

Sesión 2: Configuración del Entorno de Desarrollo para React Native

Objetivo de la sesión

- Configurar correctamente el entorno de desarrollo para React Native.
- Probar aplicaciones en dispositivos Android e iOS, entendiendo las limitaciones de emulación en Windows.
- Crear y ejecutar aplicaciones básicas con Expo.

1. Introducción

En esta sesión aprenderás a configurar las herramientas necesarias para desarrollar aplicaciones móviles con React Native, probando tanto en dispositivos físicos como en emuladores (según disponibilidad). React Native es compatible con iOS y Android, pero en sistemas Windows existen restricciones para emular iOS.

2. Configuración del entorno en Windows

2.1. Instalación de Node.js

1. Descarga Node.js desde nodejs.org. Selecciona la versión LTS (Long Term Support) para mayor estabilidad.
2. Verifica la instalación:

```
node -v
```

```
npm -v
```

2.2. Instalación de Expo

Expo simplifica el desarrollo con React Native, eliminando la necesidad de configurar herramientas avanzadas como Android Studio o Xcode en etapas iniciales.

1. Instala Expo:

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

2. Verifica la instalación:

```
expo --version
```

2.3. Crear un proyecto con Expo

1. Crea un proyecto nuevo:

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

2. Navega al directorio del proyecto:

```
cd MiPrimeraApp
```

3. Inicia el servidor de desarrollo

```
npm start
```

Esto abrirá Expo Dev Tools en tu navegador.

2.4. Configuración de Visual Studio Code

1. Descarga Visual Studio Code desde code.visualstudio.com.
2. Instala extensiones recomendadas:
 - **ES7+ React/Redux/React-Native snippets:** Para accesos directos de código.
 - **Prettier:** Para formato de código.
 - **React Native Tools:** Depuración y herramientas adicionales.

3. Probar aplicaciones en dispositivos Android e iOS

3.1. Probar en dispositivos Android

1. **Habilitar depuración USB:**
 - Ve a **Configuración > Acerca del teléfono** y toca varias veces en **Número de compilación** para activar las opciones de desarrollador.
 - Activa **Depuración USB** en **Opciones de desarrollador**.

2. Conecta el dispositivo físico por USB.
3. Usa Expo Dev Tools para ejecutar la aplicación.

3.2. Probar en dispositivos iOS

1. Instalar Expo Go:

- Descarga la app **Expo Go** desde la App Store.
2. Conecta tu dispositivo iOS y tu computadora a la misma red Wi-Fi.
 3. Escanea el código QR generado por Expo Dev Tools para ver tu app en el dispositivo físico.
-

4. Opciones para emular iOS en Windows

Limitaciones en Windows

- Xcode y los emuladores de iOS solo están disponibles en macOS.
- Windows no puede emular dispositivos iOS de manera nativa.

Soluciones disponibles

1. **Probar con Expo en un dispositivo iOS físico:** Ideal para proyectos básicos y pruebas rápidas.
 2. **Mac en la nube:**
 - Servicios como [MacStadium](#) y [MacinCloud](#) te permiten acceder a macOS desde Windows.
 - Instalando Xcode en la nube, puedes usar los simuladores de iOS.
 3. **Virtualización de macOS:**
 - Configurar macOS en una máquina virtual con software como VMware o VirtualBox.
 - **Nota:** Este método es complejo y podría infringir los términos de Apple.
 4. **Herramientas remotas:**
 - Usa plataformas como [BrowserStack](#) o [Sauce Labs](#) para probar aplicaciones en dispositivos iOS reales de manera remota.
-

5. Actividad práctica

Ejercicio 1: Crear y personalizar una aplicación básica

1. Crea un proyecto:

```
npx create-expo-app MiSegundaApp --template blank
```

2. Abre el archivo App.js y personaliza el contenido:

```
import React from 'react';
import { Text, View, StyleSheet } from 'react-native';

const App = () => {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>¡Hola, React Native!</Text>
      <Text style={styles.subtitle}>Esta es tu primera app
personalizada.</Text>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#e0f7fa',
  },
  title: {
```

```

    fontSize: 24,
    fontWeight: 'bold',
    color: '#00796b',
  },
  subtitle: {
    fontSize: 18,
    color: '#004d40',
    marginTop: 10,
  },
});

export default App;

```

3. Ejecuta la aplicación en tu dispositivo físico (Android o iOS) con Expo Go.

Ejercicio 2: Contador interactivo

Crea una aplicación con dos botones que incrementen y decrementen un contador.

Código:

```

import React, { useState } from 'react';
import { View, Text, Button, StyleSheet } from 'react-native';

const App = () => {
  const [count, setCount] = useState(0);

  return (
    <View style={styles.container}>
      <Text style={styles.counter}>Contador: {count}</Text>

```

```

        <View style={styles.buttons}>
            <Button title="Incrementar" onPress={() => setCount(count
+ 1)} />
            <Button title="Decrementar" onPress={() => setCount(count
- 1)} />
        </View>
    </View>
);
};

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  counter: {
    fontSize: 24,
    marginBottom: 20,
  },
  buttons: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    width: '60%',
  },
});
export default App;

```

6. Evaluación

Preguntas clave

1. ¿Qué herramientas instalaste y configuraste para React Native?
2. ¿Cómo probaste tu aplicación en dispositivos físicos?
3. ¿Qué opciones tienes para emular iOS desde Windows?