



Universidad de las Fuerzas Armadas

ESPE Departamento de Ciencias de la Computación

Análisis y Diseño

Integrantes:

Isaac Escobar

Eduardo Mortensen

Diego Ponce

NRC: 14571

ARQUITECTURA MVC PARA EL SISTEMA DE CALIFICACIONES

Se eligió la arquitectura Modelo-Vista-Controlador (MVC) para el desarrollo del Sistema de Gestión de Calificaciones de la Unidad Educativa Mahanaym debido a su capacidad para organizar el código de manera estructurada y modular. Esta arquitectura permite separar claramente la lógica de negocio (modelo), la interfaz gráfica (vista) y el flujo de control (controlador), lo que facilita el mantenimiento, la escalabilidad y el trabajo colaborativo. Dado que el sistema debe gestionar múltiples perfiles de usuario (administrador, docente, estudiante, padre), controlar el acceso a distintas funcionalidades y mantener una trazabilidad segura de las calificaciones, MVC proporciona una base sólida para cumplir estos requisitos de forma ordenada y eficiente.

Además, el patrón MVC se adapta perfectamente a las tecnologías propuestas en el proyecto (HTML, CSS, JavaScript, Node.js), lo que permite integrar fácilmente funcionalidades como la autenticación, registro de notas, generación de informes y control de accesos. La separación entre lógica y presentación facilita las pruebas individuales de componentes y la reutilización del código, lo cual es crucial para garantizar la mantenibilidad a largo plazo y cumplir con los requerimientos no funcionales como usabilidad, rendimiento y seguridad.

1. MODELO (Model)

Encargado de la **lógica de negocio y acceso a datos**.

Ejemplos de modelos:

- Usuario: ID, nombre, correo, contraseña, rol (administrador, docente, estudiante, padre).
- Calificacion: ID, estudianteID, materialID, tipoEvaluacion, nota, porcentaje, observaciones, fecha.
- Materia: ID, nombre, docenteID.
- Informe: estudianteID, trimestre, promedio, PDF.
- BitacoraModificaciones: usuarioID, fecha, campo_modificado, valor_anterior, valor_nuevo, comentario.
- Evaluacion: ID, tipo (examen, tarea), porcentaje, materialID.

2. VISTA (View)

Responsable de la **interfaz gráfica** con el usuario. Usa HTML, CSS, Bootstrap y JavaScript.

Vistas principales:

- login.html: formulario de acceso.
- panelDocente.html: formulario de ingreso de notas, configuración de porcentajes.
- panelEstudiante.html: consulta de calificaciones.
- panelPadre.html: vista de informes académicos.
- panelAdmin.html: gestión de usuarios, auditoría.
- reporteNotas.html: visualización/exportación de PDF.

3. CONTROLADOR (Controller)

Encargado de manejar la lógica de flujo entre la vista y el modelo. Responde a las acciones del usuario, invoca los modelos, y retorna resultados a las vistas.

Controladores clave:

- authController.js: login, logout, control de acceso por perfil (RF01, RF06).
- notasController.js: registrar y mostrar calificaciones (RF02, RF03, RF07).
- informeController.js: generación de reportes académicos (RF04).
- bitacoraController.js: registro de modificaciones (RF05).
- adminController.js: gestión de usuarios y roles.

FLUJO MVC DE FUNCIONALIDADES CLAVE

Autenticación y Control de Acceso

- **Vista:** formulario de login.
- **Controlador:** authController.login()
- **Modelo:** Usuario.findByCredentials()
- Verifica credenciales y redirige según rol (admin, docente, padre, estudiante) [RF01, RF06].

Registro de Calificaciones

- **Vista:** formulario dinámico con lista de estudiantes.
- **Controlador:** notasController.registrarNota()
- **Modelo:** Calificacion.create()
- Aplica el porcentaje automáticamente y guarda nota (RF03, RF07).

Visualización de Notas

- **Vista:** listado por materia y periodo.
- **Controlador:** notasController.verNotas()
- **Modelo:** Calificacion.find({ estudianteID })
- Carga y muestra estadísticas y detalles [RF02].

Generación de Informes Académicos

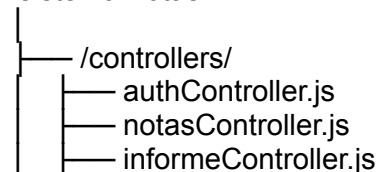
- **Vista:** filtro por curso, estudiante o trimestre.
- **Controlador:** informeController.generar()
- **Modelo:** Informe.createPDF()
- Exportación en PDF con seguridad [RF04].

