

Universidad Tecnológica de El Salvador

FACULTAD DE INFORMÁTICA Y CIENCIAS APLICADAS

TÉCNICO EN INGENIERÍA DE SOFTWARE



TEMA:

**AUTOMATIZACIÓN DEL PROCEDIMIENTO DE CONTEO DE ESTUDIANTES
Y MONITOREO DE CLASES EN LA UNIVERSIDAD TECNOLÓGICA DE EL
SALVADOR.**

TRABAJO DE GRADUACIÓN PRESENTADO POR:

RICARDO ANTONIOBELTRÁN MENA

EVER MAURICIOESCOTO VÁSQUEZ

REBECA MICHELLEMENJIVAR MERINO

PARA OPTAR AL GRADO DE:

TÉCNICO EN INGENIERÍA DE SOFTWARE

SEPTIEMBRE, 2016

SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA

PÁGINA DE AUTORIDADES

ING. NELSON ZÁRATE SÁNCHEZ

RECTOR

LIC. JOSÉ MODESTO VENTURA ROMERO

VICERRECTOR ACADÉMICO

ING. FRANCISCO ARMANDO ZEPEDA

DECANO

JURADO EXAMINADOR

ING. ROLANDO DE JESÚS MEDRANO

PRESIDENTE

ING. NOÉ ELÍAS RODRÍGUEZ MERLOS

PRIMER VOCAL

ING. RUTH MARINA FIGUEROA DE FLORES

SEGUNDO VOCAL

SEPTIEMBRE, 2016

SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA

ACTA DE EXAMEN PROFESIONAL

HABIÉNDOSE REUNIDO EL JURADO CALIFICADOR INTEGRADO POR:

Ing. Noé Elías Rodríguez Merlos, Ing. Ruth Marina Figueroa de Flores, Ing. Rolando de Jesús Medrano, a las 12:30m. del día Jueves, 23 de Junio de dos mil dieciséis.

Y LUEGO DE HABER DELIBERADO SOBRE EL EXAMEN PROFESIONAL DE LOS ALUMNOS:

1-Rebeca Michelle Menjivar Merino	Carnet 27-1982-2014
2-Ever Mauricio Escoto Vásquez	Carnet 27-6941-2012
3-Ricardo Antonio Beltrán Mena	Carnet 27-0205-2014

QUIENES PRESENTARON DEFENSA DE SU TRABAJO DE GRADUACION TITULADO:

"Automatización del procedimiento de conteo de estudiantes y monitoreo de clases en la Universidad Tecnológica de El Salvador"

PARA OPTAR AL GRADO DE:

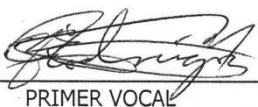
TÉCNICO EN INGENIERIA DE SOFTWARE

Y DEL CUAL TAMBIEN EVALUARON LOS CONOCIMIENTOS RELACIONADOS CON EL TEMA DEL MISMO. POR LO QUE ESTE JURADO RESUELVE DECLARAR EL EXAMEN COMO:

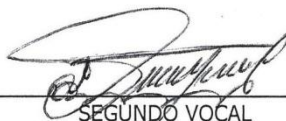
APROBADO

YA QUE CUMPLE CON LOS REQUISITOS ESTABLECIDOS EN EL REGLAMENTO DE GRADUACION DE LA UNIVERSIDAD.

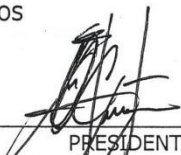
San Salvador, 23 de Junio de dos mil dieciséis.

F. 
PRIMER VOCAL

Ing. Noé Elías Rodríguez Merlos

F. 
SEGUNDO VOCAL

Ing. Ruth Marina Figueroa de Flores

F. 
PRESIDENTE
Ing. Rolando de Jesús Medrano

Índice

Introducción.....	i
-------------------	---

Capítulo I

Situación Actual

1.1 Antecedentes.....	1
1.2 Enunciado del Problema.....	3
1.3 Justificación.....	3
1.4 Objetivos.....	4
1.4.1 General.....	4
1.4.2 Específicos.....	5
1.5 Delimitaciones.....	5
1.5.1 Geográfica.....	5
1.5.2 Temporal.....	5
1.5.3 Organizacional.....	6
1.6 Alcances.....	6
1.7 Estudio de Factibilidad.....	6
1.7.1 Factibilidad Económica.....	6
1.7.2 Factibilidad Técnica.....	7

Capítulo II

Documentación Técnica

2.1 Marco Teórico de Referencia.....	9
2.1.1 Arquitectura de Software.....	9
2.1.2 Modelos de la Arquitectura de Software.....	10
2.1.3 Patrón de Arquitectura Modelo - Vista - Controlador (MVC).....	11
2.1.4 Framework.....	12
2.1.5 Framework CodeIgniter.....	13
2.1.6 Diseño Centrado en el Usuario.....	15

2.2 Marco Teórico de la Solución.	16
2.2.1 Instalación y Configuración del Servidor Web de pruebas.	16
2.2.2 Instalación y Configuración del Framework Codeigniter.	19
2.2.3 Especificación de Requisitos para la Aplicación Web.	21
2.2.4 Procesos de la Aplicación Web.	22
2.3 Marco Teórico Conceptual.	25
2.4 Documentación Técnica.	28
2.4.1 Documentación Técnica de XAMPP.	28
2.4.2 Documentación Técnica de MySQL Workbench.	29

Capítulo III

Desarrollo de la solución

3.1 Propuesta de la solución.	31
3.1.1 Análisis.	31
3.1.1.1 Actores.	32
3.1.1.2 Caso de uso del usuario registrado.	33
3.1.1.3 Caso de uso del colector.	34
3.1.1.4 Caso de uso del analista.	34
3.1.1.5 Caso de uso del usuario administrador.	35
3.1.1.6 Descripción textual de los casos de usos del sistema.	36
3.1.2 Diseño.	38
3.1.2.1 Capa de presentación.	39
3.1.2.2 Capa de negocio o lógica de aplicación.	40
3.1.2.3 Capa de persistencia o datos.	42
3.2 Conclusiones.	51
3.3 Recomendaciones.	52
Referencias.	53
Anexos.	56

Introducción

El presente documento describe la propuesta de un sistema informático para la Automatización del procedimiento de conteo de estudiantes y monitoreo de clases, ya que el sistema actual posee una serie de limitantes en cuanto a la modificación del código y al no poseer un módulo de monitoreo de clases.

En el Capítulo I Situación actual, se presenta los antecedentes que llevaron al desarrollo del sistema actual, la justificación para la realización de un nuevo sistema de conteo y monitoreo, los objetivos y alcances a lograr con este nuevo sistema. Al igual que se presenta un estudio de factibilidad económica y técnica para la viabilidad del sistema.

En el Capítulo II Documentación Técnica, se presenta el marco teórico de referencia el cual se detalla sobre el patrón de arquitectura Modelo-Vista-Controlador, y el framework de CodeIgniter.

En el Capítulo III Desarrollo de la Solución, se presenta la propuesta de solución del desarrollo del sistema, el análisis en el cual se define los actores involucrados en el sistema junto con la descripción textual de sus casos de uso, el diseño de lo que conformara el sistema de conteo de asistencia y monitoreo de clases, incluyendo el diseño de la base de datos, el respectivo diccionario de datos, que describe de forma detallada cada una de las tablas de la base de datos.

Capítulo I

Situación Actual

1.1 Antecedentes

A finales del ciclo 02-2012, el decano de la Facultad de Informática y Ciencias Aplicadas de la Universidad Tecnológica de El Salvador, propuso al Departamento de Apoyo Técnico (DAT) de la misma institución el desarrollo de un sistema para llevar un control de asistencias e inasistencias por cada facultad con la finalidad de tomar datos en tiempo real que ponen a disposición datos estadísticos de la asistencia y posible deserción de cada facultad y coordinaciones en general, permitiendo identificar las áreas con más ausencias de estudiantes en fechas específicas, y poder observar a través de una gráfica la tendencia de varianza que ocurre en determinados periodos de tiempo.

Se inició este proyecto con un encargado en la administración de la aplicación, supervisando el ingreso de datos y manteniendo la información de las aulas y secciones actualizadas, y en coordinación con el jefe del Departamento de Apoyo Técnico, se supervisa la integridad de los datos e información generada en el sistema.

Se realizó la implementación del sistema en su primera versión a principios del año 2013 sobre el servidor de CIOPS (Centro de Investigación de la Opinión Pública Salvadoreña), funcionando para los propósitos de la Facultad de Informática y Ciencias Aplicadas; pero después el proyecto fue al DAT, el cual lanzó una segunda versión en julio de 2014, lo que permitió prestar las mismas funciones a las demás facultades de la universidad, versión que se mantiene hasta la fecha.

Para la aplicación se trabajaba en una plataforma de Microsoft con SQL Server

versión 2008 para almacenar datos; pero a mediados del año 2013 se migró la base de datos a la versión SQL Server 2012 y el entorno de desarrollo ocupado fue .NET.

Actualmente, para su funcionamiento solo necesita un servidor con SQL Server y IIS configurado con el framework 4.0, el servidor actual cuenta con 4 GB de RAM, disco duro de 300GB y procesador de 4 núcleos. Utiliza las tecnologías ASP.NET MVC 4, la cual trabaja con el motor gráfico razor para el entorno visual, realiza consultas a través de JQuery y JQuery Mobile, la conexión a base de datos se realiza a través del modelo EntityFramework y consultas basadas en LinQ.

Considerando los procesos del sistema, uno de ellos son las personas, los cuales son: el Departamento de Apoyo Técnico coordinando con estudiantes de horas sociales se encarga de recolectar la información y son los responsables de su conteo, a nivel operativo existe un técnico responsable que monitorea el sistema e ingresa los datos a una base de datos. A nivel administrativo lo consultan directores, coordinadores de materias y decanos para consultar los resultados, y a nivel técnico se encuentra un responsable encargado del mantenimiento del sistema.

En cuanto a la recopilación de los datos, inicialmente se utilizaban listados impresos para conservar en físico dichos datos, pero la aplicación permite ser utilizada desde un navegador de cualquier dispositivo móvil a fin de registrarlos en tiempo real si fuere necesario. El sistema permite ingresar datos a través de un módulo el cual ofrece los filtros de edificio y turno basándose en la fecha y en los días de las secciones, permite exportar a Excel el detalle de datos ingresados de asistencia, existe un módulo que permite ver un gráfico lineal de un determinado rango de fechas el cual muestra el

porcentaje de inasistencia de estudiantes a clases por facultad y un cuadro porcentual en donde se reflejan las ausencias resumidas por materia.

En síntesis, la Universidad Tecnológica de El Salvador cuenta con cuatro facultades que encierran las carreras de estudio: Facultad de Informática y Ciencias Aplicadas, Facultad de Ciencias Empresariales, Facultad de Ciencias Sociales y Facultad de Ciencias Jurídicas. Lo que se pretende con este sistema es contabilizar la asistencia diaria de estudiantes y mantener un monitoreo de las clases por cada facultad.

1.2 Enunciado del Problema

¿Cuáles serán las mejoras que traerá consigo la Automatización del Procedimiento de Conteo de estudiantes y Monitoreo de clases en la Universidad Tecnológica de El Salvador?

1.3 Justificación

La Universidad Tecnológica de El Salvador comprometida con la innovación tecnológica, está en constante actualización de sus procesos académicos, que benefician a toda la comunidad educativa. Se cuentan con sistemas de información que cumplen con los propósitos generales para la gestión de sus estudiantes; pero a medida que van surgiendo necesidades de indicadores académicos para la toma de decisiones, se tiene que recurrir a desarrollar una aplicación. La universidad cuenta con el Departamento de Apoyo Técnico, conformado por una diversidad de profesionales especialistas en distintas materias, siendo esta la primera instancia para que satisfagan las necesidades o resuelvan los problemas que las autoridades han identificado con respecto a la gestión de la información.

