

Sistema de gestión de cursos

Ingeniería de sistemas

Ximena del Pilar Zambrano Chala
Juan Camilo Penagos Molina

Tabla de contenido

1. OBJETIVO	3
2. CONTEXTO	3
3. ALCANCE	3
4. JUSTIFICACION	4
5. DESCRIPCION GENERAL	4
5.1 Perspectiva del producto	4
5.2 Funcionalidades del producto	4
5.3 Suposiciones y dependencias	4
6. REQUISITOS ESPECIFICOS	5
6.1 Requerimientos funcionales	5
6.2 Requerimientos no funcionales	6
7. ESTRUCTURA TECNICA DEL SISTEMA	7
8. ARQUITECTURA	8
9. DISEÑO BASE DE DATOS	9

Especificación de requisitos del sistema de gestión de cursos

1. OBJETIVO

El objetivo principal del proyecto `Coursys` es desarrollar un sistema de gestión de cursos que facilite la administración y el seguimiento de estudiantes, profesores y cursos ofrecidos en una institución educativa. Este sistema busca automatizar procesos como la asignación de profesores a cursos, la inscripción de estudiantes, la gestión de horarios y el monitoreo de pagos, brindando una solución integral, eficiente y escalable.

2. CONTEXTO

Coursys surge como una herramienta para digitalizar y centralizar la gestión académica, permitiendo:

- Registrar estudiantes y profesores con sus datos clave.
- Gestionar cursos, incluyendo información sobre horarios, costos y asignaciones.
- Proveer un entorno seguro para la administración y consulta de información.

3. ALCANCE

- Integración de estudiantes y profesores en una base de datos central.
- Relación entre cursos y participantes (estudiantes/profesores).
- Gestión eficiente de horarios, costos y fechas.

4. JUSTIFICACION

En las instituciones educativas, la gestión eficiente de cursos, estudiantes y profesores representa un desafío crítico que, de no ser manejado adecuadamente, puede dar lugar a procesos administrativos complejos, errores en la asignación de recursos y dificultades en el seguimiento del progreso académico. Las herramientas tradicionales, como hojas de cálculo y sistemas manuales, suelen ser ineficientes, propensas a errores y poco escalables, especialmente en entornos con un gran volumen de estudiantes y cursos.

El sistema Coursys responde a esta problemática al ofrecer una solución tecnológica integral que automatiza y centraliza los procesos administrativos clave, como la inscripción de estudiantes, la asignación de profesores a cursos, la planificación de horarios y el monitoreo de pagos. Esto no solo reduce la carga administrativa, sino que también mejora la precisión de los datos y la toma de decisiones basada en información confiable.

5. DESCRIPCION GENERAL

5.1 Perspectiva del producto

Este sistema se caracteriza por ser intuitivo, adaptable y escalable, con una interfaz fácil de usar que garantiza una experiencia óptima para estudiantes, profesores y administradores. Al centralizar y automatizar las operaciones clave, Coursys mejora la precisión de los datos, reduce la carga administrativa y apoya la toma de decisiones estratégicas dentro de la institución.

5.2 Funcionalidades del producto

La pagina estará compuesta de los siguientes módulos:

- **Gestión de estudiantes y profesores:** Registro y actualización de datos, asignación de cursos, grupos y horarios.
- **Gestión de cursos:** Creación y administración de cursos con detalles como horarios, temarios y requisitos.
- **Gestión de Horas:** Creación automática o manual de horas para cursos, profesores y estudiantes.

5.3 Suposiciones y dependencias

- ✓ El usuario debe acceder al sitio con Internet para la visualización y descarga de los cursos y detalles de las inscripciones.
- ✓ Compatibilidad con navegadores y sistemas operativos.

- ✓ Los usuarios (profesores y estudiantes) tienen acceso a dispositivos electrónicos y conocimientos básicos en el uso de tecnología.
- ✓ Los datos iniciales de estudiantes, cursos y profesores estarán disponibles y serán proporcionados por la institución para el proceso de migración
- ✓ Acceso a servicios de bases de datos escalables como MySQL, PostgreSQL o servicios cloud.

6. REQUISITOS ESPECIFICOS

6.1 Requerimientos funcionales

RF1: Gestión de Estudiantes

- Descripción: El sistema permitirá a los administradores registrar, actualizar, eliminar y consultar información de estudiantes inscritos en los cursos.
- Entrada: Datos del estudiante (Nombre, cédula, teléfono, correo electrónico).
- Procesamiento: Validar unicidad de la cédula y almacenar los datos en la tabla `estudiantes`.
- Salida: Confirmación de registro o actualización del estudiante.

