

UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA
LICENCIATURA EN CIENCIAS DE LA COMPUTACION



PROYECTO:

SISTEMA DE ASISTENCIA POR MATERIA

EQUIPO DESARROLLADOR:

GUILLERMO ANTONIO PÉREZ FUENTES	PF01132156
DENNYS ERNESTO GARCIA RIVAS	GR01134464
JIMMY ALEXANDER GARCIA RIVAS	GR01134477
GLENDA IZAMAR GONZÁLEZ CAMPOS	GC01132280

CÁTEDRA: PROGRAMACIÓN II

HORARIO: MIERCOLES - 1:00 PM a 3:00PM

FACILITADOR: LIC. RIGOBERTO ISRAEL ORELLANA

INDICE

<i>INTRODUCCION</i>	3
<i>PROBLEMÁTICA A RESOLVER</i>	4
<i>OBJETIVOS</i>	5
<i>JUSTIFICACION</i>	6
<i>ANALISIS DEL BENEFICIO DEL PROYECTO</i>	7
<i>METODOLOGIA Y CICLO DE VIDA</i>	9
<i>DISEÑO</i>	10
<i>DESARROLLO</i>	19
<i>Técnicas de programación aplicadas</i>	19
<i>Herramientas de desarrollo utilizadas</i>	21
<i>Lenguaje de programación y gestor base de datos</i>	22
CONCLUSIONES	25
RECOMENDACIONES	26

INTRODUCCION

El presente trabajo pretende establecer una nueva herramienta de software para los docentes de la Universidad Luterana Salvadoreña, que, permita registrar la asistencia de los estudiantes en cada una de las materias impartidas en dicha universidad, también tiene como objetivo agilizar la recolección de datos y sobre todo optimizar los recursos y tiempo que se implica al tomar asistencia en cada materia, por ultimo también pretende agilizar la obtención de los porcentajes finales de asistencia en cada cátedra, generando un reporte automático en el cual se muestren los datos de cada estudiante inscrito. El software se presenta de manera simplificada, en la cual está marcada su facilidad de uso para las diferentes edades de catedráticos, con un diseño básico, minimalista y eficaz, cada apartado donde tiene que estar

PROBLEMÁTICA A RESOLVER

Hasta el día de hoy la asistencia de pasa de manera manual, teniendo en cuenta que solo hay que firmar una hoja que hace constar que el estudiante asistió a clases, más sin embargo hay que tomar en cuenta un par de factores que podrían alterar la veracidad de los datos, por ejemplo: compañeros firmando a estudiantes que faltaron a clase, hojas que los catedráticos pierden sin querer, catedráticos que olvidan pasar la hoja de asistencia; entre otros factores que alteran de alguna u otra manera los datos correctos de asistencia de los estudiantes.

OBJETIVOS

Objetivo general

- ▲ Sistematizar la asistencia por materia en la Universidad Luterana Salvadoreña

Objetivos específicos

- ▲ Organizar de mejor manera los recursos disponibles en la Universidad Luterana Salvadoreña
- ▲ Registrar de manera precisa la asistencia de los estudiantes por materia en la Universidad Luterana Salvadoreña
- ▲ Establecer una nueva herramienta tecnológica que ayuden a ser más precisa y ágil la tarea de asistencia por materia en la Universidad Luterana Salvadoreña

JUSTIFICACION

La tecnología aporta un sinfín de utilidades, solo se debe tener un poco de imaginación y saber las oportunidades que en cada ámbito se necesite mejorar, de ahí nacen nuevas ideas y proyectos determinados a cubrir esas necesidades, agilizarlas, hacerlas más precisas y optimizar recursos. La Universidad Luterana Salvadoreña cuenta con reglamentos como toda institución de educación, y nosotros como estudiantes debemos cumplir, por ello la aportación de una herramienta tecnológica que obligue al catedrático a cumplir reglamentos como lo es de pasar asistencia sus estudiantes, aportara un valor adicional de calidad académica, ya que los estudiantes se verán obligados a asistir mínimo el 80% en cada asignatura.

ANALISIS DEL BENEFICIO DEL PROYECTO

El sistema de asistencia por materia, no solo tiene como objetivo, mejorar el control de esta misma, si no, también, optimizar recursos, el gasto de papeles, folders, portafolios, entre otros recursos que se utilizan para guardar la información necesaria; tomando en cuenta que son alrededor de 50 a 60 materias impartidas por ciclo en la universidad; también, ahorrar tiempo en la búsqueda individual de asistencia por estudiantes, si un catedrático imparte alrededor de 2 a 4 materias y en cada una habrá un mínimo de 20 estudiantes, la tarea se volvería un poco tediosa al querer sacar el porcentaje de asistencia que tuvo cada estudiante en cada materia impartida por el catedra.

De esta manera el sistema muestra ser más solidario, con la utilización de recursos, y más ágil y eficaz, a la hora de sacar sus respectivos porcentajes de asistencia por estudiante.