Una de las necesidades que se ha dado a partir del año 2012 es contar con datos estadísticos de asistencias e inasistencias a clases y evaluaciones de cada facultad por cada ciclo escolar. Como parte de la respuesta a esta necesidad, se creó un sistema de información para tal propósito; pero existen una serie de mejoras que se le pueden hacer con respecto a la accesibilidad y consulta de información más precisas. En la aplicación actual, el vicerrector académico para la visualización de las estadísticas de inasistencia debe de ingresar a cada una de las facultades por separado ya que se debe tener un usuario y contraseña para cada facultad haciendo que este proceso de autenticación sea engorroso, de tal manera, que como solución se le desarrollara un usuario y una contraseña para que pueda visualizar la tendencia de ausencias del ciclo; para la consulta de datos se mantendrá un usuario y contraseña por cada facultad, además se desea incorporar un módulo para el monitoreo de clases.

El jefe encargado del Departamento de Apoyo Técnico de la Universidad Tecnológica de El Salvador para el proceso de graduación 01/2016 de la carrera Técnico en Ingeniería de Software, ha propuesto realizar un rediseño para el sistema de conteo de estudiantes, que tenga los módulos de asistencias e inasistencias a clases y evaluaciones; pero que además tenga el módulo de monitoreo de clases. Como se tomarán en cuenta las sugerencias y observaciones dadas para la accesibilidad de la aplicación y las consultas pertinentes para la toma de decisiones.

1.4 Objetivos

1.4.1 general

- Desarrollar una aplicación web para la automatización del procedimiento de

conteo de estudiantes y monitoreo de clases en cada ciclo de estudios en la Universidad Tecnológica de El Salvador.

1.4.2 específicos

- Manejar el patrón de arquitectura de software modelo-vista-controlador para potenciar la facilidad de mantenimiento y reutilización de código en el desarrollo de la aplicación web.
- Analizar los datos que se deben registrar en el sistema de información para crear de manera eficaz la estructura de la base de datos.
- Elaborar el manual de usuario de la aplicación para que sirva de referencia en algún proceso implicado en el conteo de estudiantes y monitoreo de clases.

1.5 Delimitaciones

1.5.1 geográfica

El proyecto se realizara en la Universidad Tecnológica de El Salvador, Edificio Gabriela Mistral, 3° Nivel.



Figura 1. Ubicación de la Universidad Tecnológica de El Salvador.

1.5.2 temporal

El proyecto de desarrollo se llevará a cabo en un lapso de tiempo comprendido entre los meses de febrero y junio del año 2016.

1.5.3 organizacional

El diseño de la aplicación web será desarrollado por un grupo de tres alumnos de la carrera Técnico en Ingeniería de Software, asesorados por la Ingeniero Ruth Figueroa de Flores, jefe del Departamento de Apoyo Técnico de la Universidad Tecnológica de El Salvador y con el respaldo del Ingeniero Carlos Monroy para cubrir aspectos técnicos.

1.6 Alcances

Tabla 1.
Alcances del sistema.

Promesa	Producto
1. Uso de CodeIgniter 3.0 como framework de la aplicación web bajo el patrón MVC.	1. Aplicación web desarrollada con el framework CodeIgniter MVC, gestor de bases de datos MySQL, y otros lenguajes de programación tales como JQUERY, HTML5 y CSS2.
2. Descripción de las fases para el desarrollo de la aplicación web.	2. Documento en formato PDF, el cual describe las fases de análisis, diseño, desarrollo e implementación de la aplicación web.
3. Documentar el código fuente de la aplicación web.	3. Líneas de comentarios en cada archivo para las funciones, variables y procesos que requieran una mayor atención en el desarrollo de la aplicación.

1.7 Estudio de Factibilidad

1.7.1 factibilidad económica

El desarrollo del Sistema de Conteo de estudiantes y Monitoreo de clases no representará un gasto económico para la UTEC, debido a que la mayoría de software que será utilizado para el desarrollo es software libre, mejor conocido como Open Source.

Tabla 2.
Costo del desarrollo del sistema.

Producto	Tipo de Licencia	Costo
Framework Codeigniter MVC 3.0	Licencia de Código Abierto (MIT)	\$0.00
MySQL Server 5.0.11	Licencia Pública General (GNU)	\$0.00
JQuery	Licencia de Código Abierto (MIT) Licencia Pública General (GNU)	\$0.00
Xampp 5.6.15	Licencia Pública General (GNU)	\$0.00

1.7.2 factibilidad técnica

De acuerdo a la tecnología necesaria para la implementación del sistema informático para el Conteo de estudiantes y Monitoreo de Asistencia a clase, se ha evaluado la factibilidad bajo dos enfoques: software y hardware, con la finalidad de que el sistema posea un buen rendimiento.

A continuación, se especifican los requerimientos mínimos de hardware y software:

Tabla 3.
Requerimientos mínimos de hardware y software para el desarrollo del sistema.

Atributo	Descripción
Procesador	AMD Sempron(tm) SI-42 ~ 2.10GHz
RAM	2.00 GB
Disco duro	130 GB Disco Duro
Tarjeta grafica	ATI Radeon HD 3200 Graphics
Tarjeta de red	Adaptador de red 802.11g Broadcom
Puertos externos	2 o más puertos USB 2.0.
Monitor	Monitor PnP genérico
Sistema operativo	Windows 7 Ultimate
IDE	Sublime Text 2.0
Motor de Base de Datos	MySQL 5.0.11
Framework	CodeIgniter MVC 3.0
Navegador web	Google Chrome
Servidor	Xampp 5.6.15
SGBD	MySQLWorkbench

Tabla 4.

Requerimientos mínimos de hardware y software para el usuario.

Atributo	Descripción
Procesador	Intel coreduo 1.73 GHz
RAM	4.00 GB o más.
Disco duro	130 GB Disco Duro
Tarjeta grafica	Tarjeta gráfica de 200 MHz de ratio
Tarjeta de sonido	SB 128 o compatible
Tarjeta de red	Ethernet RJ45 10/100 Mbps.
Puertos externos	2 o más puertos USB 2.0.
Monitor	Monitor con resolución 1024x768 o superior.
Sistema operativo	Windows Xp Profesional
Navegador web	Google Chrome

Tabla 5.

Requerimientos mínimos de hardware y software para el servidor.

Atributo	Descripción
Procesador	Intel coreduo 1.73 GHz
RAM	4.00 GB o más.
Disco duro	200 GB Disco Duro
Tarjeta grafica	Tarjeta gráfica de 200 MHz de ratio
Tarjeta de sonido	SB 128 o compatible.
Tarjeta de red	Ethernet RJ45 10/100 Mbps.
Puertos externos	2 o más puertos USB 2.0.
Monitor	Monitor con resolución 1024x768 o superior.
Sistema operativo	Windows Xp Profesional
IDE	Sublime Text 2.0
SGBD	MySQL 5.0.11
Framework	CodeIgniter MVC 3.0
Navegador Web	Google Chrome
Servidor	Xampp 5.6.15

En base a lo anterior, la Universidad Tecnológica de El Salvador cuenta con los requerimientos necesarios para implementar el sistema informático de Conteo de estudiantes y Monitoreo de asistencia a clase, por lo cual, el desarrollo del trabajo es factible.

Capítulo II

Documentación Técnica

2.1 Marco Teórico de Referencia

2.1.1 arquitectura de software

La Arquitectura de Software se define como un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación.

Podemos encontrar más definiciones acerca de lo que es la arquitectura de software; una de ellas se encuentra en el libro "An Introduction to Software Architecture", en donde sus autores definen que la arquitectura es un nivel de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura general del sistema se perfila como un nuevo tipo de problema. Los problemas estructurales que incluyen la organización y estructura de control global; protocolos de comunicación, sincronización y acceso a datos, asignación de funciones a elementos de diseño, distribución física, la composición de los elementos de diseño, escalado y el rendimiento; y la selección entre alternativas de diseño." (Garlan, & Shaw, 1994).

La definición oficial de Arquitectura de Software por el estándar ISO/IEEE 1471-2000, es "La arquitectura es la organización fundamental de un sistema incorporada en sus componentes, sus relaciones con el entorno y los principios que

conducen su diseño y evolución." (Arias, & Durango, 2016).

El objetivo principal de la Arquitectura del Software es aportar elementos que ayuden a la toma de decisiones y, al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participen en un proyecto. Para conseguirlo, la Arquitectura del Software construye abstracciones, materializándolas en forma de diagramas comentados.

2.1.2 modelos de la arquitectura de software

- Modelos Estructurales: Son similares a la vista estructural, pero su énfasis primario radica en la estructura coherente del sistema completo, en vez de concentrarse en su composición. Los modelos de framework a menudo se refieren a dominios o clases de problemas específicos. El trabajo que ejemplifica esta variante incluye arquitecturas de software específicas de dominios, como CORBA, o modelos basados en CORBA, o repositorios de componentes específicos, como PRISM.

- Modelos Dinámicos: Enfatizan la cualidad conductual de los sistemas. Con “dinámico” puede referirse a los cambios en la configuración del sistema, o a la dinámica involucrada en el progreso de la computación, tales como valores cambiantes de datos.

- Modelos de Proceso: Se concentran en la construcción de la arquitectura, y en los pasos o procesos involucrados en esa construcción. En esta perspectiva, la arquitectura es el resultado de seguir un argumento (script) de proceso. Esta vista se ejemplifica con el actual trabajo sobre programación de procesos para derivar arquitecturas.

2.1.3 patrón de arquitectura modelo -vista- controlador (mvc)

El Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. 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 realiza la descripción del dicho patrón:

El **Modelo** contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia. El modelo es responsable de:

- a) Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- b) Define las reglas de negocio (la funcionalidad del sistema).
- c) Lleva un registro de las vistas y controladores del sistema.
- d) Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

El **Controlador** actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno. El controlador es responsable de:

- a) Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc).
- b) Contiene reglas de gestión de eventos. Estas acciones pueden suponer peticiones al modelo o a las vistas.

La **Vista** compone la información que se envía al cliente y los mecanismos de interacción con éste. La vista es responsable de:

- a) Recibir datos del modelo y lo muestra al usuario.
- b) Tienen un registro de su controlador asociado.

2.1.4 framework

En el desarrollo de software, un framework se define como un entorno o ambiente de trabajo para desarrollo; estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

La arquitectura más utilizada en casi la mayoría de framework se basa en la conocida arquitectura Modelo-Vista-Controlador (MVC).

características de los framework

Casi todos los frameworks comparten las mismas características de acuerdo a su tipo, entre las que podemos destacar están:

- La **Autenticación** mediante login y password que permite restringir el acceso y el tipo de permiso.
- El **Acceso a los datos** en archivos txt, xml por ejemplo mediante interfaces que integran la base de datos.
- **Abstracción de URLs y Sesiones** ya que el framework se encarga de manejarlas.

- **Controladores** fácilmente adaptables a las necesidades del proyecto que gestionan las peticiones y/o eventos.

ventajas y desventajas del uso del framework

ventajas:

- Compatibilidad de lenguajes.
- Transparencia de proyectos de plataforma a plataforma.
- Portabilidad de arquitectura.
- Integración con múltiples dispositivos.
- Desarrollo de aplicaciones de manera más sencilla, ya que cuenta con los componentes necesarios incluidos.
- Reutilización de código.
- Maneja política de diseño uniforme y organizado.

desventajas:

- Curva inicial de aprendizaje: Cada framework tiene su ecosistema de componentes que el desarrollador debe aprender, no basta con conocer el lenguaje sobre el que está escrito. Por ello, los frameworks son islas de conocimiento.

2.1.5 framework codeigniter

Es un potente framework PHP que permite desarrollar aplicaciones web de manera eficaz. CodeIgniter utiliza el patrón Modelo-Vista-Controlador (MVC), un estilo de programación que divide la aplicación en tres capas:

- **Modelo:** conjunto de instrucciones para interactuar con la base de datos. Aquí se

programa todas las consultas requeridas para la manipulación de datos de la aplicación.

- Vista: elementos de la aplicación web con la que interactúa el usuario. Ventanas, cuadros de diálogo, botones de opciones, formularios, etc.
- Controlador: sirve de enlace entre el Modelo y la Vista. Es una serie de métodos que permiten controlar los datos del modelo, los procesa y se los pasa a las Vistas.

