

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 desde el diseño visual.

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 cinco módulos principales: Login, Estudiantes, Profesores, Cursos y Registro. Cada sección permite visualizar datos en tablas, registrar nuevos elementos mediante formularios con validación HTML5, y navegar fluidamente entre páginas

Aunque aún no se emplea una base de datos real, se simulan procesos como el registro y la eliminación de datos mediante redirecciones y alertas interactivas. 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 y conexión a bases de datos reales.

**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 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.
* Utilizar archivos php para simular la estructura dinámica del sistema y la interacción entre módulos.
* Permitir la navegación entre módulos mediante enlaces funcionales.
* Representar la lógica del sistema mediante un diagrama UML

**Estructura del Proyecto**

A continuación, se describe la organización de carpetas y archivos que componen el sistema.

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 su futuros 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, simulando una estructura dinámica. Solo el Login permanece como archivo HTML. Los archivos utilizan 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 actual del proyecto:

1. **Carpeta *Pages***: Contiene los archivos principales del sitio web, uno por cada módulo implementado:

* login.html: página de inicio de sesión simulado. *Ver anexo 1*
* cuenta\_Re.php: página adicional para un formulario de registro de usuario. *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 anexo4*
* registroCurso.php: formulario para registrar cursos. *Ver anexo 5*
* profesores. Php: módulo de gestión de profesores. *Ver anexo 6*
* registroProfesor.php: formulario para registrar profesores. *Ver anexo 7*
* estudiantes. Php: módulo de gestión de estudiantes. *Ver anexo 8*
* registroEstudiante.php: formulario para registrar estudiantes. *Ver anexo9*

1. **Carpeta** ***images***: Contiene los recursos visuales utilizados por las páginas, además del 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.

1. **Index**: Archivo raíz.

**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 front-end, simulando el proceso de autenticación de usuarios en la plataforma educativa. Aunque no conecta con una base de datos, es funcional.

**Estructura del Código:**

* **Encabezado institucional:** El header contiene dos títulos (h1 y h2) que identifican la universidad y el propósito de la plataforma. Se utiliza 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**: Dentro del main se encuentra un form con los siguientes elementos:

1. Campos para correo electrónico y contraseña, utilizando los atributos type="email", required y type="password" y minlength="6" para validar el ingreso.
2. Botón de envío que redirige al módulo Inicio usando el método GET.
3. Enlace para recuperar contraseña (¿Olvidaste tu contraseña?).
4. Enlace para registrarse si no se tiene cuenta (Regístrate aquí).

* **Estilos visuales**: El diseño utiliza Flexbox para centrar el contenido verticalmente. Los campos tienen bordes redondeados, sombras suaves y efectos al enfocar, focus. El botón de ingreso cambia de color al pasar el cursor, hover, para mejorar la interacción.
* **Pie de página**:  
  El incluye información de contacto y derechos reservados, manteniendo la coherencia institucional.

**Navegación**

* Al hacer clic en **"Ingresar"**, el formulario 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 completo.
* El enlace **"¿Olvidaste tu contraseña?"** está presente como elemento visual, aunque no tiene funcionalidad activa aún.

**Observaciones técnicas**

* Se utiliza method="post" para simular el envío de datos, para mayor seguridad.
* El diseño se adapta a pantallas pequeñas gracias al uso de Flexbox.
* La estética mantiene coherencia con el resto del sitio: colores institucionales en tonos azules y blanco, tipografía Poppins, y estructura clara.

1. **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, presenta una estructura completa, con entradas funcional que incluye validaciones, diseño institucional y navegación clara.

