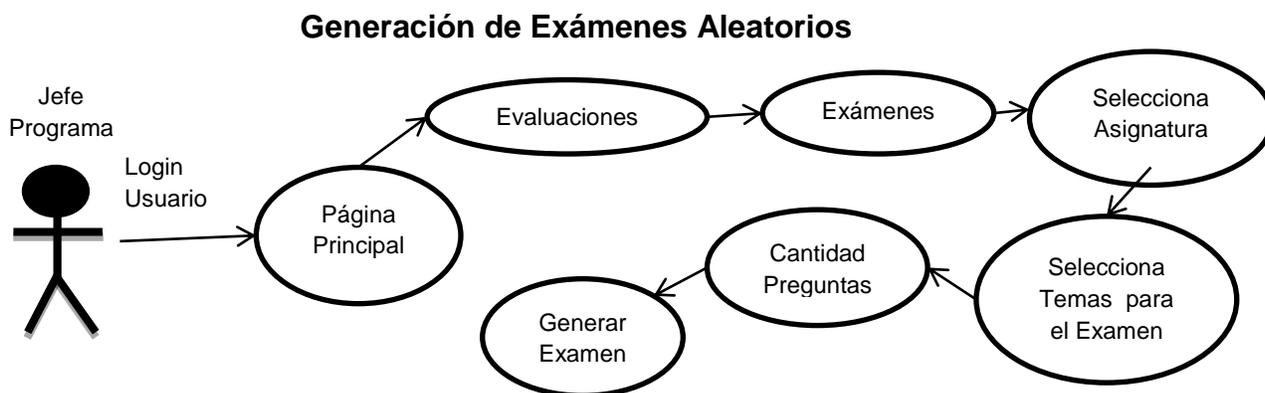


Figura 6

Al momento de desarrollar la última parte del proyecto encontramos un gran reto el cual era exportar a PDF tanto el Examen Generado como el reporte Intradía. Luego de investigar diferentes librerías encontramos TCPDF la cual es compatible totalmente al Framework utilizado y se ajustaba a nuestras necesidades.

Finalizando el proyecto tuvimos una cuarta Reunión con el Asesor y Jefe de Programa donde se le hizo entrega del software, se realizaron pruebas en conjunto y se dio por terminado el proyecto, dando por cumplidos los objetivos propuestos.

## SEGUNDA FASE

La implementación de la segunda fase se realizó con el fin de suplir dos necesidades bastante importantes para el buen manejo de la herramienta; la visualización de exámenes ya generados, al igual que las respuestas de los mismos y la generación del compromiso académico para cada asignatura ya que con la información suministrada en el software es posible extraer dicha información y así garantizar la misma estructura en el contenido temático la asignatura independiente del horario o del curso.

Para consultar los exámenes ya generados y las respuestas de los mismos, se definió agregar una opción en el módulo de administración, teniendo en cuenta que se debía consultar por el código del examen o por asignatura con el fin de visualizar todos los exámenes generados para la misma; de acuerdo a lo anterior se definió la estructura de la

