



DESARROLLO DE APPS CON REACT NATIVE



Marco Leon Mora Mendez

SENA - CEET

Contenido

DESARROLLO DE APPS.....	4
REACT NATIVE	5
ENTORNO DE DESARROLLO.....	6
Configurar el entorno de desarrollo.....	6
Crear una App.....	¡Error! Marcador no definido.
Ejecutar la App en un emulador.....	¡Error! Marcador no definido.
Ejecutar la App en un dispositivo físico.....	¡Error! Marcador no definido.
COMPONENTES CENTRALES.....	¡Error! Marcador no definido.
Crear una ventana de inicio	¡Error! Marcador no definido.
Imagen de Fondo.....	¡Error! Marcador no definido.
Formulario de ingreso	¡Error! Marcador no definido.
Elementos del formulario.....	¡Error! Marcador no definido.
Crear elementos personalizados.....	¡Error! Marcador no definido.
Pasando parámetros al componente	¡Error! Marcador no definido.
Crear estilos globales	¡Error! Marcador no definido.
NAVEGACION ENTRE PANTALLAS	¡Error! Marcador no definido.
Separando las pantallas	¡Error! Marcador no definido.
Crear pantalla de registro	¡Error! Marcador no definido.
Agregar componentes a la pantalla de registro.....	¡Error! Marcador no definido.
Capturar valores de los inputs	¡Error! Marcador no definido.
ORGANIZANDO EL CÓDIGO CON EL PATRÓN DE DISEÑO MVVM.....	¡Error! Marcador no definido.
Reorganizar las carpetas según las convenciones de Google	¡Error! Marcador no definido.
Capa de Presentación.....	¡Error! Marcador no definido.
Capa de Dominio	¡Error! Marcador no definido.
Capa de Datos	¡Error! Marcador no definido.
Crear un componente personalizado para los inputs.....	¡Error! Marcador no definido.
Separar las hojas de estilos	¡Error! Marcador no definido.
MONTANDO UN SERVIDOR NODE JS	¡Error! Marcador no definido.
Prueba de Rutas del servidor	¡Error! Marcador no definido.
CONECTANDO A LA BASE DE DATOS (MySQL)	¡Error! Marcador no definido.
Realizando la conexión desde Node.js.....	¡Error! Marcador no definido.
Creando un usuario	¡Error! Marcador no definido.

PARTE 1. INSTALACION DE RECURSOS DE SOFTWARE

DESARROLLO DE APPS

La creación de una aplicación (APP) se refiere al proceso de desarrollar software para dispositivos móviles o de escritorio que brinda funcionalidades específicas a los usuarios.

Existen dos enfoques principales para desarrollar aplicaciones móviles: plataformas nativas y plataformas multiplataforma.

Las plataformas nativas, como Android o *iOS*, permiten desarrollar aplicaciones utilizando los lenguajes de programación y las herramientas específicas de cada plataforma. Esto significa que se debe escribir código separado para cada plataforma, lo que puede requerir más tiempo y recursos.

Por otro lado, los sistemas multiplataforma, como *React Native* y *Flutter*, permiten desarrollar aplicaciones utilizando un único código base que se puede ejecutar en múltiples plataformas. Estas plataformas utilizan *frameworks* y bibliotecas que traducen el código a un lenguaje nativo para cada plataforma.

Además de las funcionalidades proporcionadas por las plataformas nativas, las aplicaciones móviles a menudo hacen uso de bibliotecas de terceros para agregar características adicionales. Estas bibliotecas pueden incluir funcionalidades como la integración con servicios en la nube, análisis de datos, autenticación de usuarios, entre otros.

En resumen, al crear una aplicación, se debe considerar si se utilizará una plataforma nativa o una plataforma multiplataforma, si se requerirá almacenamiento de datos local utilizando SQLite¹ y si se utilizarán bibliotecas de terceros para agregar funcionalidades adicionales.

¹ SQLite es una biblioteca de base de datos relacional que se utiliza comúnmente en aplicaciones móviles para almacenar y administrar datos localmente en el dispositivo del usuario. Proporciona una forma eficiente y confiable de almacenar datos estructurados.

REACT NATIVE

React Native es una opción popular para el desarrollo de aplicaciones móviles debido a varias razones:

Eficiencia de desarrollo: React Native permite escribir código una vez y ejecutarlo en múltiples plataformas, como iOS y Android. Esto reduce el tiempo y los recursos necesarios para desarrollar y mantener aplicaciones móviles.

Rendimiento nativo: A diferencia de las aplicaciones híbridas, las aplicaciones desarrolladas con React Native se ejecutan directamente en el dispositivo, lo que proporciona un rendimiento similar al de las aplicaciones nativas.

Reutilización de código: React Native permite reutilizar componentes de interfaz de usuario entre diferentes plataformas, lo que facilita el desarrollo y la actualización de aplicaciones para múltiples sistemas operativos.

Comunidad activa: React Native cuenta con una gran comunidad de desarrolladores que comparten conocimientos, recursos y bibliotecas de código abierto. Esto facilita el aprendizaje y la resolución de problemas durante el desarrollo de aplicaciones.

Integración con tecnologías existentes: React Native se puede integrar fácilmente con código nativo existente, lo que permite aprovechar las funcionalidades y características específicas de cada plataforma.

En resumen, utilizar React Native puede acelerar el desarrollo de aplicaciones móviles, ofrecer un rendimiento nativo y permitir la reutilización de código entre diferentes plataformas, además, para aquellos desarrolladores Front-End, la curva de aprendizaje es más rápida.

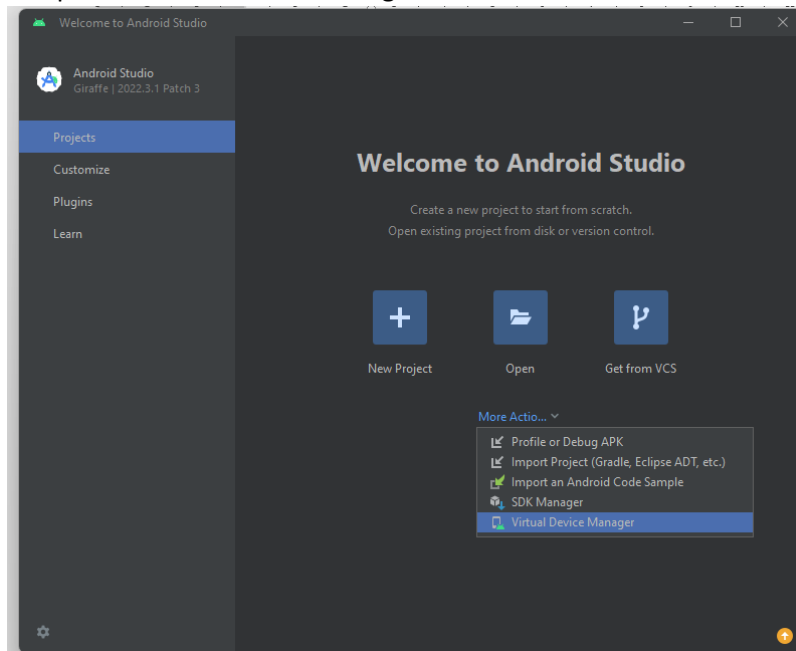
ENTORNO DE DESARROLLO

En este apartado se muestran los recursos necesarios para el desarrollo con React Native y su instalación, comprende el editor y sus dependencias necesarias, node.js como servidor, un emulador y el conversor de código para iOS y Android.

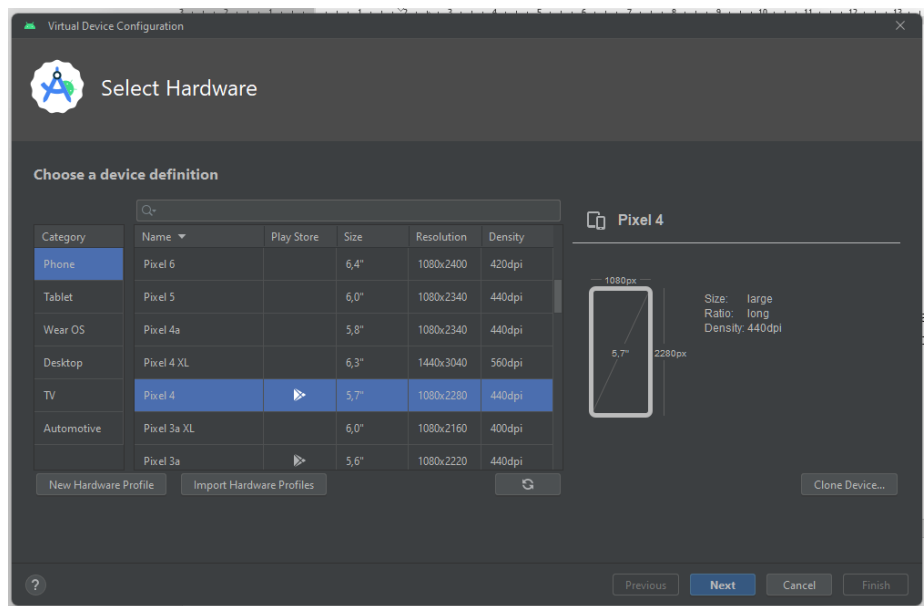
Configurar el entorno de desarrollo

1. Instalar Visual Studio Code (VSC). En el asistente de instalación dejar marcada la opción “Agregar PATH...”
2. Ejecutar VSC e instalar dependencias, en la opción “Extensiones”:
 - a. “ES7 React/Redux/React-Native snippets”, suministra funciones nativas de React y Redux
 - b. “Simple React Snippets”, es un conjunto de ayudas
 - c. “Auto Close Tag”, ayuda para la creación de etiquetas (tags)
 - d. “Paste JSON as Code” para manipular JSON, es posible que ya esté instalado.
 - e. “TypeScript importer”
3. Instalar Android Studio. Desde la página oficial de Android Studio “developer.android.com/studio”.
 - a. Dejar habilitado “Android Virtual Device”
4. Instalar Node.js, de “node.org/es/download/”. Descargar la última versión. Dejar las opciones por defecto. En consola ejecutar “node --version” para verificar su correcta instalación.
5. Instalar Postman (postman.com/downloads/), para realizar peticiones HTTP (opcional)
6. Crear un emulador
 - a. Ejecutar Android Studio

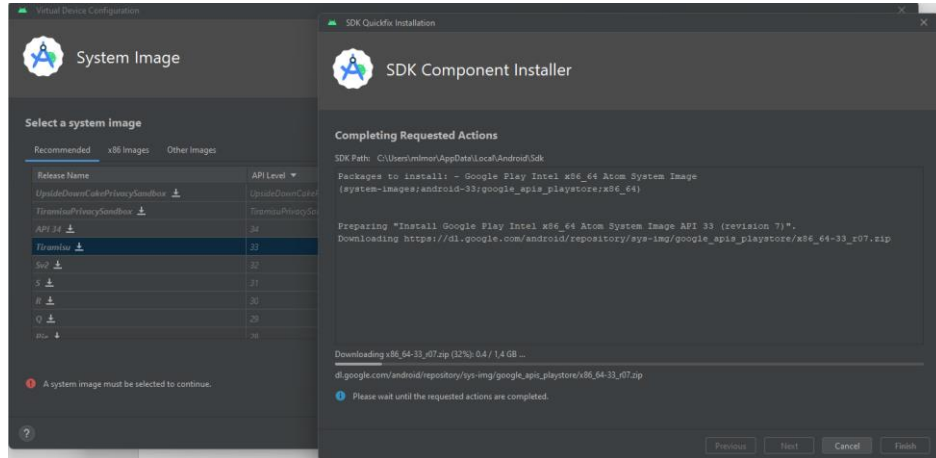
b. La opción “Virtual Device Manager”



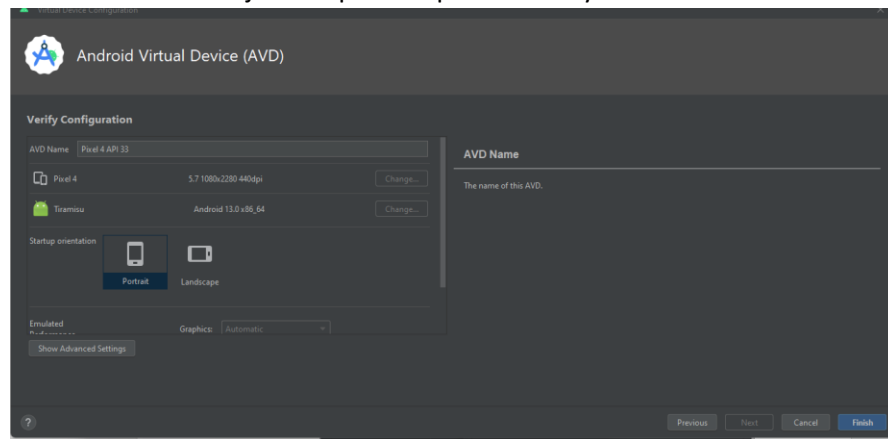
c. “Create Device”, seleccionar “Phone” y buscar un emulador que tenga los servicios de Google Service para probar las aplicaciones sin restricciones (icono de Play Store)



