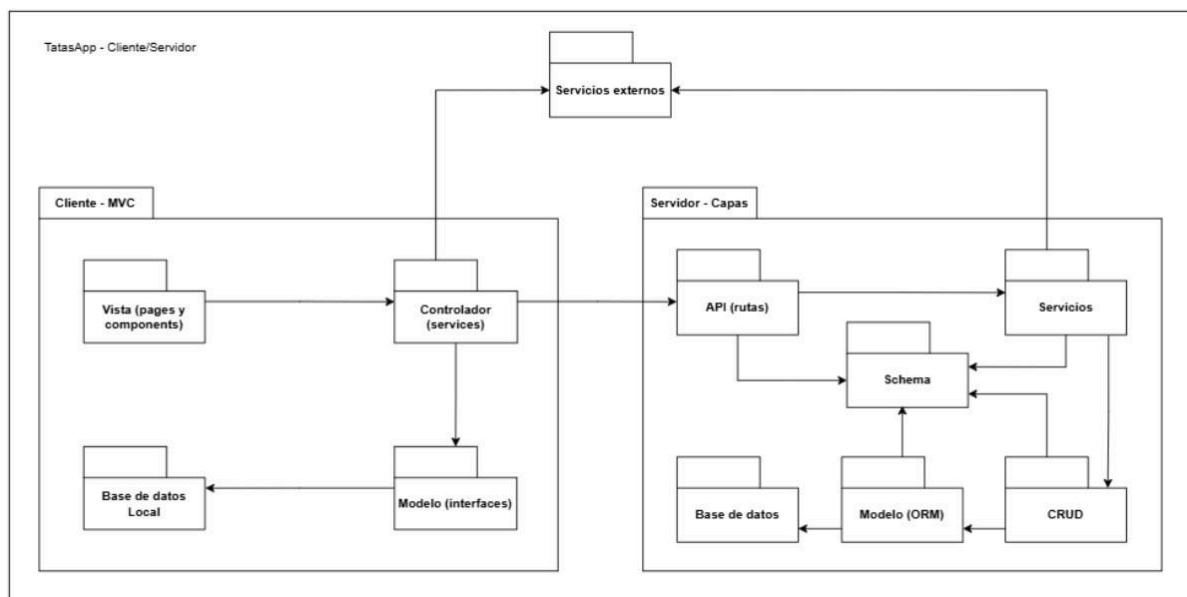


El sistema ha sido diseñado bajo una arquitectura cliente-servidor, que permite separar las responsabilidades entre la aplicación móvil (cliente) y la lógica de negocio junto con la gestión de datos (servidor). Esta decisión se fundamenta en la necesidad de mantener una comunicación eficiente y estructurada mediante el uso de una API REST, facilitando así la escalabilidad, el mantenimiento del sistema y la posibilidad de integrar servicios externos como Google Maps, Google Calendar, notificaciones push, entre otros.

A nivel del cliente, desarrollado con Ionic y Angular, se implementó una arquitectura MVC (Modelo-Vista-Controlador). Esta estructura organiza el código en componentes que representan la interfaz de usuario (Vista), los servicios que gestionan la lógica y conexión con la API (Controlador), y los modelos de datos que representan la información manejada por la aplicación (Modelo). Además, se contempla una base de datos local para operar de forma offline cuando sea necesario, lo que complementa la autonomía del usuario.

