



Universidad Central

Facultad de Ingeniería

Carrera de Ingeniería Informática

II-51 PROGRAMACIÓN INTERNET

Proyecto: Desarrollo Plataforma educativa

Estudiante:

Milagros de Jesús Hernández Jiménez

SM24006689

Profesor:

Fabian Chinchilla Mayorga

Julio 2025

Introducción

Este documento presenta el desarrollo de un sitio web académico orientado a simular gestiones institucionales básicas. La plataforma ha sido construida utilizando componentes funcionales como HTML semántico, CSS y Bootstrap, junto con funcionalidades básicas en PHP que permiten simular un entorno dinámico conectado a una base de datos.

El enfoque principal se centra en la organización y visualización de la información, mediante una interfaz clara y una estructura coherente. El sistema está compuesto por cuatro módulos principales: Login, Estudiantes, Profesores y Cursos. Cada sección permite visualizar información almacenados en la base de datos, mediante tablas dinámicas, registrar nuevos elementos mediante formularios con validación HTML5, y navegar fluidamente entre páginas.

El sistema permite realizar operaciones reales como la inserción y eliminación de datos utilizando sentencias en PHP. Además, se incluyen alertas interactivas para mejorar la experiencia del usuario, como confirmaciones de registro o eliminación. Cada módulo fue diseñado para facilitar la lectura del contenido en un entorno moderno, limpio y estructurado. Se emplearon componentes reutilizables y una barra de navegación fija que mejora la experiencia de uso.

En conjunto, el proyecto demuestra la integración entre diseño visual, estructura semántica y simulación funcional. Además, se incluye un diagrama UML que representa las clases principales del sistema, sus atributos y relaciones, estableciendo una base sólida para futuras extensiones dinámicas.

Objetivos

Objetivo General

Desarrollar una plataforma educativa web que simule las gestiones institucionales de estudiantes, profesores y cursos, mediante módulos funcionales con navegación fluida, formularios validados, conexión a base de datos y estructura visual coherente.

Objetivos Específicos

- Diseñar una estructura web utilizando HTML semántico y estilos CSS coherentes en cada módulo.
- Implementar formularios funcionales con validación HTML5 para los distintos registros.
- Usar PHP con conexión a base de datos para insertar, consultar y eliminar registros reales.
- Permitir la navegación entre módulos mediante enlaces funcionales.
- Representar la lógica del sistema mediante un diagrama UML que refleje las clases, atributos y relaciones principales.

Estructura del Proyecto

A continuación, se describe la organización de carpetas y archivos que conforman el sistema. La estructura sigue un esquema claro y ordenado, lo cual facilita sus futuras modificaciones. Cada carpeta cumple una función específica y almacena los archivos necesarios para el buen funcionamiento de la plataforma.

La mayoría de los módulos están desarrollados en PHP, utilizando conexión real a una base de datos mediante PDO. Esto permite insertar, consultar y eliminar datos en tiempo real. Solo el módulo de Login permanece como archivo HTML estático.

Los archivos emplean etiquetas semánticas como <header>, <nav>, <main>, <section>, <form>, <table> y <article>, además, los estilos se encuentran embebidos en cada archivo mediante la etiqueta <style>.

A continuación, se describe la organización de carpetas y archivos que conforman el proyecto:

1. **Carpeta *Pages*:** Contiene los archivos principales del sitio web, uno por cada módulo implementado:
 - login.html: página de inicio de sesión simulado (estática). *Ver anexo 1*
 - cuenta_Re.php: Página adicional para un formulario de registro de usuario, simulado. *Ver anexo 2*
 - Inicio.php: Página principal del sistema, muestra la bienvenida y noticias. *Ver anexo 3*
 - cursos.php: Módulo de gestión de cursos. *Ver anexo 4*
 - registroCurso.php: Formulario funcional para registrar cursos. *Ver anexo 5*
 - profesores.php: Módulo de gestión de profesores. *Ver anexo 6*
 - registroProfesor.php: Formulario funcional para registrar profesores. *Ver anexo 7*
 - estudiantes.php: Módulo de gestión de estudiantes. *Ver anexo 8*

- registroEstudiante.php: Formulario funcional para registrar estudiantes. Ver *anexo9*
- 2. **Carpeta *images***: Contiene los recursos visuales utilizados por las páginas, así como el diagrama del sistema.
 - LogoU.png: Logotipo con fondo blanco
 - LogoUtransparente.png: Logotipo con fondo transparente
 - Nuevo Diagrama UML.png: Diagrama UML actualizado con clases y atributos.
- 3. **Archivo index**: Archivo raíz, que puede servir como punto de entrada general al sistema.
- 4. **Archivo db.php**: Archivo que contiene la lógica de conexión a la base de datos utilizando PDO.

Descripción de los Módulos

A continuación, se describen los módulos implementados. Cada uno fue desarrollado utilizando etiquetas semánticas en HTML y PHP, con estilos coherentes en CSS, que mantienen una estética institucional y moderna. Se incorporaron formularios funcionales, navegación fluida y elementos visuales que mejoran la experiencia del usuario.