USUARIO: Docentes

BENEFICIOS:

- ▲ Ahorro de recursos
- ▲ Correcta información sobre la asistencia de los estudiantes
- ▲ Consultas y registro de asistencia en un solo lugar
- ▲ Orden de asistencia por materias
- ▲ Respaldo por quejas

USUARIO: Alumnos

BENEFICIOS:

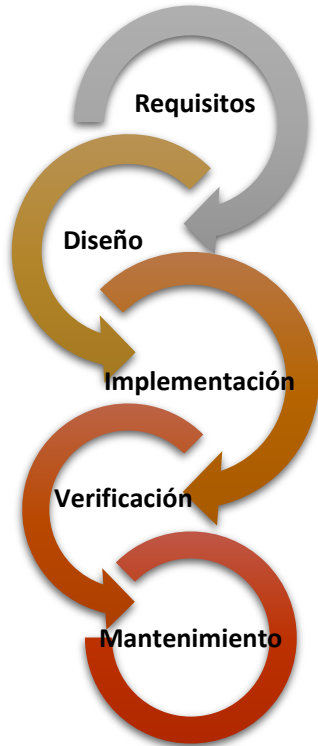
- ▲ Respaldo de asistencia
- ▲ Orden en salón clase
- ▲ Ahorro de recursos
- ▲ Consulta de días asistidos
- ▲ Solicitud de permiso registrado

USUARIO: Administración académica

BENEFICIOS:

- ▲ Correcta información de asistencia
- ▲ Respaldo de asistencia por materias

- ^ Cumplimiento de normas universitarias como la de 80% de asistencia por asignatura para aprobar.
- ^ Verificación de compromiso de los docentes
- ^ Control de asistencia de los estudiantes



METODOLOGIA Y CICLO DE VIDA

El modelo en cascada es un proceso de desarrollo secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo, como una cascada.

Requisitos del software

En esta fase se hace un análisis de las necesidades del cliente para determinar las características del software a desarrollar, y se especifica todo lo que debe hacer el sistema sin entrar en detalles técnicos. Hay que ser especialmente cuidadoso en esta primera fase, ya que en este modelo no se pueden añadir nuevos requisitos en mitad del proceso de desarrollo.

El cual se manejará por medio de entrevistas, cuestionarios, corroboración con el sistema implantado de la universidad para manejar un mismo tipo de estilo en tablas, letras y colores.

Diseño

En esta etapa se describe la estructura interna del software, y las relaciones entre las entidades que lo componen.

Implementación

En esta fase se programan los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior. La programación es el proceso que lleva de la formulación de un problema de computación, a un programa que se ejecute produciendo los pasos necesarios para resolver dicho problema.

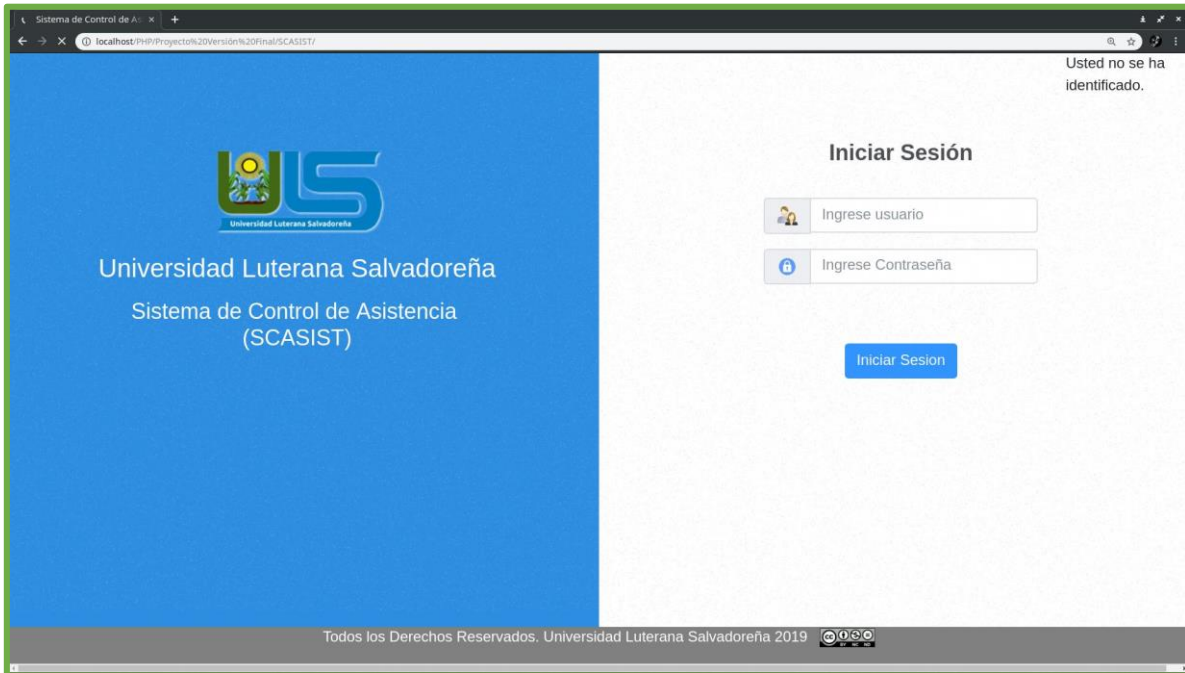
Verificación

Como su propio nombre indica, una vez se termina la fase de implementación se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.