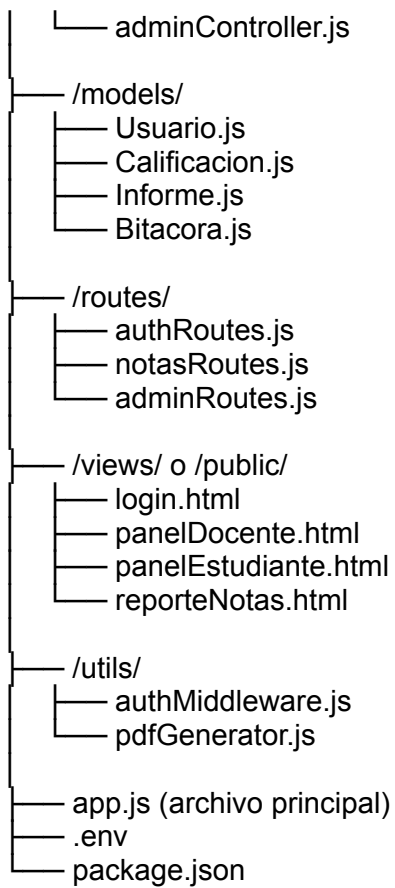
Registro de Cambios (Auditoría)

- **Controlador:** bitacoraController.registrarCambio()
- **Modelo:** BitacoraModificaciones.create()
- Se ejecuta cada vez que un docente modifica una calificación [RF05].

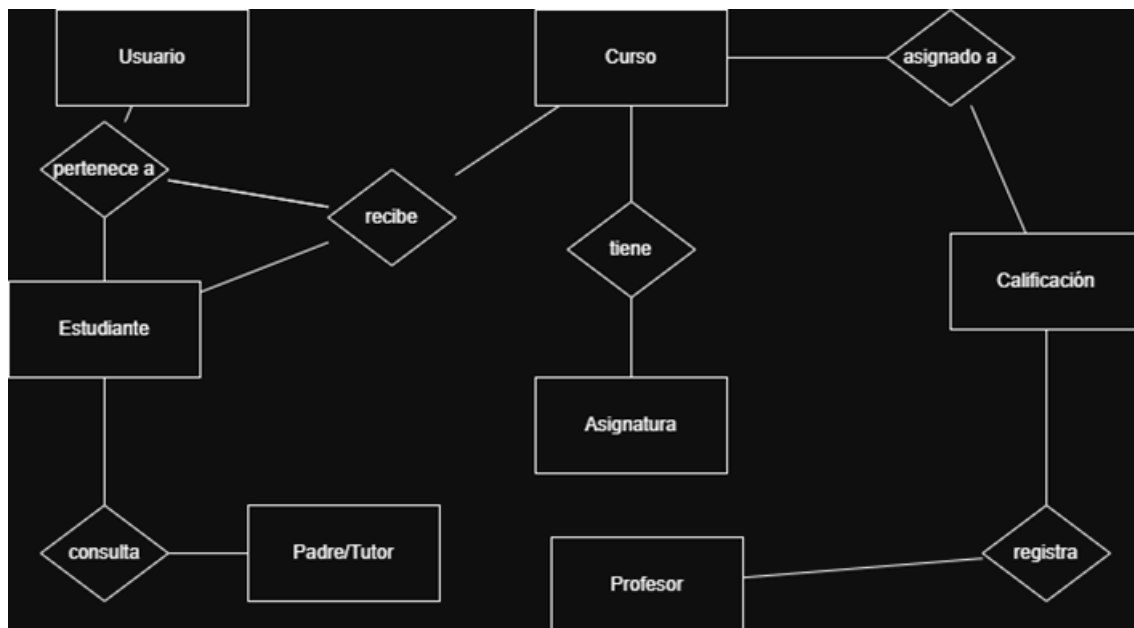
ESTRUCTURA DE CARPETAS SUGERIDA APLICANDO MVC