1. **Inicio de sesión (Login)**: Este módulo fue desarrollado únicamente para el **front-end**, simulando el proceso de autenticación de usuarios en la plataforma educativa. Aunque no realiza conexión con una base de datos, cumple una función visual y estructural funcional.

Estructura del Código:

- **Contenido HTML estático**: No usa PHP ni conexión a base de datos.
- **Formulario de inicio de sesión**: Ubicado dentro del *main*, contiene:
 1. Campos para correo electrónico (*type="email", required*) y contraseña (*type="password", required, minlength="6"*).
 2. Botón de envío que simula autenticación redirigiendo a Inicio.php mediante método POST.

3. Enlace visual para recuperar contraseña (sin funcionalidad).
4. Enlace para registro de nuevos usuarios que redirige a `cuenta_Re.php`.
5. Estilos embebidos: Uso de CSS dentro de *style* para diseño con Flexbox, bordes redondeados, sombras y efectos hover en botones.
6. HTML semántico: Uso de etiquetas estructurales como *header*, *main*, *footer*, y formularios con *fieldsets*.

Diseño web:

- **Encabezado institucional:** El header contiene dos títulos (h1 y h2) que identifican a la universidad y el propósito del sitio. Se emplea un fondo azul oscuro y sombras para resaltar el texto.
- **Logotipo institucional:** Se incluye el logo transparente y centrada que refuerza la identidad visual del sitio.
- **Formulario de autenticación.**
- **pie de página:** Se incluye un (footer) con información de contacto y derechos reservados, manteniendo la coherencia institucional del diseño.

Navegación

- El botón "**Ingresar**" redirige al archivo `Inicio.php`, simulando el acceso a la plataforma.
- El enlace "**Regístrate aquí**" lleva al módulo `cuenta_Re.php`, donde se encuentra el formulario de registro.
- El enlace "**¿Olvidaste tu contraseña?**" está presente como elemento visual, aunque no tiene funcionalidad activa aún.

2. **Registro de Usuario (cuenta_Registro):** Formulario de Bootstrap que simulan el proceso de registro de un nuevo usuario. Aunque no almacena datos en una base de datos, cuenta con una estructura completa, con entradas funcional que incluye validaciones, diseño institucional coherente y navegación clara.

Estructura del Código:

- **Conexión a la base de datos:** Se incluye un *require 'db.php'*; al inicio del archivo para establecer la conexión con la base de datos mediante PDO. Aunque no se ha implementado aún.
- **Formulario:** Estructurado con múltiples campos organizados en filas y columnas usando Bootstrap, utilizando *required* para validaciones. Cada entrada está contenida dentro de *form-group* y *form-row*, organizados con para lograr una distribución armónica. Contiene los siguientes campos:
 1. **Tipo de identificación:** menú desplegable *select* con opciones como cédula, pasaporte o refugiado
 2. **Número de identificación.**
 3. **Nombre y fecha de nacimiento.**
 4. **Correo electrónico:** Con validación HTML5 (*type="email"*)
 5. **Género:** Botones de opción (**radio**) para Hombre, Mujer y Otro
 6. **País, provincia y cantón.**
 7. **Dirección completa.**
 8. **Teléfonos:** Uno obligatorio y otro opcional.
 9. **Contraseña:** Incluye confirmación con validación (*type="password"*).
 10. **Checkbox** para aceptación de términos y condiciones
 11. **Botón de envío:** Estilizado con Bootstrap (*btn btn-primary*).
 12. **Enlace** para usuarios que ya tienen cuenta (*login.html*).
- **Botón de envío:** Se incluye un botón *btn btn-primary* que dice “Registrarse” con el atributo *action="Inicio.html"* indica que, una vez enviado, el usuario es redirigido a la página de inicio de sesión, simulando el registro.
- El formulario emplea el método POST para simular el envío seguro de datos sensibles, como la contraseña.

Diseño Web:

- **Encabezado institucional:** con el nombre de la universidad y el propósito de la plataforma.

- **Logotipo institucional:** Se incluye una imagen centrada que refuerza la identidad visual del sitio.
- **Contenedor principal:** Dentro del *main*, se emplea un fondo azul institucional con bordes redondeados y sombra para destacar el formulario.
- **Validaciones:** Se utilizan atributos HTML como *required*, *type*, *minlength* y *select* para asegurar que el usuario complete correctamente todos los campos antes de enviar el formulario.
- **Pie de página:** Contiene información de contacto y derechos reservados, manteniendo la estética institucional.

Navegación

- Al hacer clic en “**Registrarse**”, el formulario redirige a `login.html`, simulando el flujo de creación de cuenta y acceso.
- El enlace “**¿Ya tienes una cuenta?**” permite regresar al módulo de inicio de sesión.

3. **Inicio:** Este módulo constituye la página principal del sistema educativo, desarrollado con HTML y PHP.