RF2: Gestión de Profesores

- Descripción: El sistema permitirá registrar, actualizar, eliminar y consultar los datos de los profesores.
- Entrada: Datos del profesor (Nombre, especialidad, correo electrónico).
- Procesamiento: Validar los datos y relacionarlos con los cursos correspondientes.
- Salida: Confirmación del registro, actualización o eliminación del profesor.

RF3: Creación de Cursos

- Descripción: El sistema permitirá registrar nuevos cursos con asignación de profesores y estudiantes.
- Entrada: Datos del curso (Nombre, horas, precio, fecha de inicio, profesor asignado, estudiantes inscritos).
- Procesamiento: Validar la existencia de profesores y estudiantes relacionados, y guardar el curso en la tabla `cursos`.
- Salida: Confirmación de la creación del curso.

RF4: Inscripción de Estudiantes en Cursos

- Descripción: El sistema permitirá asignar estudiantes a cursos existentes.
- Entrada: Datos de inscripción (ID del estudiante, ID del curso).
- Procesamiento: Validar la existencia del estudiante y el curso, y registrar la relación en el sistema.
- Salida: Confirmación de inscripción.

RF5: Consultas de Cursos

- Descripción: El sistema permitirá listar los cursos disponibles y sus detalles, como estudiantes inscritos y profesor asignado.
- Entrada: Solicitud de consulta por curso o filtro por profesor o estudiante.
- Procesamiento: Consultar la base de datos para recuperar la información solicitada.
- Salida: Lista detallada de cursos con sus asignaciones.

6.2 Requerimientos no funcionales**RNF1: Seguridad de Acceso**

- Descripción: El sistema debe garantizar que solo usuarios autorizados puedan realizar cambios en los datos.
- Entrada: Credenciales del usuario (nombre de usuario y contraseña).
- Procesamiento: Validar las credenciales y otorgar acceso según el rol.
- Salida: Acceso permitido o denegado.

RNF2: Rendimiento

- Descripción: El sistema debe procesar solicitudes en un tiempo promedio menor a 2 segundos.
- Entrada: Solicitud de consulta o registro.
- Procesamiento: Ejecución de operaciones en la base de datos y presentación de resultados.
- Salida: Respuesta rápida a las solicitudes del usuario.

RNF3: Disponibilidad

- Descripción: El sistema debe estar disponible al menos el 99.9% del tiempo para garantizar operaciones continuas.
- Entrada: Uso de la aplicación por parte de los usuarios.
- Procesamiento: Monitoreo y mantenimiento de los servicios.
- Salida: Operación continua sin interrupciones significativas.

RNF4: Escalabilidad

- Descripción: El sistema debe ser capaz de manejar un aumento en la cantidad de usuarios y cursos sin pérdida de rendimiento.
- Entrada: Aumento en los datos y usuarios concurrentes.
- Procesamiento: Ajuste dinámico de los recursos del servidor.
- Salida: Respuesta eficiente y escalable.

RNF5: Compatibilidad

- Descripción: El sistema debe ser accesible desde navegadores web modernos y dispositivos móviles.
- Entrada: Acceso desde diversos dispositivos.
- Procesamiento: Adaptación de la interfaz al entorno del usuario.
- Salida: Interfaz funcional y adaptada.

7. ESTRUCTURA TECNICA DEL SISTEMA

7.1 Frontend

Framework: Vue.js/ Ionic

Tecnologías:

- HTML5, CSS3, JavaScript

Librerías: Axios (peticiones HTTP), Vuex/Redux (estado), Vue Router/React Router (ruteo)

Componentes clave:

- Dashboard de Administrador
- Panel de Estudiante
- Panel de Profesor

Vistas: Cursos, Inscripciones, Horarios, Pagos, Calificaciones

7.2 Backend

Lenguaje/Framework:

- Kotlin con Spring Boot
- Node.js

Responsabilidades:

- Autenticación y autorización (JWT u OAuth2)
- Gestión de usuarios (estudiantes, profesores, admins)
- CRUD de cursos, inscripciones, horarios, pagos
- Validaciones y lógica de negocio
- Comunicación con la base de datos

Patrones comunes:

- Repositorio, Servicio, Controlador

7.3 Base de Datos

Sistema Gestor: MySQL

Modelo Relacional:

- Tablas: profesores, estudiantes, cursos
- Claves foráneas y relaciones entre entidades

8. ARQUITECTURA

8.1 Frontend

Framework: Vue.js / ionic

Funciones:

- ✓ Interfaz amigable para estudiantes, profesores y admins
- ✓ Paneles personalizados por rol
- ✓ Comunicación con backend vía API REST

8.2 Backend (API REST)

Framework: Spring Boot (Java) / Express.js (Node.js)

