

# Nombre del Software

Nombre del Software: Gestión Escolar

## Autores

Melani Forsythe Matos

Olivia Ibañez Mustelier

Dayan Cabrera Corvo

Orlando R. De la Torre Leal

## Diccionario de Datos

Aquí se describen los elementos clave del sistema, incluyendo las entidades y sus atributos.

- **Administrator:** Entidad que representa la tabla administrador.
  - AdminId: Id del administrador.
  - UserId: Su id como usuario.
  - AdminName: Nombre completo del administrador.
  - AdminSalary: Salario del administrador.
  - IsDeleted: Indica si el administrador ha sido eliminado de la base de datos
- **AuxiliaryMeans:** Entidad que representa medios auxiliares.
  - IdMean: Id del medio auxiliar
  - IsAviable: Indica si el medio esta disponible para reservar.
  - IsDeleted: Indica si ha sido eliminado de la base de datos.
  - Type: Tipo de medio auxiliar.
  - NameMean: Nombre del medio.
  - State: Estado del medio.
  - Maintenances: Mantenimientos que ha recibido el medio.
  - Subjects: Asignaturas asociadas a este medio auxiliar.
  - Professors: Profesores asociados a este medio auxiliar.
- **TechnologicalMeans:** Entidad que representa medios tecnologicos.
  - IdMean: Id del medio tecnológico
  - IsAviable: Indica si el medio esta disponible para reservar.
  - IsDeleted: Indica si ha sido eliminado de la base de datos.
  - NameMean: Nombre del medio.
  - State: Estado del medio.
  - Maintenances: Mantenimientos que ha recibido el medio.

- Classroom: Aulas a las que este medio ha sido asignado.
- **ClassRoom:** Entidad que representa aulas.
  - IdClassR: Id de la entidad.
  - IsAviable: Indica si la entidad esta disponible para reservar.
  - IsDeleted: Indica si ha sido eliminado de la base de datos.
  - Location: Ubicación del aula.
  - Subjects: Asignaturas asociadas al aula.
  - Restriction: Restricciones asociadas al aula.
  - Professors: Profesores asociados a la entidad.
  - TechnologicalMeans: Medios tecnológicos asociados al aula.
- **Curso:** Entidad que representa aulas.
  - IdCourse: Id de la entidad.
  - IsDeleted: Indica si ha sido eliminado de la base de datos.
  - CourseName: Nombre de la entidad.
  - Students: Estudiantes asociados al curso.
- **Maintenances:** Entidad que representa Mantenimientos.
  - IdM: Id de la entidad.
  - IsDeleted: Indica si ha sido eliminado de la base de datos.
  - TypeOfMean: Tipo de medio.
  - Cost: Costo de los medios.
  - IdMean: Id del medio.
- **Professor:** Entidad que representa Profesores.
  - IdProf: Id de la entidad.
  - IsDeleted: Indica si ha sido eliminado de la base de datos.
  - UserId: Id del usuario.
  - IsDean: Indica si es el decano.
  - LaboralExperience: Experiencia laboral.
  - Salary: Salario.
  - Spec: Especialización.
  - Contract: Tipo de contrato.
  - NameProf: Nombre del profesor
- **ProfessorPunishment:** Entidad que representa sanciones a Profesores.
  - IdPun: Id de la entidad.
  - IdProf: Id del profesor.

- Professor: Profesor.
- PunishmentDate: Fecha de la sanción.
- **Restriction:** Entidad que representa Restricciones de aulas.
  - IdRes: Id de la entidad.
  - IsDeleted: Indica si ha sido eliminado de la base de datos.
  - NameRes: Nombre de la restricción.
  - ClassRooms: Aulas a las que ha sido aplicada.
- **Secretary:** Entidad que representa Secretarios.
  - IdS: Id de la entidad.
  - IsDeleted: Indica si ha sido eliminado de la base de datos.
  - NameS: Nombre del secretario.
  - UserId: Id de usuario de la entidad.
  - SalaryS: Salario del secretario.
- **Subject:** Entidad que representa Asignaturas.
  - IdSub: Id de la entidad.
  - IsDeleted: Indica si ha sido eliminado de la base de datos.
  - NameSub: Nombre de asignatura.
  - StudyProgram: Programa de estudio.
  - CourseLoad: Horas clases.
  - IdClassRoom: Id del aula en que se imparte dicha asignatura.
- **ClassRoomRestriction:** Relación de aula y restricción.
  - IdClassRoomRest: Id de la entidad.
  - IdClassRoom: Id del aula.
  - IdRest: Id de la restricción
- **ClassRoomTechMean:** Relación de aula y medios tecnológicos.
  - IdClassRoomTech: Id de la entidad.
  - IdClassRoom: Id del aula.
  - IdTechMean: Id de medio tecnológico.
- **ProfessorClassRoom:** Relación de aula y restricción.
  - IdProfClass: Id de la entidad.
  - IdClassRoom: Id del aula.
  - IdProd: Id del profesor.
- **ProfessorStudentSubject:** Relación de profesor y la relación StudentSubject.

