



**Universidad de las Fuerzas Armadas**

**ESPE Departamento de Ciencias de la Computación**

**Análisis y Diseño**

**Diagrama de caso uso y uso extendido**

**Integrantes:**

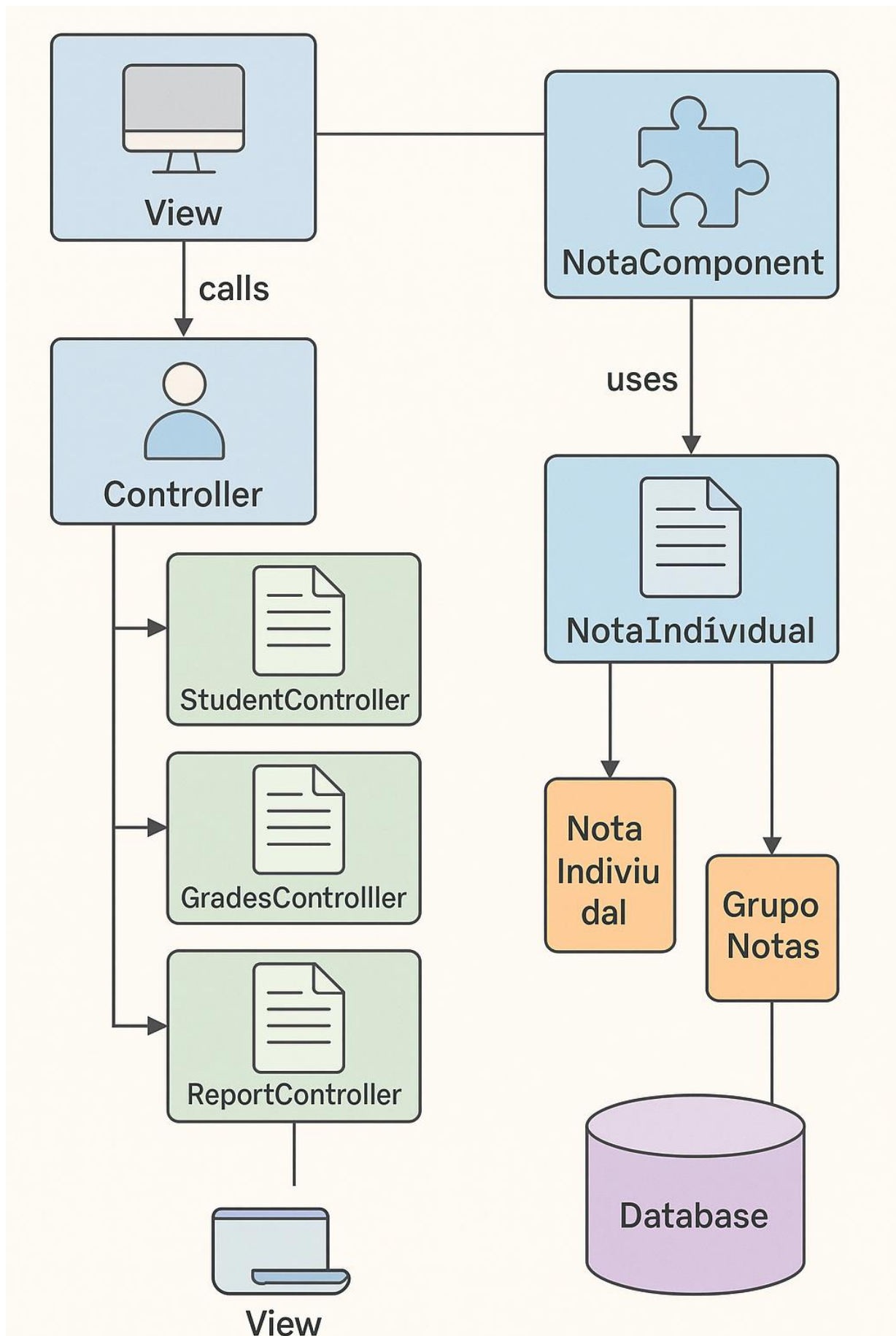
Isaac Escobar

Eduardo Mortensen

Diego Ponce

**NRC: 14571**

## Patrón de diseño



1. **Usuario → View**  
El actor “Usuario” interactúa con la **View** (React/Vite). Esto se muestra con la flecha “interactúa con”, que conecta el icono de usuario al recuadro **View**.
2. **View → Controller**  
La **View** llama al **Controller** (`server.js`) a través de HTTP. En el diagrama aparece la flecha etiquetada “llama a” que une ambos recuadros.
3. **Controller → AuthController / StudentController / GradesController / ReportController**  
Desde el **Controller** salen cuatro flechas que apuntan a cada uno de los subcontroladores:
  - **AuthController** (ruta `/auth`)
  - **StudentController** (ruta `/students`)
  - **GradesController** (ruta `/grades`)
  - **ReportController** (ruta `/reports`)
4. **Cada Controller → NotaComponent**  
De cada subcontrolador parte una flecha con la etiqueta “uses” hacia la interfaz **NotaComponent**, indicando que todos utilizan la misma abstracción de calificación.
5. **NotaComponent → NotaIndividual / GrupoNotas**  
Bajo la interfaz **NotaComponent** aparecen dos implementaciones:
  - **NotaIndividual** (una calificación atómica)
  - **GrupoNotas** (un conjunto de calificaciones)La relación de herencia se ve con la flecha de línea continua y el rombo hacia abajo para el composite.
6. **NotaIndividual / GrupoNotas → MongoDB Atlas**  
Finalmente, ambas clases tienen flechas que apuntan al icono de la **Database** (**MongoDB Atlas**), etiquetadas “save()/load()”, que muestran dónde se persisten y recuperan los datos de calificaciones.