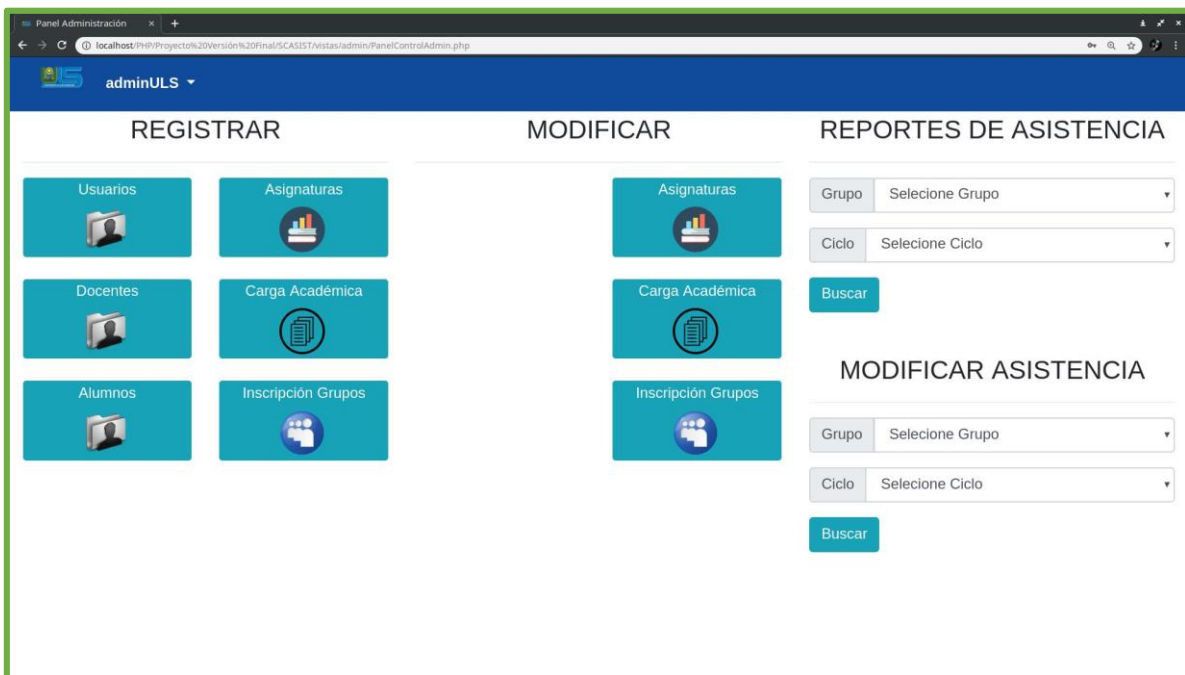
Instalación y mantenimiento

Una vez se han desarrollado todas las funcionalidades del software y se ha comprobado que funcionan correctamente, se inicia la fase de instalación y mantenimiento. Se instala la aplicación en el sistema y se comprueba que funcione correctamente en el entorno en que se va a utilizar.

DISEÑO



1 – Primera pantalla del sistema, LOGIN



2 – Primera vista al ingresar como administrador al usuario donde se encuentran los apartados de Registrar y modificar

Registrar Carga Académica: ✕

N° Grupos:

Carrera:

3 – Registro de carga académica

Registrar Nueva Asignatura: ✕

Código:

Asignatura:

Cómputo:

4 – Registro de asignatura

Registrar Nuevo Docente: ✕

Código: 123...

DUI: 00000000-0

Grado Académico: Lic., Ing., Mtr. o Mgtr., Dr. C o DR. Cs.

Nombres: Nombres

Apellidos: Apellidos

Dirección:

Telefono: 00000000

Correo: @uls.edu.sv, @gmail.com

Usuario: Seleccione usuario ▼

Procesar **Cancelar**

5 – Registro de docente

Registrar Estudiantes: ✕

N° Estudiantes:

Carrera:

6 – Registro de estudiante, primero procede a ingresar la cantidad de estudiantes que registrara el usuario y luego la carrera que el o los estudiantes estén cursando.

