
Diseño Estructural

Para

The Hive

Versión 1.0 no aprobado

Preparado por Alejandro Villarreal (Scrum Máster)

Universidad Tecnológica de Bolívar

24 de marzo de 2025

1. Introducción

El siguiente Diseño Estructural para *The Hive* responde a la necesidad de una empresa de medios de unificar el intercambio de datos entre sus negocios de streaming, publicaciones digitales, publicidad y prensa; y el manejo de su oferta de contenido y suscripciones adaptadas a cada usuario. En este, se describirá detalladamente la arquitectura, la organización de la aplicación, entre otros.

2. Arquitectura General del Sistema

La aplicación está compuesta por dos grandes componentes: un frontend desarrollado en Flutter y un backend desarrollado en FastAPI con Python. Ambos están conectados mediante una API RESTful que permite la comunicación entre la interfaz de usuario y la lógica del negocio.

2.1. MVC: Model, View, Controller

La aplicación se desarrollará bajo el modelo arquitectónico MVC.

Se eligió este modelo pensando en la escabealidad de la aplicación. Al ser una aplicación pequeña, se priorizó la rapidez, la ligereza, y conectividad entre frentes que brinda este modelo al momento de organizar los datos.

2.2. Comunicación y seguridad

El frontend y el backend se comunicarán a través de peticiones HTTP en formato JSON. FastAPI expone endpoints que Flutter consumirá utilizando librerías como http.

Se utilizará JWT (JSON Web Tokens) para la autorización de usuarios, garantizando una comunicación segura entre cliente y servidor.

3. Backend

3.1. Modelado de servicios bajo los principios SOLID

El backend de la aplicación se creará bajo el modelado de los microservicios que ofrecerá la aplicación, utilizando los principios SOLID.

3.2. Microservicios de la aplicación

The Hive dispondrá de tres dominios repartidos en nueve microservicios:

- **Dominio de Usuario:** El dominio usuario contará con dos microservicios: el microservicio encargado de controlar el usuario interno de la aplicación, y el microservicio encargado de controlar el usuario público, es decir, el perfil.
- **Dominio de Contenido:** Encargado de controlar los microservicios alrededor de la manipulación de las películas. Por parte de la aplicación, se centrará principal y

únicamente en todo lo relacionado con las listas de películas, ya que de las películas se encargará una API externa.

- **Dominio de Interacciones:** Centrado en todas las interacciones que realiza el usuario en la aplicación. Maneja: reacciones, comentarios, calificaciones, y acciones de seguimiento por parte de los perfiles.

3.3. Modelo relacional de datos y tablas derivadas

En esta sección se presenta el Modelo Entidad-Relación (MER) diseñado para representar las entidades clave de *The Hive* y las relaciones entre ellas. A partir del MER, se derivaron las tablas que conforman el modelo lógico de la base de datos relacional.

