

Investigación primer parcial.

Introducción.

En esta investigación se abarcarán temas relacionados con el lenguaje creado por la empresa Google llamado Flutter, que ha sido creado principalmente para el desarrollo de aplicaciones móviles. Se analizará lo que es un gestor de estados, para qué nos sirve y cómo es implementado en Flutter, se mencionará los widgets con estado y los widgets sin estado, para su mejor entendimiento se mencionará un ejemplo y se finalizará con la explicación de lo que es clean code, cómo es aplicado a flutter y se mencionará un ejemplo.

Gestores de estados (mencionar a lo mínimo 3).

Se utilizan los gestores de estados para poder administrar el estado de una aplicación y que no se pierda el control sobre este. Al tener un gestor de estados es más fácil controlar la información que se declara en los componentes y permite que una aplicación sea más escalable. Recordemos que el estado en lo que finalmente termina viendo el usuario de la aplicación, es recomendable usar algún gestor de estados para que no nos perdamos en el código sobre que información se le está mostrando. Ejemplos de gestores de estados en Flutter son: Bloc, Riverpod y Dash.

Mencionar los widgets con estado y sin estado, poner un ejemplo.

Los widgets con estado en Flutter los reconocemos como StatefulWidget, estos widgets son los que el usuario de la aplicación puede tener una interacción directa. Un ejemplo podría ser un Checkbox, el usuario decide que valor tendrá determinada variable al elegir una opción de este widget.

Los widgets sin estado en Flutter se reconocen como los StatelessWidget, estos a diferencia de los StatefulWidget no tienen una acción o comportamiento, son estáticos, como por ejemplo un texto que pudiera ser el Título de la aplicación, este no va a cambiar, el usuario no puede alterarlo.

Clean Code en Flutter, poner un ejemplo.

La metodología clean code nos permite como desarrolladores facilitar las revisiones de código una vez este haya sido finalizado, también esta metodología permite poder hacer actualizaciones de una manera más sencilla. Se aplica clean code en flutter también para poder hacer las aplicaciones más escalables. Clean architecture son unos principios y patrones de diseño para facilitar la construcción de aplicaciones en flutter. Las capas de clean architecture son UI que es la representación de los datos en un dispositivo, en Flutter son los Widgets, Data que se encarga de comunicarse con dependencias externas para obtener los datos, Device que se encarga de las funcionalidades nativas y Domain que es donde está la lógica del negocio.

Conclusiones.

Flutter es un lenguaje que nos va a permitir hacer aplicaciones escalables y de fácil mantenimiento y actualizaciones gracias a su capacidad de tener gestores de estados, la arquitectura Clean que podemos seguir y sus Widgets con la capacidad de tener estados o ser estáticos.

Bibliografía.

<https://medium.com/gdgeurope/clean-architecture-en-flutter-ee028a6379a5>

<https://www.hostgator.mx/blog/clean-code-codigo-limpio/>