/sistema-notas/





1. Diagrama entidad relación

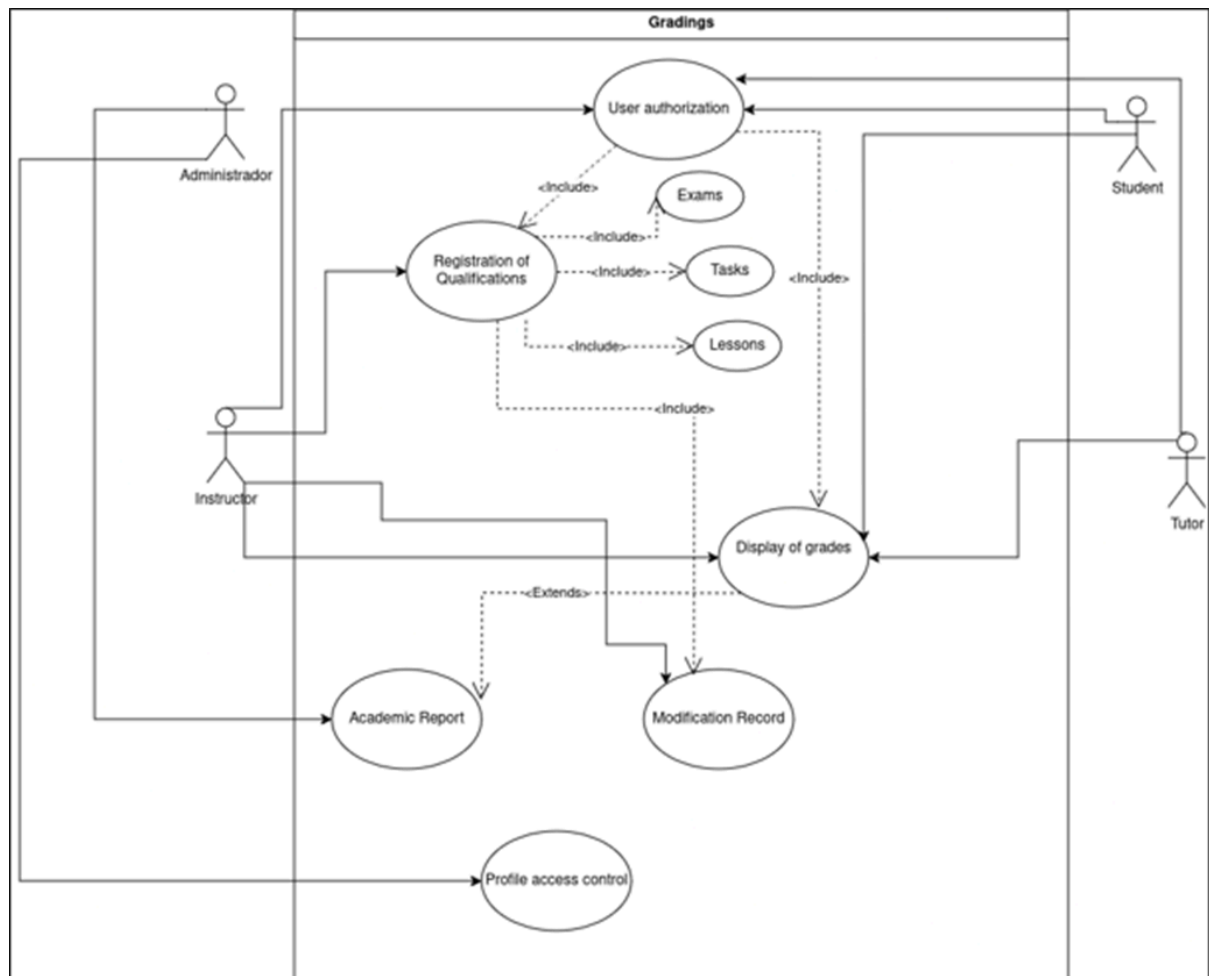


Este diagrama de entidad-relación (ER) muestra la relación entre diversas entidades en un sistema educativo. A continuación te doy una descripción de cada entidad y su relación:

1. **Usuario:** Representa a los usuarios del sistema, que pueden estar relacionados con varias entidades. Un usuario **pertenece a** un estudiante o un padre/tutor.
2. **Estudiante:** Se asocia con la entidad **Usuario**. Un estudiante **consulta** información relacionada con su curso o asignaturas.
3. **Padre/Tutor:** Se asocia con el **Estudiante**. Los padres o tutores tienen una relación con el estudiante, en donde **consultan** información sobre el progreso académico.
4. **Curso:** Un curso **recibe a** un estudiante, quien **tiene** asignaturas dentro de ese curso. Un curso también **asignado a** un profesor y tiene una relación directa con la **calificación** de los estudiantes.
5. **Asignatura:** Se encuentra asociada con el **Curso** y con el **Profesor**, que **registra** las calificaciones de los estudiantes. Un estudiante **tiene** asignaturas en su curso.
6. **Profesor:** Los profesores **registran** las calificaciones de los estudiantes en las asignaturas correspondientes.
7. **Calificación:** Esta entidad está vinculada al **Curso** y **Asignatura**, registrando las evaluaciones de los estudiantes.

Este diagrama refleja cómo interactúan los usuarios, estudiantes, padres/tutores, cursos, asignaturas, profesores y calificaciones dentro de un sistema educativo.

