|  |  |
| --- | --- |
| **Diseño y Programación de Software Multiplataforma DPS941**  **Ciclo 02-2024**  **GT 01** | Logotipo, nombre de la empresa  Descripción generada automáticamente |

**Foro 2**

**Docente: \_** **Alexander Alberto Sigüenza Campos \_\_\_**

**Integrantes del equipo:**

|  |  |  |
| --- | --- | --- |
| Apellidos | Nombres | Carné |
| Mendoza Moreno | Alberto Joseph | MM200462 |
| Orellana Murcia | German Stanley | OM230518 |
| Henríquez Salmerón | Bryam Benjamín | HS232171 |
| Ayala González | Antonia Francisca | AG161316 |

**Fecha de entrega:** noviembre (vía Aula Digital)

ÍNDICE

**No se encontraron entradas de tabla de contenido.**

# INTRODUCCION

**1. Investigación sobre Opciones de Autenticación**

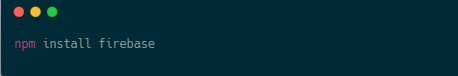
Firebase ofrece varias opciones de autenticación para aplicaciones móviles y web. Aquí tienes un resumen de cada una:

* **Correo Electrónico y Contraseña**:
  + **Características**: Proporciona autenticación mediante un formulario con email y contraseña.
  + **Ventajas**: Fácil de implementar, comúnmente usado, compatible con recuperación de contraseña.
  + **Desventajas**: Depende del usuario para recordar credenciales, requiere almacenamiento seguro.
  + **Consideraciones**: Usar conexiones seguras (HTTPS) y configurar reglas de seguridad en Firebase.
* **Autenticación con Google**:
  + **Características**: Permite a los usuarios iniciar sesión usando cuentas de Google.
  + **Ventajas**: Experiencia de usuario mejorada, evita la necesidad de crear nuevas contraseñas.
  + **Desventajas**: Requiere configuración en la consola de Google Cloud.
  + **Consideraciones**: Verifica el cumplimiento de la política de privacidad de Google.
* **Autenticación Anónima**:
  + **Características**: Crea un usuario anónimo en Firebase.
  + **Ventajas**: Útil para probar funciones sin crear una cuenta.
  + **Desventajas**: Limitada en términos de funcionalidad avanzada.
* **Autenticación por Teléfono**:
  + **Características**: Verificación mediante SMS.
  + **Ventajas**: Acceso rápido y fácil.
  + **Desventajas**: Puede ser costoso por los mensajes SMS.

# 2. Documentación de Implementación de Autenticación por Correo Electrónico

Pasos para configurar e implementar autenticación por correo electrónico:

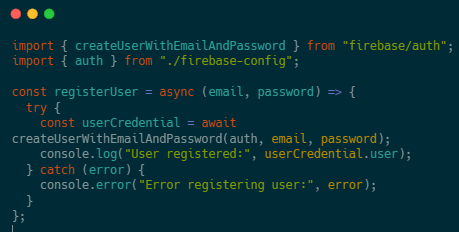
1. **Configurar Firebase**:
   * Crea un proyecto en Firebase Console.
   * Activa el método de autenticación "Email/Password" en la pestaña "Authentication".
2. **Instalar Firebase en el Proyecto React Native**:



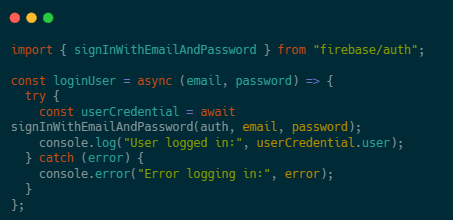
1. **Configurar Firebase en el Código**: En el archivo firebase-config.js:



1. **Registrar Usuarios**: Código para registro de usuarios:



1. **Iniciar Sesión**: Código para login:



# 3. Desarrollo de Pantalla de Login

**a) Estructura Básica:**

Crea la pantalla de login con React Native y Firebase.

