

Práctica FOTG

Diseño

Software

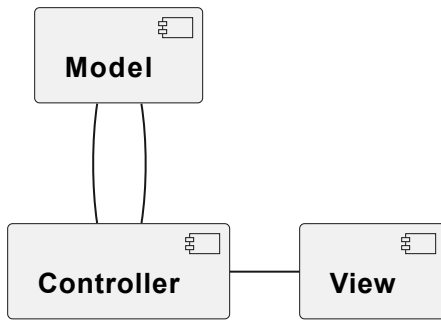
- > *Prototipo de las vistas* - Figma
- > *Diseño de bases de datos (MER y MR)* - Microsoft Whiteboard
- > *Implementación de bases de datos* - MySQL
- > *Hosting de la base de datos* - Amazon RDS
- > *Implementación del servidor HTTP* - Ktor con conectores JDBC.
- > *Hosting del servidor* - Render
- > *Implementación de controladores de vista, controladores de aplicación y DAO* - Kotlin
- > *Desarrollo de la aplicación móvil* - Android Studio con Kotlin

Versiones

- > *Conector JDBC* - MySQL Connector/J Type 4 v8.0.33
- > *Kotlin* - v1.19.0
- > *Gradle* - KTS v?.?.?

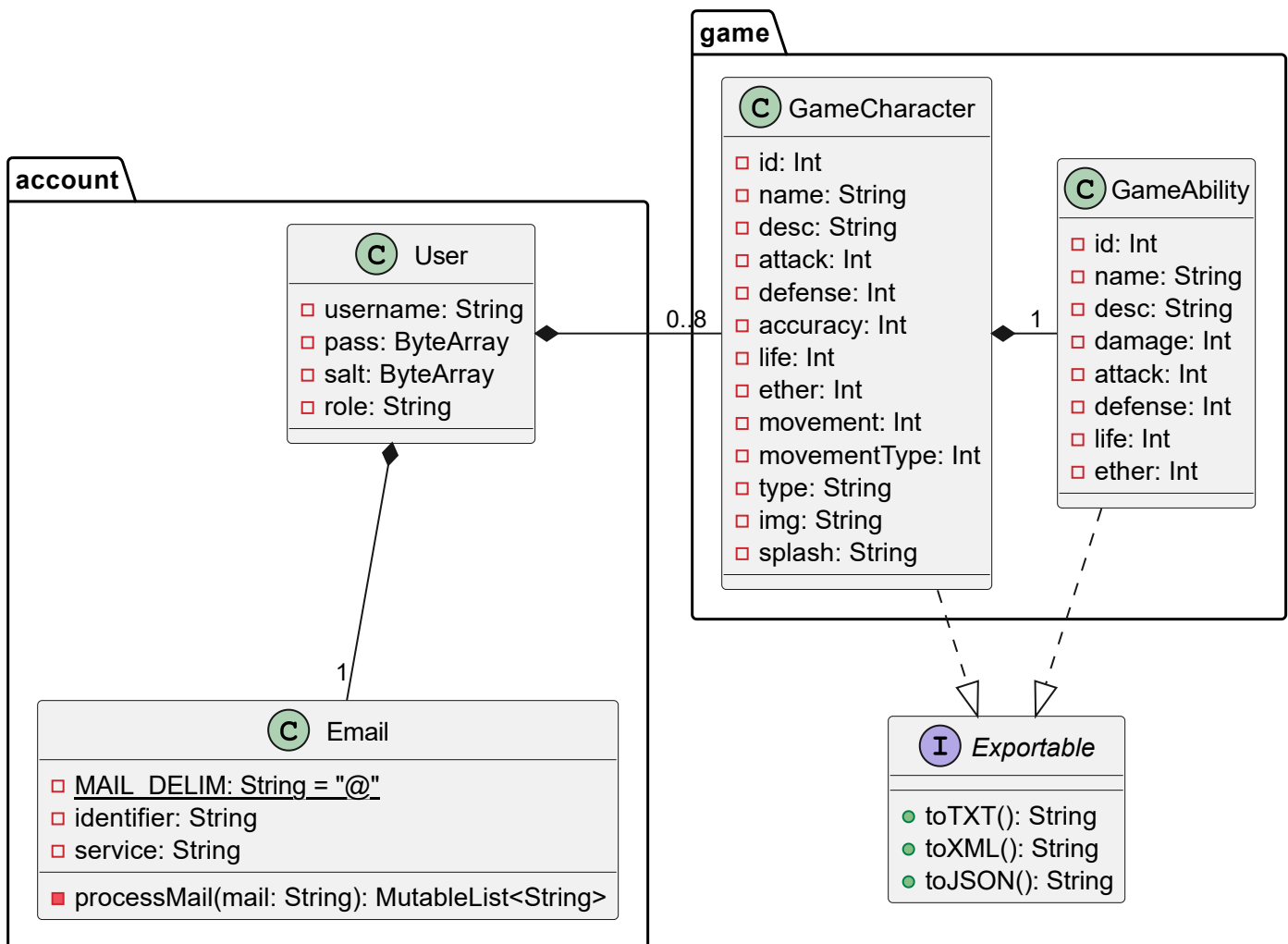
Arquitectura

La arquitectura de despliegue del proyecto se basa en una arquitectura **Cliente - Servidor** de 3 nodos (*Aplicación, servidor intermediario y base de datos*).



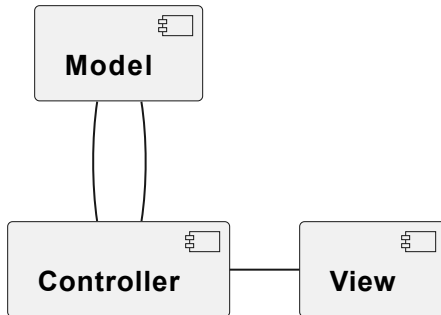
Para la arquitectura interna del servidor de **Ktor**, se ha tomado la decisión de utilizar el metapatrón de diseño **Modelo - Vista - Controlador**, ya que provee una gran adaptabilidad que es de gran utilidad para un servidor cuyo objetivo es servir de interfaz entre el alojamiento de la base de datos y la propia aplicación.

La arquitectura de dicho servidor modifica, por su naturaleza, el metapatrón de diseño, aligerando enormemente la vista, que queda relegada a ser un proveedor de servicios con acceso al controlador y al modelo del servidor.



Para la arquitectura interna de la **aplicación móvil** se ha decidido usar, de igual manera, el metapatrón de diseño **Modelo - Vista - Controlador**, esta vez no solo por su adaptabilidad, sino por su flexibilidad, permitiendo a la aplicación móvil acceder tanto a una base de datos externa, como a una interna al mismo tiempo, además de orquestar la navegación por vistas de manera muy efectiva.

En esta arquitectura, la vista cobra mucho más protagonismo y gana mucho más peso, pero el modelo también, pues la aplicación debe poseer la capacidad de acceder a dos servicios diferentes de bases de datos y procesar esa información.



Para el acceso a datos, tanto del servidor **Ktor**, como de la aplicación móvil, se utilizará el patrón de diseño **DAO** por su gran flexibilidad gracias a las factorías y su eficacia en el acceso a datos, además del patrón de diseño **Singleton** para dichas factorías de los accesos a datos.

Diagramas de Clases

