

## Temario

### Unidad 6

- a) Android Life Cycle
- b) Android Adaptadores
- c) Cambiar de Pantalla
- d) Patrón MVP
- e) Callbacks

## Unidad 6

# Android Life Cycle

- Cuando las activity se crean, se usan, y luego pierden el control pasan por diferentes estados
  - Android permite manejar lo que pasa en esos estados/cambios de estado
  - Permite que la app:
    - No falle si el usuario recibe una llamada telefónica o cambia a otra app.
    - No consuma recursos valiosos del sistema cuando el usuario no la use de forma activa.
    - No pierda el progreso del usuario si este abandona la app y regresa a ella posteriormente.
    - No falle ni pierda el progreso del usuario cuando se gire la pantalla entre la orientación horizontal y la vertical.

## Unidad 6

# Android Life Cycle

- Los métodos del ciclo de vida:
  - onCreate: cuando se crea el activity/fragment
  - onStart: cuando se entra en estado started, luego de created
  - onResume: cuando la actividad entra en el estado Resumed, pasa al primer plano
  - onPause: cuando se esta abandonando la activity/fragment
  - onStop: cuando el usuario ya no puede ver la actividad
  - onDestroy: cuando se va a finalizar la actividad, pero antes de que finalice

## Unidad 6

# Android Adaptadores

- Adapter
  - Un adaptador **adapter** es un objeto que implementa la interface adapter
  - Actúa como vínculo entre un conjunto de datos y una vista
  - Es responsable de recuperar los datos desde el conjunto de datos y generar el objeto vista basado en los datos
    - Permite un mejor manejo de memoria y procesador
- Podemos crear nuestras propias clases adaptadores
  - Comúnmente se extienden clases adaptadores incluidas en el SDK de Android

## Unidad 6

# Cambiar de Pantalla

- Para cambiar de pantalla se usan los “intent”.
- Intent
  - Objeto de mensajería que podemos usar para solicitar una acción de otro componente de una app
  - Permiten cambiar el control
    - Otra activity de la app
    - Otra app
  - Especifican qué aplicación las administrará
    - Incluyendo el nombre del paquete de la app de destino
    - Nombre de clase del componente completamente calificado

## Unidad 6

# Patrón MVP

- Patrones de diseño
  - Las mejores prácticas aplicadas en desarrollo de software
  - Documenta la mejor solución a un problema de desarrollo
  - Comúnmente presentados en catálogos
    - Ej: GOF
- MVP
  - Model View Presenter
  - Basado en MVC – Model View Controller
  - El presentador asume la funcionalidad del "hombre-intermedio"
  - Toda lógica de presentación es colocada al presentador

## Unidad 6

# Patrón MVP

- Componentes de MVP
  - Modelo: Esta capa de acceso a datos, tales como bases de datos o API remotas.
  - Vista: Se encarga de mostrar los datos. Aquí se encontrarían nuestros Fragmentos y Vistas.
  - Presentador: es la capa que provee datos desde la vista hasta el modelo. También maneja las tareas en segundo plano.

## Unidad 6

# Callbacks

- Callback:
  - Función "A" que se usa como argumento de otra función "B"
  - Cuando se llama a "B", ésta ejecuta "A"
  - Usualmente lo que se pasa a "B" es el puntero a "A"
- Presentes en muchos ámbitos del desarrollo Android
- Consideraciones
  - Android es Single thread
  - Posibilidades de error si no se maneja bien los eventos
- En Android muchos widgets presentan y definen Listeners para eventos especiales
  - Permiten construir funcionalidad en respuesta a eventos especiales



# Material complementario

- Cómo pasar de una Activity a otra - Intent en Android:  
<https://www.youtube.com/watch?v=VQcJRM6ZgHw>
- Android Studio - Arquitectura MVP (Explicación):  
<https://www.youtube.com/watch?v=iL2PbIHsS9A>
- <https://developer.android.com/training/basics/firstapp>