Estructura del Código:

- **Conexión a la base de datos:** Al inicio del archivo se incluye el archivo `db.php`, lo que permite establecer la conexión con la base de datos mediante PDO. Aunque en esta página no se realizan consultas, la conexión se encuentra disponible por si se requiere en futuras funcionalidades.
- **Configuración de zona horaria:** Se define la zona horaria para Costa Rica (*America/Costa_Rica*) utilizando `date_default_timezone_set()`. Esto asegura que la función del último acceso se muestre correctamente según el país.

- **HTML semántico:** Se estructura con etiquetas semánticas *nav*, *main*, *aside*, *footer* para barra de navegación, contenido principal, panel lateral y pie de página.
- **Contenido dinámico con PHP:** El módulo muestra dinámicamente la hora y fecha del último acceso utilizando `<?= date("d/m/Y - h:i A") ?>`, asegurando que se actualice automáticamente cada vez que se carga la página.
- **JavaScript y Librerías externas:** Se incluye la librería flatpickr para mostrar un calendario interactivo en la barra lateral derecha. Este se inicializa con JavaScript personalizado y se muestra en idioma español (*locale: "es"*), *en formato inline*.
Además, se cargan iconos desde Font Awesome para enriquecer visualmente los elementos como títulos y botones.

Diseño Web

- **Barra de navegación fija:** El nav está posicionado en la parte superior de la página (*position: fixed*), contiene:
 1. Logotipo institucional alineado a la izquierda
 2. Enlaces funcionales a los módulos: Inicio, Cursos, Profesores, Estudiantes.
 3. Un menú desplegable bajo la opción de Servicios, con accesos a Soporte Técnico y Biblioteca
 4. Un botón de “Cerrar Sesión” que redirige al login, simulando una finalización de sesión.
- **Sección de bienvenida:** En el centro se muestra un mensaje de bienvenida junto con un video incrustado de YouTube sobre la plataforma educativa.
- **Sección de avisos y novedades:** Dos bloques tipo tarjeta (*.aviso-card*) muestran avisos importantes y novedades recientes del sistema. Cada uno incluye íconos temáticos.
- **Frase motivacional:** Un bloque con la clase *frase-bienvenida* que muestra una cita inspiradora.

- **Barra lateral derecha:** Implementado mediante un *aside class=lateral-derecho*, este panel está pegado completamente al margen derecho.
Incluye:
 1. **Información del usuario:** Muestra detalles como nombre, correo, rol, ID y último acceso.
 2. **Noticias breves:** Lista de eventos importantes y anuncios institucionales.
 3. **Calendario interactivo:** Presenta un calendario fijo generado con *Flatpickr*.
- **Pie de página:** El footer contiene información de derechos reservados y un enlace de contacto institucional.

Navegación

- Todos los enlaces del menú superior, menos los de servicios son funcionales y redirigen a sus respectivos módulos.
 - El menú Servicios despliega sus opciones adicionales al pasar el cursor, sin necesidad de recargar la página.
 - El botón “Cerrar Sesión” lleva al usuario de vuelta a la página de Login simulando el cierre de sesión.
4. **Estudiantes:** Este módulo presenta la información de los estudiantes en una tabla organizada, con opciones visuales que simulan acciones administrativas básicas como el registro, filtrado y eliminación.

Estructura del Código:

- **Conexión a la base de datos:** Al inicio del archivo se incluye *db.php*, que establece la conexión a la base de datos mediante PDO.
- **Consulta y obtención de datos:** Se realiza una consulta *SQL (SELECT * FROM estudiantes)* para recuperar todos los registros de la tabla estudiantes, los cuales se almacenan en un array.
- **Eliminación de registros:** El código detecta si se ha enviado el parámetro eliminar, y ejecuta un *DELETE* para borrar el estudiante correspondiente. Este proceso está protegido por *try/catch* para manejo de errores.

- **Estilos y presentación:** Se utiliza CSS embebido dentro del archivo. Se da estilos directamente en el *style*.
- **JavaScript:** Se usa para filtrar los estudiantes en tiempo real, ocultando las filas de la tabla según lo ingresado en el campo de búsqueda.
- **SweetAlert:** Se utiliza para confirmar visualmente la eliminación de un estudiante y para mostrar alertas de éxito tras registrar uno nuevo.
- **HTML semántico:** La estructura usa etiquetas como *table*, *thead*, *tbody*, y se inserta dinámicamente cada fila de estudiante mediante *PHP* (*foreach* sobre el array *\$estudiantes*).

Diseño web:

- **Encabezado institucional** con el título “Estudiantes”
- **Barra de navegación fija:** Incluye el logotipo y enlaces a los demás módulos.
- **Campo de búsqueda** con ícono de lupa que permite filtrar estudiantes en tiempo real según lo digitado.
- **Filtros en los encabezados** que permiten ordenar las columnas al dar clic sobre ellas.
- **Tabla de datos** con contenido dinámico proveniente de la base de datos, que muestra: Carnet, Nombre, Apellido, Identificación, Correo, Carrera y Curso.
- **Botones de acción:**
 1. **Eliminar:** Botón funcional, permite borrar estudiantes de la base de datos con confirmación de SweetAlert.
 2. **Editar:** Botón solo visual, aún no se implementa la funcionalidad.
- Botón “Registrar Estudiante” que redirige al formulario de registro (registroEstudiante.php).