- IdProfStudSub: Id de la entidad.
- IdStudSub: Id de estudiante en asignatura.
- IdProd: Id del profesor.
- StudentGrade: nota del estudiante
- **ProfessorSubject:** Relación de profesor y la relación StudentSubject.
  - IdProfSub: Id de la entidad.
  - IdSub: Id de asignatura.
  - IdProd: Id del profesor.
- **ProfStudSubCourse:** Relación de profesor con estudiante, curso y asignatura; representa la evaluación del profesor.
  - IdProfStudSubCourse: Id de la entidad.
  - IdSub: Id de asignatura.
  - IdProd: Id del profesor.
  - IdStud: Id del estudiante.
  - IdCourse: Id del curso.
  - Evaluation: Evaluación del profesor.
- **SecretaryProfessorStudentSubject:** Relación de secretario con estudiante, profesor y asignatura; permite cambiar o dar notas de alumnos.
  - IdSecProfStudSub: Id de la entidad.
  - IdSub: Id de asignatura.
  - IdProd: Id del profesor.
  - IdStud: Id del estudiante.
  - IdSec: Id de secretario.
  - Evaluation: Evaluación del estudiante.
- **StudentSubject:** Relación de estudiante con asignatura
  - IdStudSub: Id de la entidad.
  - IdSub: Id de asignatura.
  - IdStud: Id del estudiante.
- **SubjectAuxMean:** Relación de asignatura con medio auxiliar.
  - IdSubAuxMean.
  - IdSub: Id de asignatura.
  - IdAuxMean: Id del medio auxiliar.
- **MeanNotification:** Notificación de medios
  - MeanNotId: Id de la entidad.

- MeanName: Nombre del medio.
  - MeanType: Tipo de medio.
  - BeenSended: Indica si ya se ha hecho la notificación.
  - MeanId: Id del medio.
- **ProfessorNotification:** Notificación de profesores.
    - ProfNotId: Id de la entidad.
    - ProfName: Nombre del profesor.
    - BeenSended: Indica si ya se ha hecho la notificación.
    - IdProf: Id del profesor.