**Estructura del Código:**

* **Encabezado institucional:** El header contiene dos títulos que identifican la universidad. Se utiliza la fuente Poppins para mantener la coherencia.
* **Logotipo institucional:** Se incluye una imagen centrada que refuerza la identidad visual del sitio.
* **Encabezado institucional:** con el nombre de la universidad y el propósito de la plataforma.
* **Contenedor principal:** El main tiene el fondo azul institucional, bordes redondeados y sombra para destacar el formulario. Dentro de él, el formulario está contenido en un blanco con estilo tipo tarjeta (registro-container).
* **Formulario**: Estructurado con múltiples campos organizados en filas y columnas usando Bootstrap. Contiene:
  + 1. **Tipo de identificación:** menú desplegable con opciones como cédula, pasaporte o refugiado
    2. **Número de identificación.**
    3. **Nombre y fecha de nacimiento.**
    4. **Correo electrónico**: cuenta 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**: Cuenta con confirmación con validación (type="password").
    10. **Checkbox** de 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).
* **Validaciones**: Se utilizan atributos como required, type, minlength y select para asegurar que el usuario complete correctamente el formulario antes de enviarlo.
* **Pie de página:** El incluye información de contacto y derechos reservados, manteniendo la estética institucional.

**Navegación**

* Al hacer clic en “Registrarse”, el formulario redirige al archivo login.html, simulando el flujo de creación de cuenta y acceso.
* El enlace “¿Ya tienes una cuenta?” permite volver al módulo de inicio de sesión.
* El formulario utiliza el método POST para simular el envío de datos sensibles, como contraseñas.

**Observaciones técnicas**

* El diseño se adapta a pantallas pequeñas sin perder legibilidad.
* El formulario está contenido dentro de un main con fondo azul y bordes redondeados.
* Los campos están agrupados con fieldset y distribuidos usando clases de Bootstrap como form-row, form-group, y col-md-\*

1. **Inicio**: Este módulo fue desarrollado en PHP y HTML como la página principal de la plataforma educativa.

**Estructura del Código:**

* **Barra de navegación fija**: - El nav está posicionado en la parte superior de la página (position: fixed) y contiene:
  + 1. Logotipo institucional alineado a la izquierda
    2. Enlaces funcionales a los módulos: Inicio, Cursos, Profesores, Estudiantes.
    3. Menú desplegable Servicios, con opciones como Soporte Técnico y Biblioteca
    4. Enlace para cerrar sesión
* **Encabezado principal:** Dentro del main, se los títulos que dan la bienvenida al usuario.
* **Video institucional:** Se incrusta un video de YouTube dentro de un div class="video-container", centrado y con dimensiones adaptadas.
* **Sección de avisos y novedades:** Utiliza la clase de avisos-novedades para presentar información relevante en formato de tarjetas.
* Pie de página: El footer contiene información de derechos reservados y un enlace de contacto institucional.

**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.

**Observaciones técnicas**

* Los estilos están embebidos en el mismo archivo mediante la etiqueta Style.
* Se incluye Font Awesome desde CDN para utilizar íconos visuales en los avisos
* La estructura semántica utiliza etiquetas como main, nav, section, article y footer.

1. **Estudiantes**: Este módulo no solo presenta los datos de estudiantes en una tabla visualmente clara, sino que también incorpora elementos interactivos que simulan funcionalidades administrativas como búsqueda, edición, eliminación y registro.

**Estructura del Código:**

* **Encabezado institucional** con el título “Estudiantes”
* **Barra de navegación fija** con logotipo y enlaces a los demás módulos.
* **Buscador visual** con ícono de lupa para filtrar estudiantes en tiempo real
* **Tabla de datos** con información simulada: Carnet, Nombre, Apellido, Identificación, Correo, Carrera y Curso
* **Botones de acción** para editar y eliminar, con íconos de Font Awesome, no funcionales por el momento.
* **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 de registro redirige al formulario de registro de estudiante.

**Funcionalidad simulada**

* **Alerta de registro exitoso** con SweetAlert, activada por el parámetro? registrado=true.
* **Confirmación de eliminación** con alerta visual que simula el proceso administrativo.
* **Filtro dinámico** que oculta filas según el texto ingresado en el buscador.

**Aspectos técnicos**

* Estructura semántica con table, thead, tbody y button.
* Uso de JavaScript para interacciones visuales.