Navegación

- Todos los enlaces del menú superior son funcionales y redirigen a sus respectivos módulos.

- El menú desplegable permite acceder a servicios adicionales sin recargar la página.
- El botón “Cerrar Sesión” simula el cierre de sesión redirigiendo al Login.
- El botón “Registrar Estudiante” redirige al formulario de registro de estudiante.

5. Registro de Estudiante: Formulario de Bootstrap que permite el ingreso de nuevos estudiantes al sistema. Presenta un formulario completo, validaciones básicas y diseño institucional coherente con el resto de la plataforma.

Estructura del Código:

- **Conexión a la base de datos:** Al inicio del archivo se incluye `db.php`, que establece la conexión a la base de datos. Esta conexión permite ejecutar las sentencias necesarias para registrar al estudiante.
- **Manejo del formulario:** El archivo detecta si el formulario fue enviado mediante el método POST. En ese caso, se recogen los datos enviados desde el formulario usando `$_POST`, con validación básica para evitar errores por campos vacíos.
- **Validación de datos obligatorios:** Antes de realizar la inserción, el código comprueba que los campos esenciales como tipo de identificación, número, carnet, nombre, apellido, correo, carrera, curso y género hayan sido completados. La fecha de nacimiento es opcional y puede almacenarse como *NULL*.
- **Consulta SQL (INSERT):** Se construye una sentencia *INSERT INTO estudiantes* utilizando marcadores nombrados (*:nombreParametro*) para evitar inyecciones. La sentencia se prepara con *prepare* y se enlazan los valores con *bindParam* o *bindValue* (en el caso de una fecha nula).
- **Ejecución y redirección:** Si la sentencia se ejecuta correctamente, el usuario es redirigido devuelta a *estudiantes.php* con el parámetro *registrado=true*, lo que se usa para mostrar un mensaje de confirmación. En caso de fallo, se guarda un mensaje de error en la variable *\$error*.

Diseño Web:

- **Encabezado institucional:** Incluye el nombre de la universidad y el propósito general del sistema.
- **Logotipo institucional**
- **Formulario:** Organizado en una estructura de fieldset con filas y columnas utilizando clases de Bootstrap como form-row y form-group. Este formulario contiene los siguientes campos:

1. **Tipo y número de identificación.**
2. **Carnet** con validación de formato `pattern= [A-Z]{2}[0-9]{7}`
3. **Nombre y Apellido.**
4. **Fecha de nacimiento** (`type="date"`).
5. **Correo electrónico:** cuenta con validación HTML5 (`type="email"`)
6. **Género:** botones de opción (radio) para Hombre, Mujer y Otro
7. **Carrera y curso** ambos seleccionables mediante listas desplegables (`select`).
8. Botones de acción: “Registrar” y “Cancelar”.

- **Estilos CSS embebidos:** Todo el estilo se define dentro del *style* en el mismo archivo.
- **Uso de Bootstrap:** Se utiliza Bootstrap a través de CDN para la estructura y diseño responsivo del formulario y sus componentes.
- **HTML semántico:** Se emplean etiquetas como *form*, *fieldset*, *header*, *main* y *footer* para organizar adecuadamente la información.
- **Pie de página:** Muestra derechos reservados y un correo de contacto con enlace funcional.

Navegación

- El botón “**Registrar**” envía los datos del formulario y registra al estudiante en la base de datos.
 - El botón “**Cancelar**” redirige al módulo de estudiantes sin guardar información.
6. **Profesores:** Este módulo presenta la información de los profesores en una tabla organizada, con opciones visuales que simulan acciones administrativas básicas como el registro, detalle, filtrado y eliminación.

Estructura del Código:

- **Conexión a la base de datos:** Al inicio del archivo se incluye `db.php`, que establece la conexión a la base de datos mediante PDO.
- **Consulta y obtención de datos:** Se realiza una consulta *SQL* (*SELECT * FROM profesores*) para recuperar todos los registros de la tabla profesores, los cuales se almacenan en un array.
- **Eliminación de registros:** El código detecta si se ha enviado el parámetro eliminar, y ejecuta un *DELETE* para borrar el estudiante correspondiente. Este proceso está protegido por *try/catch* para manejo de errores.
- **Modal de detalle:** El botón "**Detalle**" activa un **modal** que muestra información ampliada del profesor. Este modal se genera dinámicamente con PHP e incluye campos como nombre, correo, carrera, teléfono y horario.
- **Estilos y presentación:** Se utiliza CSS embebido dentro del archivo. Se da estilos directamente en el *style*.
- **JavaScript:** Se usa para filtrar a los profesores en tiempo real, ocultando las filas de la tabla según lo ingresado en el campo de búsqueda.
- **SweetAlert:** Se utiliza para confirmar visualmente la eliminación de un estudiante y para mostrar alertas de éxito tras registrar uno nuevo.
- **HTML semántico:** La estructura usa etiquetas como *table*, *thead*, *tbody*, y se inserta dinámicamente cada fila de estudiante mediante *PHP* (*foreach sobre el array \$estudiantes*).