1. **Crear la Pantalla LoginScreen**:



1. **Autenticación con Google**: Instala expo-google-auth-session y configura la autenticación en la consola de Google Cloud.

# 4.Ejercicio practico:

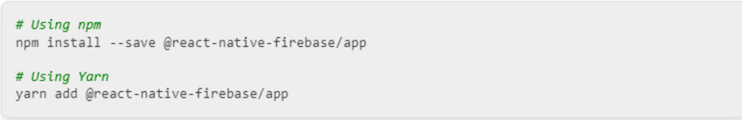
npm y paquetes expo utilizados:

npm install --save @react-native-firebase/app

npm i @react-native-firebase/auth

npx expo install @react-native-google-signin/google-signin

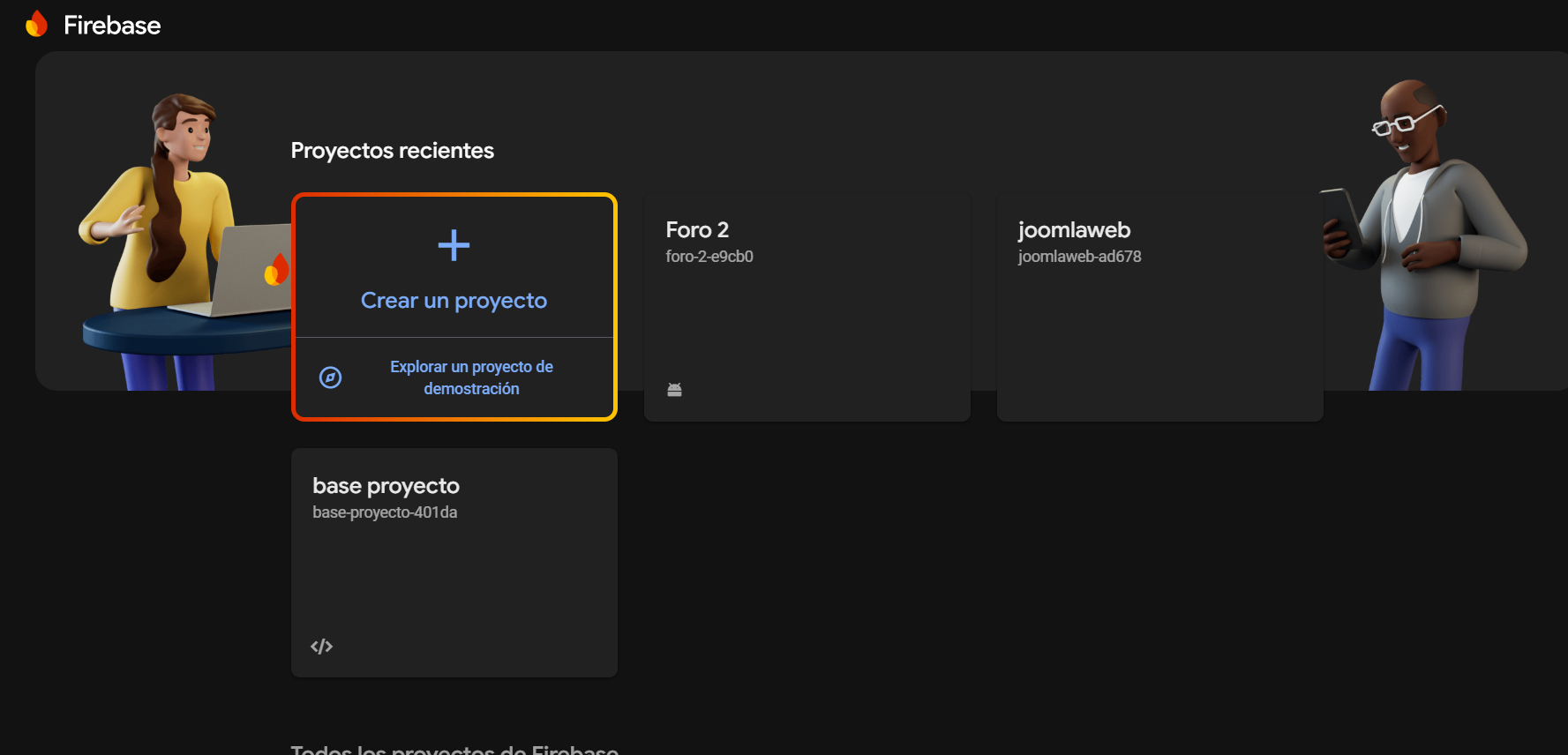
