

# UNIVERSIDAD DON BOSCO



## DISEÑO Y PROGRAMACION DE SOFTWARE MULTIPLATAFORMA

### Login en React-Native con Autenticación

### Manual de Uso Técnico

Autores

Alvarenga Vásquez, Carlos Benjamín - #AV221218

Siliezar Colocho, Carlos Eduardo - #SC202203

Versión:  
2.0 Fecha:  
27/05/2023

## Índice

<b>1. Planteamiento del Sistema.....</b>	<b>3</b>
<b>1.1 Objetivo del proyecto .....</b>	<b>3</b>
<b>1.2 Alcance del proyecto.....</b>	<b>3</b>
<b>2. Parte Técnica.....</b>	<b>4</b>
<b>2.1.1 Instalación y uso implementado de Node.js .....</b>	<b>5</b>
<b>2.1.2 Instalación y uso implementado react native .....</b>	<b>5</b>
<b>2.1.3 Instalación y uso implementado de Firebase .....</b>	<b>6</b>
<b>2.1.4 Capturas.....</b>	<b>6</b>
<b>3. Referencias.....</b>	<b>9</b>

# 1. Planteamiento del Sistema

## 1.1 Objetivo del proyecto

El objetivo de implementar un sistema de login con autenticación en una aplicación de React Native es garantizar la seguridad y privacidad de los usuarios al permitirles acceder a la aplicación solo mediante un proceso de autenticación previo. Esto se logra mediante la implementación de un sistema de autenticación sólido que incluya medidas de seguridad tales como el cifrado de contraseñas y la gestión adecuada de tokens de autenticación. Al implementar correctamente el sistema de login con autenticación, se puede garantizar que solo los usuarios autorizados tengan acceso a la aplicación y a la información confidencial de los mismos.

## 1.2 Alcance del proyecto

Implementar un sistema de login con autenticación en una aplicación de React Native incluye las siguientes actividades:

- **Análisis de requisitos:** Comprender los requisitos específicos del proyecto y definir las características necesarias para el sistema de autenticación, incluyendo la gestión de usuarios, la autenticación de usuarios y la recuperación de contraseñas.
- **Diseño de la interfaz de usuario:** Crear un diseño de interfaz de usuario intuitivo y fácil de usar para la pantalla de inicio de sesión y el proceso de autenticación.
- **Desarrollo del sistema de autenticación:** Implementar un sistema de autenticación seguro y robusto utilizando las mejores prácticas de seguridad de la industria, incluyendo el cifrado de contraseñas y la gestión adecuada de tokens de autenticación.
- **Pruebas de unidad y pruebas de integración:** Realizar pruebas de unidad y pruebas de integración para garantizar que el sistema de autenticación cumpla con los requisitos y esté libre de errores.
- **Integración con la aplicación:** Integrar el sistema de autenticación en la aplicación de React Native y garantizar que funcione correctamente en todas las plataformas móviles compatibles.
- **Documentación:** Crear documentación detallada del sistema de autenticación y proporcionar instrucciones claras para su uso y mantenimiento.

## 2. Parte Técnica

### 2.1 Listado de programas a utilizar para la realización de la aplicación.

Para comenzar a desarrollar nuestra aplicación, debemos tomar en cuenta el trabajar con herramientas que nos facilitaran el desarrollo de la aplicación, manejo de datos y la manera en que generamos estética a nuestra app, por lo tanto, nosotros hemos decidido trabajar con:

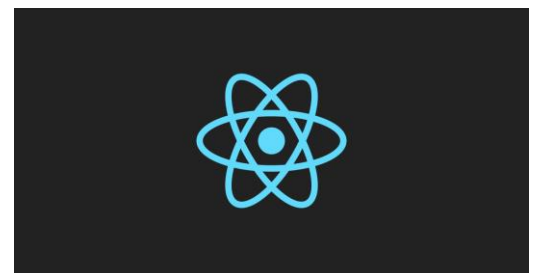
- **Node.js:** Node.js es un entorno de tiempo de ejecución de JavaScript (de ahí su terminación en “.js” haciendo alusión al lenguaje JavaScript). Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript. También aporta muchos beneficios y soluciona muchísimos problemas.



- **Firebase:** es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles lanzada en 2011 y adquirida por Google en 2014.