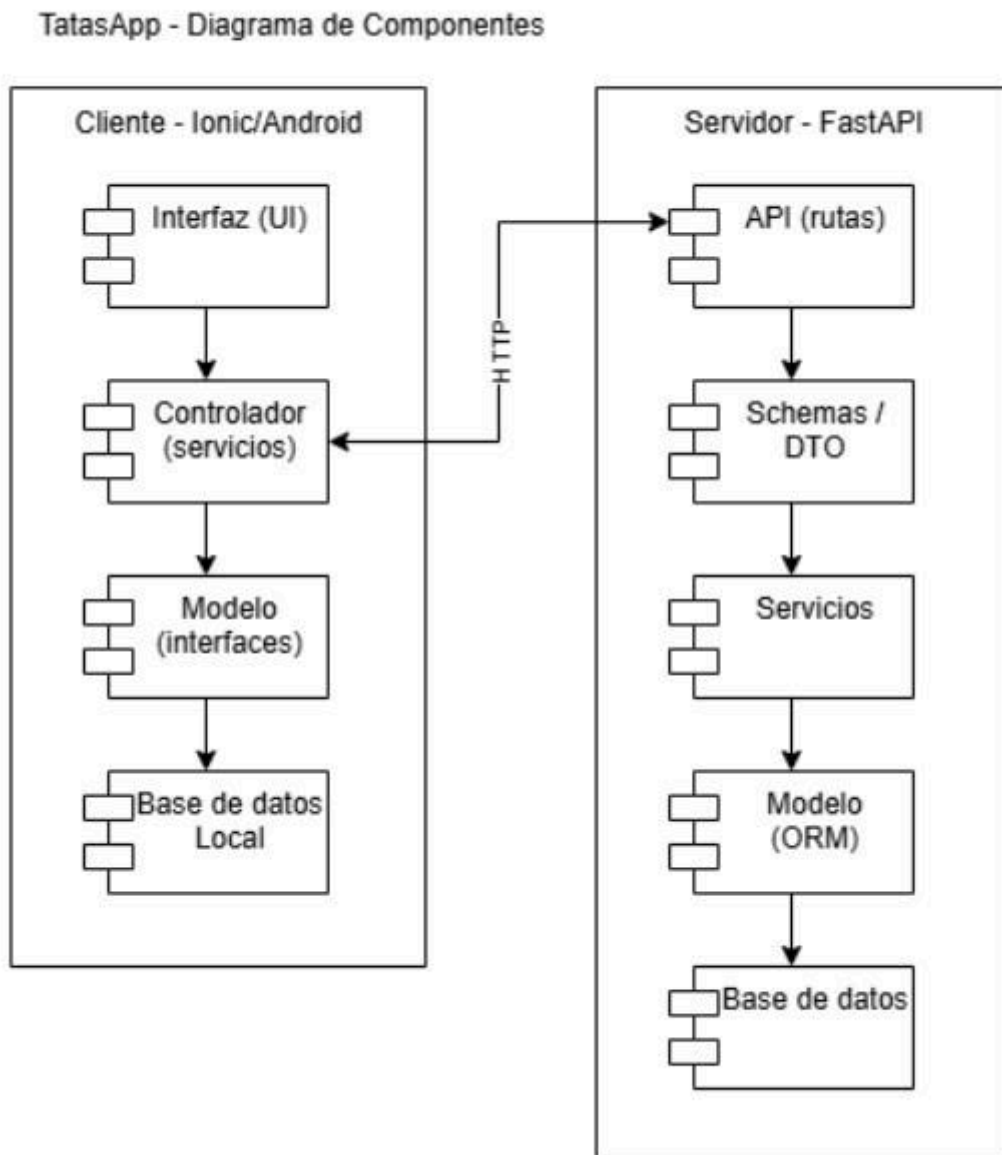
Por otro lado, en el servidor, construido con FastAPI y conectado a una base de datos relacional (MySQL o PostgreSQL), se adopta una arquitectura en capas, donde cada responsabilidad está claramente definida. La capa de rutas (API) se encarga de recibir las peticiones HTTP; la capa de servicios procesa la lógica de negocio; la capa de modelos define las estructuras de datos y la conexión con la base de datos, permitiendo una alta cohesión y bajo acoplamiento entre módulos.