Modelo Entidad-Relación

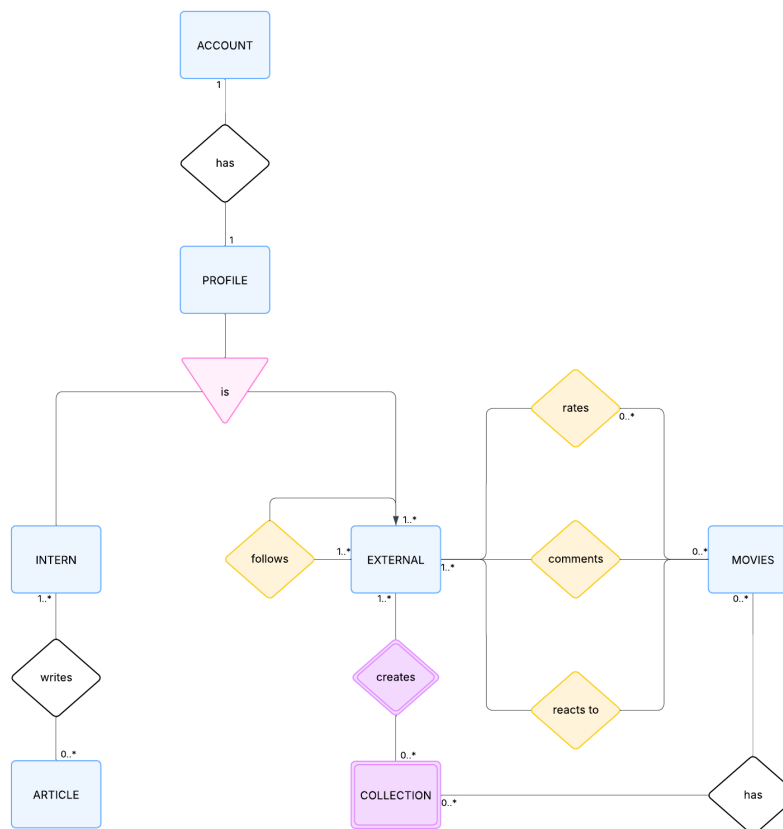


Figura 1. Modelo relacional de The Hive

Tablas derivadas

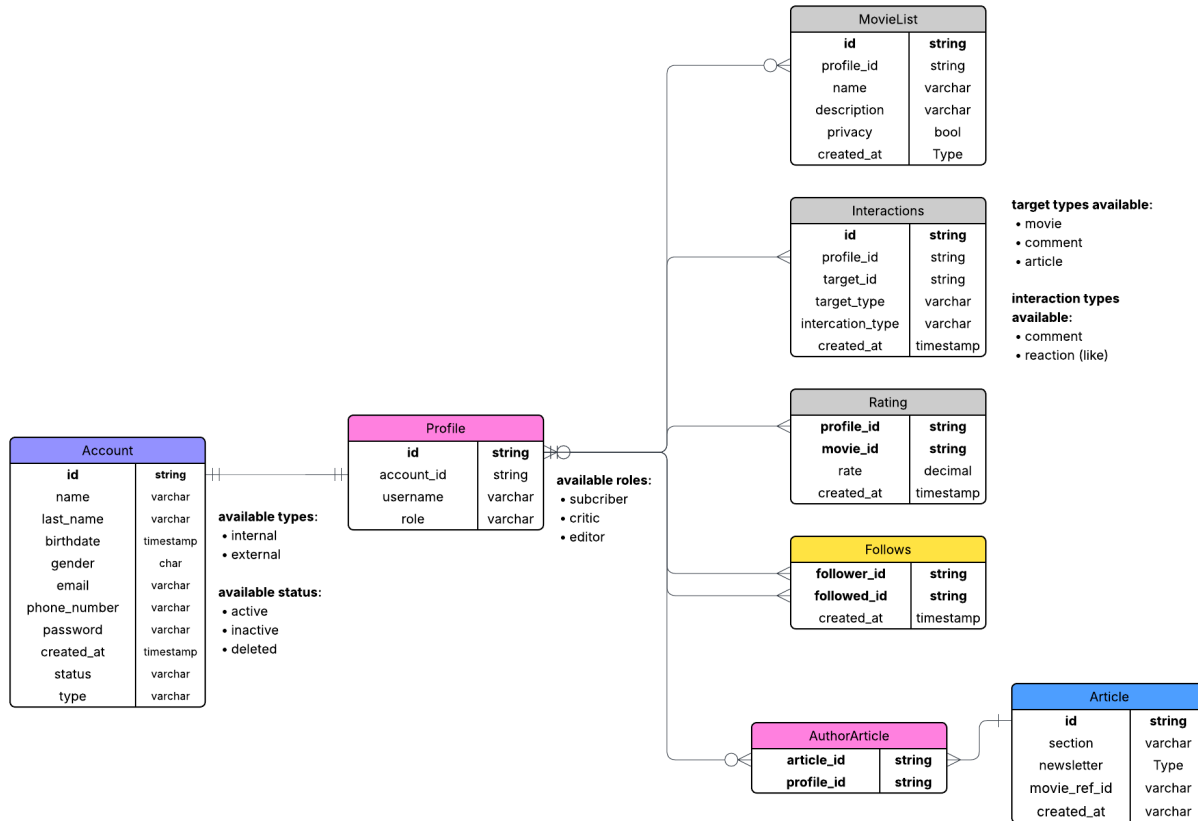


Figura 2. Tablas relacionales de The Hive

3.4. Modelo Máster-Detail