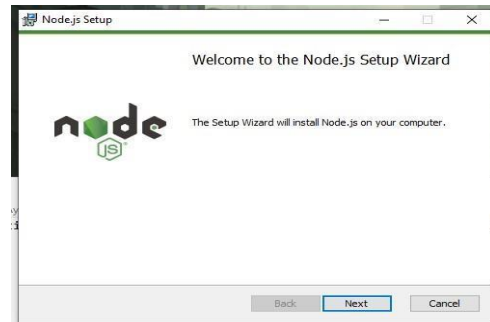


- **React Native:** es un framework de código abierto creado por Meta Platforms, Inc. Se utiliza para desarrollar aplicaciones para Android, Android TV, iOS, macOS, tvOS, Web, Windows y UWP al permitir que los desarrolladores usen React con las características nativas de estas plataformas.



### 2.1.1 Instalación y uso implementado de Node.js

Comenzaremos instalando node.js desde su web oficial para evitar programas no deseados o virus <https://nodejs.org/en/> lo descargamos y procederemos a instalar.



```
C:\WINDOWS\system32\cmd.exe - node
Microsoft Windows [Versión 10.0.19044.2251]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\carlo>node
welcome to Node.js v19.0.0.
type ".help" for more information.
> console.log("Hola Mundo")
Hola Mundo
undefined
>
```

El uso de node.js lo vemos reflejado a la hora de entrara nuestro Símbolo del sistema, ya que ahí es donde inicializamos todo, realizamos el ejemplo con un hola mundo:

Ahora bien, los que nos interesa a nosotros de node.js es el comando “npm” el cual es el manejador de paquetes de Node con él podemos instalar dependencias a nuestro proyecto o instalar programas globalmente en nuestro sistema. Es Ahi donde procedemos a la siguiente instalación.

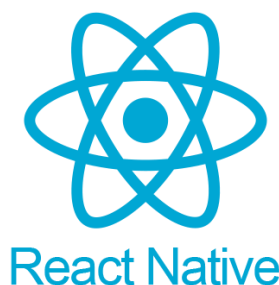
### 2.1.2 Instalación y uso implementado react native

Puede crear una aplicación de escritorio de Windows mediante React Native para Windows siguiendo estos pasos.

Abra una ventana de la línea de comandos (terminal) y vaya al directorio donde desea crear el proyecto de aplicación de escritorio de Windows.

Puede usar este comando con el ejecutor de paquetes de Node (NPX) para crear un proyecto de React Native sin necesidad de instalar herramientas adicionales de forma local o global. El comando generará una aplicación de React Native en el directorio especificado por <projectName>.

**npx react-native init <projectName>**



### 2.1.3 Instalación y uso implementado de Firebase

Para poder hacer uso del proveedor de autenticación de Firebase es necesario crear y configurar un proyecto en la plataforma de [Firebase.com](https://firebase.com), dentro del cual se registrará una nueva app de tipo web, esto con el objetivo de generar el archivo de configuración que se utilizará posteriormente dentro de la app para poder conectarla con el proveedor de autenticación.