Funciones:

- ✓ Autenticación y autorización
- ✓ CRUD completo (cursos, usuarios, pagos, horarios, etc.)
- ✓ Validaciones y lógica de negocio

8.3 Base de Datos Relacional

- MySQL

Entidades como:

- Estudiantes, profesores, cursos

8.4 Infraestructura

- Docker para contenedores
- CI/CD con GitHub Actions o GitLab

9. DISEÑO BASE DE DATOS

- Script de la Base de Datos

```
sql
DROP DATABASE IF EXISTS Coursys;
```

```
CREATE DATABASE Coursys;
```

```
USE Coursys;
```

Tabla Estudiantes

```
CREATE TABLE estudiantes (
  id SERIAL PRIMARY KEY,
  nombre VARCHAR(200) NOT NULL,
  cedula VARCHAR(100) NOT NULL,
  telefono FLOAT NOT NULL,
  email VARCHAR(200) NOT NULL
);
```

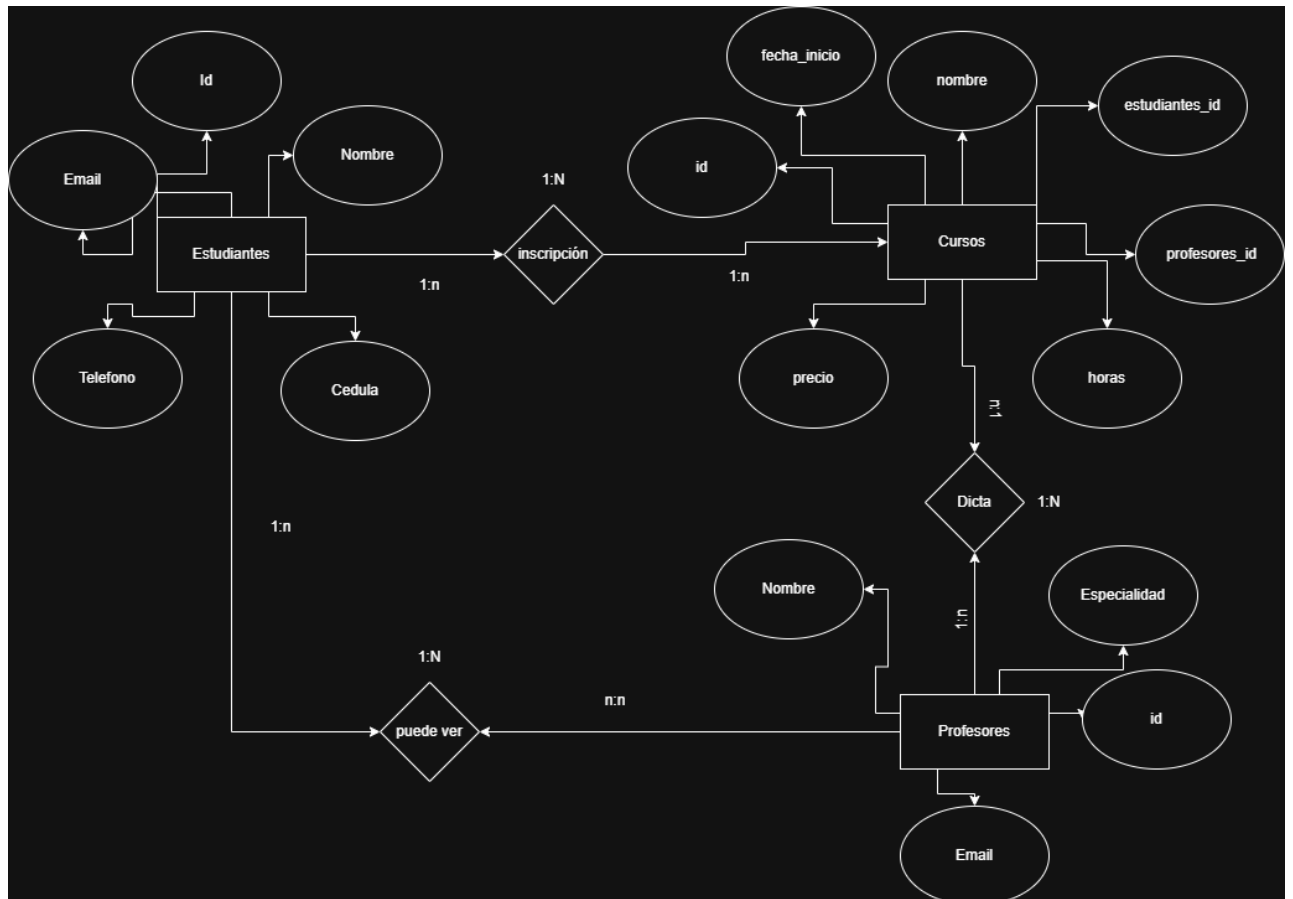
Tabla Profesores

```
CREATE TABLE profesores (
  id SERIAL PRIMARY KEY,
  nombre VARCHAR(200) NOT NULL,
  especialidad VARCHAR(200) NOT NULL,
  email VARCHAR(200) NOT NULL
);
```

