

Glosario - LiveData and UI

- **Android architecture components:** Son una colección de bibliotecas que nos ayudan a diseñar aplicaciones de forma robusta, testeables y mantenibles. Puedes obtener mas información en el siguiente enlace.
- **Android LifeCycle:** En android, a medida que un usuario navega y utiliza la aplicación, la actividad está pasando por diferentes estados en su ciclo de vida. La clase actividad entrega varios callbacks que permiten a esta saber que alguno de estos a cambiado. Es decir si el sistema está creando, deteniendo o continuando, o si es necesario destruir el proceso donde la actividad vive. A través de los métodos del los callbacks mencionados podemos declarar y manejar como queremos que la actividad se comporte en cada estado.
- **Anotación:** Para el alcance de esta lectura, nos referimos al framework que nos permite usar anotaciones, regularmente comienzan con @ en nuestro código, las cuales reducen el boiler Plate de tareas recurrentes y comunes.
- **DAO:** Significa un objeto de acceso a datos, por su sigla en inglés **data access object**, es un componente de software que suministra una interfaz común entre la aplicación y la base de datos.
- **eventBus:** Es una librería que utiliza el patrón observador, vendría a ser una alternativa a LiveData.
- **LiveData:** Es un clase contenedora de datos observable. A diferencia de otros objetos observables LiveData está consciente del ciclo de vida, lo que significa que respecta el ciclo que tenga los componentes de la app.
- **Memory leak:** Se refiere a un fallo en liberar objetos no utilizados de la memoria del dispositivo, Esto quiere decir que esos objetos no utilizados no podrán ser limpiados ni por el garbage collector de Java. Entonces la memoria estará utilizada hasta que se cierre la aplicación.
- **Objeto Dios:** En el contexto de esta lectura, será un objeto que debiera manejar todos los datos en los componentes de UI, como si fuera una variable Global.

- **Programas monolíticos:** Un programa **monolítico** hace referencia a una aplicación software en la que la capa de interfaz de usuario y la capa de acceso a datos están combinadas en un mismo programa y sobre una misma plataforma.
- **Refactorizar:** O refactorización es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.
- **ViewModel:** Es una clase diseñada para mantener y controlar datos que estén relacionados con la Interfaz de usuario de forma consciente al ciclo de vida de los componentes que la están utilizando, regularmente Actividades y fragmentos.