Registrar Estudiantes en Grupo: ✕

N° Estudiantes:

Grupo:

7 – Registro de estudiantes en grupo

Registrar Nuevo Usuario: ✕

Usuario:

Password:

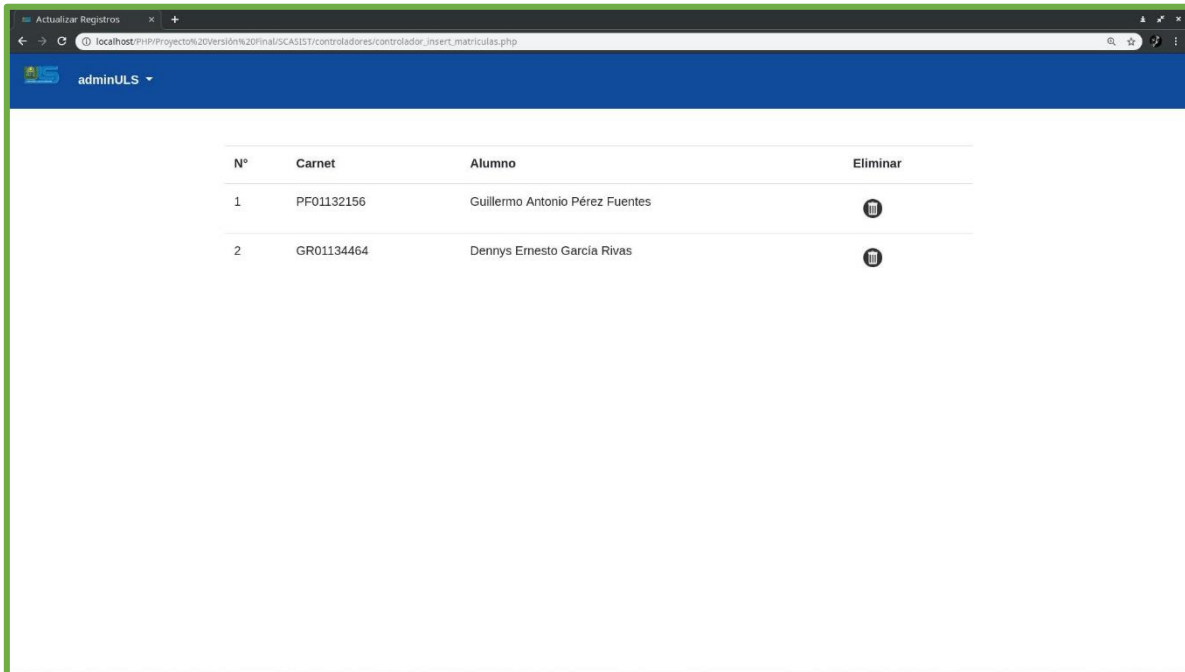
Tipo:

8 – Registro de nuevos usuarios para el sistema que pueden ser, administradores o docentes, se complementa con el registro de docentes.

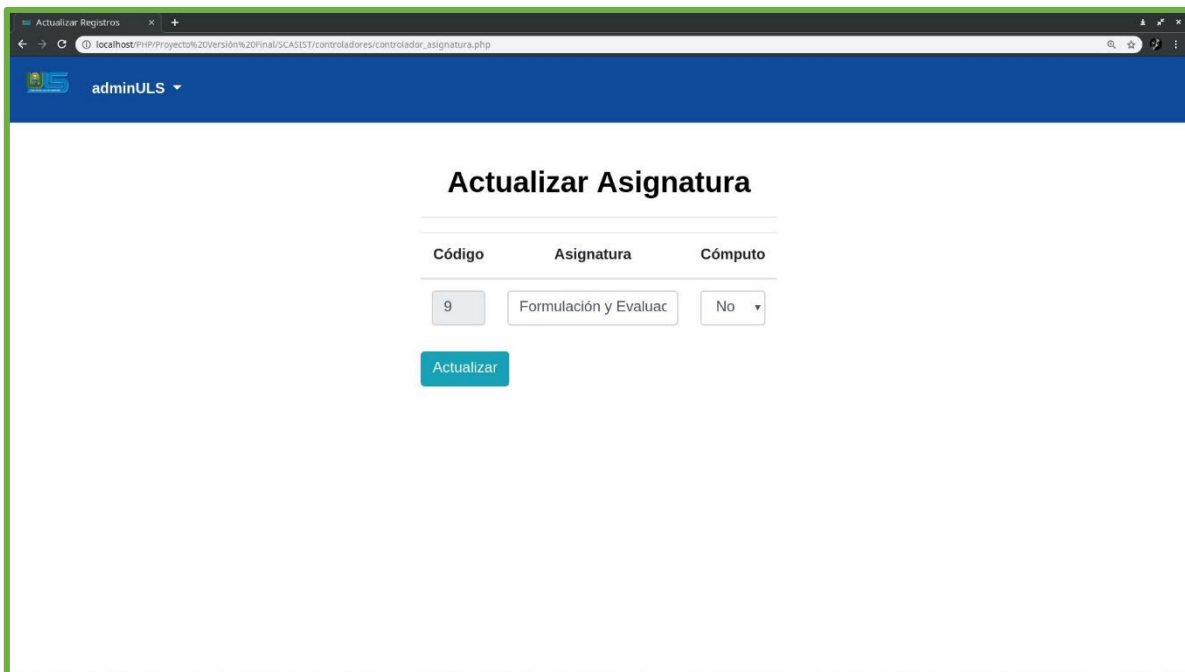
Actualizar Carga Académica

N°	Grupo	Asignatura	ID Docente	Horario	Día
1	7525	Administración de Personal	1	De:07:00:00 Hasta 09:	MIERCOLES
2	7618	Administración de Sistemas Infor	777	De:04:00:00 Hasta 07:	JUEVES
3	7613	Algoritmo II	0	De:09:40:00 Hasta 12:	MIERCOLES
4	7619	Algoritmo II	0	De:09:40:00 Hasta 12:	SABADO
5	7614	Base de datos I	387	De:09:40:00 Hasta 12:	MIERCOLES
6	7620	Base de datos I	386	De:09:40:00 Hasta 12:	SABADO
7	7648	Consultoría Profesional	0	De:04:00:00 Hasta 07:	VIERNES
8	7675	Consultoría Profesional	0	De:07:00:00 Hasta 09:	SABADO