Tabla Cursos

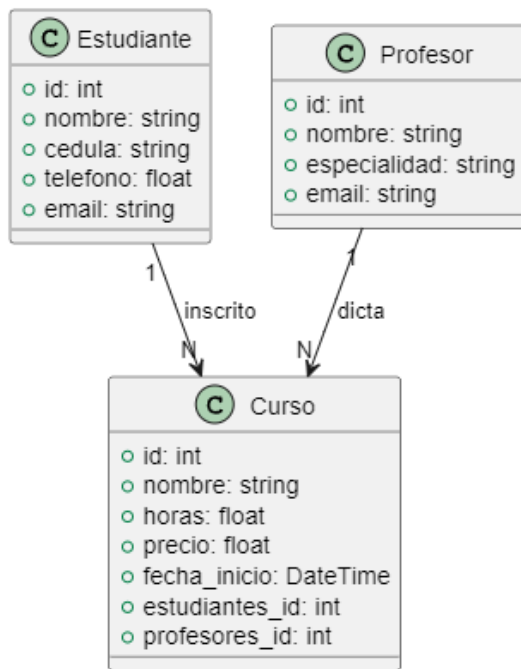
```
CREATE TABLE cursos (
  id SERIAL PRIMARY KEY,
  nombre VARCHAR(200) NOT NULL,
  estudiantes_id INT NOT NULL,
  profesores_id INT NOT NULL,
  horas FLOAT NOT NULL,
  precio FLOAT NOT NULL,
  fecha_inicio TIMESTAMP NOT NULL,
  CONSTRAINT fk_estudiantes FOREIGN KEY (estudiantes_id)
REFERENCES estudiantes (id),
  CONSTRAINT fk_profesores FOREIGN KEY (profesores_id)
REFERENCES profesores (id)
);
```

MER:

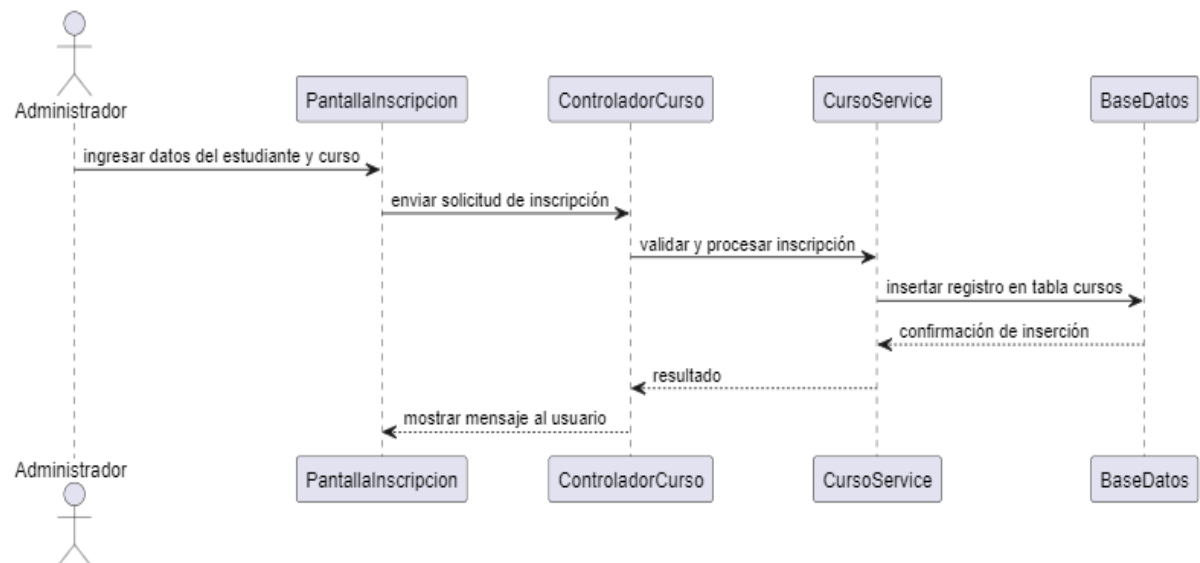


UML:

- CLASES:



- SECUENCIA:



- CASOS DE USO:

