Flutter

Scrolling

Carmelo Villegas Cruz

Scrolling

Todos estamos familiarizados con el concepto de scroll, cuya traducción al castellano podría ser algo como desplazar.

Es algo que usamos a diario cuando usamos cualquier tipo de App, ya sea web, escritorio o móvil.

Es muy normal que el contenido sea mayor que el tamaño del viewport por lo que necesitamos habilitar la posibilidad de hacer un desplazamiento de este para que el usuario lo pueda consumir al completo.

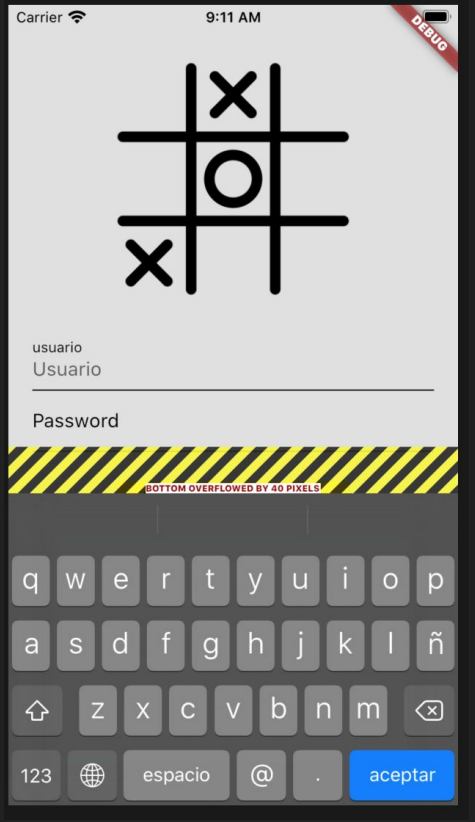
Scrolling in Flutter

Por supuesto Flutter dentro de su catálogo de Widget contiene una gran variedad de estos que implementan la posibilidad de desplazamiento (scrolling).

Ya hemos visto alguno de ellos donde implícitamente incorporan el desplazamiento como es el caso de ListView.

En las prácticas realizadas cuando el número de elementos de la lista supera el tamaño de la pantalla, se activa automáticamente el **scroll**.

Mejorando las interfaces



Aplicando scroll podemos resolver problemas cuando el contenido se ha visto desplazado por algún elemento externo, en este ejemplo a causa del teclado.

Usando SingleChildScrollView podemos resolver este problema otorgando a los elementos visibles la posibilidad de desplazarse.

Añadiendo Controller

En los widgets de tipo **scroll** entre de sus atributos podemos encontrar **controller**.

Se define como:
controller -> ScrollController?
Un objeto que puede ser usado para controlar la posición en la cual la **vista** es desplazada.

ScrollController

Puede tener múltiples propósitos, entre otros:

- Controlar la posición inicial del desplazamiento.
  `ScrollController.initialScrollOffset`
- Para leer la posición.
  `ScrollController.offset`
- Para cambiar la posición.
  `ScrollController.animateTo`
  `ScrollController.jumpTo`

ScrollController

- Obtener el offset máximo y mínimo
  `ScrollController.position.maxScrollExtent`
  `ScrollController.position.minScrollExtent`

- Si está fuera de rango
  `ScrollController.position.outOfRange`

ScrollController

Para añadir un ScrollController a un scrolling widget debemos realizar los siguientes pasos.

- Definir e inicializar
- Añadir listener en initState y definir método que será invocado.
- Asignar al atributo controller el ScrollController creado.
- Eliminar el controller en dispose.

ScrollNotifications

Se lanzan eventos cada vez que ocurre:

- **ScrollStartNotification**.Indica que el widget ha comenzado a desplazarse (scroll).
- **ScrollUpdateNotifications**. Cuando existe un desplazamiento.
- **ScrollEndNotification**. Indica cuando el widget para de desplazarse.
- **OverScrollNotifications**. Indica que el widget no ha cambiado su posición dado que el desplazamiento que se intentó era más allá de los límites.
- **UserScrollNotifications**. El usuario cambia la dirección de desplazamiento.

NotificationListener

Para poder captura notificaciones tenemos que envolver el widget que queremos escuchar con **NotificationListener.**

NotificationListener

```
NotificationListener<ScrollNotification>(
              onNotification: (scrollNotification) {
                if (scrollNotification is
ScrollStartNotification) {
                    //Do something
                } else if (scrollNotification is
ScrollUpdateNotification) {
                    //Do something
                } else if (scrollNotification is
ScrollEndNotification) {
                    //Do something
                }
              },
              child: //Scrolling widget
```

NotificationListener

“//Do something” lo podemos cambiar por ejemplo por una llamada a un método.

```
//Do somethhing
_onStartScroll(scrollNotification.metrics);


_onStartScroll(ScrollMetrics metrics) {
      setState(() {
        message = "Empieza el desplazamiento";
      });
}
```