9 – Actualizar carga academica



10 – Actualizar lista de registro, se complementa con inscripción de grupos



11 – Actualizar asignatura, nombre y sí, requiere de computo.

Reporte de Asistencia

Grupo: 7615
Materia: Programación II
Docente: Lic. Rigoberto Israel Orellana Orellana
Ciclo : I-2019
Horario : Miércoles De:01:00:00 Hasta 03:30:00

Asistio = ✓ | No asistio = ✗ | Permiso = P

N°	Alumno/a	23-01	30-01	06-02	13-02	20-02	27-02	06-03	13-03	20-03	Dias asistidos	% Asistencia
1	David Armando Cornejo Alfonso	✗	✗	P	✓	✓	P	✓	✓	✓	7	37%
2	Glenda Izamar González Campos	✗	✓	✓	✓	✓	P	✓	✓	✓	8	42%
3	Edwin Ernesto González Castillo	✓	✓	✓	✓	✓	✓	✓	✓	✗	8	42%
4	Dennys Ernesto García Rivas	✓	✓	✓	✓	✓	✓	✓	✓	✓	9	47%
5	Jimmy Alexander García Rivas	✓	✓	✓	✓	✓	✓	✓	✓	✓	9	47%
6	José Francisco Murcia Hernández	✓	✓	✓	✓	✓	✓	✗	✓	✓	8	42%
7	Luis Miguel Martir Martínez	✓	✓	✓	✓	✓	✓	✓	✓	✓	9	47%
8	Cristian Osvaldo Mejía Mendoza	✓	✓	✓	✓	✓	✓	✓	✓	✓	9	47%
9	Diego Rolando Olmedo Melgar	✓	✓	✓	✓	✓	✓	✗	✓	✓	8	42%
10	Guillermo Antonio Pérez Fuentes	✓	✓	✓	P	✓	✓	✓	P	✓	9	47%
11	Karla Arely Rodríguez Dueñas	✗	✓	✓	✓	✓	✓	✓	✓	✓	8	42%
12	Oscar Eliezar Romero Quintanilla	✓	✓	P	✓	✓	✓	✓	✓	✓	9	47%
13	Yoselin Antonieta Ramos Sandoval	✓	✓	✓	✓	✓	✓	✓	✓	✓	9	47%

Sistema de Control de Asistencia. (SCASIST)

12 – Reporte de asistencia, se selecciona el docente materia y horario, tanto como ciclo en que se imparte la asignatura

Modificar Asistencia

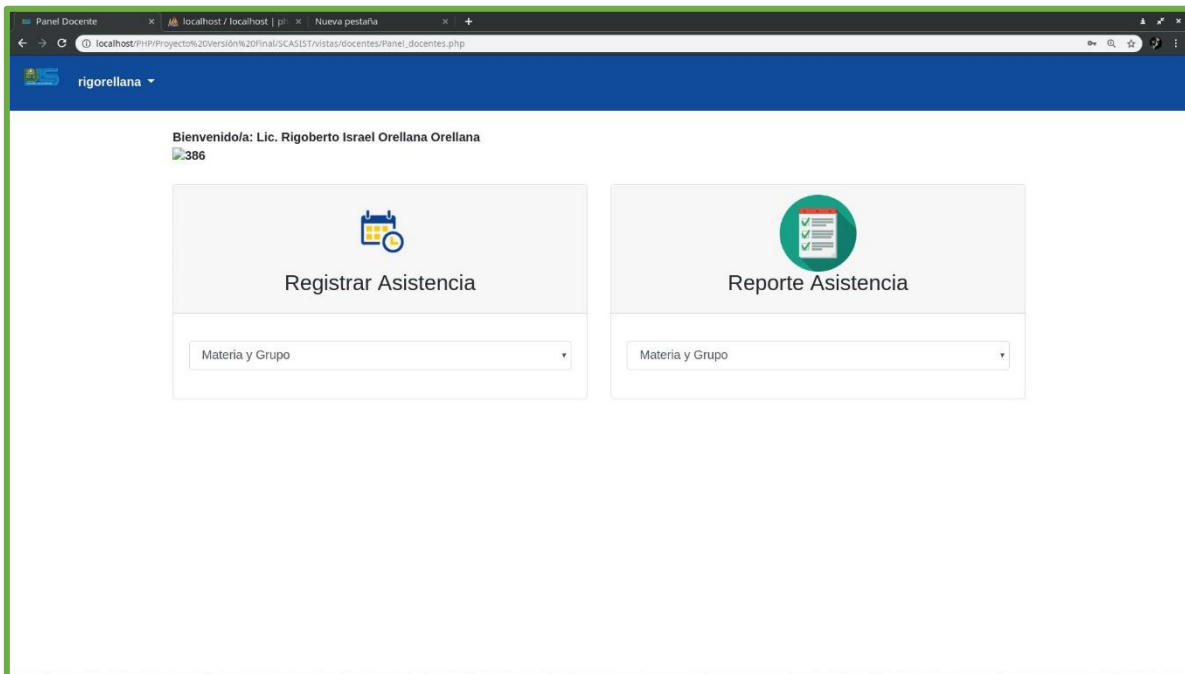
Grupo: 7615
Materia: Programación II
Docente: Lic. Rigoberto Israel Orellana Orellana
Ciclo : I-2019
Horario : Miércoles De:01:00:00 Hasta 03:30:00

