

Diagrama de clases Explicacion

```
abstract class NotaComponent {  
    +getValor(): Number  
}
```

NotaComponent

Es la clase base **abstracta** para todo tipo de notas. Declara el método `getValor()` que debe ser implementado por sus subclases. Permite diseñar una arquitectura extensible con polimorfismo.

```
class NotaIndividual {  
    -valor: Number  
    +constructor(valor: Number)  
    +getValor(): Number  
}
```

NotaIndividual

Representa una nota concreta, como una calificación numérica simple.

- `valor`: almacena la calificación.
- `getValor()`: devuelve esa nota.

Es la implementación más básica de `NotaComponent`.

```
class GrupoNotas {  
    -notas: NotaComponent[]  
    +constructor()  
    +agregarNota(nota: NotaComponent): void  
    +getValor(): Number  
    +getComponente(): Object  
}
```

GrupoNotas

Modela un conjunto de notas individuales o grupos anidados.

- `notas`: arreglo de componentes que extienden `NotaComponent`.
- `agregarNota()`: añade un componente al grupo.
- `getValor()`: calcula el promedio de los valores del grupo.
- `getComponente()`: obtiene la estructura del grupo.

Esta clase permite crear estructuras jerárquicas de calificaciones (ej. examen dividido por secciones).

```
class User {  
    +_id: ObjectId  
    +name: String  
    +email: String  
    +password: String  
    +role: Enum('admin', 'teacher', 'student', 'parent', 'guest')
```

```

+childrenIds: ObjectId[] [opcional]

+comparePassword(inputPassword: String): Promise<Boolean>
+isAdmin(): Boolean
+canRegisterUsers(): Boolean
}

```

User

Define el esquema de usuarios del sistema con roles diferenciados.

- childrenIds: útil para padres que siguen el rendimiento de sus hijos.
- Métodos:
 - comparePassword: verifica la contraseña.
 - isAdmin: valida si el usuario tiene permisos elevados.
 - canRegisterUsers: determina si puede registrar nuevos usuarios.

```

class Grade {
  +_id: ObjectId
  +studentId: ObjectId
  +subject: String
  +lecciones: Number
  +proyectos: Number
  +exámenes: Number
  +promedio: Number

  +calculatePromedio(): Number
}

```

Grade

Representa una calificación de una materia específica.

- Contiene las subnotas (lecciones, proyectos, exámenes) y el promedio.
- calculatePromedio(): aplica la fórmula ponderada 30–30–40 para obtener el valor final.

Análisis estructurado

Parte 1: Arquitectura de notas

- Usa **herencia y polimorfismo** con `NotaComponent` para permitir múltiples tipos de notas.
- `NotaIndividual` y `GrupoNotas` son implementaciones concretas. Esta estructura es escalable para incluir nuevas estrategias de evaluación.

Parte 2: Gestión de usuarios

- `User` centraliza el control de acceso y permisos por rol.
- Soporta lógica para padres (`childrenIds`) y funciones orientadas a seguridad (`comparePassword`).

- Esto facilita autorización condicional y control de vistas en los endpoints.

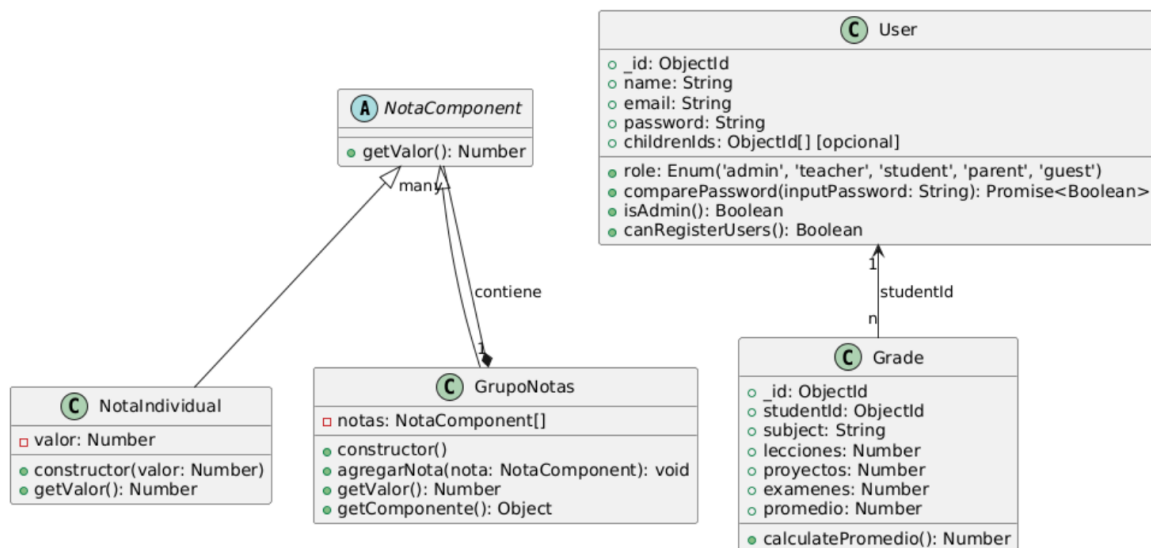
Parte 3: Modelo de calificaciones

- Grade vincula la nota con un estudiante (studentId) y una materia (subject).
- Tiene descomposición por tipos de evaluación y cálculo automático del promedio.

Parte 4: Relaciones entre clases

Relación	Tipo	Descripción	
`NotaComponent <`	-- NotaIndividual / GrupoNotas`	Herencia	Polimorfismo para implementar diferentes tipos de notas.
GrupoNotas *-- NotaComponent	Composición	Un grupo contiene muchas notas.	
User "1" <-- "n" Grade	Asociación	Un estudiante puede tener muchas calificaciones.	

DIAGRAMA DE CLASES:



ZLJ9ZjD04BttLxHoSGR8WQilE04HGhc420INA4ALxXfFeLwiNW8ZuAlu17wCxhR76uC
 q5wlyQdlrgknh7y25g1LZ5RzHu96VymFEFZWG0TY3nKNoUhwt0MwiRgr14_WFnldp1i
 DdKDQjhsg-

Z_gCZlqQoxNWW9kKvJaPIHUI4HHwDfRR3N2fjxdaOaEfr41OapX44LAFcVFgYUqR5rIB

9NOy-jRlChWln4hRugld0nvFo_wb9JGE6t0vT9tJ5bc9oCMI_0-v3Xs2CRIUdx-
Y29dtND_jpyrceT5Z4UCeR3BLwkMAFqIUbGPPlrfihkPfga1eSDGg-WYERA_F9u-
kM_0NaWyKTsKg0nfh_Z4uCar-
Hmsafa2R2dopJaunPrLAUckYNbSWDPdg1Qy2WhX3bqqVeapCifcMrbjDH1-
g8hgu8oKJlfd-P7CyyQDj1La3wilwPUcSVkWPhCcqCHmMV9Ba1sSrUTo-
iOaNc5Svb_phJ2nlf8TBHJ0Vi24Vq6LD_DGzkRHqNW8KtP80GNz-c-
7C87aUrx4VURU4OtcPYgTGv27HJy-eTVO-
nTaPYD_JSluDr2Z93bWtbX9HGS13xvr_G6Nz-Ols-OJCgX-avWDFngxHoFpl-0i0