

## TALLER 1

1. Cree una nueva aplicación con una actividad vacía utilizando el asistente de Android Studio.
2. Utilizando un `ConstraintLayout` o un `RelativeLayout`, defina tres botones para las opciones "Explorar Destinos", "Favoritos", y "Recomendaciones", así como un spinner con 6 opciones de categorías de viajes. Las categorías son: Todos, Playas, Montañas, Ciudades Históricas, Maravillas del Mundo, Selvas. La actividad debe verse bien tanto en *portrait* como en *landscape* tal como se muestra en la Figura 1.

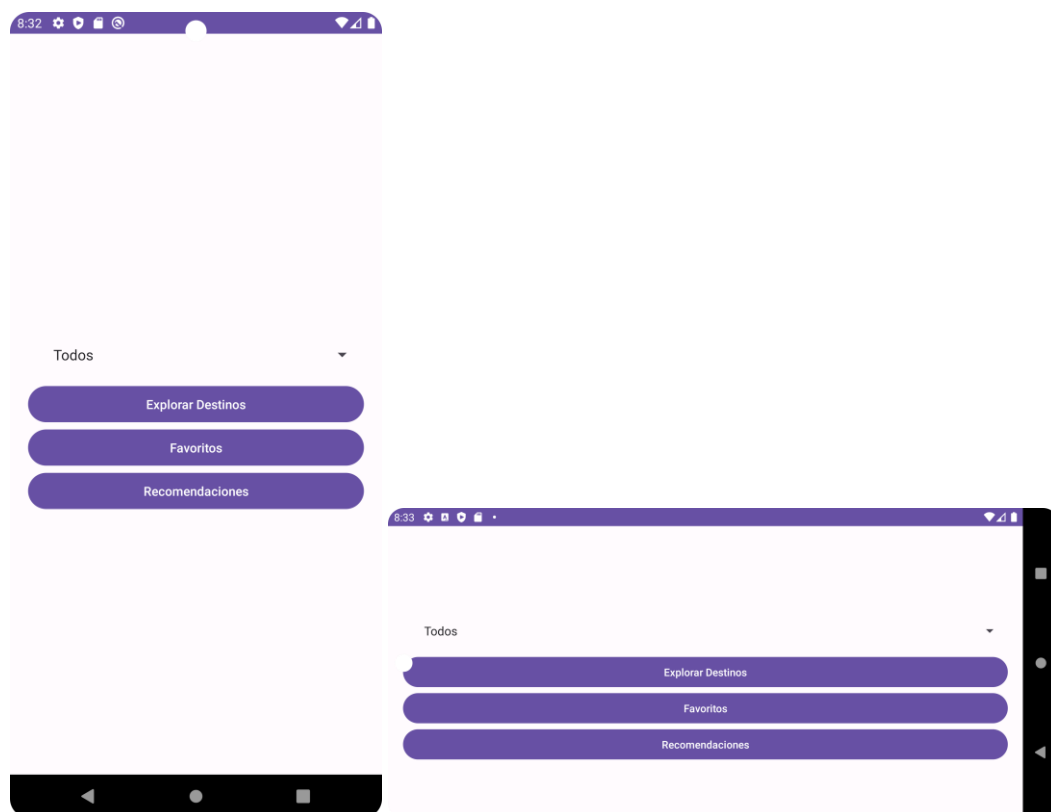


Figura 1. Actividad Principal

3. Programe la actividad para que cada vez que se pulse el botón de "Explorar Destinos", abra una nueva actividad que muestre una lista de destinos turísticos obtenidos del archivo JSON adjunto al taller. La lista desplegada debe responder al filtro de búsqueda para encontrar destinos de la categoría seleccionada en el spinner de la actividad principal.
4. Programe un *listener* para la lista de tal manera que al presionar un destino de la lista se abra una nueva actividad que muestre todos los detalles del destino seleccionado disponibles en el archivo JSON. Agregue al final un botón para añadir el destino a la lista de favoritos que al ser pulsado se desactive y envíe un Toast con la frase "Añadido a favoritos". La Figura 2 muestra el comportamiento de la lista.