La aplicación manejará un modelo Master-Detail, lo que permitirá establecer relaciones jerárquicas entre entidades. Para representar y manejar estas estructuras jerárquicas de datos, se utilizarán archivos en formato JSON.

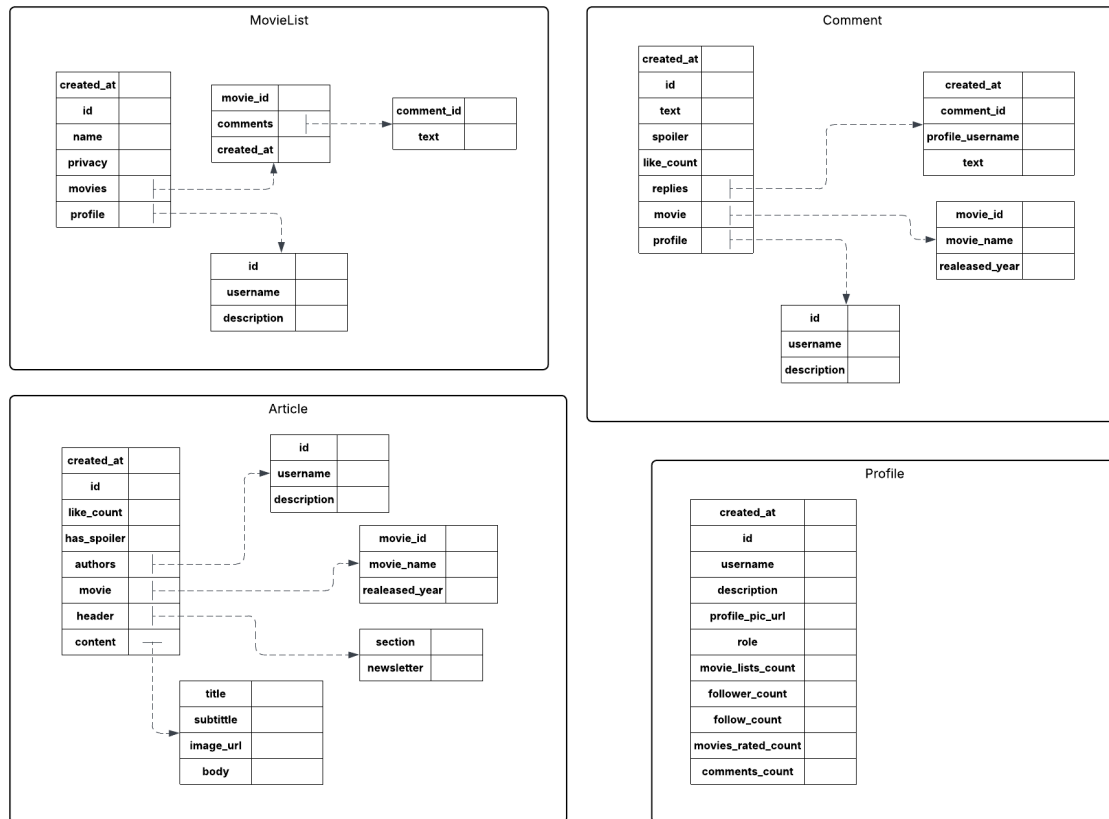


Figura 3. Entidades en formato JSON

4. Frontend