1. **Registro de Estudiante:** Formulario de Bootstrap que simulan 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:**

* **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. **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.**
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 con menú desplegable)

**Navegación**

* Al hacer clic en “Registrar Estudiante”, el formulario redirige al formulario correspondiente.
* Al darle “Registrar” o “cancelar” redirige al módulo estudiante.

**Observaciones técnicas**

* El diseño se adapta a pantallas pequeñas sin perder legibilidad.
* El formulario está contenido dentro de un main con fondo azul y bordes redondeados.
* Los campos están agrupados con fieldset y distribuidos usando clases de Bootstrap como form-row, form-group, y col-md-\*

1. **Profesores**: Este módulo fue desarrollado en PHP y HTML, simulando la visualización de docentes asignados en la plataforma. Presenta una tabla clara, navegación institucional y botones de acción.

**Estructura del Código:**

* **Encabezado institucional** con el título “Profesores”
* **Barra de navegación fija** con logotipo y enlaces a los demás módulos.
* **Buscador visual** con ícono de lupa para filtrar profesores en tiempo real.
* **Tabla de datos con información simulada:** Nombre, Apellido, Curso Asignado, Correo y Horario de Atención.
* **Botones de acción para ver detalles, editar y eliminar**, con íconos de Font Awesome.
* **Botón “Registrar Profesor”** que redirige al formulario correspondiente (registroProfesor.php).

**Funcionalidad simulada**

* **Alerta de registro exitoso** con SweetAlert, activada por el parámetro ?registrado=true.
* **Confirmación de eliminación** con alerta visual que simula el proceso administrativo
* **Filtro dinámico** que oculta filas según el texto ingresado en el buscado
* **Card emergente de detalle** que muestra información adicional del profesor:

1. Título profesional según el curso
2. Correo, teléfono y horario de atención
3. Íconos visuales para cada dato
4. Botón para cerrar el card.

**Aspectos técnicos**

* Estilos embebidos
* Uso de JavaScript para interacciones visuales.
* SweetAlert desde CDN para alertas modernas

1. **Registro de Profesor:** Formulario de Bootstrap que simulan 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:**

* **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. **Tipo de identificación y número de identificación.**
2. **Nombre y Apellido.**
3. **Fecha de nacimiento.**
4. **Teléfono**
5. **Correo electrónico**: cuenta con validación HTML5 (type="email")
6. **Curso Asignado**.
7. **Género.**

**Navegación**

* Al hacer clic en “Registrar Profesor”, el formulario redirige al formulario correspondiente.
* Al darle “Registrar” o “cancelar” redirige al módulo profesor.

**Observaciones técnicas**

* El diseño se adapta a pantallas pequeñas sin perder legibilidad.
* El formulario está contenido dentro de un main con fondo azul y bordes redondeados.
* Los campos están agrupados con fieldset y distribuidos usando clases de Bootstrap como form-row, form-group, y col-md-\*

1. **Cursos**: Este módulo fue desarrollado en PHP y HTML, simulando la visualización de cursos. Presenta una tabla clara, navegación institucional y botones de acción.

**Estructura del Código:**

* **Encabezado institucional** con el título “Cursos”
* **Barra de navegación fija** con logotipo y enlaces a los demás módulos.
* **Buscador visual** con ícono de lupa para filtrar profesores en tiempo real.
* **Tabla de datos con información simulada:** Nombre, Apellido, Curso Asignado, Correo y Horario de Atención.
* **Botones de acción para editar y eliminar**, con íconos de Font Awesome.
* **Botón “Registrar Curso”** que redirige al formulario correspondiente (registroCurso.php).

**Funcionalidad simulada**

* **Alerta de registro exitoso** con SweetAlert, activada por el parámetro ?registrado=true.
* **Confirmación de eliminación** con alerta visual que simula el proceso administrativo.
* **Filtro dinámico** que oculta filas según el texto ingresado en el buscado
* **Carga dinámica de datos** mediante un arreglo PHP ($cursos) que se recorre con foreach para generar las filas de la tabla.