## TALLER 1

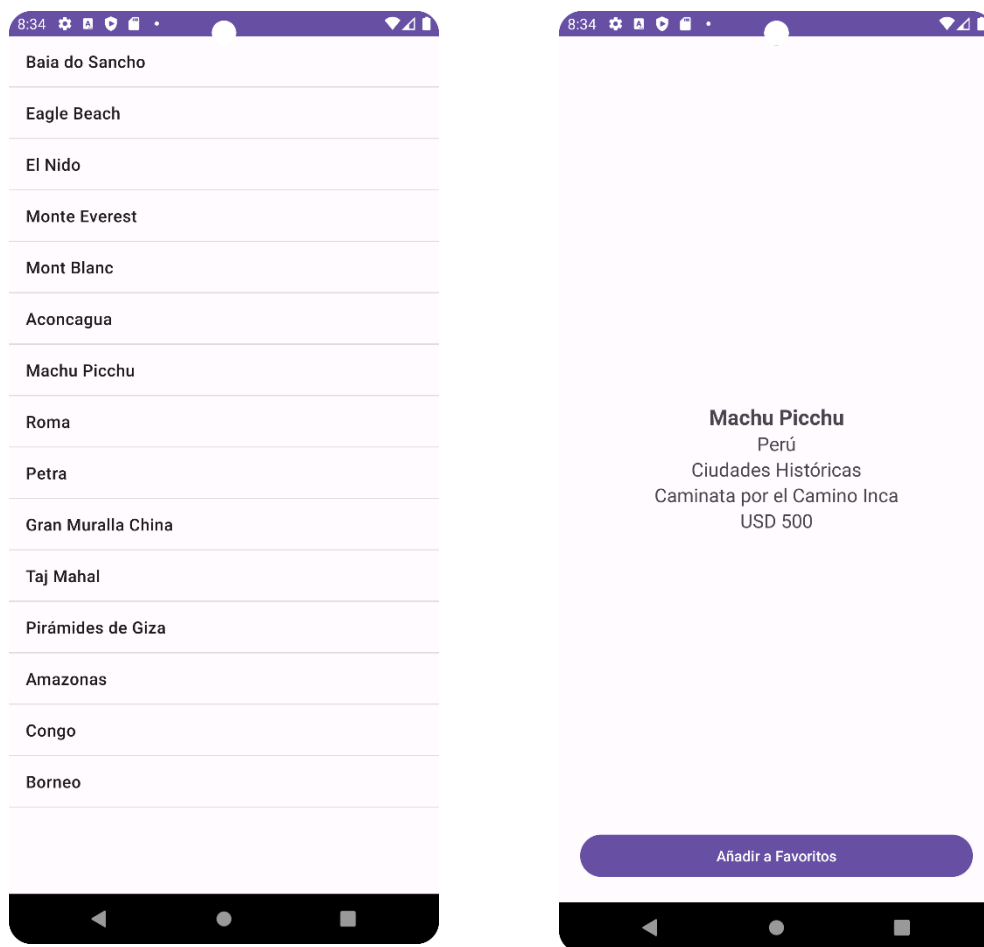


Figura 2. Actividad Lista y Detalles

5. Programe el botón Favoritos para que se muestre una lista de todos los nombres de los destinos marcados como favoritos. (Puede usar companion objects definidos en el main Activity para guardar los destinos añadidos)
6. Por último, programe el botón "Recomendaciones" para lanzar una actividad que muestre el nombre y la actividad recomendada de un destino aleatorio, pero que pertenezca a la categoría más frecuente de la lista de favoritos. Si no hay favoritos, debe mostrar NA. La Figura 3 ilustra el funcionamiento de este botón.



## TALLER 1



Figura 3. Actividad Recomendaciones

### Bono! (0.5pts sobre la nota del taller)

En la actividad de detalle de destinos del punto 4, agregue un atributo que diga el clima actual en ese destino usando una API de clima gratuita.

#### Recomendaciones.

- Las clases deben respetar las reglas de nombramiento de Kotlin y en caso de que extiendan de AppCompatActivity deben tener el sufijo Activity.
- El método onCreate no debe superar las 20 líneas de código en ninguna de las actividades.

#### Entregables.

- Crear un video de **máximo 4 minutos**, donde se evidencie la demostración de todas las funcionalidades de la aplicación del Taller 1. En el video deben aparecer los rostros en vivo de todos los integrantes del grupo mientras se hace la demostración. El video se debe cargar a YouTube y compartir el enlace por la entrega de BrighSpace.
- Cargar en BrighSpace **un archivo .zip** con la exportación del proyecto del taller.



**TALLER 1**

- Cargar en BrighSpace **un único enlace al repositorio GitHub** de alguno de los integrantes del grupo con el desarrollo del Taller. En el repositorio debe existir el archivo README.md el cual debe contener los nombres de los integrantes del grupo, el nombre del grupo, y el enlace al video demostrativo de YouTube. El video de demostrativo **no** debe estar físicamente en el repositorio de GitHub. El repositorio debe ser de alguno de los integrantes del grupo, **NO** está permitido cargar desarrollos de talleres en el repositorio del proyecto.

**Se deben entregar todos los entregables, la no entrega de alguno de ellos bajo las condiciones especificadas generará una calificación automática del taller sobre 0.0**