Asistio = ✓ | No asistio = ✗ | Permiso = P

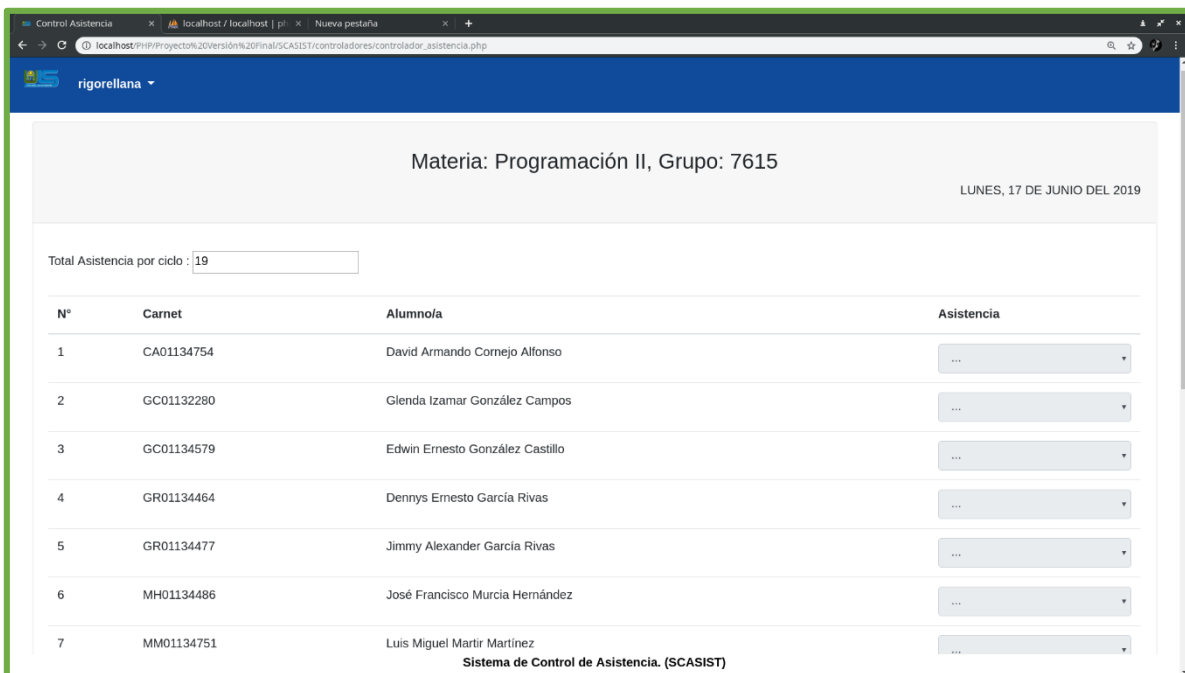
N°	Alumno/a	23-01	30-01	06-02	13-02	20-02	27-02	06-03	13-03	20-03
1	David Armando Cornejo Alfonso	✗	✗	P	✓	✓	P	✓	✓	✓
2	Glenda Izamar González Campos	✗	✓	✓	✓	✓	P	✓	✓	✓
3	Edwin Ernesto González Castillo	✓	✓	✓	✓	✓	✓	✓	✓	✗
4	Dennys Ernesto García Rivas	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	Jimmy Alexander García Rivas	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	José Francisco Murcia Hernández	✓	✓	✓	✓	✓	✓	✗	✓	✓

Sistema de Control de Asistencia. (SCASIST)

13- Modificar asistencia por materia.



14 – Vista del docente, solamente se le permite editar el perfil registrar asistencia y ver o imprimir el reporte de asistencia





15 – Registro de asistencia, se registra la cantidad de días que se impartirá la materia por ciclo, y luego se procede a llenar la asistencia, tomando en cuenta que solamente se habilitará el llenado de asistencia el día y hora específica en la que se imparte la materia para llevar un control exacto.

