

Diseño de Base de la Datos

PAMN - Programación de Aplicaciones Mviles Nativas

Chamil José Cruz Razeq

20 de noviembre de 2023

1. Introducción

Todas las tareas propuestas se encuentran disponibles en el siguiente repositorio de GitHub.

2. Identificación de la capa de persistencia

Para el desarrollo de la aplicación planteada en el [?, Lab 3] se planteó aplicar el modelo MVVM [Figura ??], identificando los siguiente elementos:

- VistaAreas, ModeloVistaAreas y ModeloAreas: con información reducida de cada area [Figura ??].
- VistaAreaEspecifica, ModeloVistaAreaExpecifica y ModeloAreaEspecifica: con información detallada de un área especificada.
- VistaSOS, ModeloVistaSOS y ModeloSOS: con información de los medios de seguridad y emergencias.
- VistaSolicitud, ModeloVistaSolicitud y ModeloSolicitud: gestionando el formulario de solicitudes de uso de áreas recreativas.
- VistaSolicitudes, ModeloVistaSolicitudes y ModeloSolicitudes: gestionando las solicitudes de uso de áreas recreativas.

Figura 1: Diagrama de MVVM

Figura 2: Diagrama de MVVM

Donde encontramos los modelos relacionados con la persistencia de datos:

- ModeloAreas: conjunto de ModeloAreaEspecifica.
- ModeloAreaEspecifica: información sobre un area.
- ModeloSOS: información de los medios de seguridad y emergencias.
- ModeloSolicitudes: información sobre las solicitudes.

3. Selección de tecnología

De los modelos anteriores hemos identificado que algunos constituirán información que deberá ser actualizada en tiempo real, y otros que conformarán datos de carácter específico que no sufrirá cambios, al menos en corto plazo. Por este motivo se ha decidido utilizar “Rooms” [?, Rooms] para la capa local; por su flexibilidad y rapidez, además de gran documentación y aceptación en el desarrollo en android; y “Firebase” para la capa externa; por su fiabilidad y facilidad de aplicación.

4. Definición de un Modelo

Continuando con el apartado anterior, se ha tomado como referencia los modelos De advertencias y áreas, respectivamente. Como podemos observar en la imagen [Figura ??], para uno de los modelos se tiene en cuenta el uso de una API para solicitar la información a la base de datos alojada en Firebase, mientras que en el otro modelo se hace uso de la arquitectura de “Rooms”, definiendo una entidad y un “DAO” que se comunica con la base de datos en “SQLite”.

Figura 3: Diagrama de MVVM

5. Conclusión

Finalmente, respondiendo a las preguntas formuladas en el enunciado de la actividad:

- **¿Qué pasa si en un futuro se quisiera cambiar el motor de base de datos?** Localmente una migración sería parcialmente sencilla ya que “Rooms” actúa como una especie de puente con “SQLite”, por lo que podría migrarse a otra estructura similar con cierta facilidad. Respecto a “Firebase”, utilizamos una “API”, lo cual podría facilitar el uso de otros servicios, pero con el matiz de tener que realizar una migración desde un servicio a otro, con los costes que ello implica, en caso de que ambas plataformas lo permitan con cierta facilidad.
- **¿Qué partes de tu aplicación tendrías que modificar?** En el caso local, dependiendo de la nueva tecnología, sustituir las “entidades” y

“DAO” por sus equivalentes. Por otro lado con “Firebase”, dependiendo del servicio destino, que herramientas facilitan para la migración, y los ajustes oportunos que sean necesarios en cuanto a la “API”.

- **¿Qué dificultades anticipas?** La posible curva de aprendizaje con las nuevas tecnologías a usar, las nuevas estructuras a desarrollar y la migración de los datos persé (en caso de poderse realizar).
- **¿Cómo podrías diseñar tu aplicación para minimizar el impacto de tal cambio?** Reduciendo la variedad de servicios a utilizar, y utilizando tecnologías que posean herramientas que faciliten la migración de datos. Adicionalmente, el uso de estructuras que apliquen modelos estándar, replicables en otras tecnologías.

Referencias

- [1] PAMN Lab3 Arquitectura
- [2] Rooms - <https://medium.com/dvt-engineering/android-room-versus-sqlite-which-is-best-32ff651bc361>