npx expo install expo-dev-client

Primero instalamos los paquetes requeridos.

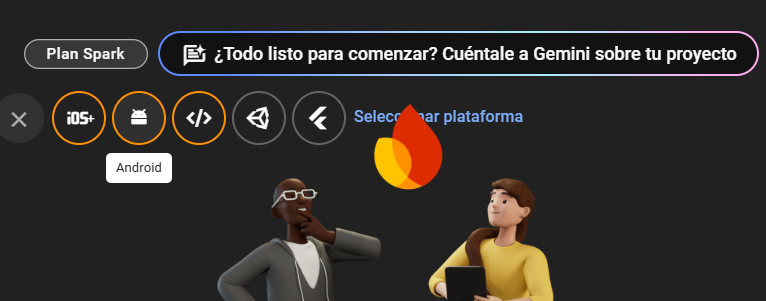


Y así sucesivamente.

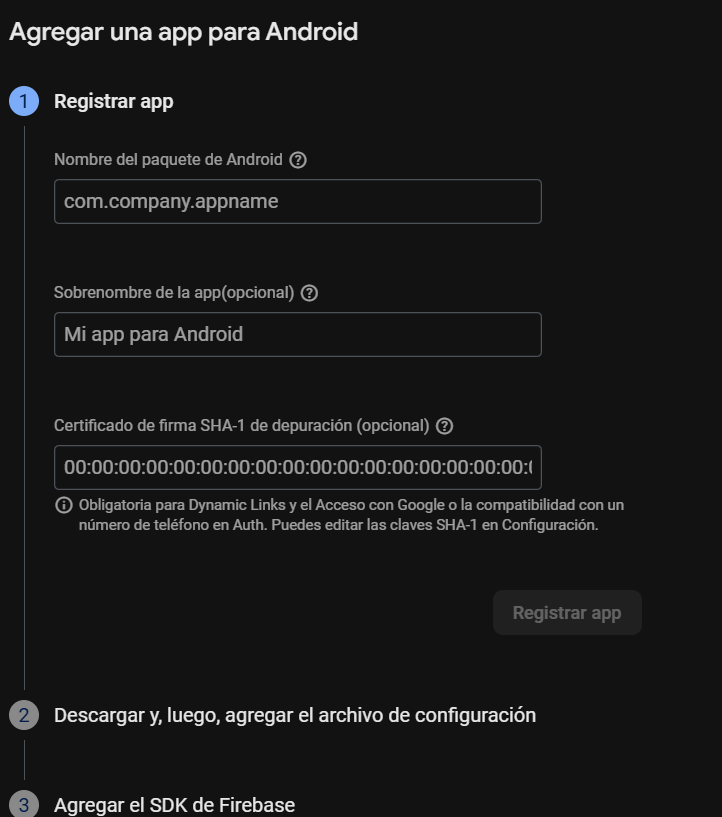
Para poder utilizar los servicios de Firebase para iniciar sesión con Google, es necesario que creemos un proyecto. Podemos hacerlo ingresando al siguiente enlace: <https://console.firebase.google.com/>.

Una vez creado el proyecto, necesitamos registrar nuestra aplicación. 