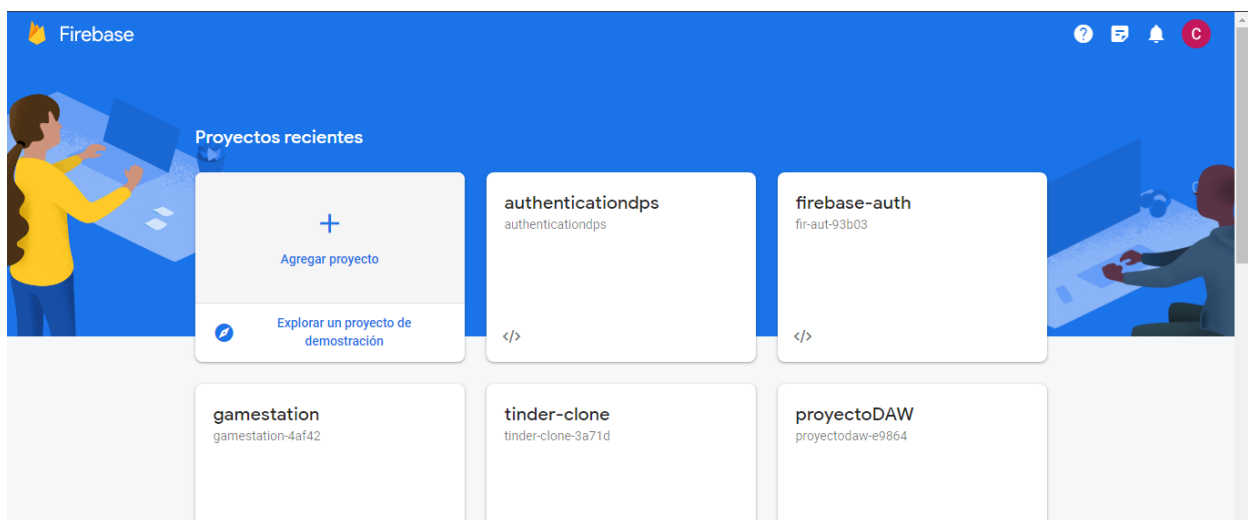
Una vez generadas estas claves, se deberán guardar en un documento dentro de la aplicación, donde también se instalará el paquete de firebase, el cual incluye los módulos de autenticación y de inicialización del consumo del proveedor, al cual se envía como parámetro el archivo de configuración generado anteriormente.

Así mismo, del lado del proyecto firebase, en el apartado de autenticación, se debe habilitar el método de ingreso con correo electrónico y contraseña, para posteriormente utilizar las funciones `createUserWithEmailAndPassword` y `signInWithEmailAndPassword` del lado del frontend, las cuales son proporcionadas por el paquete de firebase previamente instalado dentro del proyecto. A estas funciones se les pasará como parámetro el email y la contraseña, para que puedan ser validados por el proveedor de autenticación.

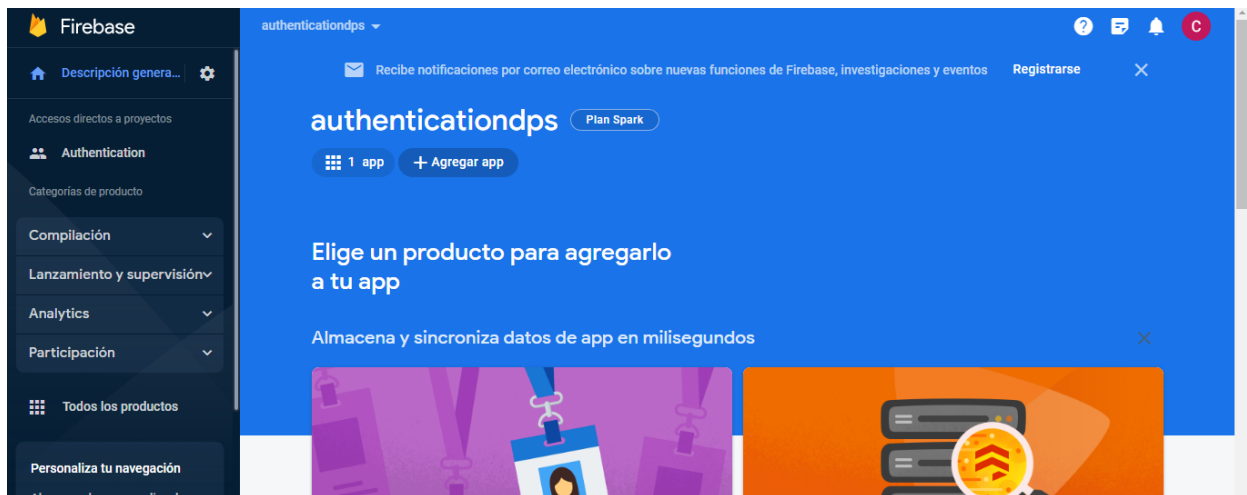
Una vez validados y recibida la respuesta positiva en la app por parte del servidor, se navegará desde la pantalla de inicio de sesión hasta la vista principal, donde, por medio de un contexto compartido, se obtendrán los datos del usuario para imprimir el email en pantalla, y para poder cerrar la sesión del usuario desde esa pantalla.