Diseño web:

- **Encabezado institucional** con el título “Profesores”
- **Barra de navegación fija:** Incluye el logotipo y enlaces a los demás módulos.
- **Campo de búsqueda** con ícono de lupa que permite filtrar profesores en tiempo real según lo digitado.
- **Filtros en los encabezados** que permiten ordenar las columnas al dar clic sobre ellas.

- **Tabla de datos** con contenido dinámico proveniente de la base de datos, que muestra: Nombre, Apellido, Curso Asignado y Correo.
- **Botones de acción:**
 1. **Detalle:** Abre un **modal** con información ampliada del profesor.
 2. **Eliminar:** Botón funcional, permite borrar profesores de la base de datos con confirmación de SweetAlert.
 3. **Editar:** Botón solo visual, aún no se implementa la funcionalidad.
- Botón “Registrar Profesor” que redirige al formulario de registro (registroProfesor.php).

Navegación

- Todos los enlaces del menú superior son funcionales y redirigen a sus respectivos módulos.
- El menú desplegable permite acceder a servicios adicionales sin recargar la página.
- El botón “Cerrar Sesión” simula el cierre de sesión redirigiendo al Login.
- El botón “Registrar Profesor” redirige al formulario de registro de profesor.

7. **Registro de Profesor:** Formulario de Bootstrap que permite el ingreso de nuevos profesores al sistema. Presenta un formulario completo, validaciones básicas y diseño institucional coherente con el resto de la plataforma.

Estructura del Código:

- **Conexión a la base de datos:** Al inicio del archivo se incluye db.php, que establece la conexión a la base de datos. Esta conexión permite ejecutar las sentencias necesarias para el registro.
- **Manejo del formulario:** El archivo detecta si el formulario fue enviado mediante el método POST. En ese caso, se recogen los datos enviados desde el formulario usando `$_POST`, con validación básica para evitar errores por campos vacíos.
- **Validación de datos obligatorios:** Se comprueba que los campos obligatorios estén completos: tipo de identificación, identificación,

nombre, apellido, correo, teléfono, curso, género y horario. El campo fechaNacimiento es opcional, y si está vacío, se establece como *NULL*.

- **Inserción en la base de datos:** Si la validación es exitosa, se construye una sentencia *INSERT INTO profesores* utilizando una sentencia preparada *bindParam*. Se insertan todos los campos del formulario en la tabla profesores.
- **Ejecución y redirección:** Si la sentencia se ejecuta correctamente, el usuario es redirigido devuelta a *profesores.php* con el parámetro *registrado=true*, lo que se usa para mostrar un mensaje de confirmación. En caso de fallo, se guarda un mensaje de error en la variable *\$error*.

Diseño Web:

- **Encabezado institucional:** Incluye el nombre de la universidad y el propósito general del sistema.
- **Logotipo institucional**
- **Formulario:** Este formulario contiene los siguientes campos:
 1. **Tipo de identificación:** Menú desplegable con opciones como cédula nacional, jurídica, pasaporte, etc.
 2. **Número de identificación.**
 3. **Nombre y Apellido.**
 4. **Fecha de nacimiento** (*type="date"*) opcional.
 5. **Correo electrónico:** cuenta con validación HTML5 (*type="email"*)
 6. **Teléfono**
 7. **Curso asignado:** Selector con cursos predefinidos.
 8. **Horario:** Campo de texto para indicar el horario asignado.
 9. **Género:** Selección mediante botones de opción (radio) para Hombre, Mujer u Otro.
- **Botones:**
 1. **“Registrar”** para enviar el formulario.

2. “Cancelar” redirige a la página *profesores.php*.

- **Estilos CSS embebidos:** Todo el estilo se define dentro del *style* en el mismo archivo.
- **Uso de Bootstrap:** Se utiliza Bootstrap a través de CDN para la estructura y diseño responsivo del formulario y sus componentes.
- **Pie de página:** Muestra derechos reservados y un correo de contacto con enlace funcional.

Navegación

- El botón “**Registrar**” envía los datos del formulario y registra al profesor en la base de datos.
- El botón “**Cancelar**” redirige al módulo de profesor sin guardar información.

8. **Cursos:** Este módulo presenta la información de los cursos en una tabla organizada, con opciones visuales que simulan acciones administrativas básicas como el registro, filtrado y eliminación.

Estructura del Código:

- **Conexión a la base de datos:** Al inicio del archivo se incluye *db.php*, que establece la conexión a la base de datos mediante PDO.
- **Consulta y obtención de datos:** Se realiza una consulta *SQL (SELECT * FROM cursos)* para recuperar todos los registros de la tabla de cursos, los cuales se almacenan en un array.
- **Eliminación de registros:** El código detecta si se ha enviado el parámetro eliminar, y ejecuta un *DELETE* para borrar el estudiante correspondiente. Este proceso está protegido por *try/catch* para manejo de errores. Esto se hace dentro de un bloque *if* para verificar su existencia y se redirige con header para mostrar una alerta usando SweetAlert.
- **Estilos y presentación:** Se utiliza CSS embebido dentro del archivo. Se da estilos directamente en el *style*.

- **JavaScript:** Se usa para filtrar los cursos en tiempo real, ocultando las filas de la tabla según lo ingresado en el campo de búsqueda.
- **SweetAlert:** Se utiliza para confirmar visualmente la eliminación de un estudiante y para mostrar alertas de éxito tras registrar uno nuevo.
- **HTML semántico:** La estructura usa etiquetas como *table*, *thead*, *tbody*, y se inserta dinámicamente cada fila de estudiante mediante *PHP* (*foreach* sobre el array *\$estudiantes*).

Diseño web:

- **Encabezado institucional** con el título “Estudiantes”
- **Barra de navegación fija:** Incluye el logotipo y enlaces a los demás módulos.
- **Campo de búsqueda** con ícono de lupa que permite filtrar los cursos en tiempo real según lo digitado.
- **Filtros en los encabezados** que permiten ordenar las columnas al dar clic sobre ellas.
- **Tabla de datos** con contenido dinámico proveniente de la base de datos, que muestra: Código, Nombre, Requisitos, Créditos, Profesor Asignado y Horario.
- **Botones de acción:**
 1. **Eliminar:** Botón funcional, permite borrar cursos de la base de datos con confirmación de SweetAlert.
 2. **Editar:** Botón solo visual, aún no se implementa la funcionalidad.
- Botón “Registrar Curso” que redirige al formulario de registro (registroCurso.php).

Navegación

- Todos los enlaces del menú superior son funcionales y redirigen a sus respectivos módulos.
- El menú desplegable permite acceder a servicios adicionales sin recargar la página.
- El botón “Cerrar Sesión” simula el cierre de sesión redirigiendo al Login.

9. **Registro de Curso:** Formulario de Bootstrap que permite el ingreso de nuevos cursos al sistema. Presenta un formulario completo, validaciones básicas y diseño institucional coherente con el resto de la plataforma.

Estructura del Código:

- **Conexión a la base de datos:** Al inicio del archivo se incluye db.php, que establece la conexión a la base de datos. Esta conexión permite ejecutar las sentencias necesarias para registrar el curso.
- **Manejo del formulario:** El archivo detecta si el formulario fue enviado mediante el método POST. En ese caso, se recogen los datos enviados por el usuario: código, nombre, requisitos, créditos, profesor y horario.
- **Validación de datos obligatorios:** Se valida que los campos obligatorios código, nombre, créditos, profesor y horario, no estén vacíos. El campo código utiliza una expresión regular en HTML (*pattern="[A-Z]{2,3}-[0-9]{2,3}"*) para validar su formato y los créditos deben estar entre 1 y 6.
- **Inserción en la base de datos:** Si los datos están completos, se prepara una sentencia *INSERT INTO cursos* y se ejecuta usando parámetros enlazados (*bindParam*) para prevenir inyecciones SQL.
- **Ejecución y redirección:** Si la sentencia se ejecuta correctamente, el usuario es redirigido devuelta a *cursos.php* con el parámetro *registrado=true*, lo que se usa para mostrar un mensaje de confirmación. En caso de fallo, se guarda un mensaje de error en la variable *\$error*.
- **Estilos CSS embebidos:** Todo el estilo se define dentro del *style* en el mismo archivo.
- **Uso de Bootstrap:** Se utiliza Bootstrap a través de CDN para la estructura y diseño responsivo del formulario y sus componentes.
- **HTML semántico**
- **Pie de página:** Muestra derechos reservados y un correo de contacto con enlace funcional.

Diseño Web:

- **Encabezado institucional:** con el nombre de la universidad y propósito de la plataforma
- **Logotipo institucional**
- **Formulario:** Estructurado con múltiples campos organizados en filas y columnas usando Bootstrap. Contiene:
 1. **Código del curso** con validación de formato.
 2. **Nombre del curso.**
 3. **Requisitos**, campo opcional.
 4. **Créditos**, entre 1 y 6
 5. **Profesor asignado**
 6. **Horario**, formato libre con ejemplo sugerido.
 7. Botón **“Registrar”** para enviar los datos.
 8. Botón **“Cancelar”**, que redirige a la vista de cursos.

Navegación

- El botón **“Registrar”** envía los datos del formulario y registra el curso en la base de datos.
- El botón **“Cancelar”** redirige al módulo de cursos sin guardar información.