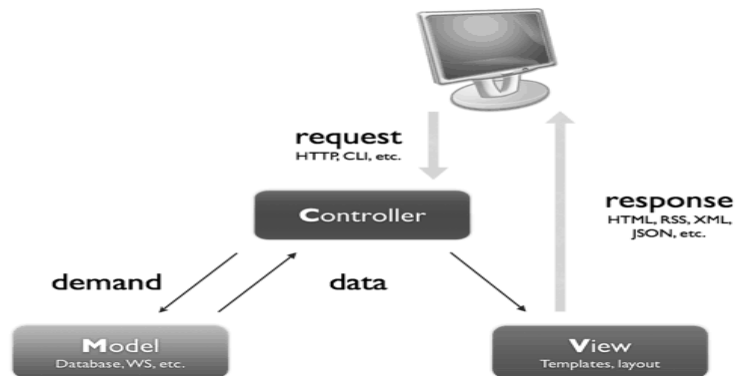


Figura 2. Patrón Modelo-Vista-Controlador.

características generales de codeigniter

- Versatilidad: trabaja en la mayoría de entornos o servidores web.
- Compatibilidad: trabaja desde la versión PHP4 en adelante, el cual es un requisito básico para que la aplicación web funcione en un servidor web.
- Facilidad de instalación: solamente se copia el directorio principal del framework CodeIgniter en el directorio público del servidor web.
- Flexibilidad: es un framework no tan riguroso en comparación de otros. Se puede seguir sus reglas; pero también se pueden saltar y codificar de acuerdo a la

necesidad que se tenga.

Lo que destaca a este framework es su accesibilidad para que el programador o el diseñador de la aplicación web se sientan a gusto y desarrollen un producto en el menor tiempo posible.

conocimientos previos para uso del framework codeigniter

- Conocer cómo funciona la programación orientada a objetos.
- Manejo del patrón de desarrollo MVC.
- Dominio básico de la sintaxis del lenguaje de programación PHP.
- Dominio del lenguaje de prototipado HTML y CSS.

2.1.6 diseño centrado en el usuario

La mayoría de las personas utilizan ordenadores como herramientas de trabajo, esto ayuda a mejorar las condiciones de trabajo. Por este motivo, el desarrollo de aplicaciones debe acompañarse de un conocimiento de la conducta humana y de la funcionalidad de dicha aplicación.

El proceso de interacción se da cuando existe transferencia de información entre el ser humano y la computadora. La interfaz es el recurso mediante el cual se lleva a cabo la interacción. Asimismo la interfaz incluye dispositivos de entrada, es decir, aquellos que permiten introducir datos y los dispositivos de manipulación, los cuales se utilizan para alterar el estado de la información y navegar por la aplicación.

fases del proceso de diseño centrado en el usuario

- Análisis: se reúne información sobre los objetivos de la aplicación, las

características de los usuarios y los requisitos técnicos del desarrollo.

- **Diseño:** se basa en el modelado de usuario, diseño conceptual, diseño de contenidos y diseño visual. El diseño de la aplicación debe de responder a las características definidas en la fase de análisis.
- **Prototipo:** es el esquema o esqueleto de la interfaz de usuario. Aquí se establecen la posición de todos los elementos que forman parte de la aplicación. Este prototipo puede ser diseñado en papel y lápiz o en computadora.
- **Evaluación:** la evaluación de la usabilidad es la fase más importante del proceso de diseño centrado en el usuario. Se puede dar con usuarios reales o sin usuarios reales.

2.2 Marco Teórico de la Solución

2.2.1 instalación y configuración del servidor web de pruebas

Para el desarrollo de la aplicación web, se necesita tener previamente instalado y configurado un servidor web que soporte el lenguaje PHP y un gestor de base de datos MySQL. Dicho servidor web va a permitir evaluar el funcionamiento de la aplicación.

Se hará uso de una distribución completamente gratuita y fácil de instalar, conocida como XAMPP. A continuación, se explica el proceso para instalar y configurar XAMPP para el sistema operativo Windows.

instalación de xampp en windows

1. Descargar el paquete de instalación desde la siguiente dirección:
<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/5.6.19/>
2. Haga clic en el vínculo xampp-portable-win32-5.6.19-0-VC11.zip

3. Extraer el contenido del archivo en la Unidad C: del sistema operativo. Dejando como directorio principal xampp.
4. Dentro de la carpeta xampp, haga clic derecho sobre el archivo xampp_control.exe > Ejecutar como administrador.
5. El idioma por default es en inglés, haga clic en Save.

configuraciones básicas de xampp

Antes de iniciar el servicio de Apache se debe de configurar el puerto principal de Apache para evitar conflictos con otras aplicaciones dentro de Windows.

1. En el panel de control de XAMPP, haga clic en el botón Config.
2. Haga clic en el botón Service and Port Settings.
3. En el Main Port, cambie el puerto 80 por 89 y en SSL Port cambie el 443 por 4433.
4. Haga clic en Save.



Figura 3. Configuración de puertos del servidor web Apache.

También se debe de especificar estos puertos en el archivo de configuración de apache httpd.conf

1. En el panel de control de XAMPP, haga clic en el botón Config del servicio de Apache.
2. Haga clic en Apache (httpd.conf). El archivo se abrirá en un editor de textos.
3. Buscar la línea 58 del archivo de texto, y reemplazar el número 80 por 89. Listen 89
4. Buscar la línea 220 y reemplazar el número 80 por 89. ServerName localhost:89
5. Guarde los cambios y cierre el archivo.
6. Ahora, repita el paso uno; pero se abrirá el archivo httpd-ssl.conf
7. En la línea 36 de este archivo, cambie el número 443 por 4433. Listen 4433
8. Guarde los cambios y cierre el archivo.

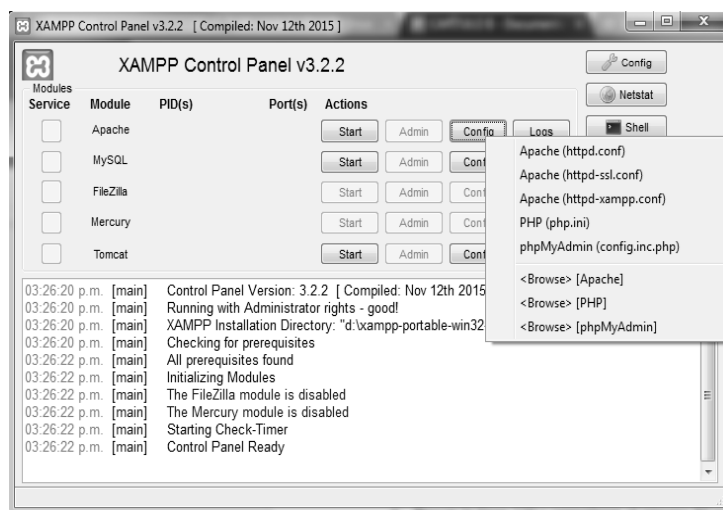


Figura 4. Panel de Control de XAMPP.

iniciar servicios de apache y mysql

Ya con los nuevos puertos, con toda seguridad puede iniciar estos servicios que son necesarios para el desarrollo de la aplicación. Simplemente haga clic en el botón Start de cada servicio.

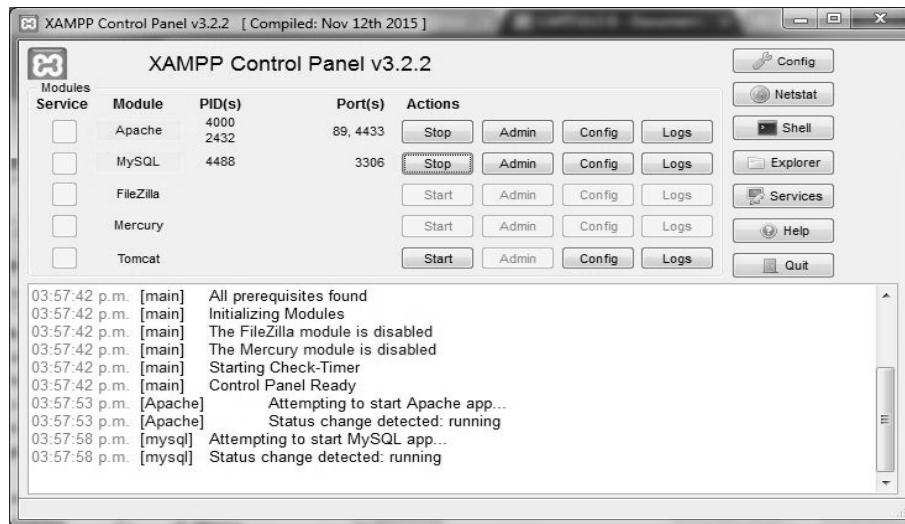


Figura 5. Servicios de Apache y MySQL iniciados y funcionando.

prueba de ejecución del servidor apache

Abra el navegador web, en la barra de dirección escriba: `http://localhost:89` y presione la tecla Enter. El navegador le mostrará la página que está alojada por defecto en el servidor web.

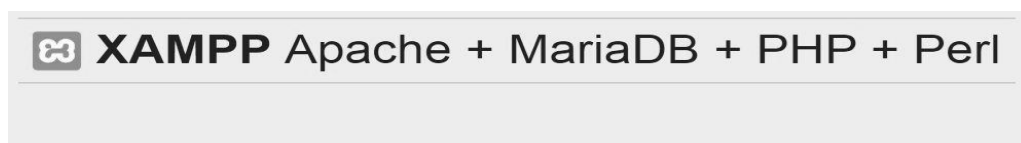


Figura 6. Página por defecto de XAMPP.

2.2.2 instalación y configuración del framework codeigniter

El framework contiene un conjunto de librerías que son de mucha utilidad para el desarrollo de la aplicación. A continuación se explica el proceso de instalación de este framework.

1. Descargue el paquete desde <https://www.codeigniter.com/>. Haga clic en el enlace Download.
2. Extraer el paquete en el directorio X:\xampp\htdocs\ el cual es el que permite ejecutar la aplicación web en el navegador.
3. Cambie el nombre del directorio CodeIgniter-3.0.6 por el que se ha considerado como requisito previo.
4. En el navegador escriba <http://localhost:89/directorio/>

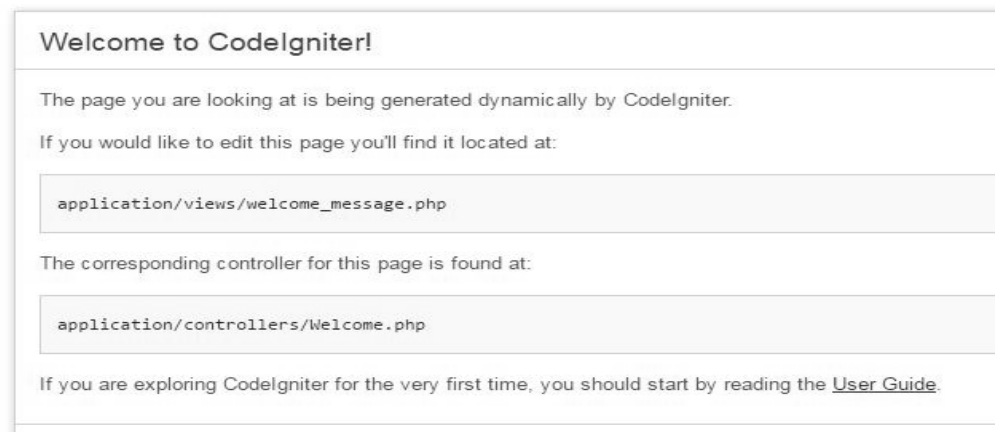


Figura 7. Página de bienvenida al instalar satisfactoriamente CodeIgniter.

configuración de la aplicación en codeigniter

Ingresa a la siguiente ruta de directorio X:\xampp\htdocs\fica\application\config\.

En dicha ruta encontrará una serie de archivos, de los cuales, únicamente se van a modificar los que se detallan a continuación:

- **autoload:** En este archivo se va a activar las librerías de base de datos y el manejo de sesiones, las cuales se van a utilizar en todo el desarrollo de la aplicación, así como también las utilidades de url y form, empleadas en la maquetación de la interfaz de usuario.

- **config:** Se harán cambio a dos parámetros de este archivo para crear urls accesibles o amigables para el usuario que va a manipular la aplicación web.
- **database:** Se asignan valores a un arreglo para establecer la conexión a la base de datos de la aplicación web, así como el manejo de errores y excepciones.
- **routes:** Se define la página que por defecto cargará al ingresar a la aplicación desde el navegador. Se pretende que sea la página de Acceso o Autenticación de usuario la que se tiene que establecer en el parámetro `$route['default_controller']`.