Elegimos la opción para Android



Llenamos los datos con la información de nuestro proyecto:



Para obtener el nombre del paquete Android, debemos hacer un prebuild de nuestro proyecto expo utilizando el siguiente comando:

npx expo prebuild --platform android

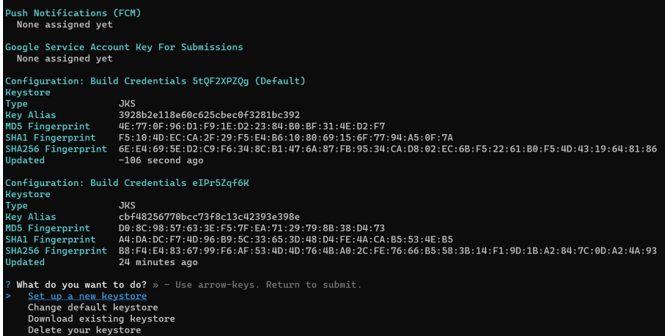
Luego nos pedirá el nombre del paquete, esto generara el Código de app nativa para poder obtener el nombre del paquete.

En nuestro archivo app.json obtenemos el nombre del paquete para luego podes usarlo al registrar nuestra app en firebase.



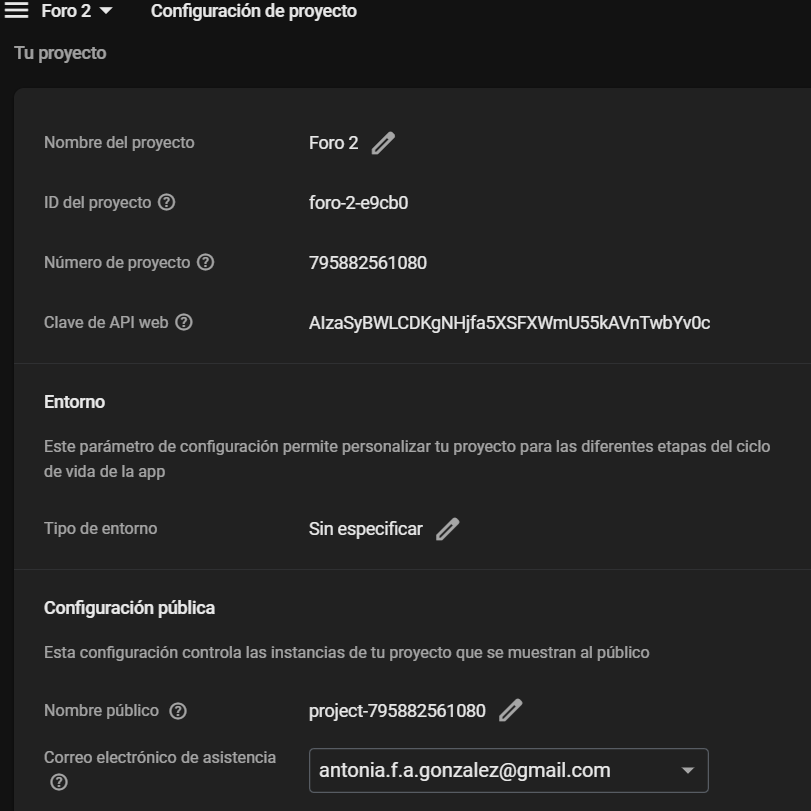
También nos solicita una clave SHA1 para poder registrar la app, para este caso con la ayuda de EAS de expo, creamos una llave para nuestro proyecto con el siguiente comando:

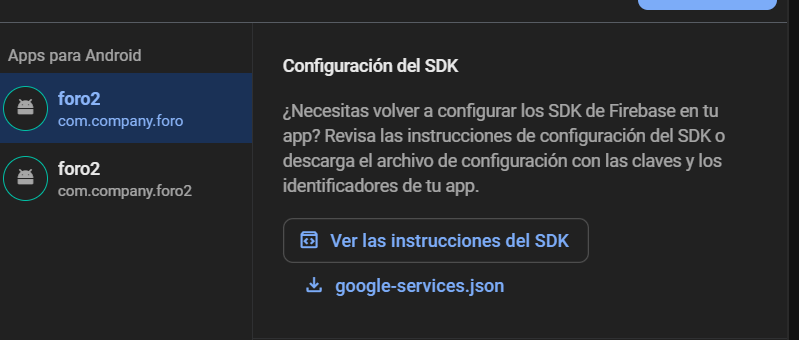
EAS credentials



Y copiamos el dato SHA1 Figereprint del segundo bloque, ese lo usamos para registrar el proyecto FireBase.