**Aspectos técnicos**

* Estilos embebidos
* Uso de JavaScript para interacciones visuales.
* SweetAlert desde CDN para alertas modernas

1. **Registro de Curso:** Formulario de Bootstrap que simulan el ingreso de cursos. Presenta un formulario completo, validaciones básicas y diseño institucional coherente con el resto de la plataforma.

**Estructura del Código:**

* **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 (pattern="[A-Z]{2,3}-[0-9]{2,3}")
    2. **Nombre del curso** con mínimo de 3 caracteres.
    3. **Requisitos,** campo opcional.
    4. **Créditos,** entre 1 y 6
    5. **Profesor asignado**
    6. **Horario,** formato libre con ejemplo sugerido

**Navegación**

* Al hacer clic en “Registrar Curso”, el formulario redirige al formulario correspondiente.
* Al darle “Registrar” o “cancelar” redirige al módulo curso.

**Observaciones técnicas**

* El diseño se adapta a pantallas pequeñas sin perder legibilidad.
* El formulario está contenido dentro de un main con fondo azul y bordes redondeados.
* Los campos están agrupados con fieldset y distribuidos usando clases de Bootstrap como form-row, form-group, y col-md-\*

**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 fluida, formularios completos y simulaciones interactivas. 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 simular acciones administrativas como edición, eliminación y búsqueda. Aunque el sistema sigue siendo estático, se incorporaron elementos dinámicos mediante JavaScript y alertas visuales que enriquecen la interfaz.

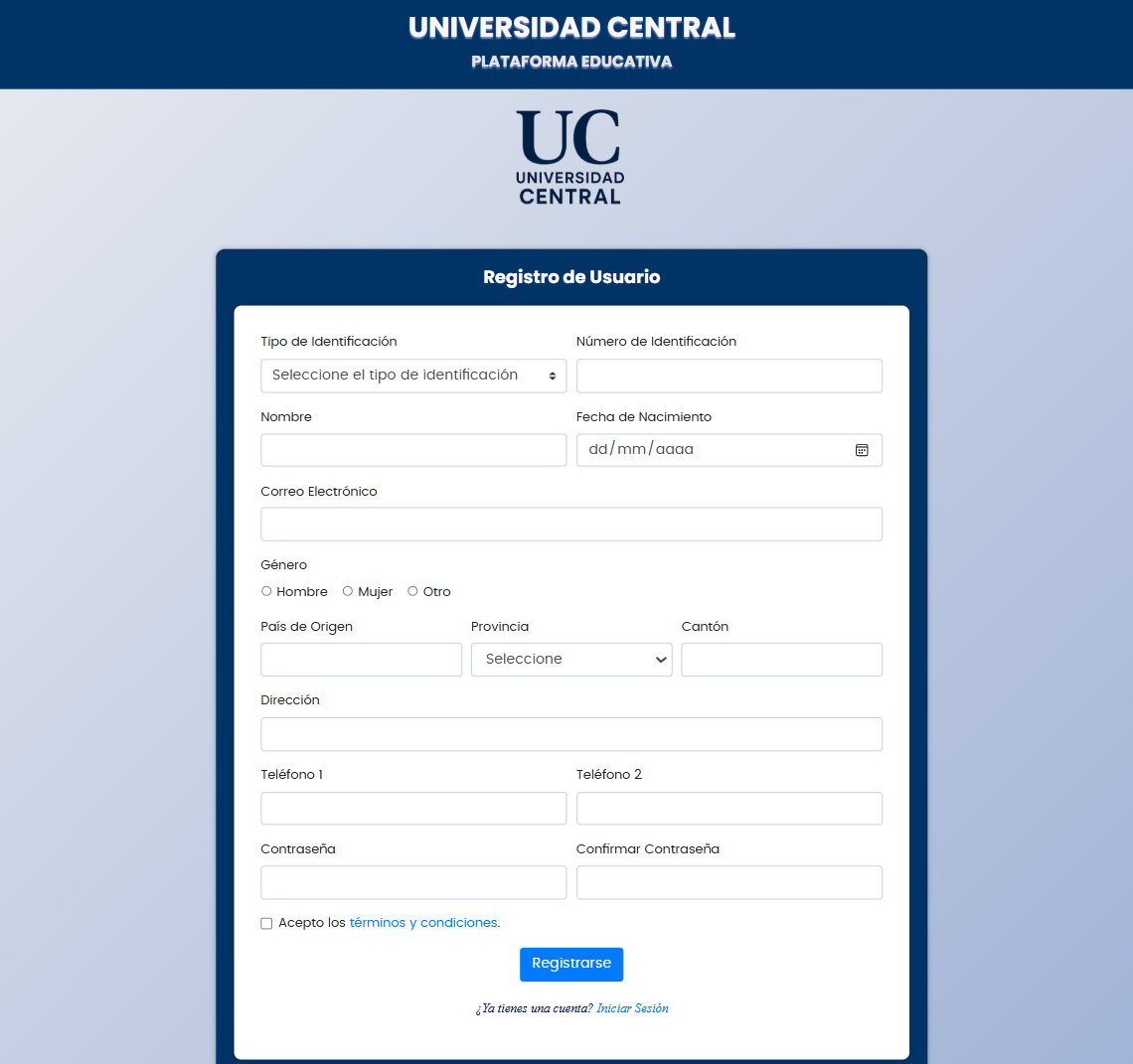
Este trabajo representa una base firme que para poder seguir construyendo más adelante. Gracias a la forma en la que está organizado, será posible incorporar mejoras o nuevas funciones cuando lo necesite, sin tener que empezar desde cero. Más allá del código, este avance refleja el crecimiento técnico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