2.2.3 especificación de requisitos para la aplicación web

funciones generales para todos los usuarios:

- Autenticación: visualiza un formulario de acceso a la aplicación, donde debe introducir nombre de usuario y contraseña.
- Cerrar sesión: se le habilita una opción para que el usuario cierre sesión. Cuando este haya terminado de usar la aplicación, ocupa esta opción para redirigirlo a la página de Inicio.

funciones para el usuario colector:

- Ingreso de asistencia: utiliza un formulario donde debe especificar la facultad, ciclo, fecha, turno, edificio para filtrar las materias para registrar el conteo de alumnos que asisten a clases.

funciones para el usuario analista de datos:

- Módulo de inasistencia: visualización de la tendencia de ausencia de clases del ciclo vigente.

- Módulo de evaluaciones: visualización de la deserción durante los exámenes.

funciones para el usuario administrador:

- Mantenimiento de entidades: altas, bajas o modificaciones de edificios, docentes, materias, facultades, escuelas, secciones, usuarios, roles, etc.

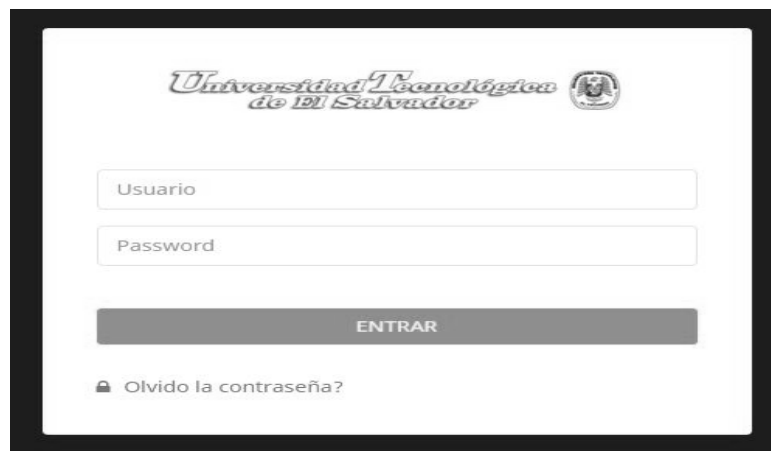
2.2.4 procesos de la aplicación web

proceso: ingresar a la aplicación web

Permite ingresar a la aplicación web “ACOMUTEC” el cual es autenticado y autorizado para la utilización de este.

flujo del proceso:

- El usuario digita en la barra de navegación del navegador <http://conmoutec/>
- La aplicación presenta el formulario de acceso.
- Escribe el nombre de usuario y contraseña asignada.
- La aplicación autentica al usuario mediante una variable de sesión.
- La aplicación autoriza al usuario proporcionándole su perfil de entrada.
- La aplicación le redirecciona a la página principal de su perfil.



The image shows a login interface for the Universidad Tecnológica de El Salvador. At the top, the university's name and logo are displayed. Below this, there are two input fields: one labeled 'Usuario' and another labeled 'Password'. Underneath the password field is a large, dark gray button with the word 'ENTRAR' in white capital letters. At the bottom of the form, there is a small icon of a key and the text 'Olvido la contraseña?'.

*Figura 8.*Formulario de acceso a la aplicación.

proceso: ingreso de asistencia

Permite al usuario con perfil de colector agregar la asistencia de clases de acuerdo a los datos recopilados previamente.

flujo del proceso:

- a. El usuario realiza el proceso de iniciar sesión con sus datos de acceso.
- b. En el menú de navegación pulsa la opción Ingresar asistencia.
- c. El usuario debe seleccionar la facultad, ciclo, fecha, turno y edificio para que filtre todas las materias para ingresar la asistencia de cada una.
- d. Digita la cantidad de asistencia de alumnos(as) en el campo respectivo de cada materia.
- e. Presiona el botón guardar.
- f. La aplicación muestra una notificación si la operación ha sido realizada satisfactoriamente.

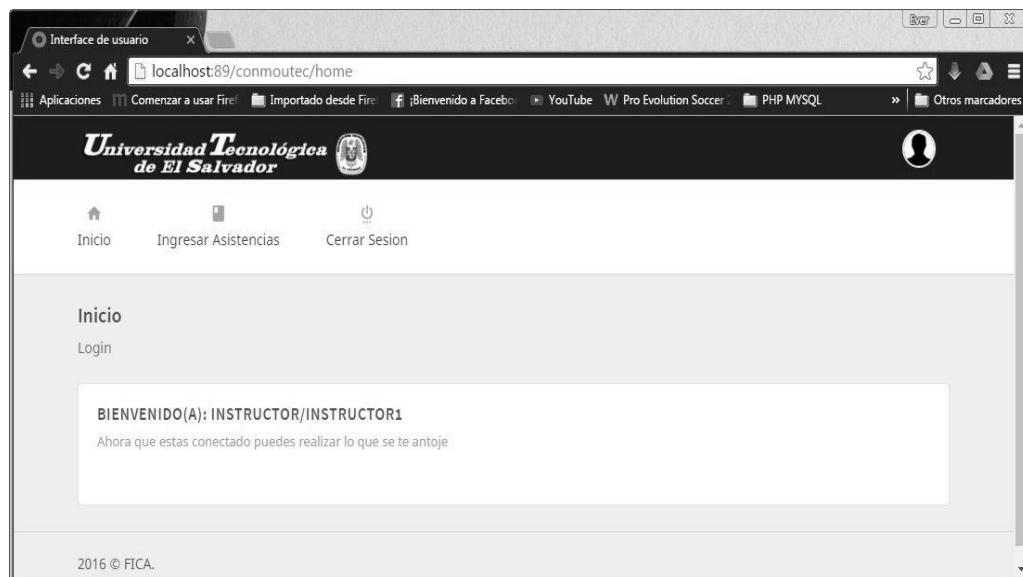


Figura 9. Perfil del usuario colector para ingresar asistencias.

The screenshot shows a web browser window with the address bar displaying 'localhost:89/conmoutec/home/asistencias'. The page header features the logo of 'Universidad Tecnológica de El Salvador' and a user profile icon. The main content area is titled 'Ingresar Asistencia' and 'Perfil Colector'. It contains a form with the following fields: 'Fecha' (text input with placeholder 'dd/mm/aaaa'), 'Facultad' (dropdown menu with 'Elegir facultad'), 'Ciclo' (dropdown menu with 'Elegir ciclo'), 'Turno' (dropdown menu with 'Elegir turno'), and 'Edificio' (dropdown menu with 'Elegir edificio'). A 'Filtrar Listado' button is located at the bottom of the form.

Figura 10. Formulario para filtrar materias para ingresar cantidad de asistencias.

proceso: visualización del módulo de inasistencias

El usuario con perfil de analista, es aquel que podrá tener acceso para visualizar la tendencia de ausencias de clases de su facultad.

flujo del proceso:

- a. El usuario realiza el proceso de iniciar sesión con sus datos de acceso.
- b. En el menú de navegación decide elegir Asistencias de clases o Asistencias de evaluaciones.
- c. Seleccionar la fecha de inicio y fecha de finalización para filtrar la información.
- d. Presiona el botón Mostrar.
- e. La aplicación web muestra un gráfico y una tabulación de datos con el promedio de Asistencias, inasistencias de cada escuela o departamento de la facultad.

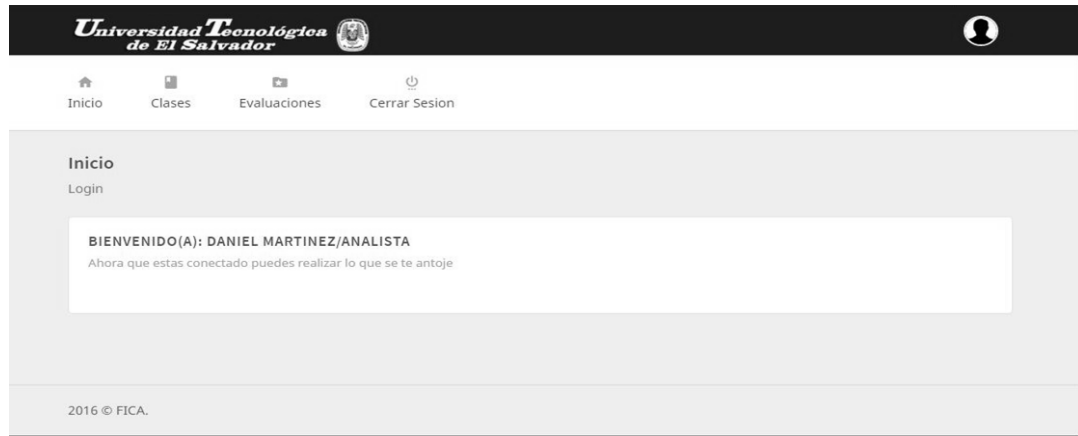


Figura 11. Perfil de usuario analista para visualizar asistencia de clases y evaluaciones.

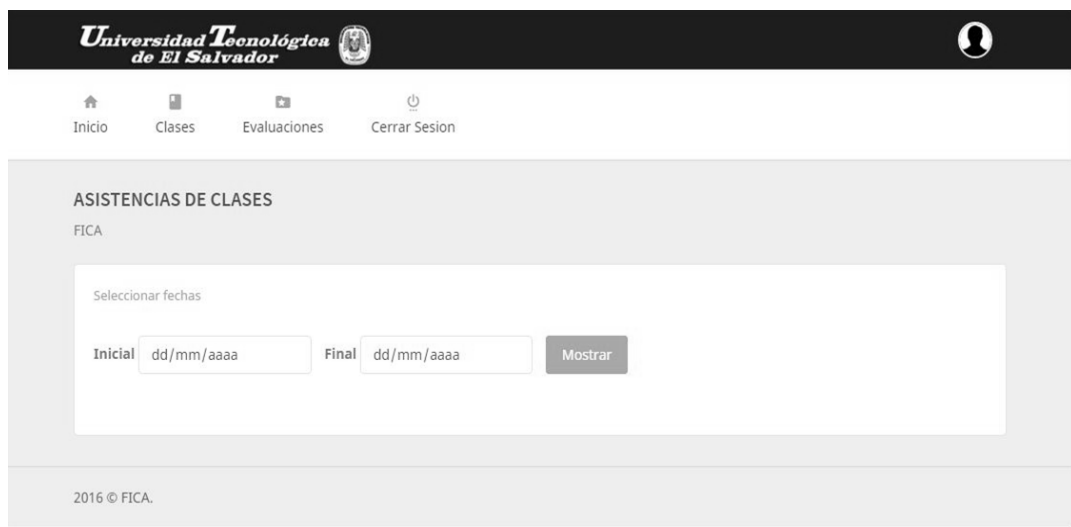


Figura 12. Formulario para filtrar asistencias de clases por rango de fechas.

2.3 Marco Teórico Conceptual

- **Accesibilidad:** Posibilidad de que un producto o servicio web pueda ser accedido y usado por el mayor número posible de personas, independiente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios.
- **Arquitectura Informática:** Es el diseño conceptual y la estructura operacional

fundamental de un sistema de computadora. Es decir, es un modelo y una descripción funcional de los requerimientos y las implementaciones de diseño para varias partes de una computadora, con especial interés en la forma en que la Unidad Central de Proceso (UCP) trabaja internamente y accede a las direcciones de memoria.