Ahora agregamos la llave (default) SHA1 a nuestro proyecto

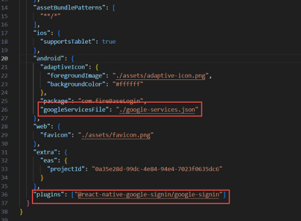


Una vez agregada, descargamos el archivo .json que nos genera firebase.

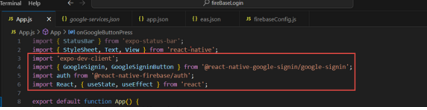
Este lo colocamos en la raíz de nuestro proyecto expo

# 

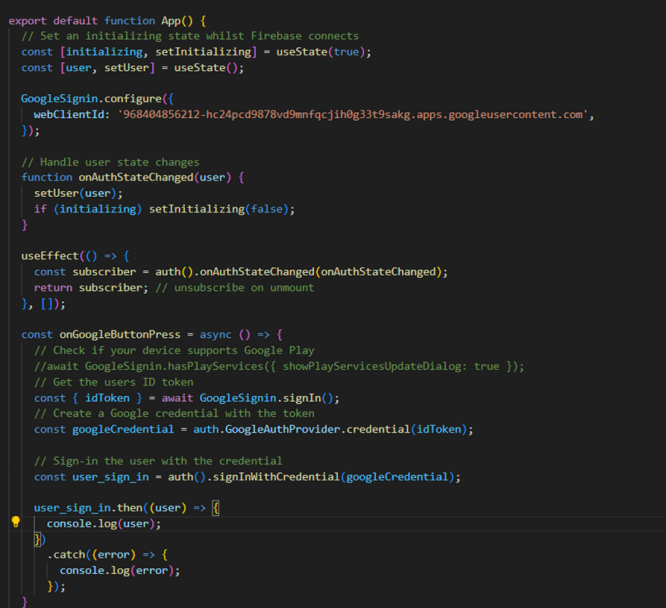
En nuestro archivo app.json agregamos las siguientes líneas.



En el archivo app.js agregamos los siguientes imports



Ahora implementamos los métodos de firebase para registrarnos, junto con el botón Sde inicio de sesión de Google que ya posee la librería.





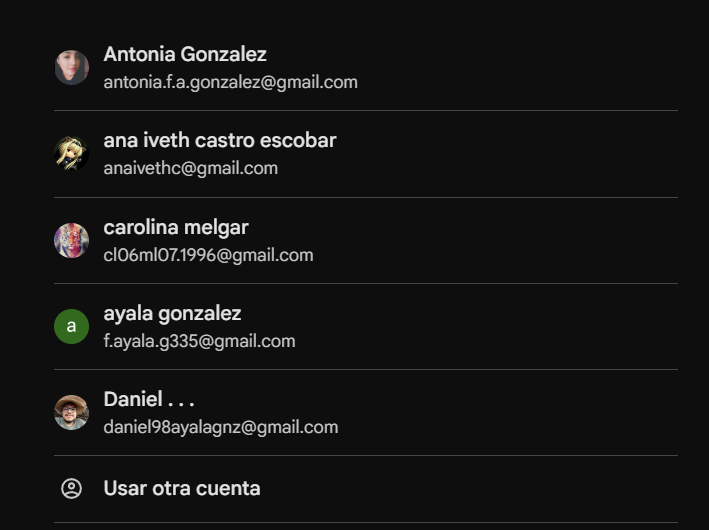
Resultado final:

Damos click en acceder:





Y elegimos la cuenta:





Resultado final:

