UNIVERSIDAD DE EL SALVADOR

Facultad Multidisciplinaria De Occidente
Departamento de Ingeniería y Arquitectura
Ingeniería en Desarrollo de Software

Carrera:

Ingeniería en Desarrollo de Software/Educación en Línea

Ciclo 2

Materia:

Programación Orientada a Objetos

Tema:

Entrega #1 de Proyecto de ciclo

Tutor/a GT03:
Francisco Javier Morales Ayala

Alumnos/as:	Carnet:
Cindy Ariana Reyes Molina	RM24001
Jefferson Stanley Ramos Damas	RD24003
Bryan José Moreno Villanueva	MV24050
Diego Oswaldo Meza Argueta	MA20081
Irvin Adonay Ramírez Linares	RL22020

Fecha:
28/09/25.

Introducción

Rúbricas de evaluación:

ENTREGA #1 20%- (28 Sept) (Nota Grupal)- Se habilitará en campus un espacio para que un solo integrante de su grupo pueda entregar un documento PDF con:

- Diagrama de clases de su proyecto asignado
- Diagrama Entidad Relación de su proyecto asignado
- Casos de Uso de su proyecto asignado
- URL de proyecto creado en github, debe ser un proyecto Gradle o Maven, deben estar como colaboradores del proyecto todos los integrantes, si es un repo privado debe dar acceso al tutor asignado.Deberá editar el readme para al menos entender de qué tratará el proyecto.

Este documento presenta la documentación completa del Sistema de Gestión de Cursos, una plataforma educativa que permite a instructores crear y gestionar cursos, mientras que los estudiantes pueden inscribirse y participar en ellos.

Objetivos del Sistema

- Facilitar la creación y gestión de cursos por parte de los instructores
- Permitir a los estudiantes inscribirse y participar en cursos
- Mantener un registro organizado de inscripciones y participantes
- Proporcionar una interfaz clara para la gestión académica

Casos de uso

1. Crear curso

- Actor principal: Instructor
- Descripción: El instructor registra un nuevo curso proporcionando título y descripción.
- Flujo principal:
  - El instructor solicita crear un curso.
  - El sistema valida los datos
  - El curso se almacena en la base de datos.
  - El sistema confirma la creación.

- Flujos alternativos:
  - Si falta un campo obligatorio, el sistema devuelve un error.

2. Editar curso

- Actor principal: Instructor
- Descripción: El instructor actualiza título o descripción de un curso ya existente.
- Flujo principal:
  - El instructor selecciona el curso a editar.
  - El sistema valida la existencia del curso.
  - Se actualizan los datos en la base.
  - El sistema confirma la edición.
- Flujos alternativos:
  - Si el curso no existe, se devuelve error.

3. Eliminar curso

- Actor principal: Instructor
- Descripción: El instructor elimina un curso de la plataforma.
- Flujo principal:
  - El instructor solicita eliminar un curso
  - El sistema valida existencia y permisos
  - Se elimina el curso y las inscripciones asociadas.
  - El sistema confirma la eliminación.
- Flujos alternativos:
  - Si el curso no existe, el sistema notifica error.

4. Inscribirse en un curso

- Actor principal: Estudiante
- Descripción: El estudiante se inscribe en un curso activo de la plataforma.
- Flujo principal:
  - El estudiante solicita inscribirse.
  - El sistema valida si el curso está activo.
  - El sistema registra la inscripción.
  - Se confirma al estudiante.
- Flujos alternativos:
  - Si el estudiante ya está inscrito, se devuelve advertencia.

- Si el curso está inactivo, se notifica error.

5. Salir de un curso

- Actor principal: Estudiante
- Descripción: El estudiante cancela su inscripción en un curso.
- Flujo principal:
  - El estudiante selecciona el curso del que desea salir.
  - El sistema valida inscripción.
  - El sistema elimina la relación en la base de datos.
  - El sistema confirma la salida.
- Flujos alternativos:
  - Si no estaba inscrito, se devuelve error.

