

EVALUACIÓN

HERRAMIENTAS DE
PROGRAMACIÓN MOVIL
Semana 3

Michel Brevis
03-11-2024

Técnico en Análisis y
Programación Computacional



DESARROLLO:

Eres parte de un equipo de desarrollo del Ministerio de Cultura, el cual está en proceso de creación de una nueva aplicación móvil en Android, para poder gestionar las reservas de entradas a los distintos museos del país. Dado lo anterior se han fijado los siguientes requerimientos:

- El nombre del proyecto debe ser su nombre_apellido_fecha_seccioncurso (la fecha en formato `yyyymmdd`).
- La aplicación debe ser desarrollada en lenguaje Kotlin (aplicativo en lenguaje Java no será considerado).
- Cada vez que la aplicación sea abierta o cerrada debe quedar un registro de debug en la consola de desarrollador.
- Debe considerar elementos básicos para el ingreso de la reserva (persona responsable, cantidad de personas y fecha).
- Debe contener un botón que, al pincharlo, muestre la información de la reserva solicitada (persona responsable, cantidad de personas y fecha) como mensaje de alerta y en la consola del desarrollador.
- Debe utilizar layout para el ordenamiento de los controles en pantalla.

A continuación, responde las siguientes preguntas:

1. ¿Qué elementos de la anatomía de una aplicación móvil son parte de la aplicación desarrollada? Explica.

1. Activity Principal

Es la actividad principal de la aplicación y es el punto de entrada cuando la aplicación se abre. Esta actividad es responsable de cargar el contenido principal, incluyendo la interfaz de usuario para el formulario de reserva.

Al abrir y cerrar la aplicación, esta actividad registra mensajes en la consola de debug, lo cual permite rastrear la apertura y cierre de la app en el sistema.

2. Interfaz de Usuario (UI) con Jetpack Compose

La interfaz de usuario fue creada utilizando **Jetpack Compose**, una moderna biblioteca de herramientas de interfaz de usuario en Android. Compose facilita la construcción de interfaces declarativas y permite organizar los elementos de la aplicación en layouts de forma intuitiva.

Se utilizó una **Column** como contenedor principal para organizar verticalmente los elementos del formulario.

3. Componentes de Entrada de Datos

La aplicación incluye **campos de texto** para que el usuario ingrese el nombre de la persona responsable y la cantidad de personas en la reserva. Estos campos son interactivos y permiten al usuario ingresar la información necesaria.

También se añadió un **DatePicker** para seleccionar la fecha de la reserva, implementado mediante un DatePickerDialog.

4. Componentes de Control de Validación y Mensajería

La aplicación incluye un botón "**Mostrar Reserva**" que valida los datos ingresados. Si hay errores, se muestra un mensaje emergente (Toast) que notifica al usuario sobre la necesidad de completar correctamente el formulario.

Cuando los datos son válidos, se muestra un Toast con la información completa de la reserva, y la misma información se registra en la consola de debug para fines de rastreo.

5. Registro en la Consola de Debug

- La aplicación implementa registros en la consola para las acciones de apertura y cierre de la aplicación, así como para mostrar detalles de la reserva al hacer clic en el botón de "Mostrar Reserva". Este uso de logs es esencial en el desarrollo móvil, ya que permite monitorear el comportamiento de la aplicación durante las pruebas y detectar posibles errores.
2. ¿Qué layout utilizaste para el ordenamiento de los controles? Justifica tu elección indicando por qué se utilizó este layout sobre los otros disponibles.

Para el ordenamiento de los controles en la aplicación de reservas, se utilizó el layout **Column** de Jetpack Compose. Aquí están las razones detrás de esta elección:

Justificación del Uso de Column

Ordenamiento Vertical: Column permite organizar los elementos en una disposición vertical, que es la estructura natural para un formulario en el cual los usuarios ingresan datos secuencialmente, de arriba hacia abajo. Esto hace que la experiencia de usuario sea más intuitiva y fácil de seguir.