Diagrama UML

Se presenta un diagrama conceptual que representa las clases principales del sistema. Este modelo permite visualizar de forma simple la estructura general y las conexiones entre los módulos implementados. *Ver Anexo 10.*

Las clases son:

- **Usuario**
 - ✓ Atributos: usuario, contraseña
 - ✓ Gestiona el acceso y navegación entre módulos
- **Estudiante**
 - ✓ Atributos: id Estudiante, nombre, apellido, correo, carnet, carrera
 - ✓ Se relaciona con la clase Curso mediante matrícula
 - ✓ Es gestionado por Usuario
- **Profesor**
 - ✓ Atributos: nombre, apellido, curso Asignado, correo, teléfono, horario Atención, título.
 - ✓ Se relaciona con Curso como responsable de impartir el curso.
 - ✓ Es gestionado por Usuario.
- **Curso**
 - ✓ Atributos: id, nombre, código, créditos, profesor Asignado, requisitos, horario
 - ✓ Se relaciona con Estudiante (matriculado) y Profesor (asignado)
 - ✓ Es gestionado por Usuario.

Conexiones

- Usuario actúa como administrador principal del sistema
- Estudiante y Profesor son entidades gestionadas que interactúan con Curso
- Curso funciona como punto de encuentro entre docentes y estudiantes
- **Relación: Usuario -Estudiante / Profesor / Curso:**
 - ✓ **1..1**
 - ✓ Un usuario accede a cada módulo de forma individual
 - ✓ Controla la navegación y gestión de datos

- **Relación: Estudiante - Curso:**
 - ✓ 1..1
 - ✓ Cada estudiante está matriculado en un único curso
 - ✓ La relación es directa y se refleja en la tabla de estudiante
- **Relación: Curso - Estudiante:**
 - ✓ 0..*
 - ✓ Un curso puede tener cero o muchos estudiantes
- **Relación: Profesor - Curso:**
 - ✓ 1..*
 - ✓ Un profesor puede impartir uno o varios cursos
 - ✓ Se visualiza en la tabla de profesores.
- **Relación: Curso - Profesor :**
 - ✓ 1..1
 - ✓ Cada curso tiene un único profesor asignado
 - ✓ La relación es directa y se muestra en la tabla de cursos

Conclusiones

Durante este proyecto se consolidó la estructura del sistema educativo, integrando módulos funcionales con navegación entre páginas, formularios interactivos y conexión real a una base de datos utilizando PHP. Se aplicaron buenas prácticas en HTML semántico, estilos CSS institucionales y validaciones básicas que refuerzan la experiencia del usuario.

Cada módulo fue diseñado con claridad, permitiendo visualizar datos, registrar información y ejecutar acciones administrativas reales como eliminación y consulta. Se incorporaron elementos dinámicos mediante JavaScript y alertas visuales, mejorando la interacción y facilitando el uso del sistema.

Este trabajo representa una base firme para seguir construyendo más adelante. Gracias a la forma en la que está organizado, será posible incorporar mejoras o nuevas funciones cuando se necesiten, sin tener que empezar desde cero. Más allá del código, este avance refleja el crecimiento técnico y la capacidad de aplicar lo aprendido en un entorno profesional.

Anexos

Anexo 1. Login

UNIVERSIDAD CENTRAL

PLATAFORMA EDUCATIVA

UC

UNIVERSIDAD
CENTRAL

Inicio de Sesión

Correo Electrónico:

Contraseña:

¿Olvidaste tu contraseña?

Ingresar

¿No tienes una cuenta? [Regístrate aquí](#)

© 2025 Plataforma Educativa Universidad Central

Contacto: [Milagros Hernández](#)

Anexo 2. cuenta_Re

UNIVERSIDAD CENTRAL

PLATAFORMA EDUCATIVA

UC

UNIVERSIDAD
CENTRAL

Registro de Usuario

Tipo de Identificación

Seleccione el tipo de identificación

Número de Identificación

Nombre

Fecha de Nacimiento

dd/mm/aaaa

Correo Electrónico

Género

☐ Hombre

☐ Mujer

☐ Otro

País de Origen

Provincia

Seleccione

Cantón

Dirección

Teléfono 1

Teléfono 2

Contraseña

Confirmar Contraseña

☐ Acepto los términos y condiciones.

Registrarse

¿Ya tienes una cuenta? [Iniciar Sesión](#)

Anexo 3.Inicio

UC

UNIVERSIDAD

CENTRAL

Inicio

Cursos

Profesores

Estudiantes

Servicios

Cerrar Sesión

Bienvenido al sistema educativo

Explorá la plataforma educativa

UC

Quando matriculás, empezás la carrera de tu vida

Mirar en

YouTube

Ver más

Ver menos

Compartir

Avisos Importantes

Mantenimiento web programado: 3 de agosto

Taller de investigación académica: 7 de agosto

Usuario

Nombre: Usuario 1

Correo: Admin@ejemplo.com

Rol: Administrador

ID: 1023

Último acceso: 30/07/2025 - 02:42 PM

Noticias

Capacitación docente el 10 de agosto.

Mejoras en la plataforma de gestión académica.

Actualización de horarios para el próximo cuatrimestre.

Nuevo curso de habilidades blandas disponible.