**Anexos**

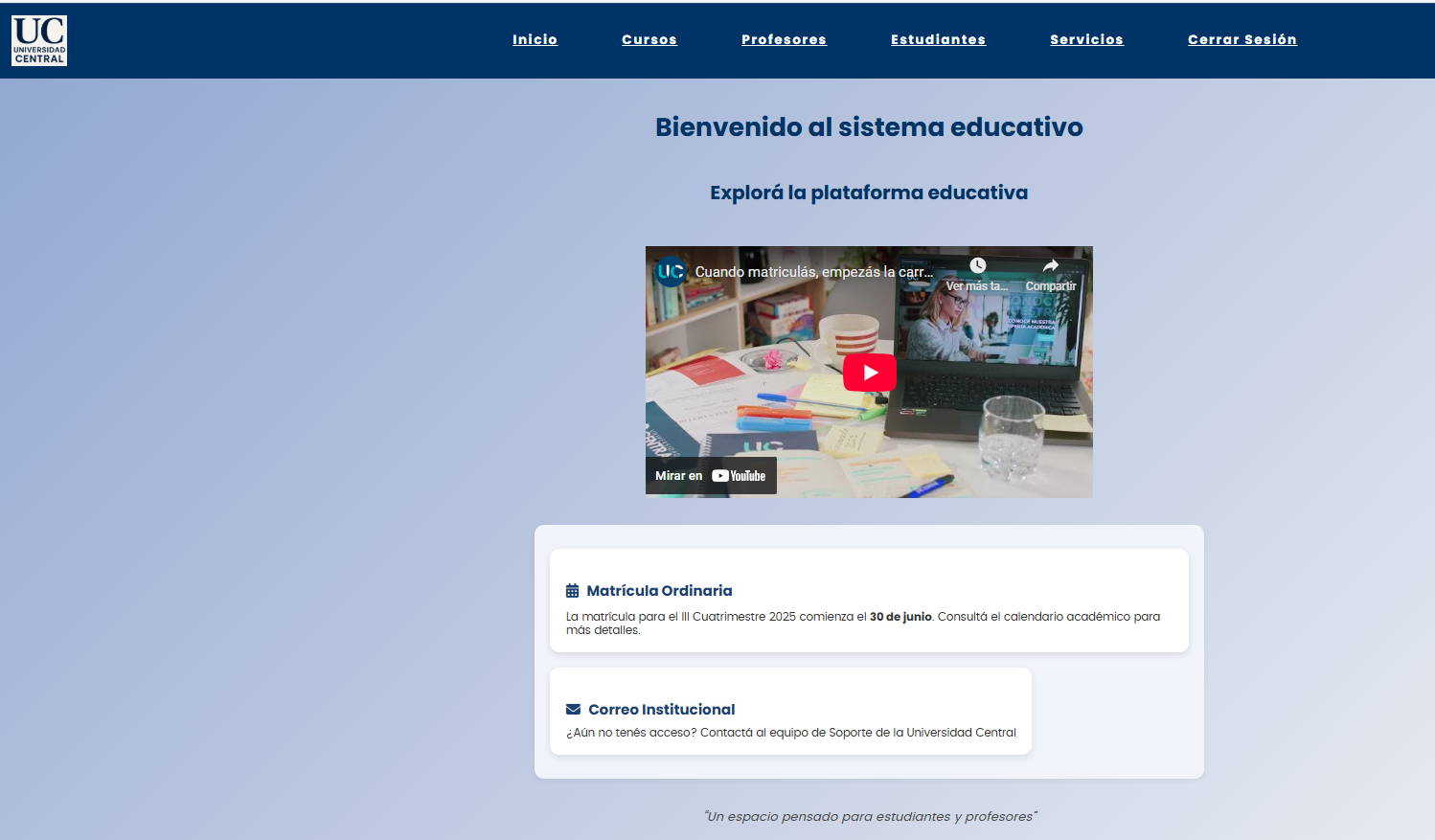
**Anexo 1. Login**

****

**Anexo 2. cuenta\_Re**

****

**Anexo 3.Inicio**

****

**Anexo 