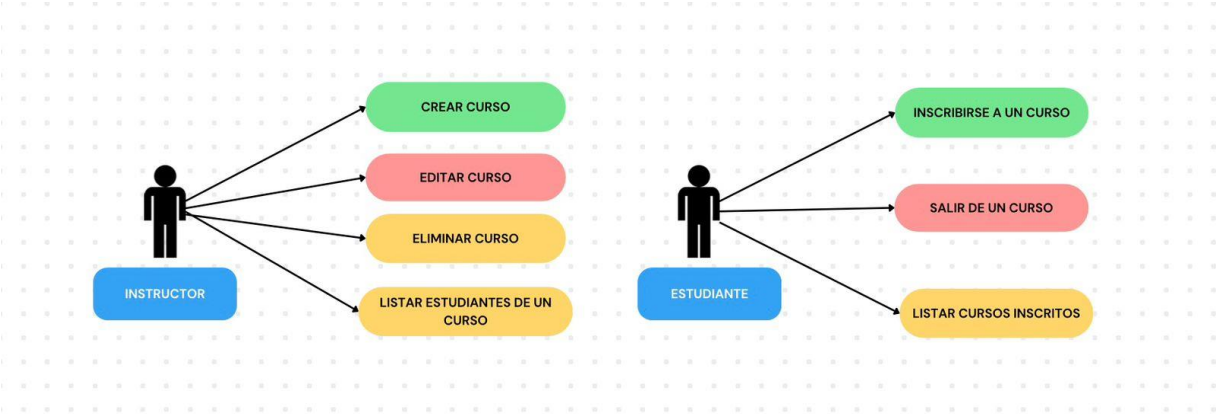
6. Listar cursos de un estudiante

- Actor principal: Estudiante
- Descripción: El estudiante consulta todos los cursos en los que está inscrito.
- Flujo principal:
  - El estudiante solicita ver sus cursos.
  - El sistema obtiene la lista de cursos desde la base.
  - El sistema devuelve la información.

7. Listar estudiantes de un curso

- Actor principal: Instructor
- Descripción: El instructor consulta los estudiantes inscritos en un curso.
- Flujo principal:
  - El instructor selecciona un curso.
  - El sistema obtiene los estudiantes inscritos.
  - El sistema devuelve la lista.
- Flujos alternativos:
  - Si el curso no tiene estudiantes, se devuelve lista vacía.

- **Diagrama de Casos de Uso**



- **Modelo de Clases**

Actor: Instructor

- Crear Curso
- Editar Curso
- Eliminar Estudiantes
- Listar Estudiantes de un Curso

Actor: Estudiante

- Inscribirse a un Curso
- Salir de un Curso
- Listar Cursos Inscritos

Modelo de Clases

Clase: Usuario

Atributos:

- id: int (BD: auto_incremental)
- nombre: String (BD: varchar(100))
- email: String (BD: varchar(100))
- rol: Rol

Métodos:

- getID(): int
- getNombre(): String
- setNombre(nombre: String): void
- getEmail(): String
- setEmail(email: String): void

- getRol(): Rol
- setRol(rol: Rol): void

Clase: Estudiante (hereda de Usuario)

Atributos:

- cursosInscritos: List<Curso>

Métodos:

- inscribirCurso(cursoID: int): boolean
- salirCurso(cursoID: int): boolean
- listarCursos(): List<Curso>

Clase: Instructor (hereda de Usuario)

Atributos:

- cursosCreados: List<Curso>

Métodos:

- crearCurso(titulo: String, descripcion: String): boolean
- editarCurso(cursoID: int, nuevoTitulo: String, nuevaDescripcion: String): boolean
- eliminarCurso(cursoID: int): boolean
- listarCursos(): List<Curso>

Clase: Curso

Atributos:

- idCurso: int (BD: auto_incremental)
- titulo: String (BD: varchar(200))
- descripcion: String (BD: varchar(1000))
- instructorId: int (BD: instructor_id)
- estado: EstadoCurso
- estudiantesInscritos: List<Estudiante>

Métodos:

- getID(): int
- getTitulo(): String
- getDescripcion(): String
- getInstructor(): int
- listarEstudiantes(): List<Estudiante>

Clase: Inscripcion

Atributos:

- idInscripcion: int

- idCurso: int
- idEstudiante: int
- fecha: Date

Enumeraciones

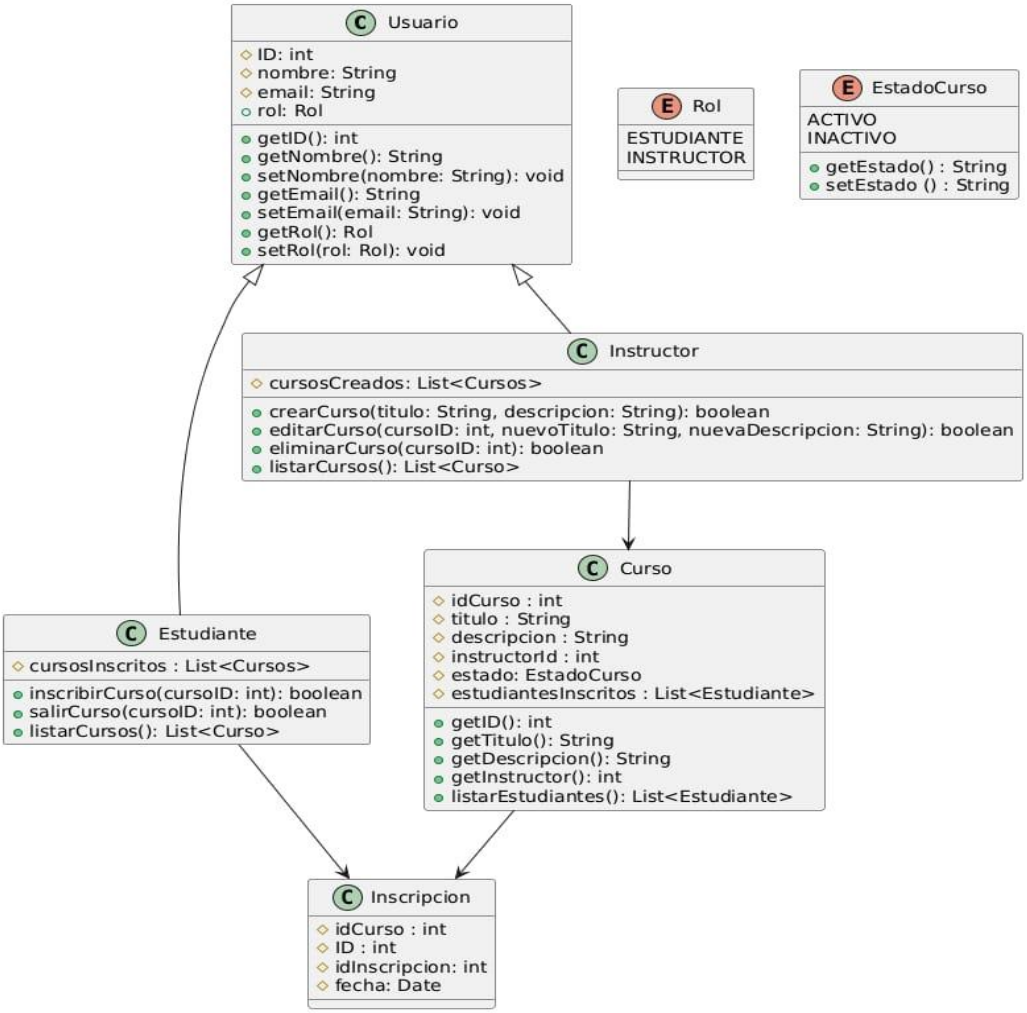
Enum: Rol

- ESTUDIANTE
- INSTRUCTOR
- ADMIN

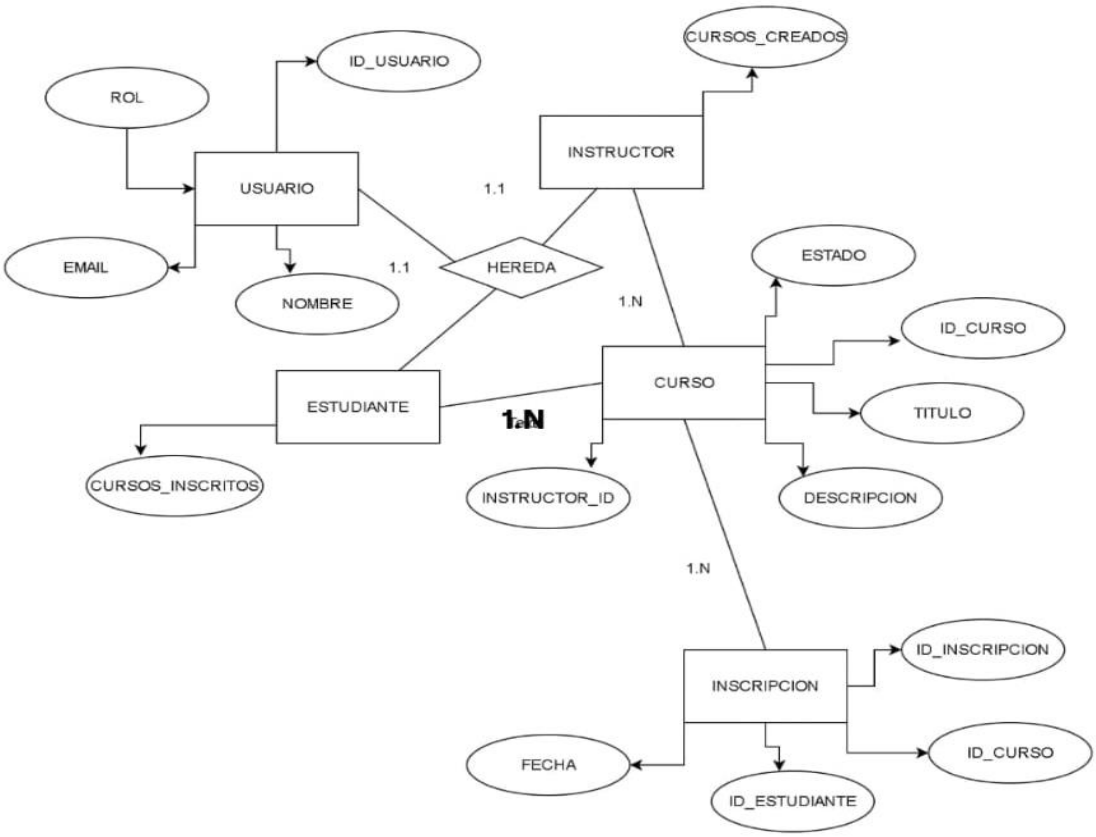
Enum: EstadoCurso

- ACTIVO
- INACTIVO

- **Diagrama de Clases**



- **Diagrama Entidad-Relación**



- **Especificaciones de Base de Datos**

Tabla: USUARIO

Campo	Tipo	Características
ID_USUARIO	INT	Primary Key, Auto-increment
NOMBRE	VARCHAR(100)	Not Null
EMAIL	VARCHAR(100)	Not Null, Unique
ROL	ENUM	ESTUDIANTE, INSTRUCTOR, ADMIN

Tabla: CURSO

Campo	Tipo	Características
ID_CURSO	INT	Primary Key, Auto-increment
TITULO	VARCHAR(200)	Not Null
DESCRIPCION	VARCHAR(1000)	Not Null
INSTRUCTOR_ID	INT	Foreign Key → USUARIO(ID_USUARIO)
ESTADO	ENUM	ACTIVO, INACTIVO

Tabla: INSCRIPCION

Campo	Tipo	Características
ID_INSCRIPCION	INT	Primary Key, Auto-increment
ID_CURSO	INT	Foreign Key → CURSO(ID_CURSO)
ID_ESTUDIANTE	INT	Foreign Key → USUARIO(ID_USUARIO)
FECHA	DATE	Not Null

- **Relaciones y Cardinalidad**

Instructor - Curso (1:N)
Descripción: Un instructor puede tener múltiples cursos.

Estudiante - Curso (N:N)
Descripción: Un estudiante puede inscribirse en múltiples cursos y un curso puede tener múltiples estudiantes.