

**TERCER PARCIAL**

Práctica Firebase Realtime Database

**T.S.U. TECNOLOGÍAS DE LA INFORMACIÓN**  
**Área Desarrollo de Software Multimedia**

**REALIZADO POR:**

Higuera Sanchez Dulce Mariela

**MATRICULA:**

0322103734

**GRUPO: 5A**

**MATERIA:**

Desarrollo móvil multiplataforma

**DOCENTE:**

Dr. Parra Galaviz Ray Brunett

## Contenido

1. INTRODUCCIÓN.....	4
2. REALIZAR UN APP EN REACT NATIVE.....	4
3. CONECTARSE A FIREBASE REALTIME DATABASE.....	4
4. ESCRIBIR DATOS EN CUALQUIER COLECCIÓN.....	6
5. LEER DATOS DE CUALQUIER COLECCIÓN.....	8
6. EVIDENCIAR LAS TRANSACCIONES A LA BASE DE DATOS.....	8
7. SUBIR LAS EVIDENCIAS.....	9
8. CONCLUSIÓN.....	10
9. FUENTES CONSULTADAS.....	10

## Índice de ilustraciones.

Ilustración 1. Creación de app en react native. ....	4
Ilustración 2. Instalando firebase.....	4
Ilustración 3. Creando proyecto.....	4
Ilustración 4. Realtime database activation. ....	5
Ilustración 5. Creando base de datos. ....	5
Ilustración 6. Configurando la región.....	5
Ilustración 7. Base de datos sin registros.....	6
Ilustración 8. Permisos de usuario . ....	6
Ilustración 9. Función de escribir datos en la coleccion usuarios.....	7
Ilustración 10. Escribir datos en la colección usuarios. ....	7
Ilustración 11. Leyendo datos de la colección usuarios. ....	8
Ilustración 12. Colección usuarios con datos. ....	9
Ilustración 13. Transacciones a la base de datos.....	9

## 1. INTRODUCCIÓN.

Firebase Realtime Database es una base de datos NoSQL alojada en la nube que permite a los desarrolladores almacenar y sincronizar datos en tiempo real entre usuarios., siendo una de las soluciones de Firebase, una plataforma de desarrollo de aplicaciones de Google. Este producto esta diseñado para almacenar y sincronizar datos de usuarios en tiempo real, permitiendo que los usuarios accedan a los datos desde cualquier dispositivo, ya sea web o móvil.

## 2. REALIZAR UN APP EN REACT NATIVE.

I. Crear una aplicación en el Visual Studio Code, para después conectarla a firebase y hacer modificaciones en tiempo real.

```
PS C:\Users\Lenovo\Desktop\Firebase> npx create-expo-app realtimeDB --template
✓ Choose a template: » Blank
✓ Downloaded and extracted project files.
> npm install
[#####] ..... - idealTree:react-native: sill placeDep ROOT chalk@2.4.2 OK for: @babel/highlight@7.24.7 want: ^2.
```

*Ilustración 1. Creación de app en react native.*

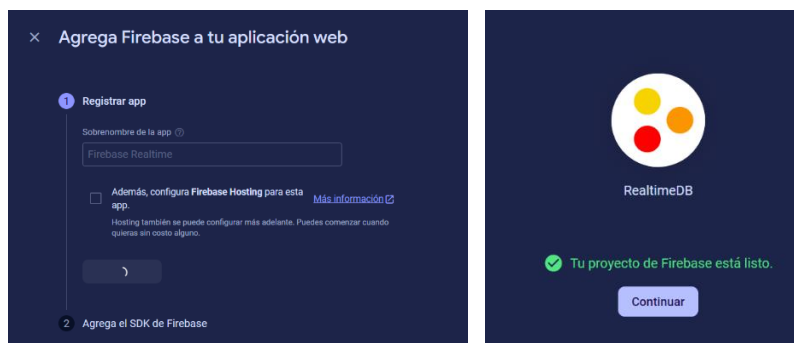
II. Ahora se agregan la siguiente línea de comandos para tener el proyecto completo y comenzar a configurar.

```
532-debug-0.10g
PS C:\Users\Lenovo\Desktop\Firebase> npm install firebase @react-native-firebase/app @react-native-firebase/auth
```