- **CORBA (Common Object RequestBroker Architecture):** Estándar definido por Object Management Group (OMG) que permite que diversos componentes de software escritos en múltiples lenguajes de programación y que corren en diferentes computadoras, puedan trabajar juntos; es decir, facilita el desarrollo de aplicaciones distribuidas en entornos heterogéneos.
- **Framework:** Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.
- **Interfaz:** Parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario.
- **Intranet:** Parte privada de la aplicación donde sólo tendrán acceso a la información los usuarios que estén registrados.
- **Navegador Web:** Permite al usuario recuperar y visualizar páginas web a través de Internet.
- **Patrón de Diseño:** Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

- **Procesos:** Unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados.
- **Servidor Web:** Se trata de un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.
- **Sitio Web:** Conjunto de archivos electrónicos y páginas web referentes a un tema en particular que incluye una página inicial de bienvenida, con un nombre de dominio y dirección en internet específicos.
- **Software:** Conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.
- **Usabilidad:** Calidad de la página web o del programa informático que son sencillos de usar porque facilitan la lectura de los textos, descargan rápidamente la información y presentan funciones y menús sencillos, por lo que el usuario encuentra satisfechas sus consultas y cómodo su uso.
- **Usuario:** Individuo que utiliza una computadora, sistema operativo, servicio o cualquier sistema, además se utiliza para clasificar a diferentes privilegios, permisos a los que tiene acceso un usuario o grupo de usuario, para interactuar o ejecutar con el ordenador o con los programas instalados en este.

2.4 Documentación Técnica

2.4.1 documentación técnica de xampp

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

características de xampp

- Para Windows existen dos versiones, una con instalador y otra portable (comprimida) para descomprimir y ejecutar.
- Otra característica no menos importante, es que la licencia de esta aplicación es GNU (General Public License), está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.
- Una de las características sobresalientes de este sistema es que es multiplataforma, es decir, existen versiones para diferentes sistemas operativos, tales como: Microsoft Windows, GNU/Linux, Solaris, y MacOS X. Existen versiones para Linux (testado para SuSE, RedHat, Mandrake y Debian), Windows (Windows 98, NT, 2000, XP y Vista), MacOS X y Solaris (desarrollada y probada con Solaris 8, probada con Solaris 9).
- XAMPP solamente requiere descargar y ejecutar un archivo zip, tar, exe o fkl,

con unas pequeñas configuraciones en alguno de sus componentes que el servidor web necesitará.

- XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin.

2.4.2 documentación técnica de mysql workbench

MySQL Workbench es una herramienta que permite modelar diagramas de entidad-relación para bases de datos MySQL. Puede ser utilizada para diseñar el esquema de una base de datos nueva, documentar una ya existente o realizar una migración compleja.

La aplicación elabora una representación visual de las tablas, vistas, procedimientos almacenados y claves extranjeras de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real.

características de mysql workbench

Algunas de las características más interesantes de MySQLWorkbench son:

- Edición de de diagramas basada en Cairo, con posibilidad de realizar una salida en los formatos como OpenGL, Win32, X11, Quartz, PostScript, PDF.
- Proporciona una representación visual de las tablas, vistas, procedimientos, funciones almacenadas y claves foráneas.
- Permite acceso a bases de datos e ingeniería inversa de las mismas para crear los SQL de creación.
- Ofrece sincronización con la base de datos y el modelo.

- Permite generar los scripts SQL a partir del modelo creado.
- Ofrece una arquitectura extensible.
- Tiene soporte para exportar los datos como script SQL CREATE.
- Permite importar modelos de DBDesigner4.
- Ofrece soporte completo a las características de MySQL 5.

Capítulo III

Desarrollo de la Solución

3.1 Propuesta de la Solución

A continuación se describen las distintas etapas que conforman el proceso de desarrollo de la aplicación web.

3.1.1 análisis

Para realizar el análisis de la aplicación web se ha optado por seguir las recomendaciones definidas por UML (Unified Modeling Language o Lenguaje Unificado de Modelado en castellano).

Este modelo dispone de multitud de diagramas que ayudan a comprender la complejidad del futuro sistema, permitiéndonos plasmar en un lenguaje estándar aquellas funcionalidades, requisitos y demás características que se han detectado en el sistema.

Aunque UML define una gran cantidad de diagramas para representar los distintos aspectos del desarrollo de la aplicación, para este caso se ha utilizado el diagrama de casos de usos.

Los diagramas de casos de uso permiten diferenciar los actores que interactúan con la aplicación, las relaciones entre ellos y las acciones que puede realizar cada uno dentro del sistema.

Este tipo de diagramas son fácilmente comprensibles tanto por clientes como por usuarios, representan los requisitos funcionales del sistema y se utilizan como base para un desarrollo iterativo e incremental.

Los diagramas de casos de uso tienen tres elementos:

- **Actores:** Son los usuarios del sistema. Un actor puede ser una persona, un conjunto de personas, un sistema hardware o un sistema software. Los actores representan un rol, que puede desempeñar alguien que necesita intercambiar información con el sistema.
- **Casos de uso:** Un caso de uso describe una forma concreta de utilizar parte de la funcionalidad de un sistema. La colección de todos los casos de uso describe toda la funcionalidad del sistema.
- **Comunicación entre actores y casos de uso:** Cada actor ejecuta un número específico de casos de uso en la aplicación. Por eso se dice que hay comunicación entre actores y casos de uso.

3.1.1.1 actores

Tabla 6.
Descripción del actor Usuario Registrado.

Nombre:	Usuario Registrado
Descripción: Representa a un usuario que se encuentra almacenado en la base de datos, los cuales pueden estar autenticados como colectores, analistas o administrador del sistema.	

Tabla 7.
Descripción del actor Colector.

Nombre:	Colector
Descripción: Representa a un usuario encargado de ingresar el conteo de asistencia a clases y el monitoreo de la clase.	

Tabla 8.

Descripción del actor Analista.

Nombre:	Analista
Descripción:	Representa a un usuario que consulta datos estadísticos de asistencias, inasistencias, promedios y las respuestas obtenidas en los indicadores del monitoreo de clases.

Tabla 9.

Descripción del actor Administrador.

Nombre:	Administrador
Descripción:	Representa a un usuario que tiene el control total de la aplicación para realizar los mantenimientos de usuarios, docentes, materias, secciones, edificios, facultades, ciclos, preguntas.

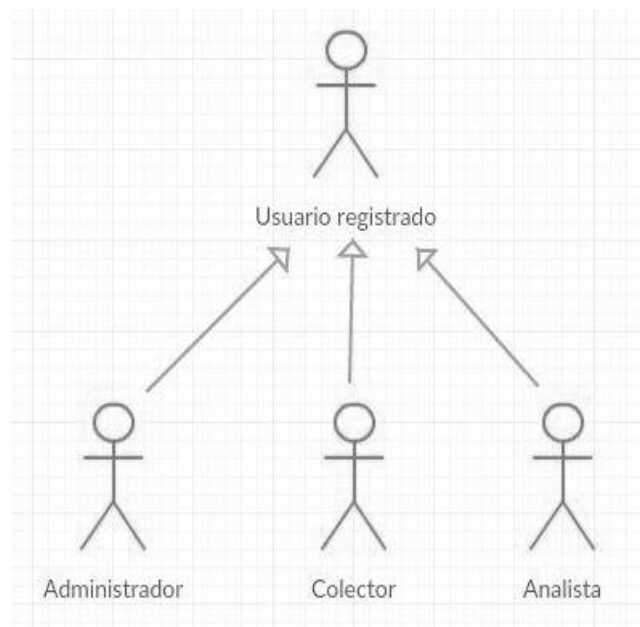


Figura 13. Actores dentro del sistema.

3.1.1.2 caso de uso del usuario registrado

Este tipo de usuario sólo puede realizar la acción de cerrar su sesión porque es la única funcionalidad que tienen en común los usuarios involucrados en la aplicación web.



Figura 14. Diagrama de caso de uso del usuario registrado.

3.1.1.3 caso de uso del usuario colector

Este tipo de usuario representa a un alumno de servicio social, el cual podrá realizar las acciones de ingresar los datos recolectados de la cantidad de asistencia a clases y del monitoreo de clases.

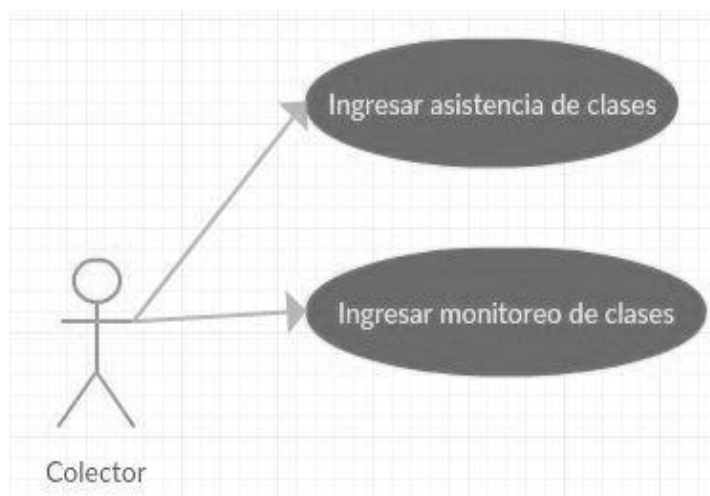


Figura 15. Diagrama de caso de uso del usuario colector.

3.1.1.4 caso de uso del usuario analista

Este tipo de usuario representa un decano de la facultad, el cual podrá realizar las acciones de consultar datos estadísticos de inasistencia de clase, también datos

estadísticos de inasistencias de clases en evaluaciones parciales y los resultados obtenidos del monitoreo de clases.

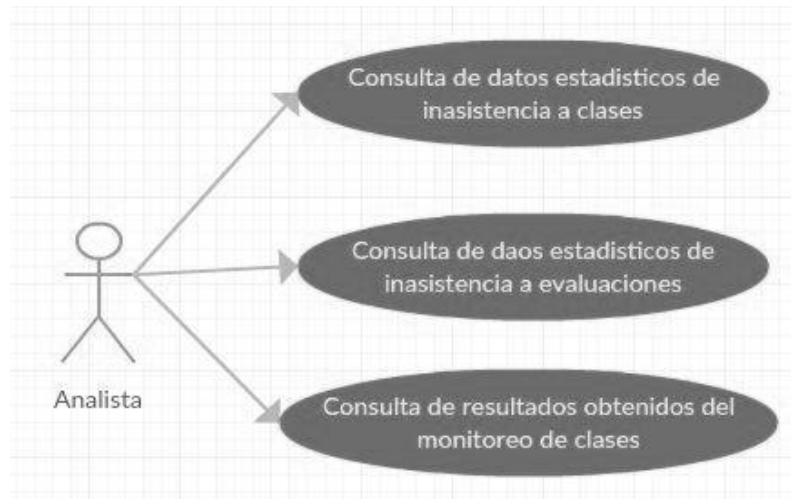


Figura 16. Diagrama de caso de uso del usuario analista.

3.1.1.5 caso de uso del usuario administrador

Este tipo de usuario representa al administrador de la aplicación. Podrá realizar las acciones de dar el alta, modificaciones y bajas de usuarios, docentes, materias, secciones, facultades, edificios, escuelas, ciclos, preguntas para el monitoreo de clases.



Figura 17. Diagrama de caso de uso del usuario administrador del sistema.

3.1.1.6 descripción textual de los casos de usos del sistema

Tabla 10.

Descripción del caso de uso para ingresar al sistema.

Nombre:	Ingresar al sistema
Descripción:	Permite el acceso a la aplicación web por medio de un usuario y contraseña asignado.
Actores:	Colector / Analista / Administrador
Precondiciones:	El usuario debe estar almacenado en la base de datos y habilitado por el administrador del sistema.
Flujo normal:	1. El actor escribe el usuario y contraseña en el formulario de acceso. 2. El actor pulsa sobre el botón para ingresar. 3. El sistema comprueba la validez de los datos. 4. El sistema verifica el tipo de usuario para mostrar las acciones de acuerdo a su rol.
Flujo alternativo:	3A. El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija.
Poscondiciones:	El actor ha iniciado sesión en el sistema.

Tabla 11.

Descripción del caso de uso para ingresar asistencia de clases.

Nombre:	Ingresar asistencia de clases
Descripción:	Permite ingresar la cantidad de asistencia a clases de una materia y sección.
Actores:	Colector / Administrador
Precondiciones:	El usuario debe estar autenticado en el sistema.
Flujo normal:	1. El actor pulsa sobre el vínculo Ingresar asistencia. 2. El sistema le muestra un formulario para que seleccione la facultad, el ciclo, la fecha, el turno, el edificio y el botón para filtrar las materias y secciones. 3. El actor selecciona todas las opciones del formulario. 4. El sistema le muestra las materias, secciones y un cuadro de textos para introducir la cantidad de asistencias de cada una.