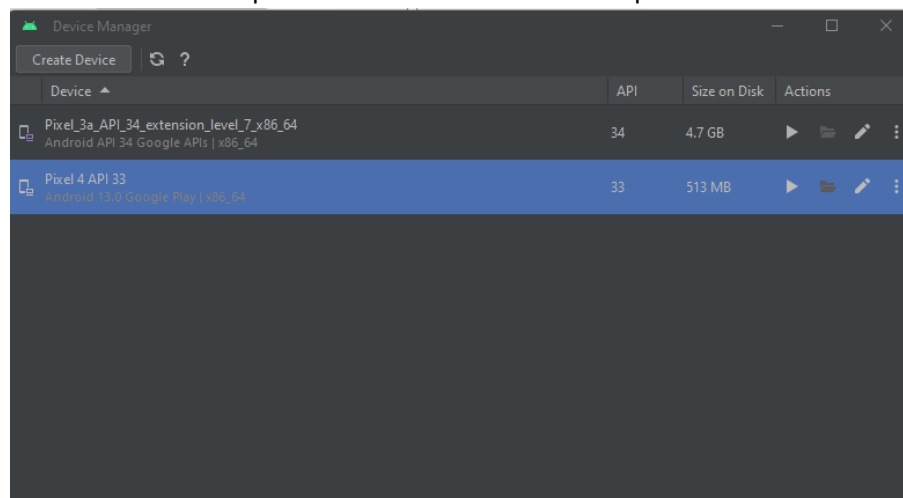
- d. Botón “Next” y descargar una versión, en este caso “Tiramisu”, descargar la API correspondiente

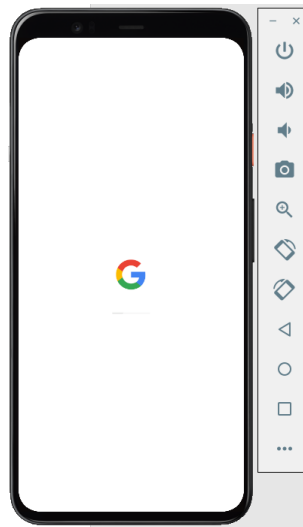


- e. Puede dejar las opciones por defecto y botón “Finish”

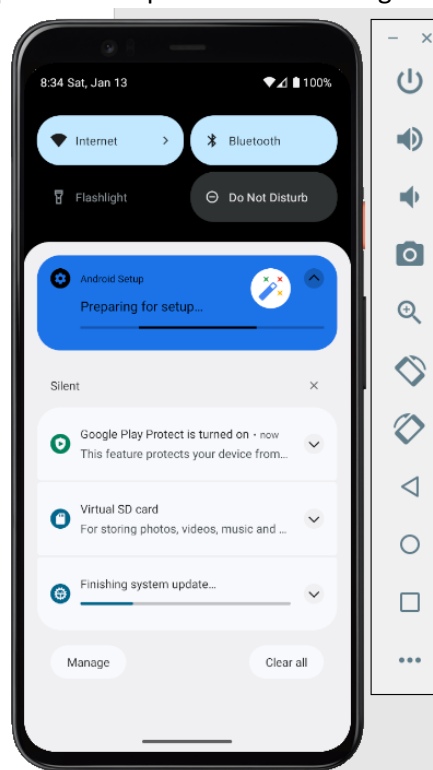


- f. Para ejecutar el emulador en la punta de flecha de la columna “Actions”, la primera vez demorará más tiempo.

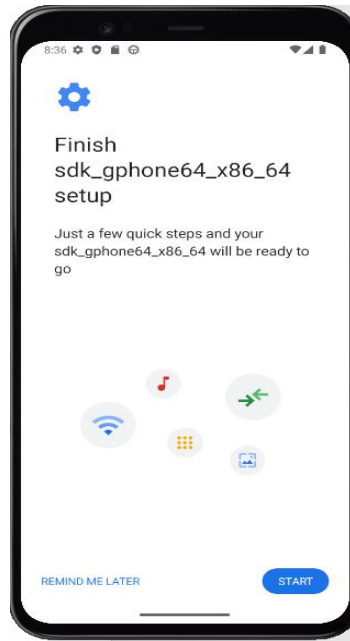




- g. En la parte superior del emulador, deslizar para visualizar las notificaciones, esperar hasta que termine la configuración.



- h. Clic en el área azul, lleva a la siguiente pantalla



- i. Mas adelante, ingresar en el emulador, su correo Gmail
7. Instalar Expo CLI, Una interfaz de línea de comandos que permite montar un servidor para correr el proyecto localmente, construir proyectos sencillos y publicarlo. En Google buscar “expo install” y el enlace “Como instalar Expo (React Native) – gists – GitHub”
- a. Copiar el comando “npm install -g expo-cli exp” y ejecutarlo en Power Shell, como administrador

```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\WINDOWS\system32> npm install -g expo-cli exp
```

- b. Verificar con “expo --version”.