*Ilustración 2. Instalando firebase.*

## 3. CONECTARSE A FIREBASE REALTIME DATABASE.

I. Para ello, se crea un nuevo proyecto en firebase, y una vez creado se copian las credenciales generadas.



*Ilustración 3. Creando proyecto.*

II. Una vez creado el proyecto, se va a todos los productos, y se activa la parte de Realtime database.

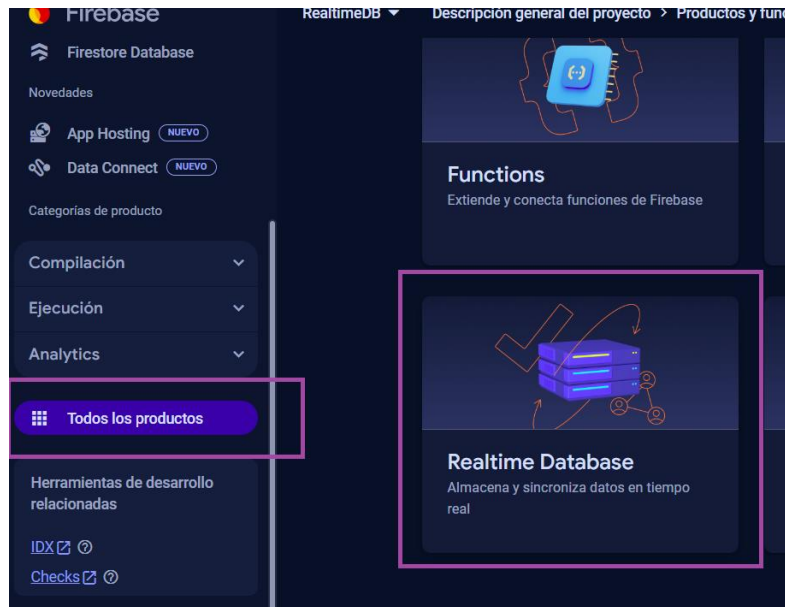


Ilustración 4. Realtime database activation.

III. Al ser por primera vez, se debe de realizar ciertas configuraciones, entonces para ello le damos en crear una base de datos, seguido de la región donde serán almacenados dichos datos, y los permisos de lectura y escritura, como esta práctica es con fines de poder escribir y leer los datos, no es necesario el identificarse / loguearse para poder hacerlo.

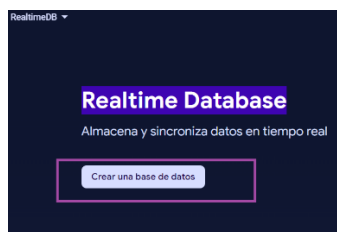


Ilustración 5. Creando base de datos.

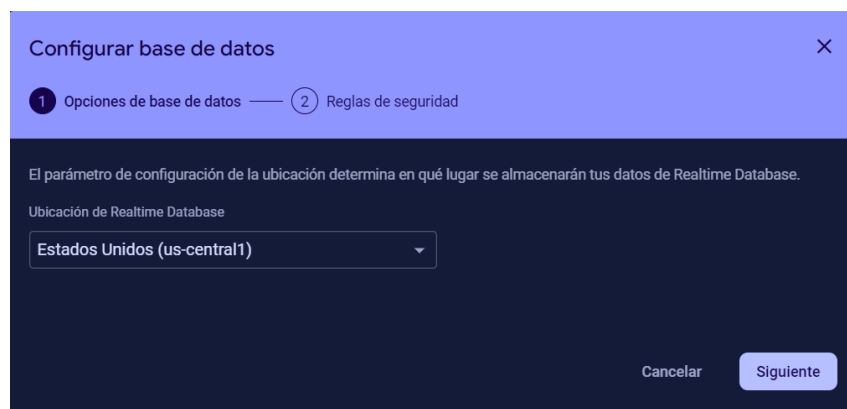
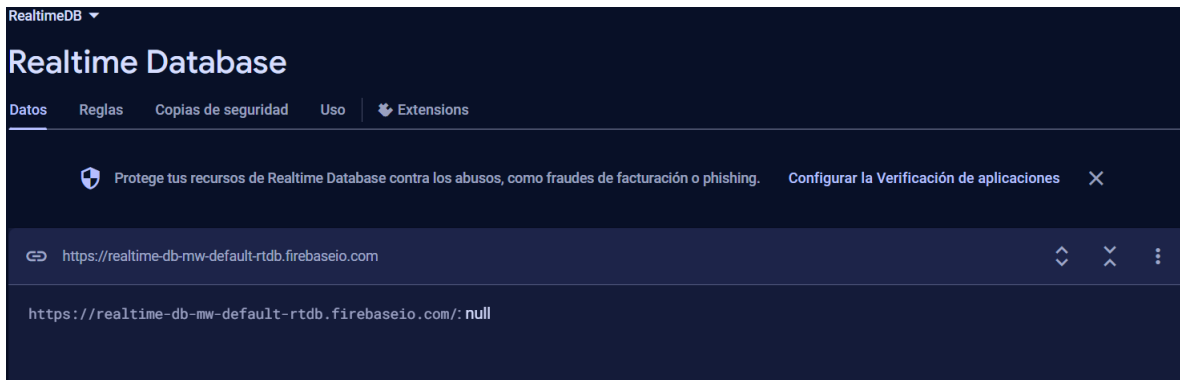