### 2.1.4 Capturas

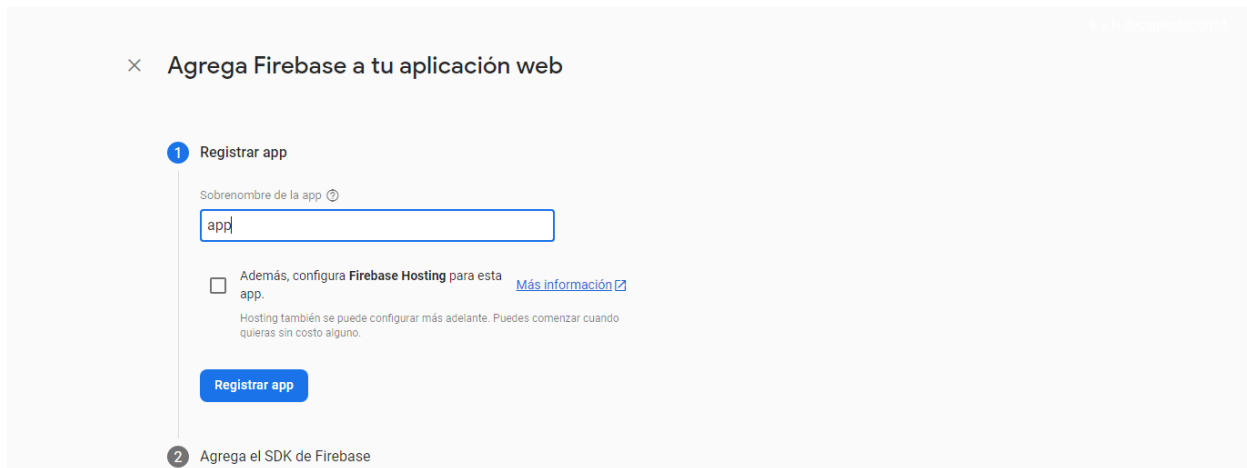
- Crear un nuevo proyecto de firebase.



- Agregar una app dentro del proyecto.



- Registrar app.



- Copiar el archivo de configuración proporcionado y guardarlo dentro del proyecto.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCan5XyEmBPVYFKwIjc-f2TYVG7euYCU9E",
  authDomain: "authenticationdps.firebaseio.com",
  projectId: "authenticationdps",
  storageBucket: "authenticationdps.appspot.com",
  messagingSenderId: "579618828004",
  appId: "1:579618828004:web:da25edc172846746b8c6f1"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

- Implementando el modulo de autenticación y de inicio de la app del paquete de firebase.

```
// Autenticación
import { getAuth, createUserWithEmailAndPassword, signInWithEmailAndPassword } from 'firebase/auth'
import { initializeApp } from 'firebase/app'
import { firebaseConfig } from '../..../firebase_config'
```

```
const app = initializeApp(firebaseConfig)
const auth = getAuth(app)

const [user, setUser] = useUser()

const handleCreateAccount = () => {
  createUserWithEmailAndPassword(auth, email, password)
    .then(userCredential => {
      console.log('Account Created')
      const newUser = userCredential.user
      flush()
      setUser(newUser)
      navigation.navigate('TabNavigator', {screen:'Home'})
    }).catch(err => {
      console.log(err)
      Alert.alert(err.message)
    })
}
```

```
const handleSignIn = () => {
  signInWithEmailAndPassword(auth, email, password)
    .then(userCredential => {
      console.log('Signed In')
      const newUser = userCredential.user
      flush()
      setUser(newUser)
      navigation.navigate('TabNavigator', {screen:'Home'})
    }).catch(err=>{
      console.log(err)
      Alert.alert(err.message)
    })
}
```



### 3. Referencias

Referencia	Título
<a href="https://firebase.google.com/docs?hl=es-419">https://firebase.google.com/docs?hl=es-419</a>	<b>Manual de Firebase</b>
<a href="https://docs.expo.dev/get-started/installation/">https://docs.expo.dev/get-started/installation/</a>	<b>Usar expo</b>
<a href="https://learn.microsoft.com/es-es/windows/devenvironment/javascript/nodejs-beginners-tutorial">https://learn.microsoft.com/es-es/windows/devenvironment/javascript/nodejs-beginners-tutorial</a>	<b>Node.js en Windows para principantes</b>
<a href="https://reactnative.dev/">https://reactnative.dev/</a>	<b>Como instalar React Native</b>