Sesión 1: Introducción a las Aplicaciones Móviles y React Native

Objetivo de la sesión

Introducir a los estudiantes al desarrollo de aplicaciones móviles multiplataforma utilizando React Native. Al finalizar, los estudiantes comprenderán los conceptos básicos de aplicaciones móviles, las características principales de React Native, y tendrán su entorno configurado para el desarrollo inicial.

---

1. Introducción a la programación de dispositivos móviles con React Native

¿Qué es React Native?

React Native es un framework de desarrollo creado por Meta (anteriormente Facebook) que permite desarrollar aplicaciones móviles utilizando JavaScript y React.

- **Multiplataforma**: Una sola base de código para Android e iOS.
- **Uso de componentes nativos**: Las aplicaciones tienen una apariencia y rendimiento similar a las nativas.
- **Ventaja principal**: Reduce el tiempo y los costos al no requerir dos desarrollos separados.

---

2. Conceptos básicos de aplicaciones móviles multiplataforma

¿Qué son aplicaciones multiplataforma?

Son aplicaciones que se desarrollan una vez y pueden ejecutarse en múltiples sistemas operativos. React Native se encarga de traducir el código JavaScript en componentes nativos.

Ventajas de aplicaciones multiplataforma

1. Ahorro de tiempo y costos de desarrollo.
2. Código centralizado, lo que simplifica el mantenimiento.
3. Comunidad activa que ofrece soporte y librerías.

---

3. Características principales de React Native

- **Hot Reloading**: Permite visualizar los cambios en tiempo real sin reiniciar la aplicación.
- **Compatibilidad con bibliotecas nativas**: Es posible integrar código nativo de Android (Java/Kotlin) e iOS (Objective-C/Swift).
- **Rendimiento cercano al nativo**: Ideal para la mayoría de las aplicaciones, aunque no para videojuegos de alto rendimiento.

---

4. Diferencias entre React Native y desarrollo nativo

Aspecto	**React Native**	**Desarrollo Nativo (Kotlin/Swift)**
Lenguaje	JavaScript	Kotlin (Android), Swift (iOS)
Código multiplataforma	Sí	No
Rendimiento	Bueno en la mayoría de casos	Óptimo para tareas intensivas
Comunidad y recursos	Amplia, con muchas librerías	Limitado a Android o iOS

Ejemplo Comparativo: Crear un botón

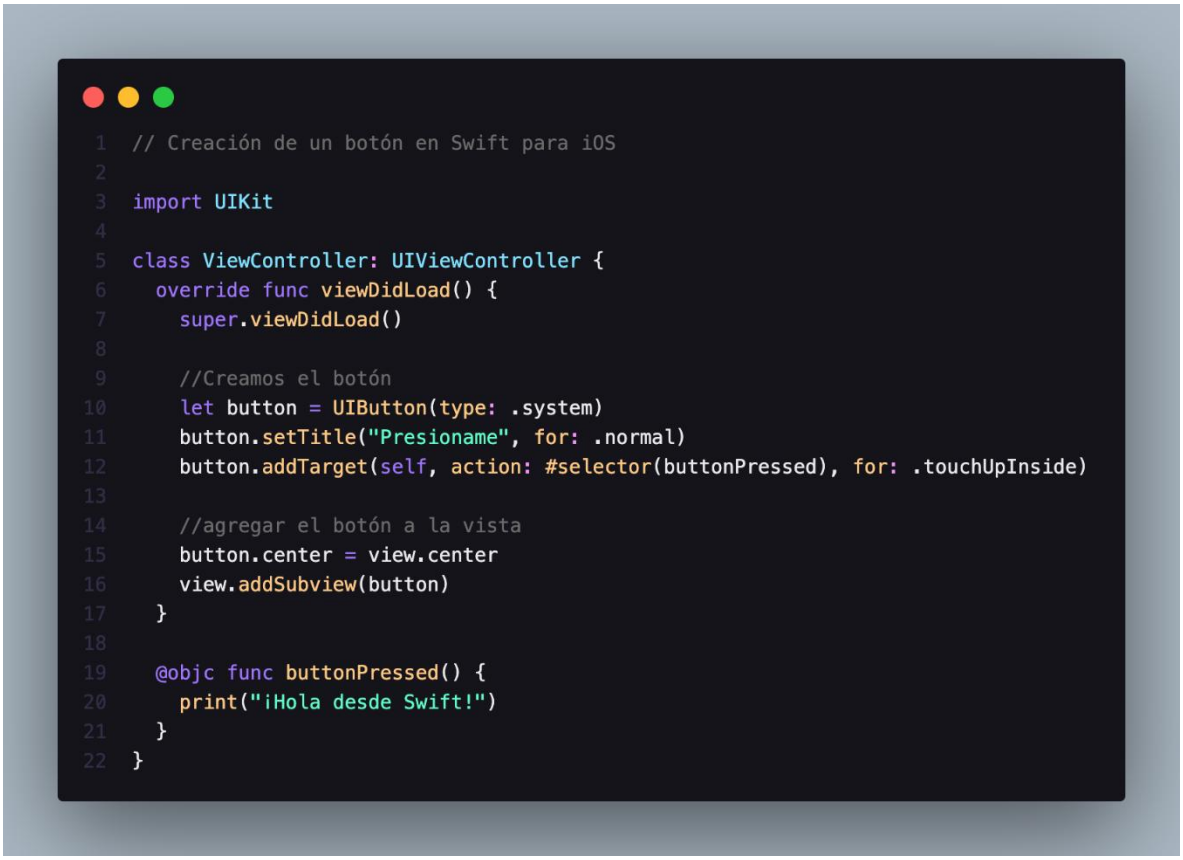
React Native:

```jsx
// Importamos los módulos necesarios de React y React Native
import React from 'react';
import { Button, Alert } from 'react-native';

// Definimos un componente funcional que muestra un botón
const App = () => {
  return (
    <Button
      title="Presióname" // Texto visible en el botón
      onPress={() => Alert.alert('¡Hola desde React Native!')} // Acción al presionar
    />
  );
};

export default App; // Exportamos el componente para que pueda ser usado
```

Kotlin:

```kotlin
// Creación de un botón en un contexto de Android nativo
Button(this).apply {
  text = "Presióname" // Texto visible en el botón
  setOnClickListener {
      Toast.makeText(context, "¡Hola desde Kotlin!", Toast.LENGTH_SHORT).show()
  }
}
```

Swift:



```swift
// Creación de un botón en Swift para iOS

import UIKit

class ViewController: UIViewController {
  override func viewDidLoad() {
    super.viewDidLoad()

    //Creamos el botón
    let button = UIButton(type: .system)
    button.setTitle("Presioname", for: .normal)
    button.addTarget(self, action: #selector(buttonPressed), for: .touchUpInside)

    //agregar el botón a la vista
    button.center = view.center
    view.addSubview(button)
  }

  @objc func buttonPressed() {
    print("¡Hola desde Swift!")
  }
}
```

4. Configuración del entorno con Expo

Expo es la forma más sencilla de comenzar con React Native, ya que simplifica la configuración inicial y elimina la necesidad de instalar herramientas complejas como Xcode o Android Studio.

Instalar Node.js

1. **Descargar desde nodejs.org.**
2. **Verificar instalación:**

`node -v`

`npm -v`

Instalar Expo

Desde 2023, Expo CLI fue integrado en el paquete expo. Para instalar:

```
npm install --global expo
```

Verificar instalación:

```
expo --version
```

Crear un proyecto con Expo

1. Crear un nuevo proyecto:

   ```
   npx create-expo-app MiPrimeraApp
   ```

- Seleccionar la plantilla **"Blank"** cuando se solicite.
- Navegar al directorio:

2. Navegar al directorio:

   ```
   cd MiPrimeraApp
   ```

3. Ejecutar la aplicación:

   ```
   npm start
   ```

   Esto abrirá Expo Dev Tools en el navegador.

Ejecutar la app en un dispositivo físico

1. Instalar la app Expo Go desde la App Store (iOS) o Google Play (Android).
2. Escanear el código QR generado en Expo Dev Tools para previsualizar la app.

5. Actividad práctica

Objetivo: Crear una aplicación básica que muestre un texto personalizado.

1. Editar el archivo App.js:

```
// Importamos módulos básicos de React Native
import React from 'react';
import { Text, View } from 'react-native';

// Componente principal
const App = () => {
  return (
    // Contenedor principal
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      {/* Texto centrado en pantalla */}
      <Text style={{ fontSize: 20, color: 'blue' }}>
        ¡Hola, React Native!
      </Text>
    </View>
  );
};

export default App;
```

6. Reflexión y evaluación

Preguntas clave

- ¿Qué es React Native y cómo se compara con Kotlin y Swift?
- ¿Cuáles son las ventajas del desarrollo multiplataforma?

- ¿Qué herramientas configuraste en esta sesión?

Entrega de evidencia

- Captura de pantalla de la app corriendo en el dispositivo o emulador.