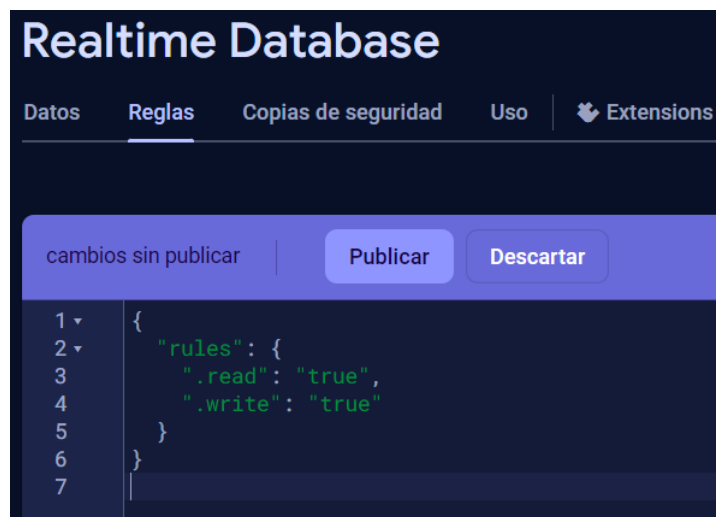
Ilustración 6. Configurando la región.

IV. Una vez creada la base de datos, aparecerá con el parámetro de null, esto se debe a que aún no existen datos registrados, puesto que la base de datos se acaba de crear.



*Ilustración 7. Base de datos sin registros.*

V. Para ello, se hará el código correspondiente en el proyecto creado, pero antes los usuarios deben de tener los permisos de escritura y lectura, para ello ir a reglas, y ver que están activados.



*Ilustración 8. Permisos de usuario*

#### 4. ESCRIBIR DATOS EN CUALQUIER COLECCIÓN.

I. Se hace el código correspondiente, donde después de inicializar e importar las librerías en el código, así como las dependencias que se ocupan, se hace la función que permite escribir datos en la tabla que se creará llamada “Usuarios”. Se hace la promesa de que, si se registra el campo nombre, y edad, entonces es un registro exitoso, en caso contrario la promesa no se cumple y no se guardan los datos.

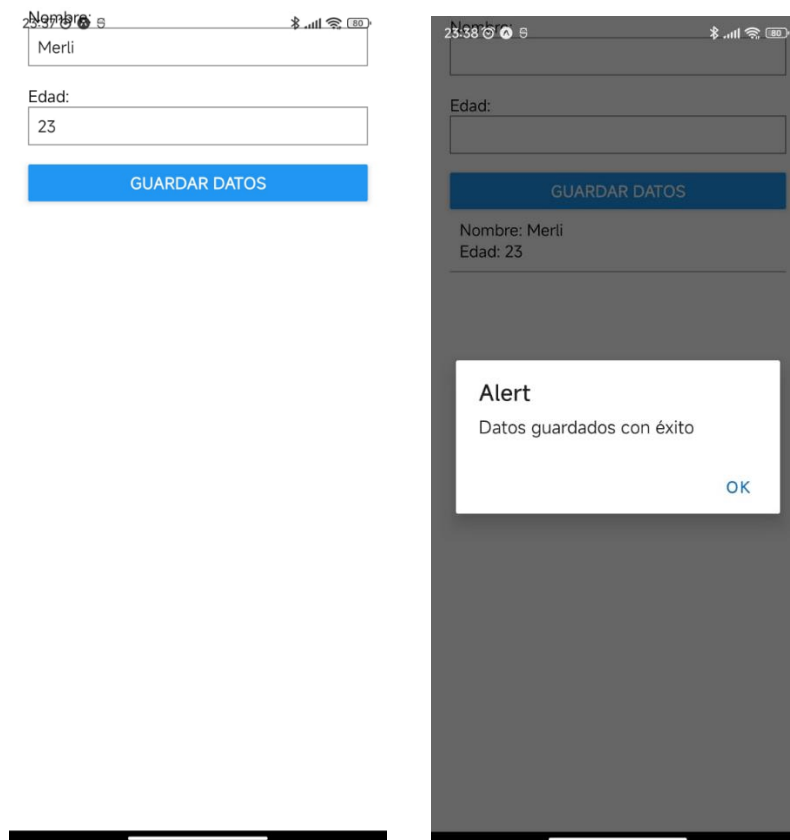
```

40 //Escribir datos en la base de datos
41 const escribirDatos = () =>
42 {
43   const usuariosRef = ref(database, 'usuarios/');
44   push(usuariosRef, {
45     nombre,
46     edad,
47   }).then(() =>
48   {
49     alert('Datos guardados con éxito');
50     setNombre('');
51     setEdad(''); //Limpian los nombres
52   }).catch(error =>
53   {
54     alert('Error al guardar los datos: ' + error.message);
55   });
56 };
57