4.Curso**

****

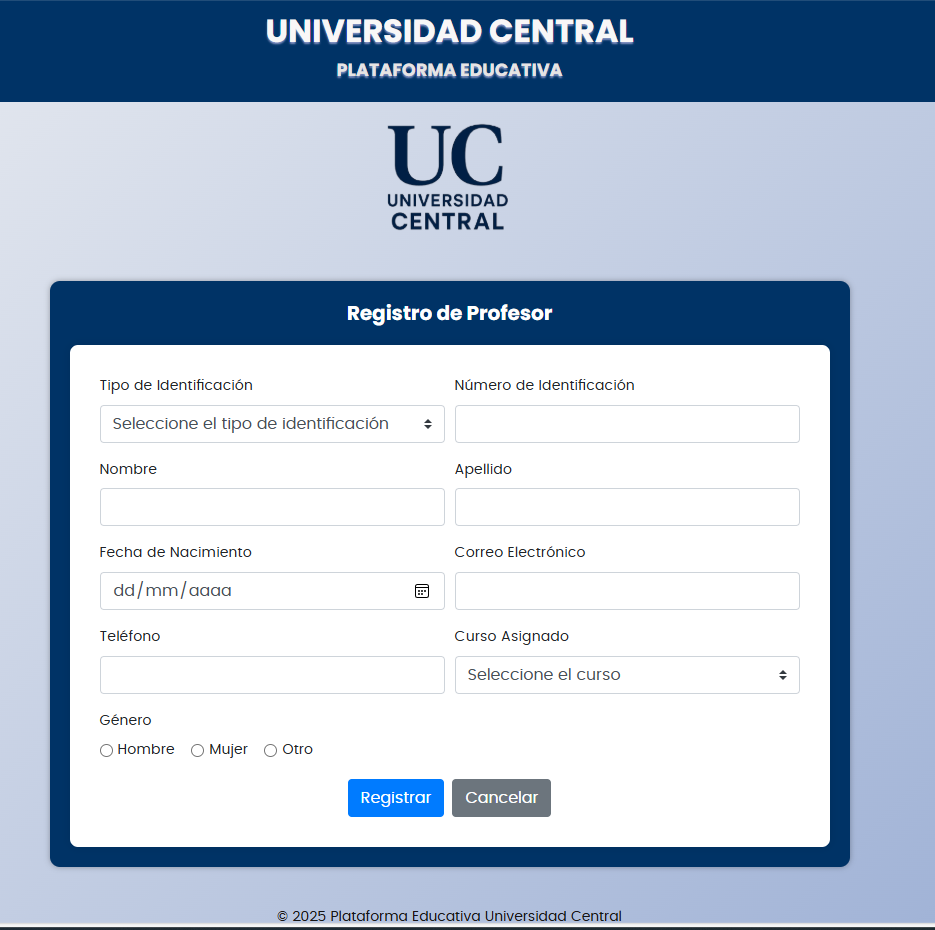
**Anexo 5.registroCurso**

****

**Anexo 6.Profesores**

****

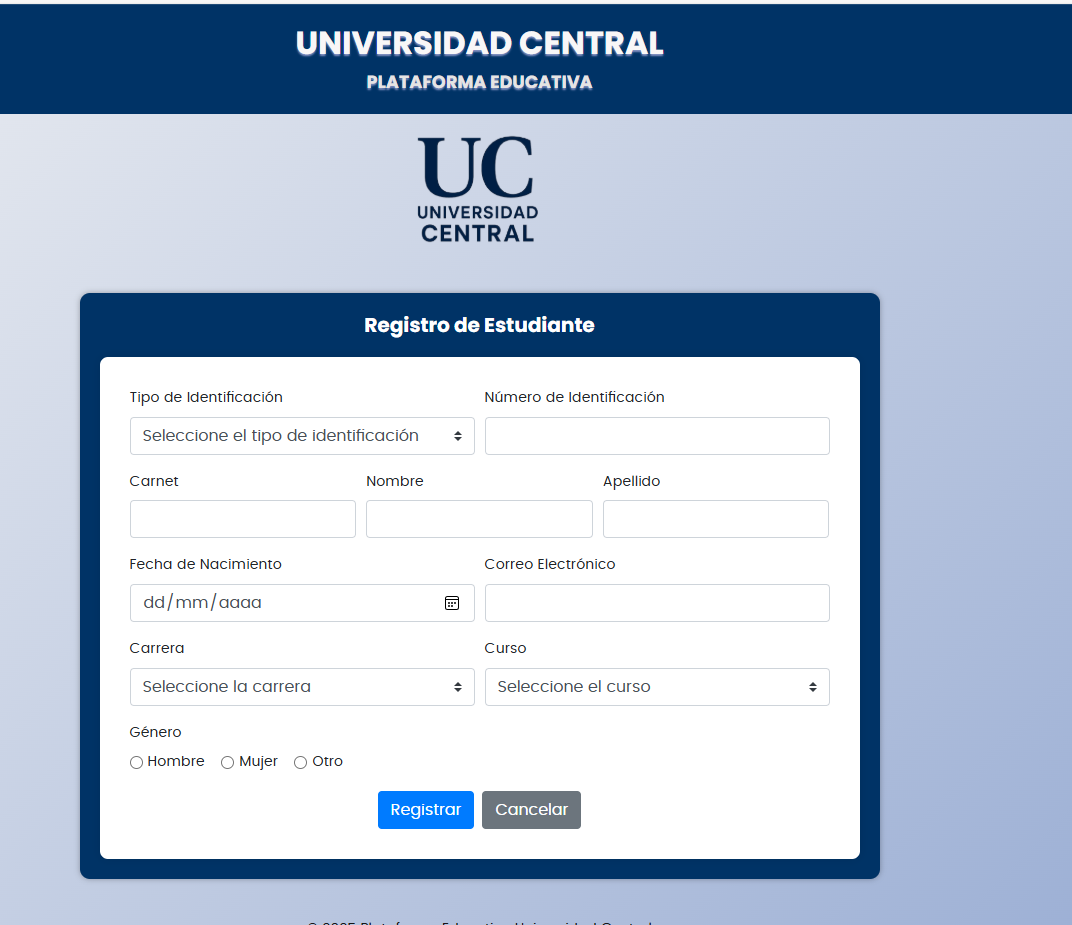
**Anexo 7 registroProfesor**

****

**Anexo 8 estudiantes**

****

**Anexo 9 registroEstudiante**

****

**Anexo 10. Diagrama UML**

