

## INGENIERÍA DE SOFTWARE

## DISEÑO

EQUIPO 3

XIN WEN ZHANG LIU 422087995

MENCHACA CARRILLO RODOLFO JOSUÉ 315294165

CERVANTES DUARTE JOSE FERNANDO 422100827

EVANGELISTA TIBURCIO JOSE MANUEL 422007939

ZAVALA HERNANDEZ ANGEL BORIS 318100021

**I Clases**

Como se describió dentro de las especificaciones buscamos 4 clases principales para la funcionalidad principal de nuestra aplicación, estas siendo *Cards*, *Notes*, *Decks*, *Notebooks*.

Otra clase principal es la del usuario, a la cual se le asocian como dueño de instancias específicas de las clases mencionadas anteriormente.

Los datos que se necesitan para el algoritmo de *spaced repetition* está contenido dentro de la clase *Card*, a la cual se le modifica la fecha en la que será estudiada, de acuerdo a la respuesta provista por el usuario. De manera simplificada y somera, si el usuario responde bien esta carta (y la califica como fácil), entonces el algoritmo no volverá a mostrarla hasta después de un rango de tiempo prolongado; y viceversa.

Para que el algoritmo funcione correctamente se crean múltiples clases temporales para realizar cálculos y almacenar información necesaria para estos. Estos no se incluyen en el diagrama de clases ya que son instancias temporales que no quedan almacenadas dentro de la base de datos.

Cabe destacar que la clase *Flashcard* hereda de *Card* ya que una tarjeta puede tener múltiple formas, y hacer que *Card* sea una entidad padre que guarde información común entre diferentes tarjetas nos ayuda a en el futuro expandir y poder agregar formatos diferentes de tarjetas que se puedan estudiar con el algoritmo de *spaced repetition*.