Diagrama de paquetes



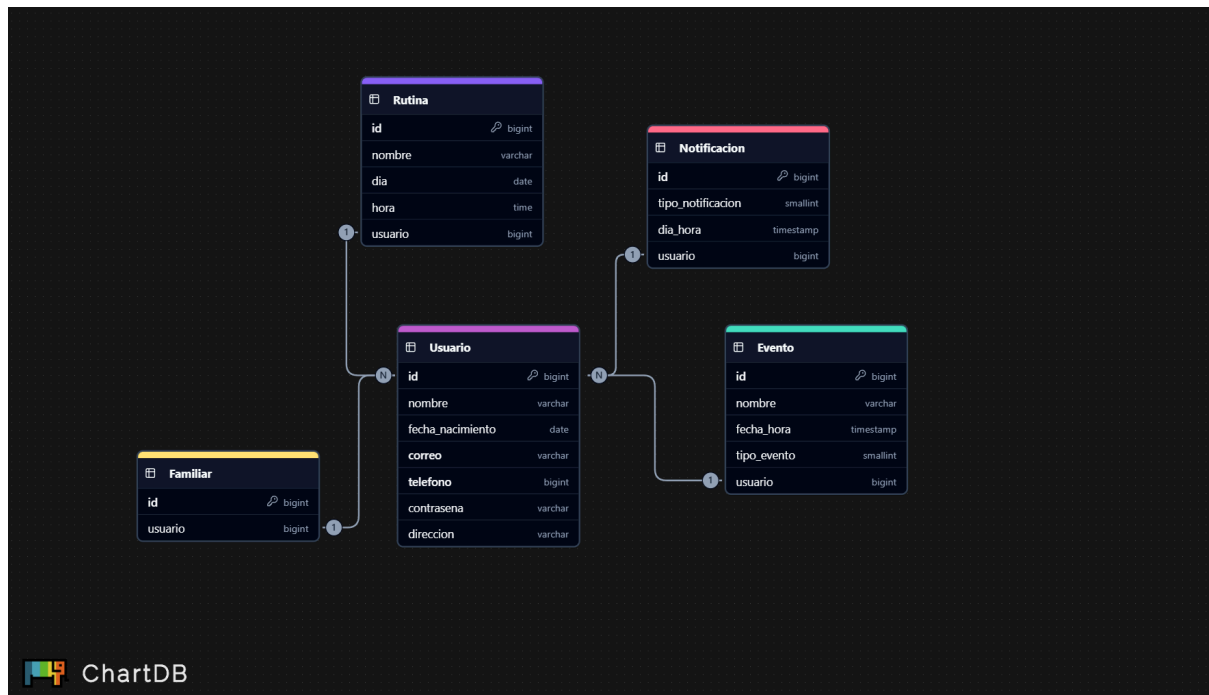
El diagrama de paquetes representa la organización modular del sistema TatasApp bajo una arquitectura cliente-servidor. Se divide en dos bloques: el cliente, estructurado con el patrón MVC, y el servidor, con arquitectura en capas. Cada paquete agrupa responsabilidades como interfaz, lógica de negocio, modelos y acceso a datos. La comunicación entre cliente y servidor se da desde el Controlador (services) del cliente hacia la API (rutas) del servidor mediante solicitudes HTTP. Además, ambos pueden interactuar con servicios externos, como APIs de Google o mensajería, para ampliar sus funcionalidades.

Diagrama de componentes



El diagrama de componentes muestra la distribución lógica de los principales módulos de TatasApp en su arquitectura cliente-servidor. En el cliente (Ionic/Android) se representan componentes como la interfaz, los controladores (services), modelos e interacción con una base de datos local. En el servidor (FastAPI), se distinguen rutas (API), esquemas/DTO, servicios, modelo ORM y la base de datos. La conexión entre ambos se realiza mediante llamadas HTTP desde el controlador del cliente hacia la API del servidor, permitiendo intercambiar información de manera estructurada.

Modelo Entidad-Relación



El modelo entidad-relación representa la estructura lógica de los datos utilizados en TatasApp. Se destacan las entidades principales: Usuario, Familiar, Rutina, Notificación y Evento, todas relacionadas a través de claves foráneas. La entidad Usuario actúa como eje central, conectándose con las demás mediante relaciones de uno a muchos. Este modelo permite organizar y acceder eficientemente a la información personal, actividades y alertas del usuario, facilitando la gestión de datos en la aplicación.

En conclusión, la arquitectura seleccionada busca mantener una clara separación de responsabilidades, facilitando el desarrollo colaborativo, la integración de tecnologías, y el cumplimiento de los requerimientos funcionales y no funcionales del proyecto, asegurando así una solución robusta, mantenible y escalable en el tiempo.