5. El sistema verifica que no existan datos ingresados de acuerdo al filtro. 6. El actor ingresa la cantidad de asistencias en cada cuadro de texto de las secciones. 7. El actor pulsa sobre el botón guardar. 8. El sistema muestra un mensaje satisfactorio al guardar los datos.
Flujo alternativo: 3A. El actor selecciona todas las opciones del formulario, si no lo hace, se avisa al actor de ello permitiéndole que los corrija.
Poscondiciones: El sistema debe limpiar el formulario al guardar los datos de asistencias.

Tabla 12.

Descripción del caso de uso para ingresar monitoreo de clases.

Nombre:	Ingresar monitoreo de clases
Descripción:	Permite ingresar el monitoreo de clases de una materia y sección.
Actores:	Colector / Administrador
Precondiciones:	El usuario debe estar autenticado en el sistema.
Flujo normal:	1. El actor pulsa sobre el vínculo Ingresar Monitoreo. 2. El sistema le muestra un formulario para que seleccione la facultad, el ciclo, la fecha, el docente, la materia, la sección y el botón para filtrar las preguntas y las opciones de respuestas. 3. El actor selecciona todas las opciones del formulario. 4. El sistema verifica que no existan datos ingresados de acuerdo al filtro. 5. El sistema le muestra las preguntas y opciones de respuesta para el monitoreo de clases. 6. El actor selecciona la respuesta obtenida de cada pregunta. 7. El actor pulsa sobre el botón guardar. 8. El sistema muestra un mensaje satisfactorio al guardar los datos.
Flujo alternativo:	3A. El actor selecciona todas las opciones del formulario, si no lo hace, se avisa al actor de ello permitiéndole que los corrija.
Poscondiciones:	El sistema debe limpiar el formulario al guardar los datos del monitoreo de clases.

Tabla 13.

Descripción del caso de uso para consultar inasistencia de clases.

Nombre:	Consultar inasistencia de clases
Descripción:	Permite consultar inasistencia de clases de acuerdo a la facultad que pertenece el usuario autenticado.
Actores:	Analista
Precondiciones:	El usuario debe estar autenticado en el sistema.
Flujo normal:	1. El actor pulsa sobre el vínculo Clases. 2. El sistema le muestra un formulario para que seleccione la fecha de inicio, fecha de finalización y el botón para filtrar las estadísticas de asistencia, inasistencia y promedio de clases. 3. El actor selecciona la fecha de inicio, fecha de finalización y pulsa sobre el botón filtrar. 4. El sistema comprueba la existencia de datos de acuerdo al filtro. 5. El sistema muestra un mensaje satisfactorio de la consulta de los datos.
Flujo alternativo:	3A. El actor selecciona todas las opciones del formulario, si no lo hace, se avisa al actor de ello permitiéndole que los corrija.
Poscondiciones:	El sistema debe mostrar una opción para detallar por materia la inasistencia.

3.1.2 diseño

El diseño es el proceso que extiende, refina y reorganiza los aspectos detectados en el proceso de modelado conceptual para generar una especificación rigurosa del sistema de información siempre orientada a la obtención de la solución del sistema software.

Para el diseño de la aplicación web se ha escogido una arquitectura de tres capas (presentación, aplicación y persistencia). La utilización de esta arquitectura se debe a que los distintos niveles son independientes unos de otros de manera que, por ejemplo,

se puede cambiar fácilmente el comportamiento de las clases en el nivel de aplicación sin que ello influya en las otras capas.

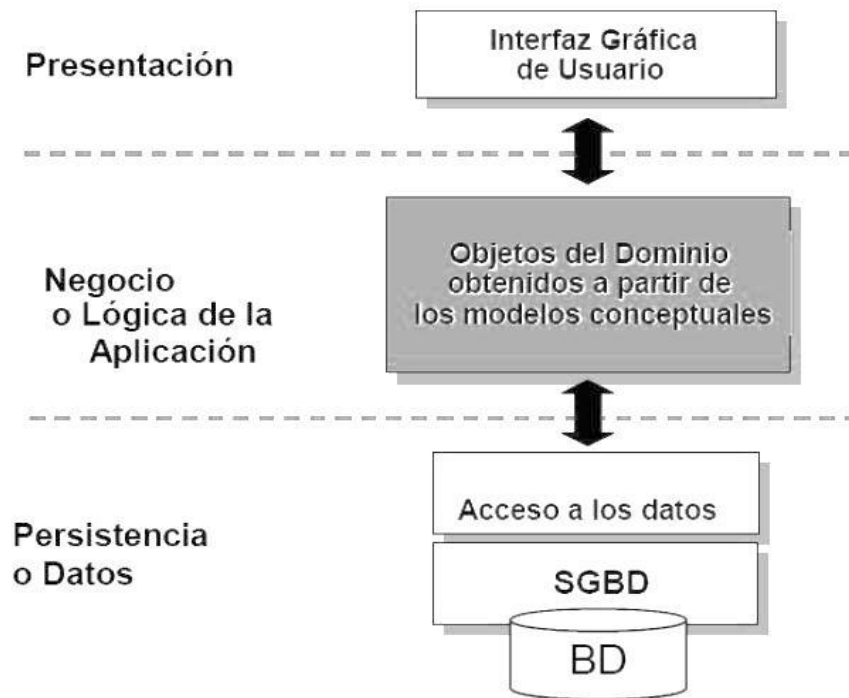


Figura 18. Esquema de la arquitectura de tres capas.

3.1.2.1 capa de presentación

La capa de presentación se puede definir como el conjunto de componentes de software que implementan la interacción con los usuarios a través de una representación visual de la aplicación, proporcionando a los usuarios una forma de acceder y controlar los datos y los servicios de los objetos. A partir de la interfaz gráfica, el usuario podrá navegar por las distintas secciones para poder obtener toda la información que desee, o aportarla en caso de ser necesario.

A continuación se muestra la representación visual para la aplicación web cuando el usuario se ha autenticado:

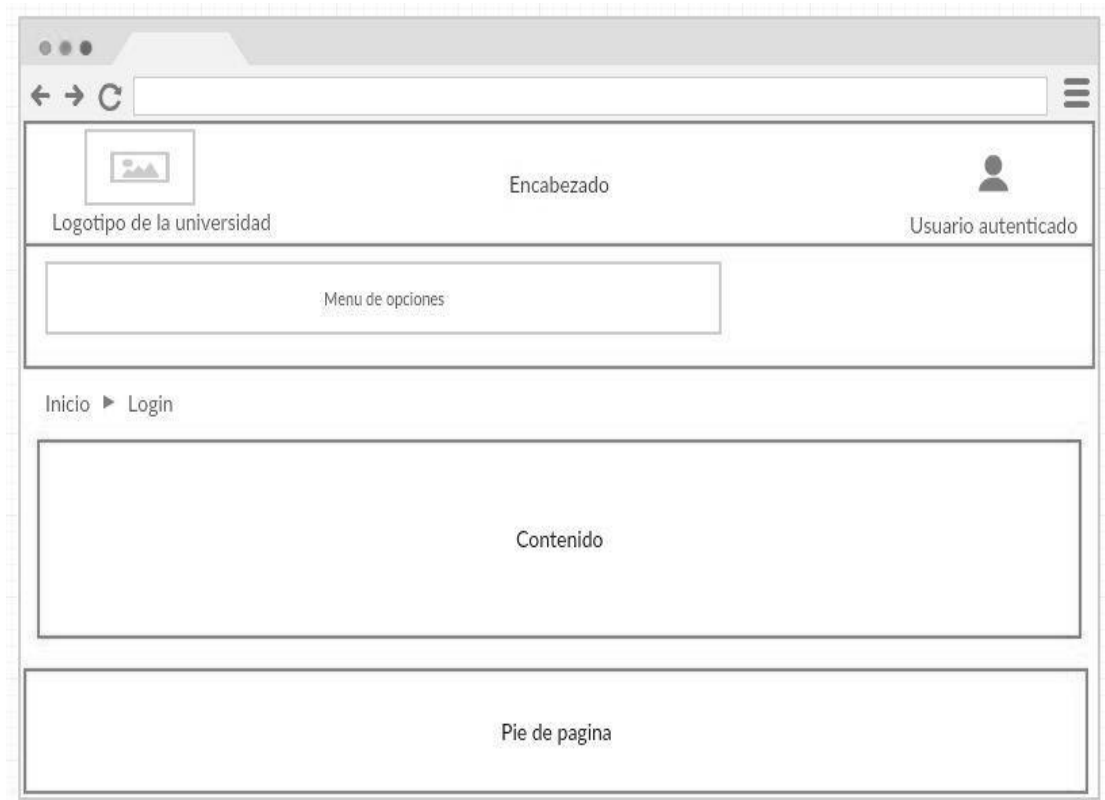


Figura 19. Prototipo de la interfase gráfica de la aplicación Web.

3.1.2.2 capa de negocio o lógica de aplicación

La capa de negocio se puede definir como el conjunto de componentes software que implementan completamente el comportamiento de las clases del dominio, especificadas en la fase de modelado conceptual. Es en este nivel, por tanto, donde se implementa la funcionalidad de la aplicación web.

Esta capa sirve de enlace entre los niveles de presentación y de persistencia, ya que la capa de presentación no accede a la base de datos directamente, sino que se comunica con la capa de aplicación para demandarle el servicio deseado y es la capa de aplicación la que se comunica con la capa de persistencia para recuperar los datos necesarios.

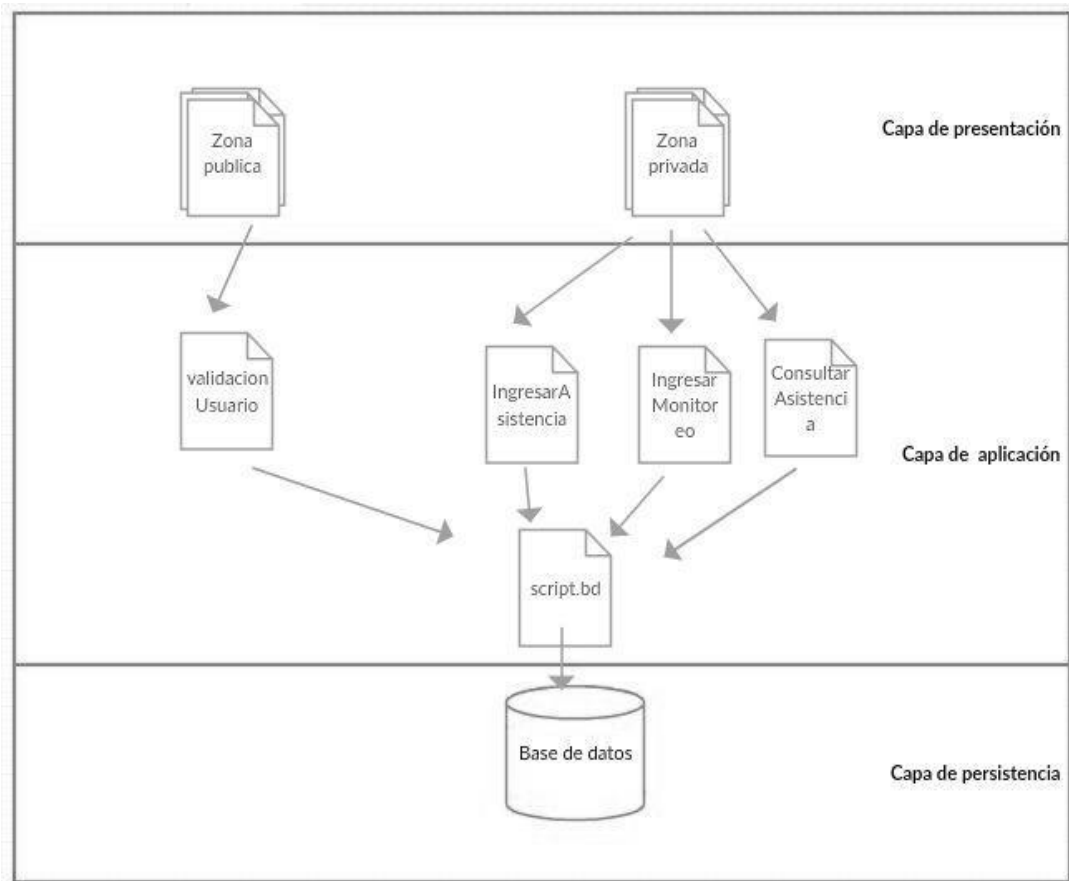


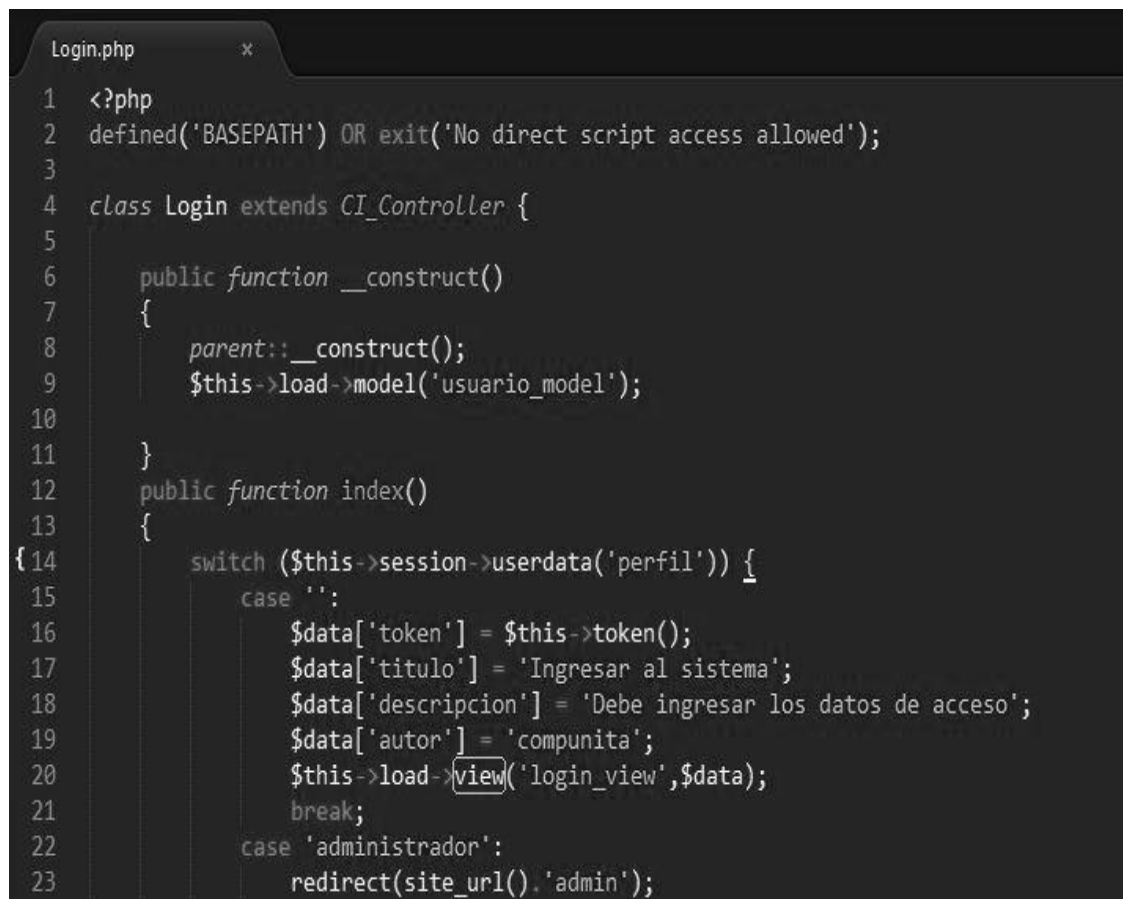
Figura 20. Capas del proyecto.

En el proyecto la capa de negocio corresponde al uso del framework CodeIgniter 3.0, el cual está diseñado para implementar el patrón Modelo Vista Controlador MVC.

Los controladores, serán los archivos creados para ejecutar las funciones del sistema, los archivos son almacenados en la subcarpeta Controllers, con el nombre de archivo: NombreArchivo.php.

Las vistas, serán los archivos encargados de mostrar la interfaces gráfica de la aplicación, es decir, todos los elementos con los que el usuario interactúa. Estos archivos son almacenados en la subcarpeta Views y con la nomenclatura nombrearchivo_view.php

Los modelos, son los archivos encargados de ejecutar las consultas a la base de datos. Estos son almacenados en la subcarpeta Models, con la nomenclatura Nombrearchivo_model.php



```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Login extends CI_Controller {
5
6     public function __construct()
7     {
8         parent::__construct();
9         $this->load->model('usuario_model');
10
11     }
12     public function index()
13     {
14         switch ($this->session->userdata('perfil')) {
15             case '':
16                 $data['token'] = $this->token();
17                 $data['titulo'] = 'Ingresar al sistema';
18                 $data['descripcion'] = 'Debe ingresar los datos de acceso';
19                 $data['autor'] = 'compunita';
20                 $this->load->view('login_view',$data);
21                 break;
22             case 'administrador':
23                 redirect(site_url().'admin');
```

Figura 21. Código fuente del Controlador para ingresar al sistema.

3.1.2.3 Capa de persistencia o datos

La capa de persistencia se puede definir como el conjunto de componentes software que proporcionan una serie de servicios que permiten a los objetos del dominio interactuar con su repositorio permanente asociado.

En el proyecto la capa de persistencia se corresponde con la base de datos de la aplicación y las distintas tablas que la conforman.

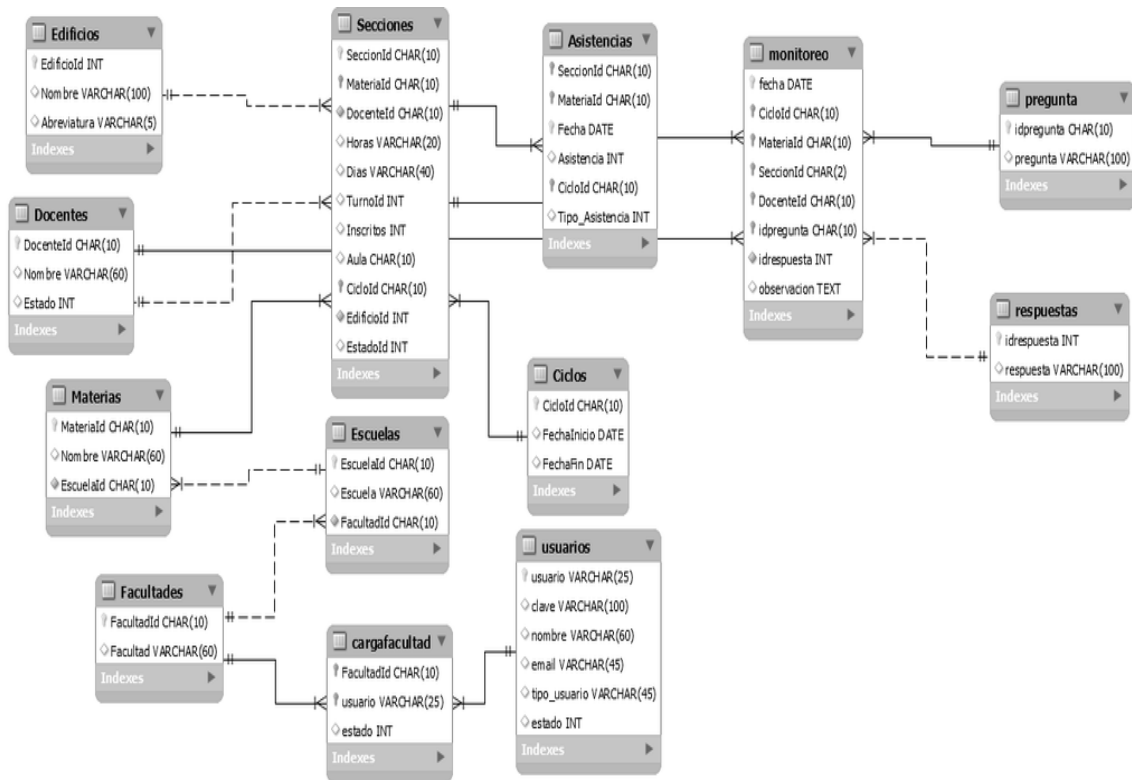


Figura 22. Modelo Entidad-Relación para la Base de Datos.

diccionario de datos

El diccionario de datos se define como un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, contenido, etc.

El objetivo de un diccionario de datos es dar precisión sobre los datos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades.

Tabla 14.

Diccionario de Datos. Tabla Edificios.

Tabla: Edificios

Descripción: Esta tabla almacenará los datos de cada uno de los edificios pertenecientes a la Universidad Tecnológica.

Llave primaria: EdificioId

Llave foránea: No posee.			
Campo	Tipo	Longitud	Descripción
EdificioId	Entero	8	Este campo contiene el Id de los edificios.
Nombre	Carácter	100	En este campo se encuentra almacenado los nombres de cada edificio de la universidad.
Abreviatura	Carácter	5	Este campo almacena la abreviatura del nombre de cada edificio perteneciente a la universidad.

Tabla 15.
Diccionario de Datos. Tabla Docentes.

Tabla: Docentes Descripción: Esta tabla almacenará el dato de cada docente perteneciente a la UTEC. Llave primaria: DocenteId Llave foránea: No posee.			
Campo	Tipo	Longitud	Descripción
DocenteId	Carácter	10	Se almacena el Id que se le asignará a cada docente.
Nombre	Carácter	60	Almacenará el nombre del docente.
Estado	Entero	8	Este campo contendrá el estado en que se encuentra el docente. Puede ser activo o inactivo.

Tabla 16.
Diccionario de Datos. Tabla Materias.

Tabla: Materias Descripción: Esta tabla contendrá la información de cada materia impartida por las diferentes escuelas de la UTEC. Llave primaria: Materiald Llave foránea: EscuelaId			
Campo	Tipo	Longitud	Descripción

MaterialId	Carácter	10	Almacena el Id de cada una de las materias.
Nombre	Carácter	60	Almacena el nombre correspondiente de cada una de las materias.
EscuelaId	Carácter	10	Almacenará el Id correspondiente de cada escuela que poseen las distintas facultades.

Tabla 17.
Diccionario de Datos. Tabla Facultades.

Tabla: Facultades Descripción: Esta tabla almacena los datos de las diferentes facultades de la que posee la UTEC. Llave primaria: FacultadId Llave foránea: No posee.			
Campo	Tipo	Longitud	Descripción
FacultadId	Carácter	10	Almacena el Id de que cada Facultad que posee la universidad.
Facultad	Carácter	60	Almacena el nombre de cada una de las facultades.

Tabla 18.
Diccionario de Datos. Tabla Secciones.

Tabla: Secciones Descripción: Esta tabla almacena las secciones que puede tener una materia que se está impartiendo actualmente. Llave primaria: SeccionId Llave foránea: MaterialId, DocenteId, CicloId, EdificioId.			
Campo	Tipo	Longitud	Descripción
SeccionId	Carácter	10	Esta almacena el Id de las distintas secciones.
MaterialId	Carácter	10	Este campo almacena el Id de las materias impartidas.

DocenteId	Carácter	10	Este campo almacena el Id de cada docente.
Horas	Carácter	20	Este campo almacena la hora clase de la sección.
Dias	Carácter	40	Este campo almacena el día clase de la sección.
TurnoId	Entero	8	Este campo almacena el Id del turno que la sección.
Inscritos	Entero	8	Este campo almacenará la cantidad de estudiantes inscritos en la sección correspondiente.
Aula	Carácter	10	Este campo almacena el aula donde se está impartiendo dicha sección.
CicloId	Carácter	10	Este campo almacena el Id del ciclo.
EdificioId	Entero	8	Este campo almacena el Id del edificio.
EstadoId	Entero	8	Este campo almacena el Id del estado en el cual se encuentra la sección.

Tabla 19.
Diccionario de Datos. Tabla Escuelas.

Tabla: Escuelas Descripción: Esta tabla almacena las escuelas pertenecientes a cada facultad de la UTEC. Llave primaria: EscuelaId Llave foránea: FacultadId			
Campo	Tipo	Longitud	Descripción
EscuelaId	Carácter	10	Este campo almacena el Id de cada una de las escuelas.
Escuela	Carácter	60	Este campo almacena el nombre de la escuela.
FacultadId	Carácter	10	Este campo almacena el Id de la Facultad donde se encuentra dicha

			escuela.
--	--	--	----------

Tabla 20.