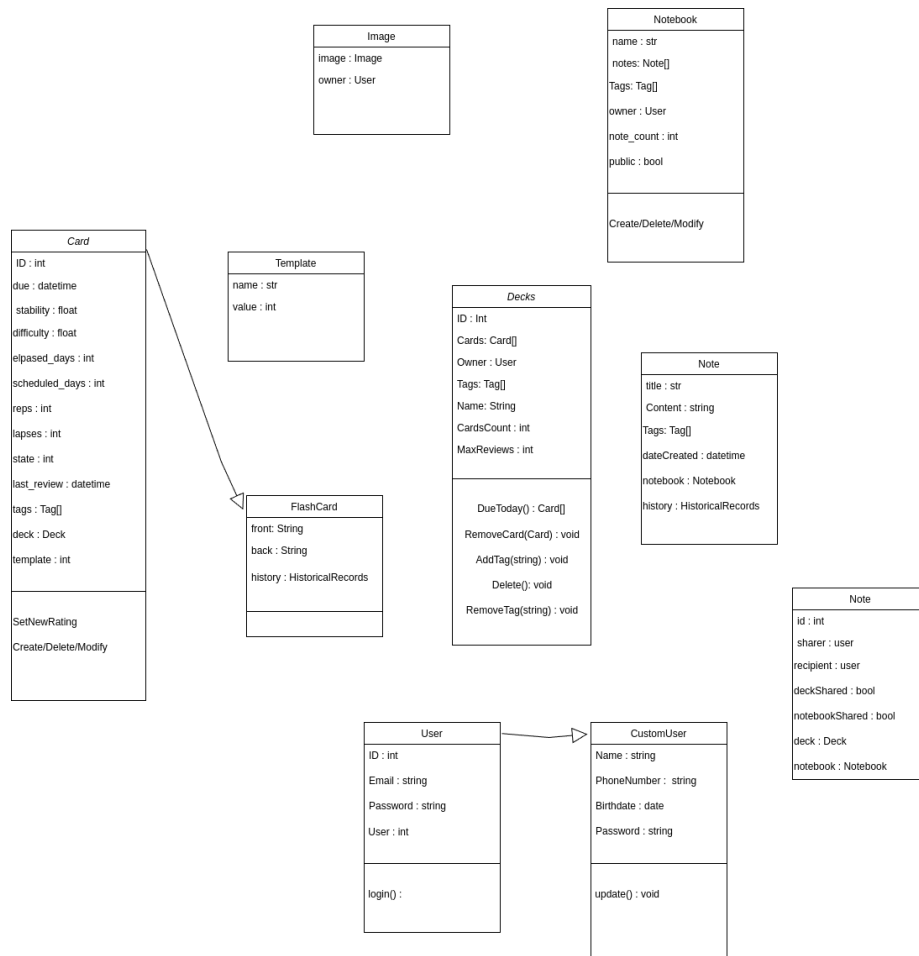


Figure 1: Diagrama de clases

## 2 Modelo Entidad Relación

El siguiente es un simple diagrama Entidad-Relación el cual omite las columnas de las entidades ya que son las mismas descritas en el diagrama de clases. Lo que se busca principalmente mostrar en este diagrama son las relaciones que existen entre las entidades principales presentes en la aplicación.

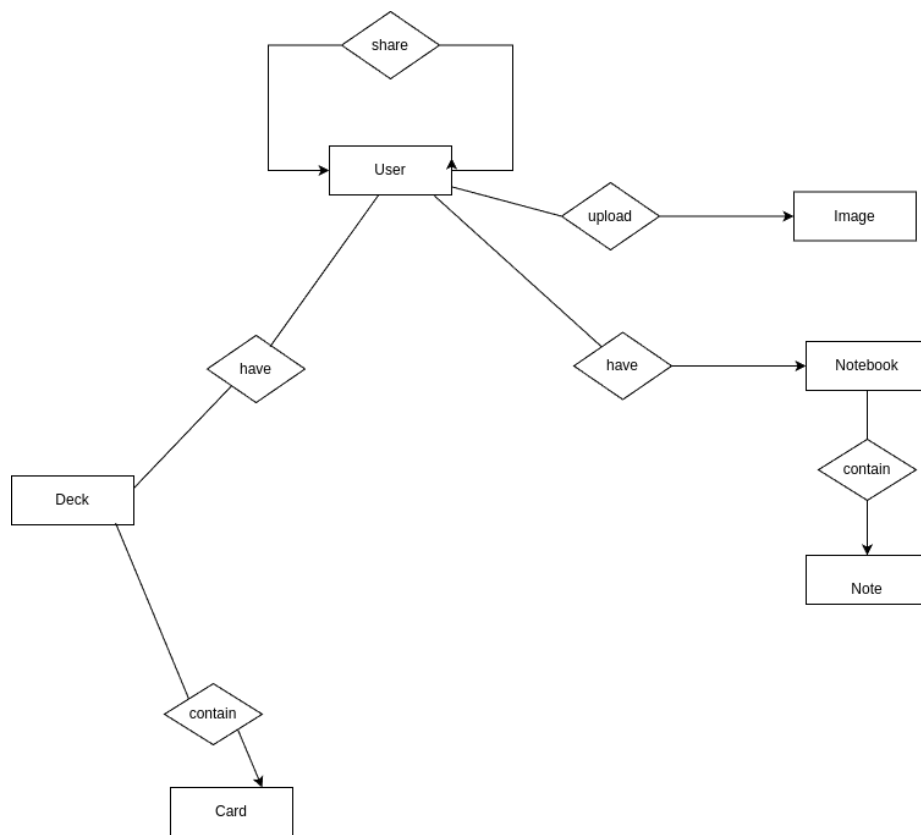


Figure 2: Diagrama Entidad-Relación

### 3 Casos de Uso

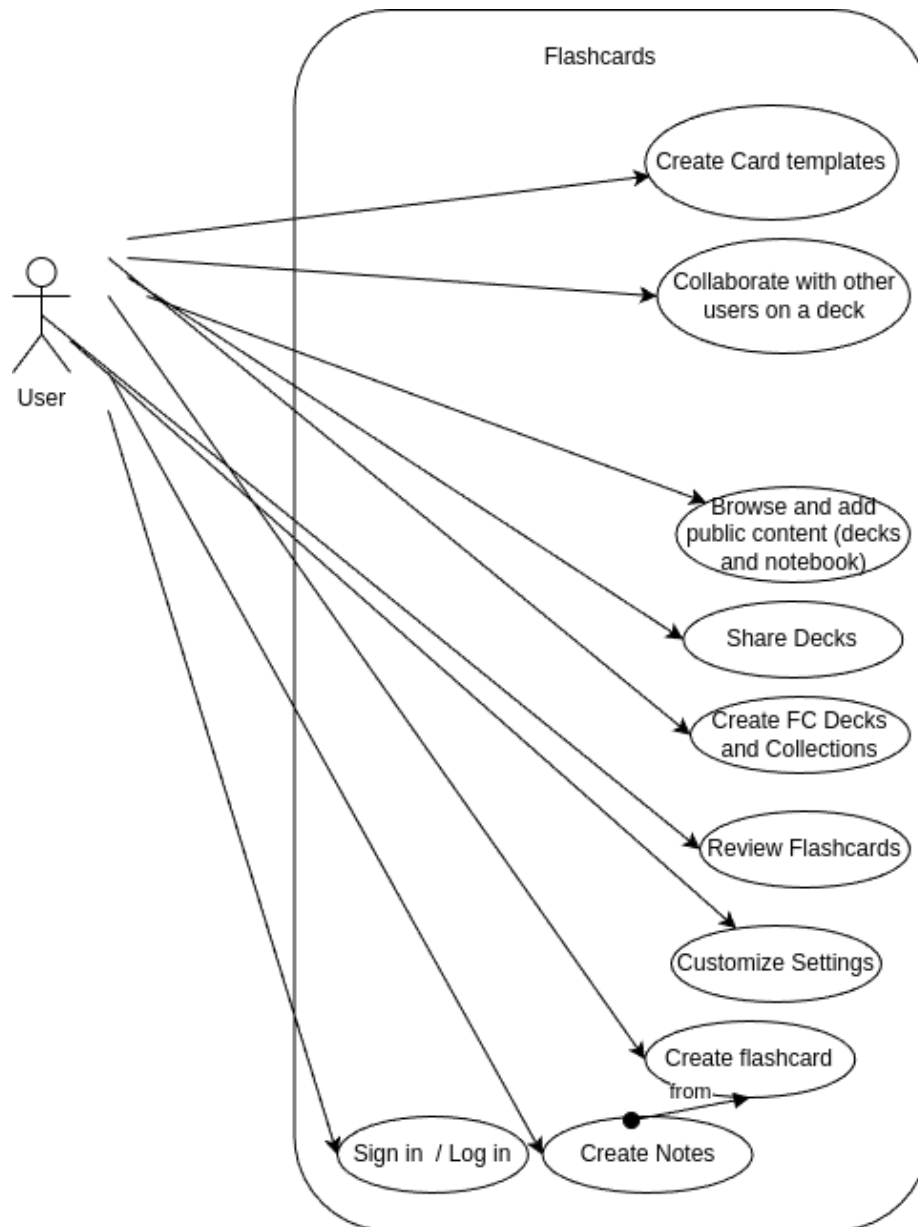


Figure 3: Casos de uso

Los principales casos de uso giran alrededor de el poder crear materiales de estudio como notas y tarjetas. Además de la habilidad de poder interactuar con otros usuario compartiendo y publicando sus propios *decks* y *notebooks*.

En la creación de cartas, existe la posibilidad de elegir plantillas que comprenden los diver-

sos casos de uso. Markdown, para temas y/o conceptos generales, sin necesidades específicas.  $\LaTeX$ , para notaciones más rigurosas y asociadas a las matemáticas. E imágenes, para satisfacer las necesidades de usuarios cuyo método principal de estudio esté asociado a la percepción visual.

A pesar de que no existe una colaboración en tiempo real, la edición de cartas, en conjunto con el control de versiones, hacen de este sistema uno capaz de proveer eficientemente una red de organización para el estudio conjunto.

El usuario podrá acceder a la aplicación a través de un correo electrónico. Sin embargo, no existe verificación alguna de la validez de este correo. A la cuenta de usuario se le asocian los *decks* y *notebooks* los cuales contienen las tarjetas y notas creadas, de la misma manera cada imagen agregada por el usuario es asociado a este en su respectiva carpeta dentro de *media*.

La funcionalidad más importante está comprendida por el repaso de cartas. Es aquí donde el algoritmo FSRS es usado para brindar una base sólida de teoría sobre modelos de estudio al sistema.