Espaciado entre Elementos: Column facilita el ajuste del espaciado entre elementos usando `Arrangement.spacedBy`, lo que ayuda a mantener una separación uniforme entre los campos de texto, botones y otros componentes. Esto contribuye a un diseño más limpio y fácil de leer.

Compatibilidad con Jetpack Compose: Como Column es un componente nativo de Jetpack Compose, es fácil de configurar y personalizar sin tener que recurrir a layouts más complejos. Además, Column es altamente compatible con los modificadores de Compose, como `fillMaxWidth` y `padding`, lo cual simplifica el diseño de la interfaz.

Por qué Column sobre Otros Layouts

ConstraintLayout: Aunque ConstraintLayout permite organizar elementos en posiciones más complejas y precisas, no era necesario en este caso, ya que el formulario sigue una disposición lineal y sencilla.

Row y Box: Estos layouts tienen aplicaciones específicas (organización horizontal y superposición de elementos, respectivamente). Column fue preferido porque permite una estructura vertical, que es la disposición natural para un formulario de entrada de datos.

3. ¿Qué elementos básicos adecuados consideraste para construir la aplicación sobre la base de los requerimientos? Explica.

Para construir la aplicación según los requisitos, se seleccionaron los elementos básicos de interfaz y funcionalidad en Jetpack Compose. Estos elementos permiten una experiencia de usuario sencilla y cumplen con los requerimientos de captura y visualización de datos de la reserva.

1. OutlinedTextField (Campo de Texto)

Persona Responsable: Se incluyó un campo de texto para que el usuario ingrese el nombre de la persona responsable de la reserva. Este campo captura la información básica necesaria para identificar a quién pertenece la reserva.

Cantidad de Personas: Otro campo de texto permite al usuario ingresar la cantidad de personas en la reserva. Este campo incluye validación para asegurar que el valor ingresado sea numérico y positivo, evitando errores de datos y mejorando la precisión de la información capturada.

2. DatePickerDialog (Selector de Fecha)

Para la selección de la fecha de la reserva, se usó un DatePickerDialog. Al presionar el botón de selección de fecha, el DatePickerDialog permite al usuario escoger una fecha de forma intuitiva mediante un calendario visual.

3. Button (Botones de Acción)

Botón de Selección de Fecha: Este botón activa el DatePickerDialog, permitiendo al usuario seleccionar una fecha cuando lo necesite. Su texto cambia dinámicamente para mostrar la fecha seleccionada o "Seleccionar Fecha" si no se ha elegido ninguna.

Botón de Confirmación "Mostrar Reserva": Este botón permite al usuario enviar la información de la reserva. Antes de proceder, realiza una validación de los campos y, si todo está correcto, muestra los detalles en un Toast y los registra en la consola de debug.

4. Toast y Log.d (Mensajería y Registro en Consola)

Toast: Un Toast muestra un mensaje emergente con los detalles de la reserva si los campos están completos o un mensaje de error si hay información faltante. Esto proporciona una retroalimentación rápida sin interrumpir el flujo de la aplicación.

Log.d: La aplicación registra en la consola mensajes que indican cuando la aplicación se abre, se cierra y cuando se muestra una reserva. Esto ayuda a monitorear el comportamiento de la aplicación para desarrolladores.

Muestra de app en dispositivo Android

The screenshot shows an Android application titled "Reserva de Entradas a Museos". The interface includes a status bar at the top with the time 6:15, signal strength, and battery level. The app has two input fields: "Persona Responsable" with the value "mike" and "Cantidad de Personas" with the value "5". Below these is a date field showing "Fecha: 12/11/2024" and a button labeled "Mostrar Reserva". At the bottom, a dark notification bubble displays a green icon and the text "Reserva: Persona Responsable: mik...".

6:15

Reserva de Entradas a Museos

Persona Responsable

mike

Cantidad de Personas

5

Fecha: 12/11/2024

Mostrar Reserva

Reserva:
Persona Responsable: mik...

REFERENCIAS BIBLIOGRÁFICAS:

- **Ejemplo texto de lectura de IACC:**

IACC. (2024). *Herramientas de programación móvil*
Semana 3