Diccionario de Datos. Tabla cargafacultad.

Tabla: cargafacultad Descripción: Es una tabla pivote para asignar una o varias facultades a un usuario. Llave primaria: FacultadId, Usuario. Llave foránea: No posee.			
Campo	Tipo	Longitud	Descripción
FacultadId	Carácter	10	Este campo almacena el Id de la facultad.
Usuario	Carácter	25	Este campo almacenará el usuario que se encuentre en el sistema.
Estado	Entero	8	Este campo almacenará la habilitación o deshabilitación del registro.

Tabla 21.

Diccionario de Datos. Tabla Asistencias.

Tabla: Asistencias Descripción: La tabla almacenará los datos de asistencia de estudiantes a cada materia durante el ciclo. Llave primaria: Fecha Llave foránea: SeccionId, MateriaId, CicloId, DocenteId.			
Campo	Tipo	Longitud	Descripción
SeccionId	Carácter	10	Este campo almacena el Id de la Sección.
MateriaId	Carácter	10	Este campo almacenará el Id de cada materia.
Fecha	Fecha	3	Este campo contendrá la fecha en la cual se realice la toma de asistencia.
Asistencia	Entero	8	Este campo almacenará la cantidad de alumnos presentes.
CicloId	Carácter	10	Este campo contendrá el Id del ciclo

			en el cual se tome la asistencia.
Tipo_Asistencia	Entero	8	Este campo contiene el tipo de asistencia que se tome. (Clase o Evaluación)

Tabla 22.
Diccionario de Datos. Tabla Ciclos.

Tabla: Ciclos Descripción: Almacenará la información del ciclo académico en curso. Llave primaria: CicloId Llave foránea: No posee.			
Campo	Tipo	Longitud	Descripción
CicloId	Carácter	10	Este campo contiene el Id del ciclo.
FechaInicio	Fecha	3	Este campo contendrá la fecha de inicialización del ciclo académico.
FechaFin	Fecha	3	Este campo contendrá la fecha de finalización del ciclo académico.

Tabla 23.
Diccionario de Datos. Tabla Usuarios.

Tabla: usuarios Descripción: Esta tabla almacenará la información de cada usuario que sea agregado al sistema. Llave primaria: usuario Llave foránea: No posee.			
Campo	Tipo	Longitud	Descripción
usuario	Carácter	25	Este campo contendrá el usuario que se registrará.
Clave	Carácter	100	Este campo contendrá la clave del usuario.
nombre	Carácter	60	Este campo contendrá el nombre del usuario que se registrará.
Email	Carácter	45	Este campo contendrá el email del

			usuario a registrar.
tipo_usuario	Carácter	45	Este campo contendrá el tipo de usuario que se le asignará.
Estado	Entero	8	Este campo almacena el estado en el cual se encuentre dicho usuario.

Tabla 24.
Diccionario de Datos. Tabla Monitoreos.

Tabla: monitoreo Descripción: Esta tabla almacena la información obtenida del monitoreo de clase. Llave primaria: fecha Llave foránea: CicloId, MateriaId, SeccionId, DocenteId, idrespuesta.			
Campo	Tipo	Longitud	Descripción
fecha	Fecha	3	Este campo contendrá la fecha en la cual se realiza el monitoreo.
CicloId	Carácter	10	Este campo almacena el Id del ciclo en el cual se realiza el monitoreo.
MateriaId	Carácter	10	Este campo almacena el Id de la materia en el cual se realiza el monitoreo.
SeccionId	Carácter	10	Este campo almacena el Id de la sección en la cual se realiza el monitoreo.
DocenteId	Carácter	10	Este campo almacena el Id del de docente al cual se le realiza el monitoreo.
idpregunta	Carácter	10	Este campo almacenará el Id de las preguntas.
idrespuesta	Entero	8	Este campo almacenará el Id de las respuestas.
observacion	Carácter	64	Este campo almacena los datos de la observación que se realicen durante el monitoreo de clase.

Tabla 25.

Diccionario de Datos. Tabla respuesta.

Tabla: respuesta Descripción: Esta tabla almacenará las distintas opciones posibles de respuesta que el usuario con permisos ingrese al sistema para el monitoreo de clases. Llave primaria: idrespuesta Llave foránea: idopcionrespuesta.			
Campo	Tipo	Longitud	Descripción
idrespuesta	Entero	8	Este campo almacenará el Id de las distintas respuestas.
Respuesta	Carácter	100	Este campo almacena los datos de respuesta seleccionada.

Tabla 26.

Diccionario de Datos. Tabla pregunta.

Tabla: pregunta Descripción: Esta tabla almacenará las preguntas a responder para el proceso del monitoreo de clase. Llave primaria: idpregunta Llave foránea: No posee.			
Campo	Tipo	Longitud	Descripción
Idpregunta	Carácter	10	Este campo registra el Id de las preguntas correspondientes.
Pregunta	Carácter	100	Este campo almacena las distintas opciones de preguntas a responder.

3.2 Conclusiones

Se ha podido desarrollar una aplicación Web eficaz que permite la automatización del procedimiento de conteo de estudiantes y monitoreo de clases mediante la implementación de herramientas de uso libre, las cuales han permitido agilizar el diseño del producto en el menor tiempo posible.

El uso de un framework en la lógica de negocios permite reutilizar funciones, mantener un orden en la programación y agilizar las peticiones de los usuarios.

Finalmente, el desarrollo de la aplicación Web ha mejorado significativamente la usabilidad en los procesos y funciones para el conteo de estudiantes, también se logró agregar el módulo de monitoreo de clases, el cual es muy beneficioso para la toma de decisiones en la mejora de los oferta académica de la universidad y del proceso de enseñanza aprendizaje de los estudiantes y los docentes.

3.3 Recomendaciones

- Realizar copias de seguridad de la base de datos al finalizar cada ciclo de estudios para tener un respaldo de la información.
- Se recomienda el mantenimiento y actualización del sistema dado en el caso que por cada nuevo pensum o la actualización de este, algunas materias sean retiradas o su nombre sea modificado, esto con el fin de evitar posibles conflictos en la integridad de los datos en un futuro.
- El sistema es de uso únicamente para la Universidad Tecnológica de El Salvador, ya que es un sistema desarrollado en base a las materias que proporciona cada facultad según las normas de la UTEC.
- Para la visualización de la aplicación en dispositivos móviles se recomienda como mínimo el uso de una pantalla de 4.7 pulgadas.

Referencias

Álvarez, M. (2009). *Manual de Codeigniter: Desarrollo Web*. Recuperado de <http://www.desarrolloweb.com/articulos/codeigniter.html>

Arias, A. & Durango, A. (2016). *Ingeniería y Arquitectura del Software*. (2ª ed.). Madrid: CreateSpaceAmazonEuropa. Recuperado de https://books.google.com.sv/books?id=cixpCwAAQBAJ&dq=arquitectura+de+software&source=gbs_navlinks_s

Casanova, J. (2004). *Usabilidad y Arquitectura del Software*. Recuperado de <http://www.desarrolloweb.com/articulos/1622.php>

Climont, C. (2013). *Ventajas e inconvenientes de los Framework*. Recuperado de <https://github.com/carlesclimont/curso-symfony2/blob/master/1-introduccion/ventajas-e-inconvenientes-de-los-frameworks.md>

Ecured. *Arquitectura de Software*. Recuperado de http://www.ecured.cu/Arquitectura_de_software

Fontán, M. (2012). *CodeIgniter, un framework PHP para el desarrollo rápido de aplicaciones Web*. Recuperado de <http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>

Fundación Wikimedia; Wikipedia la Enciclopedia Libre. (2016). *Modelo-vista-controlador*. Recuperado de <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>

Fundación Wikimedia; Wikipedia la Enciclopedia Libre.(2016).*Framework*. Recuperado de <https://es.wikipedia.org/wiki/Framework>

Morelos, C. (2014). *Qué son los Framework*. Recuperado de <http://www.nubelo.com/blog/que-son-los-frameworks/>

OpenUp. (2011). *Concepto: Software Architecture*. Recuperado de http://epf.eclipse.org/wikis/openupsp/openup_basic/guidances/concepts/software_architecture,___O7tAMVvEduLYZUGfgZrkQ.html

Rios, A. (2012). *Introducción a CodeIgniter*. Recuperado de <http://www.antoniorios.net/blog/?p=373>

Universidad de Alicante. *Modelo-Vista-Controlador*. Recuperado de <http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

Universidad de Cataluña. *Diseño Centrado en el Usuario*. Recuperado de http://delreguero.com/wordpress/wpcontent/uploads/DAW/UOC_Disen%C3%B3_centrado_en_el_usuario.pdf

Anexos

Anexo 1. Matriz de congruencia.

MATRIZ DE CONGRUENCIA		
TEMA: Automatización del Procedimiento de Conteo de Estudiantes y Monitoreo de Clases en la Universidad Tecnológica de El Salvador.		
ENUNCIADO DEL PROBLEMA: ¿Cuáles serán las mejoras que traerá consigo la Automatización del Procedimiento de Conteo de estudiantes y Monitoreo de Clases en la Universidad Tecnológica de El Salvador?		
OBJETIVO GENERAL: Desarrollar una aplicación web para la automatización del procedimiento de conteo de estudiantes y monitoreo de clases en cada ciclo de estudios en la Universidad Tecnológica de El Salvador.		
OBJ. ESPECÍFICO 1: Manejar el Patrón de Arquitectura de Software Modelo-Vista-Controlador para potenciar la facilidad de mantenimiento y reutilización de código en el desarrollo de la aplicación web.	OBJ. ESPECÍFICO 2: Analizar los datos que se deben registrar en el sistema de información para crear de manera eficaz la estructura de la base de datos.	OBJ. ESPECÍFICO 3: Elaborar el manual de usuario de la aplicación para que sirva de referencia en algún proceso implicado en el conteo de estudiantes y monitoreo de clases.
ALCANCE 1: Uso de CodeIgniter 3.0 como framework de la aplicación web bajo el patrón MVC.	ALCANCE 2: Descripción de las fases para el desarrollo de la aplicación web.	ALCANCE 3: Documentar el código fuente de la aplicación web.
PRODUCTO 1: Aplicación web desarrollada con el framework CodeIgniter MVC, gestor de bases de datos MySQL, y otros lenguajes de programación tales como JQUERY, HTML5 y CSS2.	PRODUCTO 2: Documento en formato PDF, el cual describe las fases de análisis, diseño, desarrollo e implementación de la aplicación web.	PRODUCTO 3: Líneas de comentarios en cada archivo para las funciones, variables y procesos que requieran una mayor atención en el desarrollo de la aplicación.

Anexo 2. Carta de autorización.

San Salvador, jueves 3 de marzo de 2016

Ing. Jorge Armando Aparicio Lemus
Director de la Escuela de Informática
Universidad Tecnológica de El Salvador

Presente:

Reciba un cordial saludo de nuestra parte deseándole éxitos en sus labores cotidianas y muchas bendiciones para su familia.

El motivo de la presente es para solicitarle en calidad de estudiantes egresados de la carrera Técnico en Ingeniería de Software, su autorización para la realización del proyecto titulado "Sistema Informático para el Conteo y Monitoreo de Clases", el cual es parte de nuestro proceso de graduación.

Sin más por el momento nos suscribimos:

Ricardo Antonio Beltrán Mena

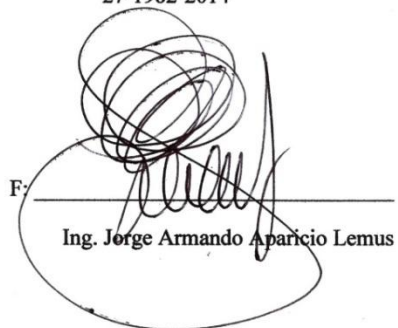
27- 0205-2014

Ever Mauricio Escoto Vasquez

27- 6941-2012

Rebeca Michelle Menjivar Merino

27-1982-2014

F: 
Ing. Jorge Armando Aparicio Lemus