```

*Ilustración 9. Función de escribir datos en la colección usuarios.*

II. En la aplicación móvil, aparece un cuadro para ingresar los datos, que al ser guardados de manera exitosa, avisa como una notificación, además de ir guardando los valores ingresados.



*Ilustración 10. Escribir datos en la colección usuarios.*

## 5. LEER DATOS DE CUALQUIER COLECCIÓN.

I. Es posible leer los datos de esa colección, dato que se van almacenando en y mostrando conforme se llenan los campos ingresados, siendo el campo nombre de tipo cadena y el campo edad únicamente numérico.

Nombre: Ray B

Edad:

GUARDAR DATOS

Nombre: Merli  
Edad: 23

Nombre: Edwin G  
Edad: 26

Nombre: Emilio H  
Edad: 18

Nombre: Noel A  
Edad: 21

Nombre:

Edad:

GUARDAR DATOS

Nombre: Melissa B  
Edad: 31

Nombre: José Andrea  
Edad: 51

Nombre: Javier Z  
Edad: 52

Nombre: Patricia T  
Edad: 49

Nombre: Leo Jiménez  
Edad: 43

Nombre: Txus D  
Edad: 60

Nombre: Ramón W  
Edad: 76

Ilustración 11. Leyendo datos de la colección usuarios.

II. Al final se hicieron más pruebas para tener múltiples datos y que se muestren en la colección usuarios.

## 6. EVIDENCIAR LAS TRANSACCIONES A LA BASE DE DATOS.

I. Al tener varios datos, ya no se encuentra la tabla en null directo en firebase, al contrario, se tienen múltiples registros con un ID proporcionado por la plataforma.



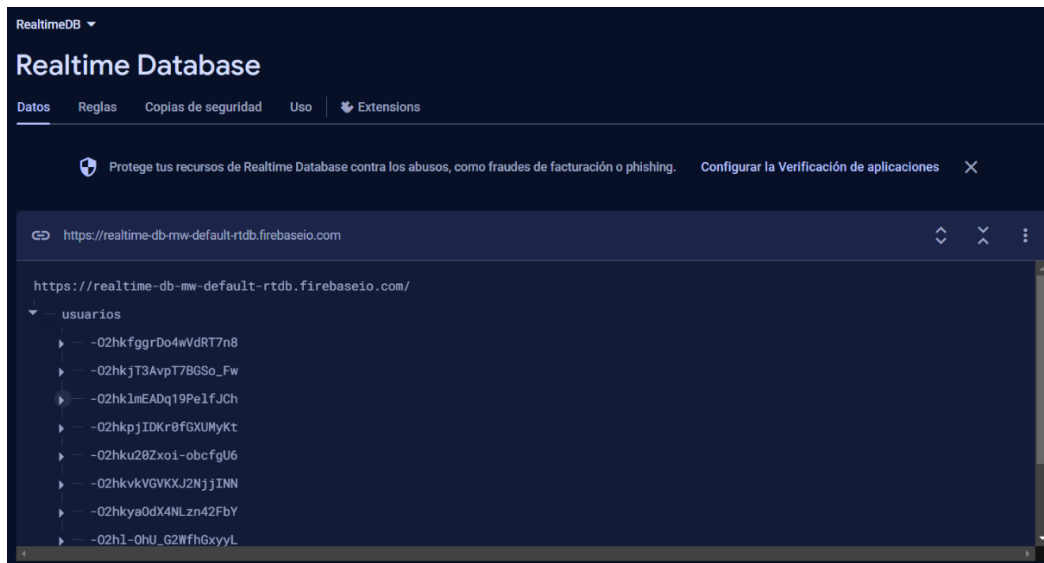


Ilustración 12. Colección usuarios con datos.