2. Diagrama de casos de uso

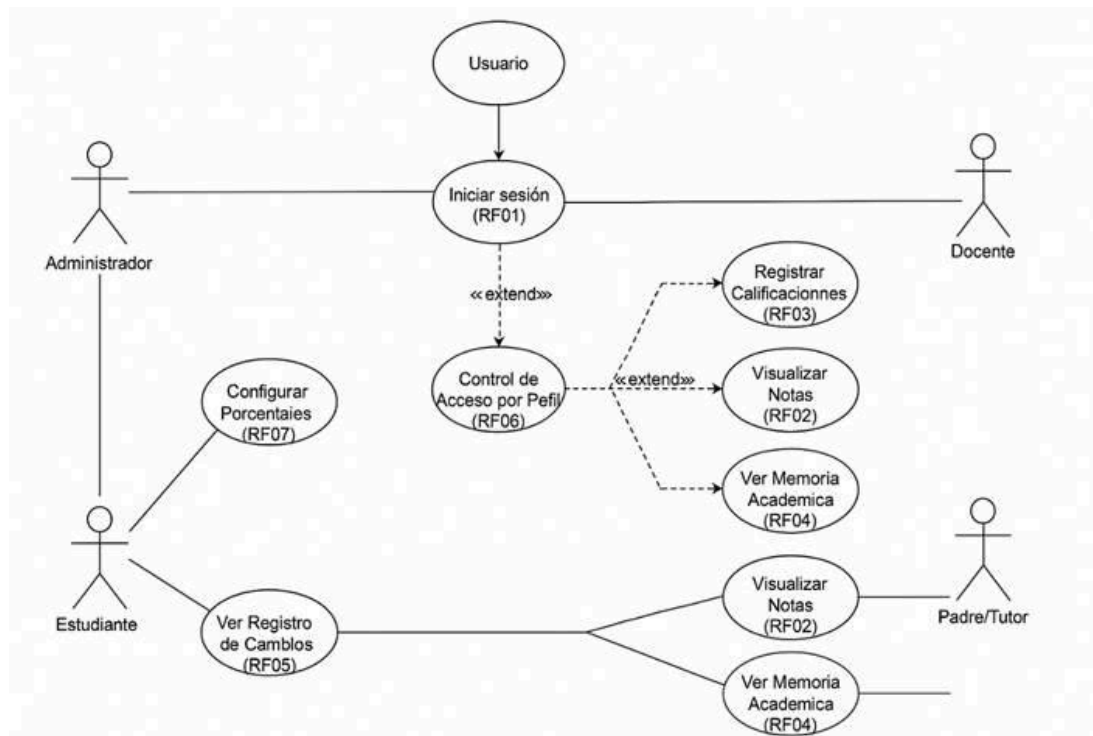


Este diagrama de casos de uso representa un sistema de calificaciones académico, donde varios actores interactúan con diferentes funcionalidades del sistema.

1. Administrador: Tiene la capacidad de gestionar la autorización de usuarios y controlar el acceso al perfil. Además, tiene acceso a las funcionalidades de registro de calificaciones.
2. Instructor: Se encarga de generar el informe académico, y también puede interactuar con las funcionalidades de registro de calificaciones, exámenes, tareas, y lecciones.
3. Estudiante: Puede ver sus calificaciones a través de la opción mostrar calificaciones.
4. Tutor: También tiene acceso a la funcionalidad de mostrar calificaciones.

El sistema integra distintos procesos a través de relaciones de inclusión (include) y extensión (extend). Las funcionalidades básicas incluyen el registro de calificaciones, la autorización de usuarios, y la visualización de calificaciones por parte de estudiantes y tutores. El sistema permite que los instructores generen informes académicos y registren detalles de tareas, lecciones y exámenes.

3. Diagrama de caso de uso extendido



Este diagrama de caso de uso extendido muestra cómo los usuarios interactúan con el sistema de gestión académica y las funcionalidades que se ofrecen según su rol.

1. **Usuario:** Es el actor principal y debe **iniciar sesión** (RF01) para poder acceder al sistema. Después de iniciar sesión, puede acceder a diferentes funcionalidades según su rol.
2. **Administrador:** Tiene la capacidad de **configurar porcentajes** (RF07) y **controlar el acceso por perfil** (RF06), lo que le permite gestionar qué funcionalidades están disponibles para otros usuarios.
3. **Estudiante:** Puede **ver el registro de cambios** (RF05), lo que le permite conocer las modificaciones realizadas en sus calificaciones o datos académicos.
4. **Docente:** Tiene la capacidad de **registrar calificaciones** (RF03), lo que se extiende desde la acción de iniciar sesión.
5. **Padre/Tutor:** Puede **visualizar notas** (RF02) y **ver la memoria académica** (RF04) del estudiante, proporcionando acceso a la información académica del alumno.

Las relaciones **extendidas** muestran cómo ciertas funcionalidades se activan solo después de realizar una acción previa, como iniciar sesión para acceder a los casos de uso relacionados.