## Esquema con el Diseño de la App

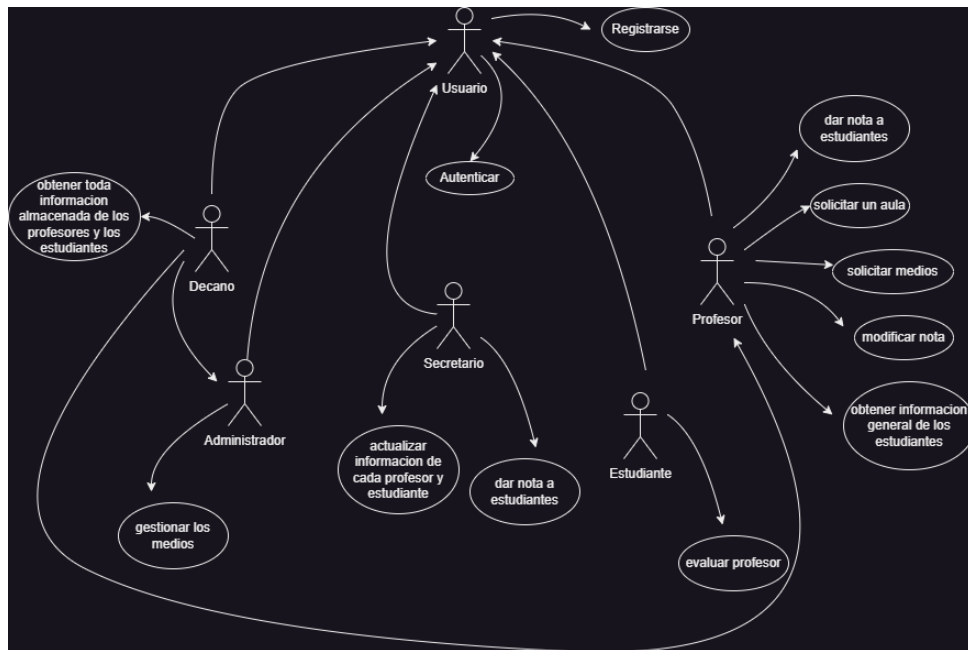


Figure 1: Esquema del diseño de la app

## 1 Esquema de Clases Definidas

Presentar un diagrama de clases que muestre las relaciones entre las clases definidas en el software.

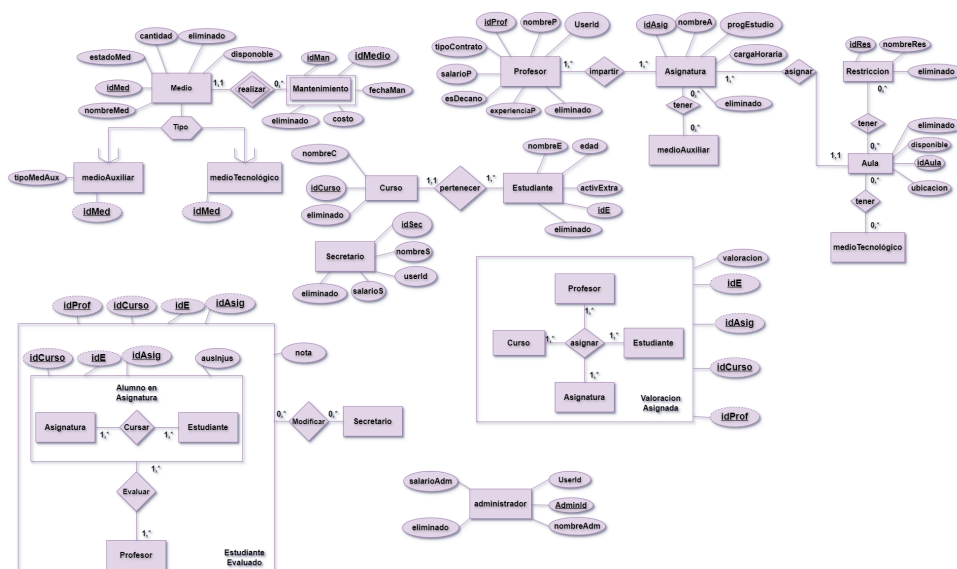


Figure 2: Esquema de clases definidas

## Solución Propuesta

## Arquitectura Empleada

Para el proyecto se decidió utilizar la arquitectura limpia (Clean Architecture), debido a la magnitud del software en cuanto a cantidad de entidades y sus interrelaciones, además de múltiples funcionalidades. Se considera que esta arquitectura ayudaría a organizar mejor su comportamiento interno, así como a identificar con mayor precisión la funcionalidad de cada uno de los scripts. A continuación, se mostrará una descripción de la funcionalidad de cada capa que utilizaremos en el software.

- **Capa de Api:** La capa API o de presentacion es la más externa de la arquitectura Clean. Se encarga de la interfaz entre el sistema y los usuarios finales. En el contexto de este proyecto de gestion escolar, esta capa podría incluir controladores web, vistas moviles, interfaces gráficas, etc. Su función principal es recibir solicitudes entrantes y procesarlas para producir respuestas adecuadas. La capa API actua como puente entre el sistema y los clientes, adaptando los datos de entrada segun lo necesario y asegurandose de que se siguen las reglas de validación establecidas. Es aquí donde se definen las rutas de las APIs, se manejan los formatos de datos, se gestionan las sesiones de usuario y se configuran los aspectos relacionados con la interfaz de usuario. La capa API debe ser independiente de las otras capas internas, permitiendo cambiar tecnologías o frameworks sin afectar el core del sistema.
- **Capa de Aplicación:** La capa de aplicaciones es la encargada de coordinar entre las capas del dominio e infraestructura. Contiene la logica de aplicación que ejecuta

las operaciones de negocio definidas en la capa del dominio. Maneja la logica empresarial compleja, como calcular promedios de las valoraciones de los profesores, notas generales de los estudiantes, etc., utilizando los servicios de infraestructura necesarios. Esta separacion permite que la lógica de negocios sea independiente de la implementacion específica de la infraestructura.

- **Infraestructura:** La capa de infraestructura se encarga de la implementacion de los componentes necesarios para que el sistema funcione. Esta capa interactua con dependencias externas como bases de datos, sistemas operativos y hardware. Aquí se implementan interfaces definidas en la capa del dominio utilizando herramientas y tecnologías específicas. Tambien puede incluir clases para gestionar el acceso a archivos, configuraciones del sistema, etc.
- **Capa de Dominio:** La capa del dominio es el nucleo más interno de la arquitectura Clean. Contiene las entidades fundamentales de la aplicacion. Esta capa no depende de ninguna otra parte del sistema, en ella se definen los conceptos clave del dominio, como clases e interfaces. La capa del dominio es independiente de cualquier tecnología o framework específico, lo que permite una mayor flexibilidad y reutilizacion del codigo. Es aquí donde se definen los conceptos basicos de la aplicación, como en un sistema de gestion escolar se definen clases como Estudiante, Profesor, Asignatura, etc.

## Patrones de diseño

- **Repository:** El patron Repository permite separar claramente la lógica de acceso a datos de la logica de negocio. Esto fomenta un diseño más limpio y modular en nuestra aplicacion al aislar la lógica de acceso a datos en una capa específica (esto concuerda con la arquitectura empleada para el desarrollo de nuestro software). Al utilizar el patron Repository, se centralizan todas las operaciones de acceso a datos en un solo lugar, lo cual evita la duplicacion de código y promueve un mejor mantenimiento y escalabilidad del sistema. Ademas, con este patrón se encapsula la logica de acceso a datos, lo que facilita la transición entre diferentes tecnologías de almacenamiento de datos sin afectar la logica de negocio de la aplicación. Esto es esencialmente util para nuestro software ya que nuestro cliente no pone restricciones específicas de entorno, luego la gestion de los datos que utiliza nuestro programa no debe estar sujetos a tecnologías específicas de almacenamiento
- **Data Mapper:** Como patrón de acceso usamos el Data Mapper, un patrón de diseño que tiene como objetivo principal desacoplar la lógica de negocio de la persistencia de datos. Como ventajas tiene facilitar el mantenimiento y la escalabilidad al mantener la logica de negocio limpia y soportar operaciones complejas en la base de datos sin exponer detalles internos al dominio

## Descripción de los Módulos

Aquí se detallan los diferentes módulos que componen el software:

- **Microsoft.EntityFrameworkCore:** Este es el núcleo del Entity Framework Core. Proporciona las funcionalidades básicas para interactuar con bases de datos, incluyendo la configuración del contexto y la definición de entidades.



- **Microsoft.EntityFrameworkCore.SqlServer:** Este módulo permite a EF Core interactuar con bases de datos SQL Server. Es necesario si tu aplicación utiliza SQL Server como sistema de gestión de bases de datos.
- **Microsoft.EntityFrameworkCore.Tools:** Este paquete proporciona herramientas para trabajar con EF Core desde la línea de comandos o desde Visual Studio, incluyendo la capacidad de realizar migraciones y generar código.
- **AutoMapper:** AutoMapper es una biblioteca de .NET diseñada para simplificar el mapeo entre objetos, facilitando la transferencia de datos entre diferentes modelos, como entidades y objetos de transferencia de datos (DTOs)
- **Microsoft.AspNetCore.Authentication.JwtBearer:** Es un componente de middleware en ASP.NET Core que facilita la autenticación de portador utilizando JSON Web Tokens (JWT). Este módulo permite a las aplicaciones validar y procesar tokens JWT que se envían en el encabezado de autorización de las solicitudes HTTP, asegurando que solo los usuarios autenticados puedan acceder a recursos protegidos.