Calendario

Anexo 4.Curso

UC

UNIVERSIDAD

CENTRAL

Inicio

Cursos

Profesores

Estudiantes

Servicios

Cerrar Sesión

CURSOS

Buscar

CÓDIGO	NOMBRE	REQUISITOS	CRÉDITOS	PROFESOR ASIGNADO	HORARIO	ACCIONES
INF-101	Introducción a la informática		3	Carlos Esquivel Bolaños	L 6:00 PM A 9 PM	<div>Editar</div> <div>Eliminar</div>
ARQ-112	Diseño Arquitectónico	INQ	4	Mateo Ovando Salazar	S 10:00 AM a 1:00 PM	<div>Editar</div> <div>Eliminar</div>
II-60	Sistemas Computacionales	II-21	4	Karol Hernández Serrano	K 3:00 PM a 6:00 PM	<div>Editar</div> <div>Eliminar</div>
EI-48	Arquitectura de Computadores	EI-21	4	José Jacamo Tellez	L 6:00 PM a 9:00 PM	<div>Editar</div> <div>Eliminar</div>
IN-20	Inglés II	IN-10	3	Karina Zamora Tercero	V 11:00 AM a 1:00 PM	<div>Editar</div> <div>Eliminar</div>
AE-10	Administración I	EG-10	3	Marcela Barahona Fernández	M 3:00 PM a 6:00 PM	<div>Editar</div> <div>Eliminar</div>

+ Registrar Curso

Anexo 5.registroCurso

UNIVERSIDAD CENTRAL
PLATAFORMA EDUCATIVA

UC
UNIVERSIDAD
CENTRAL

Registro de Curso

Código

Nombre del curso

Requisitos

Créditos

Profesor asignado

Horario

Ej: L 6:00 pm a 9:00 pm

Registrar

Cancelar

Anexo 6.Profesores

UNIVERSIDAD CENTRAL					
Inicio Cursos Profesores Estudiantes Servicios Cerrar Sesión					
PROFESORES					
Buscar					
NOMBRE	APELLIDO	CURSO ASIGNADO	CORREO	ACCIONES	
Marcela	Barahona Fernández	Administración I	mbarahonafa@uc.cr	Detalle	Editar Eliminar
Carlos	Esquivel Bolaños	Introducción a la Informática	cesquivelb@uc.cr	Detalle	Editar Eliminar
Karol	Hernández Serrano	Sistemas Computacionales	khernandezs@uc.cr	Detalle	Editar Eliminar
José	Jacamo Tellez	Arquitectura de Computadores	jtellezj@uc.cr	Detalle	Editar Eliminar
Mateo	Ovando Salazar	Diseño Arquitectónico	movandos@uc.cr	Detalle	Editar Eliminar
Karina	Zamora Tercero	Inglés II	kzamorat@uc.cr	Detalle	Editar Eliminar
Registrar profesor					

Anexo 7 registroProfesor

UNIVERSIDAD CENTRAL

PLATAFORMA EDUCATIVA

UC

UNIVERSIDAD
CENTRAL

Registro de Profesor

Tipo de Identificación

Seleccione el tipo de identificación

Número de Identificación

Nombre

Apellido

Fecha de Nacimiento

dd/mm/aaaa

Correo Electrónico

Teléfono

Curso Asignado

Seleccione el curso

Horario

Ej: L 6:00 pm a 9:00 pm

Género

☐ Hombre ☐ Mujer ☐ Otro

Registrar

Cancelar

Anexo 8 estudiantes

UC

UNIVERSIDAD
CENTRAL

Inicio

Cursos

Profesores

Estudiantes

Servicios

Cerrar Sesión

ESTUDIANTES

Buscar

CARNET	NOMBRE	APELLIDO	IDENTIFICACIÓN	CORREO	CARRERA	CURSO	ACCIONES
SM1975364	María Fernanda	Angulo Pérez	118744597	MperezA@uc.cr	Administración de Empresas	Administración I	<div>EditarEliminar</div>
AL2175413	Juan Carlos	Gómez Rodríguez	118744598	Jgomezp@uc.cr	Ingeniería Informática	Sistemas Computacionales	<div>EditarEliminar</div>
SM1025785	Ana María	Pereira Sandoval	158197648871	Apereiras@uc.cr	Administración de Empresas	Arquitectura de Computadores	<div>EditarEliminar</div>
SM1239753	Carlos Andrés	Ramírez Torres	118744601	Cramirez@uc.cr	Contaduría	Inglés II	<div>EditarEliminar</div>
SM1987342	Jimena Patricia	Zuñiga Primero	119744602	Jzunigap@uc.cr	Ingeniería Informática	Introducción a la Informática	<div>EditarEliminar</div>

+ Registrar Estudiante

Anexo 9 registroEstudiante

UNIVERSIDAD CENTRAL

PLATAFORMA EDUCATIVA

UC

UNIVERSIDAD
CENTRAL

Registro de Estudiante

Tipo de Identificación

Seleccione el tipo de identificación

Número de Identificación

Carnet

Nombre

Apellido

Fecha de Nacimiento

dd/mm/aaaa

Correo Electrónico

Carrera

Seleccione la carrera

Curso

Seleccione el curso

Género

☐ Hombre

☐ Mujer

☐ Otro

Registrar

Cancelar

Anexo 10. Diagrama UML