Control Asistencia x localhost / localhost | p... Nueva pestaña x +

localhost/PHP/Proyecto%20Versión%20Final/SCASIST/controladores/controlador_asistencia.php

rigorellana

 **Reporte de Asistencia**

 Grupo: 7615
Materia: Programación II
Docente: Lic. Rigoberto Israel Orellana Orellana
Ciclo : I-2019
Horario : Miércoles De:01:00:00 Hasta 03:30:00

Asistio = ✓ | No asistio = ✗ | Permiso = P

Nº	Alumno/a	23-01	30-01	06-02	13-02	20-02	27-02	06-03	13-03	20-03	Días asistidos	% Asistencia
1	David Armando Cornejo Alfonso	✗	✗	P	✓	✓	P	✓	✓	✓	7	37%
2	Glenda Izamar González Campos	✗	✓	✓	✓	✓	P	✓	✓	✓	8	42%
3	Edwin Ernesto González Castillo	✓	✓	✓	✓	✓	✓	✓	✓	✗	8	42%
4	Dennys Ernesto García Rivas	✓	✓	✓	✓	✓	✓	✓	✓	✓	9	47%
5	Jimmy Alexander García Rivas	✓	✓	✓	✓	✓	✓	✓	✓	✓	9	47%
6	José Francisco Murcia Hernández	✓	✓	✓	✓	✓	✓	✗	✓	✓	8	42%

Sistema de Control de Asistencia. (SCASIST)

16 – Reporte generado de asistencia de una materia en un 47%

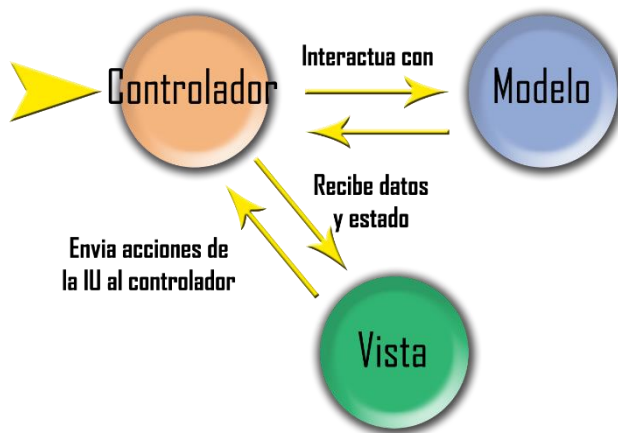
DESARROLLO

Técnicas de programación aplicadas

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

- ▲ El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- ▲ La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- ▲ El Controlador, que 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.



MODELO

El modelo es el responsable de:

- ▲ Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- ▲ Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- ▲ Lleva un registro de las vistas y controladores del sistema.

- ^ 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.).

CONTROLADOR

El controlador es responsable de:

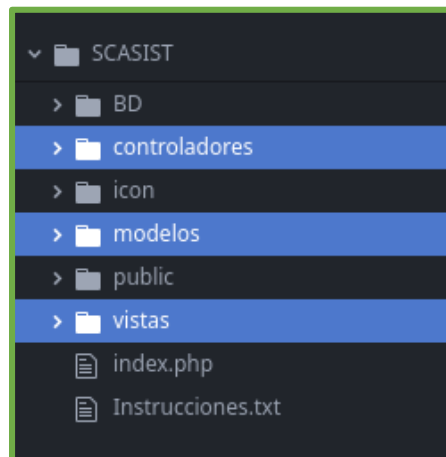
- ^ Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- ^ Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega (nueva_orden_de_venta)".

VISTAS

Las vistas son responsables de:

- ^ Recibir datos del modelo y los muestra al usuario.
- ^ Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- ^ Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

Representación visual de MVC, implementado en el "Sistema de asistencia por materia"



Servicio de Informática ASP.NET MVC 3 Framework. (2018). Modelo vista controlador (MVC). 2019, de Universidad de Alicante Sitio web: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

Herramientas de desarrollo utilizadas

Editor de texto: ATOM

Versión: 1.36.0 UV (11 de abril 2019)

Características:



^ *Interfaz gráfica y personalización con temas visuales.*

Atom dispone por defecto de una interfaz de usuario amigable, con un diseño moderno y visualmente atractivo. Esto que parece algo secundario, es muy importante a la hora de programar, para sentirnos cómodos y que no se canse la vista.

Para gestionar las diferentes combinaciones de colores, Atom emplea los llamados temas visuales. Estos permiten definir el diseño de la interfaz y de la sintaxis, aportando la flexibilidad necesaria para que el usuario elija entre temas oscuros o claros.

^ *Facilidad para añadir funcionalidades extra con extensiones.*

Una de las opciones estrella que nos ofrece este editor de texto, es la posibilidad de añadir funcionalidades extra de forma sencilla mediante la instalación de extensiones. Estas extensiones, son como paquetes o plugins desarrollados para añadir una característica nueva al editor.

^ *Gestor de paquetes atom*

Para facilitar la administración de las extensiones, Atom dispone de un gestor de paquetes propio con el que podemos instalar, desinstalar, habilitar/deshabilitar o actualizar las extensiones. La comunidad es muy activa en el desarrollo y mantenimiento de extensiones, por lo que podemos encontrar extensiones de todo tipo.

