Los fragments son como miniactivities. Como ocurre con los activities, hay un ciclo de vida. Los fragments permiten cambiar la vista de la pantalla sin cambiar de activity. De esta forma, podríamos utilizar un único activity, y trabajar con diversos fragments.

En los fragments, además de onCreate, tenemos el método de onCreateView(), que lo que hace es devolvernos la carga de ese fragment, es decir, el inflater.

Además, los fragments tienen un método muy interesante que es el **onViewCreated**(). Este nos garantiza que ya ha cargado el activity junto al fragment. Hay garantía de que tenemos accesibles los elementos gráficos que hemos definido en el fragment. Aquí se recomienda hacer el control de los eventos.

Los fragments se ponen dentro de FrameLayout, que es el tipo de contenedor específico para los fragments.

Gráfico

Descripción generada automáticamente con confianza media

El OnCreateFView() es equivalente al setContentView de un activity. Y el onViewCreated equivaldría al onCreate de una activity.

También se pueden desasociar los fragments de un activity.

**Fragments estáticos**: no dan nada más que para dividir una pantalla en dos:

Pantalla de computadora con un celular en la mano

Descripción generada automáticamente

**Fragments dinámicos**: son aquellos donde tenemos un único activity y cargamos diferentes fragments.

**Paso de información de un fragment a otro:**

* Con el atributo arguments

Bundle = map (objetos clave-valor)

Texto

Descripción generada automáticamente

Companion object = static

Texto

Descripción generada automáticamente

**Comunicación entre el activity y el fragment:**

Es muy fácil pasar info del fragment al activity, pero no del activity al fragment.

Ya que fragment tiene otro atributo, llamado activity, que contiene el activity al que pertenece:

Texto

Descripción generada automáticamente

Pero el activity no conserva ningún objeto fragment.

Lo que se hace en este caso es crear una interfaz:

Texto

Descripción generada automáticamente

Hay que implementar dos de ellos en el activity y dos de ellos en el fragment.

\*En Kotlin, a diferencia de Java, se puede implementar un cuerpo de un método en el propio interfaz.

# Como implementar

Texto

Descripción generada automáticamente



supportFragmentManager es un atributo incluido en los activities, que se encarga de gestionar los fragments.

Aparte del findViewById, también hay findFragmentById, que lo que hace

Texto

Descripción generada automáticamente