II. Al desglosar un ID, se pueden observar los campos de nombre y edad llenados según el orden en que fueron ingresados desde el dispositivo móvil.

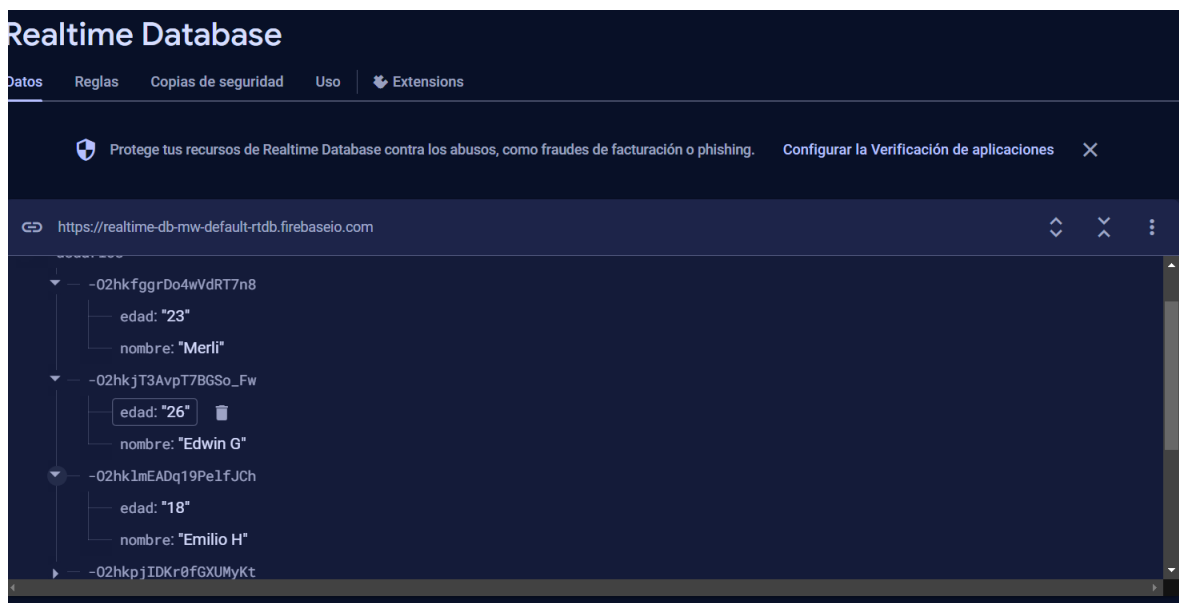


Ilustración 13. Transacciones a la base de datos.

## 7. SUBIR LAS EVIDENCIAS.

Se encuentra el código correspondiente a esta práctica, junto con el documento en pdf en: [https://github.com/MerliGH/Desarrollo\\_Movil/tree/main/Parcial3](https://github.com/MerliGH/Desarrollo_Movil/tree/main/Parcial3)

## 8. CONCLUSIÓN.

El uso de esta base de datos esta diseñada para tipo nosql, puesto que los registros los convirtió en objetos JSON, y su estructura es igual a un JSON, este tipo de uso en firebase permite mandar datos en tiempo real y subirlos al momento, sus configuraciones son más sencillas en lugar de una BD local, por lo que simplifica procesos y ahorra mucho tiempo el usarla.

## 9. FUENTES CONSULTADAS.

- Firebase (2024). Firebase Google Documentation. Recuperado el 26 de julio de 2024, de: <https://firebase.google.com/products/realtime-database?hl=es-419>
- Vazquez, A. (25 de septiembre, 2021). Bases de datos en tiempo real con Firebase y React Native | React Native en Español. Youtube. Recuperado el 26 de julio de 2024, de: <https://www.youtube.com/watch?v=xDGpTY7cwhU>
- Navarro, A. (25 de febrero, 2020). Conecta FIREBASE con REACT o REACT NATIVE. Youtube. Recuperado el 26 de julio de 2024, de <https://www.youtube.com/watch?v=pfyC1VSzVHY>