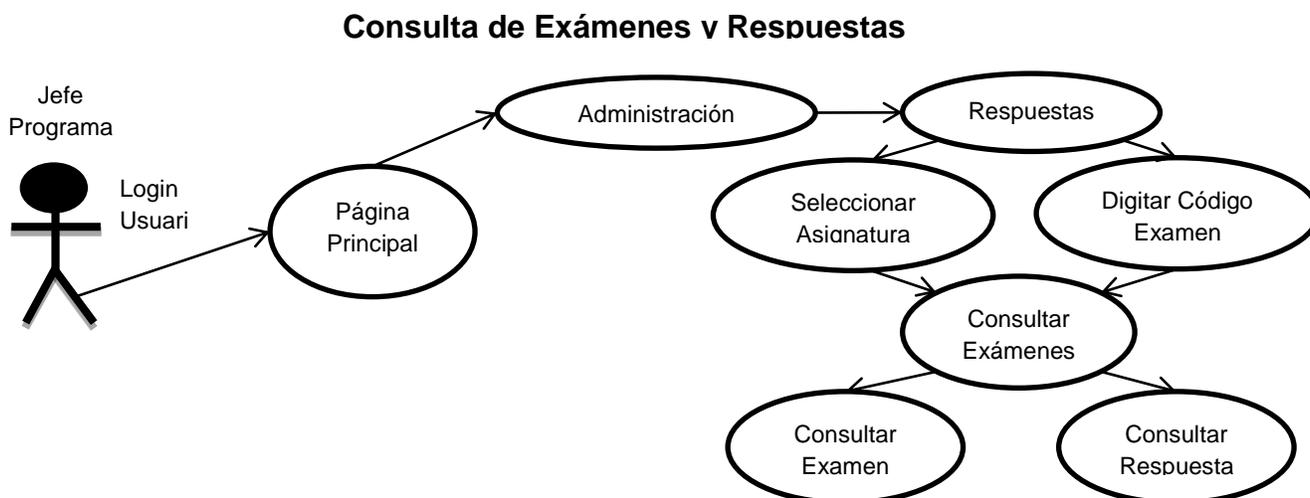


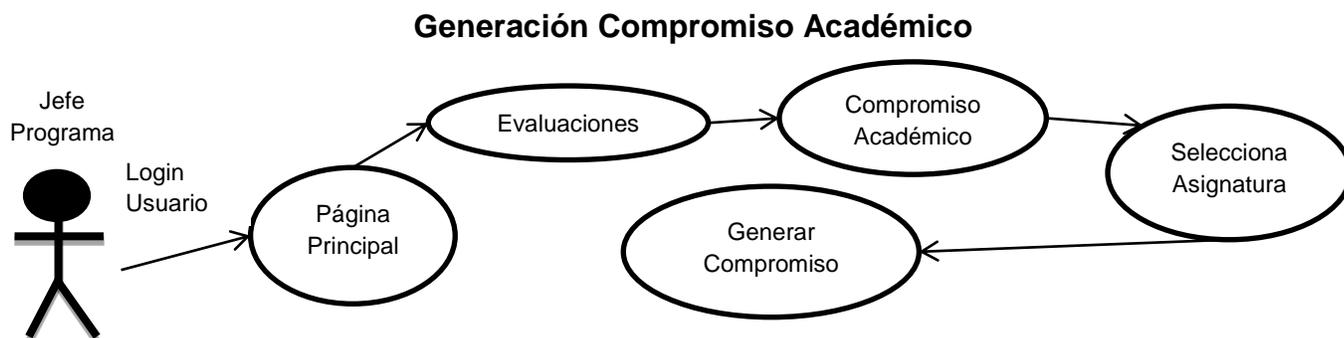
Figura 7

Para la generación del compromisos académicos, tuvimos en cuenta la información que ya se había ingresado al sistema, como las asignaturas y sus respectivos contenidos temáticos, adicional a esto debían agregar las competencias, los logros y los ejes temáticos de cada una.

El compromiso académico original tiene dos opciones que no se encuentran registradas en la aplicación, por lo cual se van a agregar como opciones al momento de generar dicho compromiso.

El compromiso se dividió en dos partes, Compromiso Docente el cual contiene la información como porcentajes y forma de evaluación al igual que las reglas del profesor para el curso, y el Compromiso Académico que es el que contiene la información de la asignatura y los contenidos temáticos que se extraen de la aplicación.

La generación de este compromiso se detalla en la *Figura 8*.



## 4. Resultados Y Discusión

La generación de exámenes aleatorios busca como resultado mejorar el nivel académico de los estudiantes y estandarizar los procesos de evaluación para las diferentes asignaturas. Donde los docentes sean quienes alimenten el software de preguntas, para luego generar el examen de manera aleatoria; esto se realiza con el fin que los estudiantes tengan un conocimiento unificado y profundo sobre los diferentes temas como lo proyecte la universidad.

Al estandarizar los procesos de evaluación en la facultad de sistemas, se abrirá la posibilidad para que este modelo sea implementado en las demás facultades, así poder generar mayor control, no solo en el nivel académico de los estudiantes, sino también en la calidad de evaluación de los docentes ya que las preguntas ingresadas podrán ser monitoreadas por cada jefe de programa.

Este tema puede presentar controversia ya que los docentes tienen diferentes metodologías de enseñanza, pero a la vez los ayudaría a que el aprendizaje de los estudiantes este ceñido al alcance que de la universidad en las diferentes facultades.

### **SEGUNDA FASE**

Con la segunda fase se concluye la administración de la generación de los exámenes, ya que uno de las dificultades que existían con la Fase I, se veía reflejada en que los exámenes solo se podían ver al momento de generarlos y tampoco existía ninguna opción donde visualizar las respuestas de estos; con la implementación se pueden consultar en cualquier momento obteniendo así una solución más completa del requerimiento inicial.

Los compromisos académicos de cada asignatura se diligenciaban manuales por cada docente, quedando abierta la posibilidad que se incluyeran temas que no corresponden a la asignatura o dejando por fuera alguno; con esta implementación se genera un control efectivo ya que un archivo PDF, el cual no se puede modificar el cual contiene las temáticas estipuladas para dicha asignatura

## 5. Conclusiones, Recomendaciones Y Trabajo Futuro

- **Conclusión General**

- Se crea una aplicación web tal como la solicita la jefatura de sistemas de la Universidad, la cual genera preguntas de manera aleatoria para una o varios temas de una asignatura, además cuenta con el módulo de reporte Intradía, donde se visualizan las fechas de compromiso adquiridas por los docentes al momento de dictar sus Temas de trabajo a los estudiantes.

Una de las limitaciones que tiene el proyecto es no poder visualizar la fecha real en la cual el docente dictó su tema de trabajo y que tiempo demora para hacerlo. También tiene como restricción que la generación de preguntas es solo para asignaturas de la Facultad de Sistemas, las demás facultades no podrán tener acceso esta herramienta web.

- **Conclusiones Específicas**

- Los docentes de la facultad de sistemas ya no tendrán la necesidad de realizar las evaluaciones de los estudiantes en forma manual como lo realizan en la actualidad.
- El jefe de programa tendrá la posibilidad de visualizar las fechas de compromiso de cada tema y realizar retroalimentación a los docentes que no cumplan con estas fechas fijadas por la Jefatura de Sistemas, también pueden llevar un control de las evaluaciones que se deben generar cada día, dependiendo de las fechas de cada tema.
- La Facultad de sistemas puede delegar el personal idóneo para administrar la aplicación, donde se asignen roles y responsabilidades a personas determinadas para que estas puedan ingresar los Usuarios, Asignaturas, Temas, Exámenes, etc.
- Nuestra recomendación es crear un módulo aparte donde se puedan calificar los exámenes que se generaron de manera aleatoria, dando como resultado que los docentes no tengan que calificar de forma manual los exámenes.
- Para un trabajo futuro se puede integrar la aplicación que hay actualmente del ITM ([www.itm.edu.co](http://www.itm.edu.co)) con este proyecto, de esta manera poder obtener una información más completa de los estudiantes de la Universidad.

## 6. Referencias

Aguilar, J(2008-2015) Generar un archivo PDF con contenido HTML. Recuperado de <http://www.jose-aguilar.com/blog/generar-archivo-pdf-con-contenido-html>

EllisLab, Inc. (2006-2012) Guía del Usuario de CodeIgniter Versión 2.1.4. Recuperado de [http://escodeigniter.com/guia\\_usuario/libraries/security.html](http://escodeigniter.com/guia_usuario/libraries/security.html)

Canós J, Letelier P y Penadés M,(junio 2006). Metodologías Ágiles en el Desarrollo de Software. Tomado de [http://noqualityinside.com/nqi/nqifiles/XP\\_Agil.pdf](http://noqualityinside.com/nqi/nqifiles/XP_Agil.pdf)

Asuni,N (2001-2013) TCPDF es una clase de software libre PHP para generar documentos PDF. Recuperado de <http://www.tcpdf.org/>

Uno de Piera (2014), TCPDDF en Codeigniter, Recuperado de <http://uno-de-piera.com/creacion-de-pdf-con-codeigniter-la-libreria-tcpdf/>

Grupo php,(Marzo de 2015) Documentación PHP, Recuperado de <http://php.net/docs.php>

Trujillo, J(29 de Enero de 2015 ) Presentación de trabajos de grado según Normas APA, Biblioteca de la Corporación Universitaria Lasallista. Recuperado de <https://prezi.com/sn0z0uefxep6/presentacion-de-trabajos-de-grado-segun-normas-apa-biblioteca-de-la-corporacion-universitaria-lasallista/>

FIRMA ESTUDIANTES \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

FIRMA ASESOR \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD \_\_\_\_\_

RECHAZADO \_\_\_\_\_

ACEPTADO \_\_\_\_\_

ACEPTADO CON  
MODIFICACIONES \_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_

FIRMA CONSEJO DE FACULTAD \_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_

Académico