^ *Herramientas para desarrolladores.*

Aunque Atom es un editor de texto apto para todo tipo de usuarios, serán los desarrolladores quienes logren exprimirle todo el potencial. Veremos que se integran una gran colección de extensiones que permiten añadir las funcionalidades que el programador necesita en su día a día.

Algunas de las funcionalidades más destacables que encontraremos son:

- ^ Resaltado de sintaxis para infinidad de lenguajes de programación (PHP, HTML, SQL, C#, Python, Java, CSS, Markdown, Javascript, Go, Ruby, etc).

- ▲ Autocompletado con soporte para los lenguajes de programación más conocidos. Podremos escribir código de una forma más eficiente gracias a un sistema de autocompletado.
- ▲ Integración con Git y Github. Disponemos de herramientas de control de versiones para poder gestionar el código alojado en Github o a través de Git.
- ▲ A la hora de programar, podemos depurar código directamente y buscar errores en la sintaxis.
- ▲ Posibilidad de editar código de forma colaborativa a través de la extensión Teletype.
- ▲ Viendo las características básicas, no es de extrañar que este editor se pueda convertir en un auténtico entorno de desarrollo. Han sido los propios desarrolladores quienes han creado un conjunto de extensiones que podemos instalar libremente y convertir Atom en un IDE.



Lenguaje de programación y gestor base de datos



Lenguaje de programación: PHP

Versión: 7.1 (1 de diciembre 2016)

Características:

- ▲ Gran extensión de documentación

Ya sea creada de manera oficial como por los usuarios, PHP tiene la ventaja de ser usado por millones de usuarios en muchos proyectos. De cara al nuevo programador es una gran ventaja porque no hay nada mejor que un lenguaje con una gran comunidad detrás, sobre todo para los no tan afines al idioma anglosajón.

- ▲ Variedad de herramientas para aprender

Tenemos varios métodos para iniciarnos: podemos contratar un servidor remoto para subir nuestros archivos, como haríamos con cualquier web, lo cual tiene lógica ya que si tenéis vuestro sitio en la red ya tenéis este recurso con vosotros; podemos crear un servidor local con una máquina virtual, o lo más popular entre los alumnos: descargar la herramienta libre WAMP (XAMP o LAMP en función de si es para Mac OSX o Linux). Esta herramienta simula que vuestra máquina local es un servidor web para poder ejecutar los ficheros PHP. Es decir, sea cual sea vuestra

situación, es imposible que os quedéis fuera, sobretodo teniendo en cuenta que no requiere un entorno de desarrollo complejo, como ocurre con Titanium.

Hoy en día las webs más complejas funcionan bajo PHP, por lo que son muchas las empresas que requieren los servicios de un desarrollador de este lenguaje.

▲ Permite programación orientada a objetos

Esto significa que podréis dividir vuestros scripts en métodos, clases, etc. para hacer más ágil de cara al servidor el procesamiento de los datos.

▲ Módulos externos para mejorar la aplicación web

Gracias a esto, podréis implementar funcionalidades que de por sí no existen, siendo una gran solución a vuestras necesidades. Por ejemplo, en vez de programar de cero una pantalla de registro, podréis usar un módulo ya hecho. Esto no sería posible si no fuese orientado a objetos, como menciono en el punto anterior.

▲ Se puede separar la estructura

Con esto quiero decir que podréis tener de manera independiente el código que se encarga de mover los datos del que se enlaza a la interfaz. Esto se conoce como Modelo Vista/Controlador (MVC). Gracias a esta característica, no “contaminaremos” código con líneas innecesarias y lo mantendremos limpio y ordenado.

▲ La libertad es un bien muy preciado

Si, PHP es totalmente libre y multi-plataforma, por lo que no importa que sistema operativo uséis. Su desarrollo es posible en cualquier sitio (siempre y cuando la máquina sea capaz de ejecutar el código).

Gestor de base de datos: MySQL

Versión: 5 (24 de agosto 2010)

Características:

- ▲ El principal objetivo de MySQL es velocidad y robustez.
- ▲ Soporta gran cantidad de tipos de datos para las columnas.
- ▲ Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.



- ^ Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- ^ Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- ^ Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- ^ El servidor soporta mensajes de error en distintas lenguas.

CONCLUSIONES

La nueva era de la informática, provee un sinfín de nuevas tecnologías, por ende la seguridad, responsabilidad, ahorro de recursos y exactitud en los datos, son parte de los beneficios que adquieren quienes trabajan con ella, como grupo consideramos necesaria la implementación de SCASIST, no solo para brindar estos beneficios, si no uno también muy importante como es la transparencia de la asistencia de los estudiantes.

RECOMENDACIONES

- △ Poner en ejecución el sistema, ya que facilitara la administración de asistencia por estudiantes
- △ Crear dos o tres administradores
- △ Implementar el sistema como prueba un ciclo con los docentes de informática, para verificar y constatar el resultado de esta herramienta
- △ Capacitar el personal que se utilizara como prueba, por los creadores del sistema para su buen uso
- △ Crear una red de wifi para los docentes más extensa que permita abarcar todo el campus para la conexión a internet, ya que será necesaria para el registro de los datos