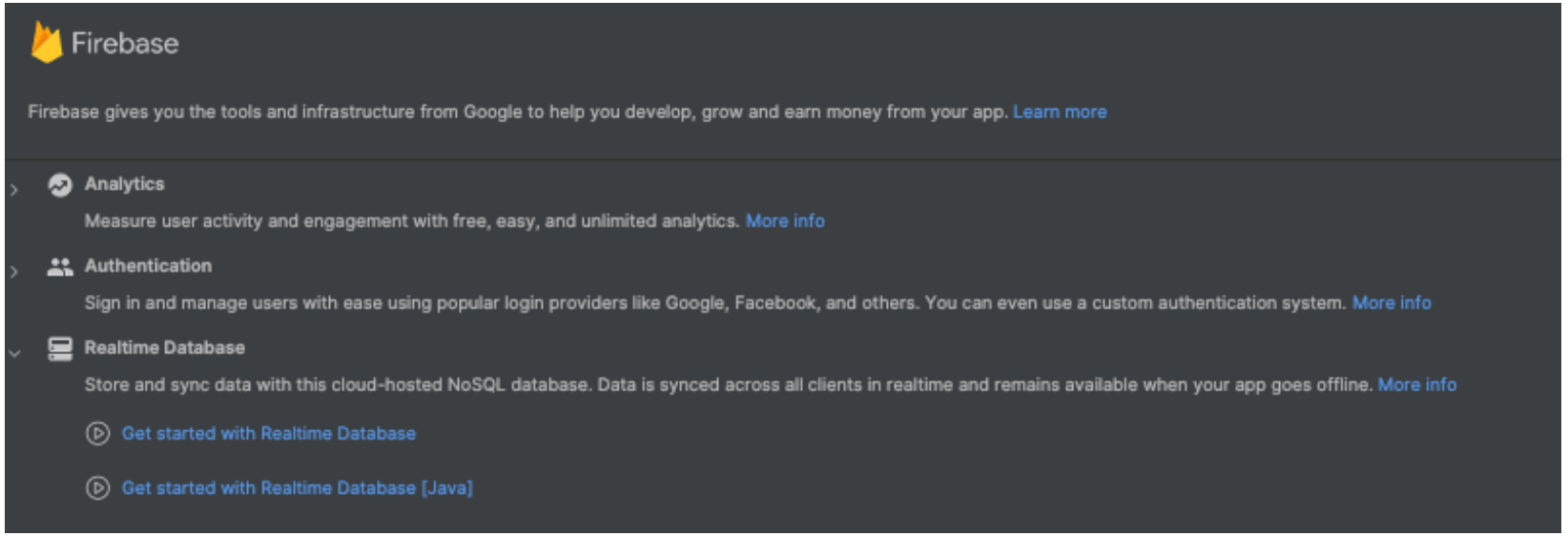
SANTO TOMÁS
INSTITUTO PROFESIONAL

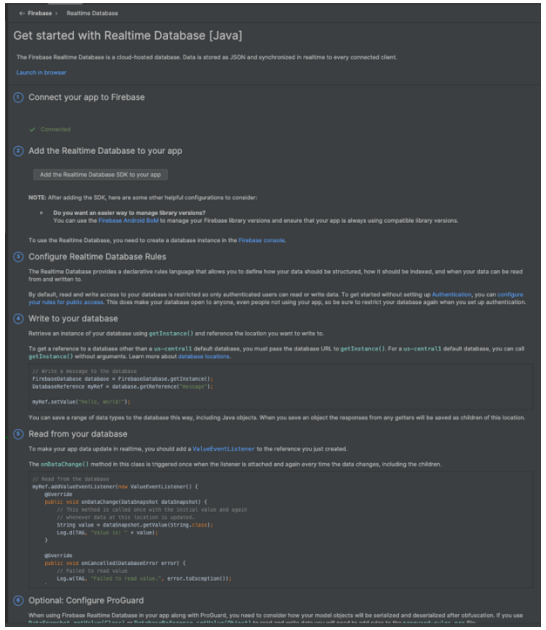
PROGRAMACIÓN ANDROID

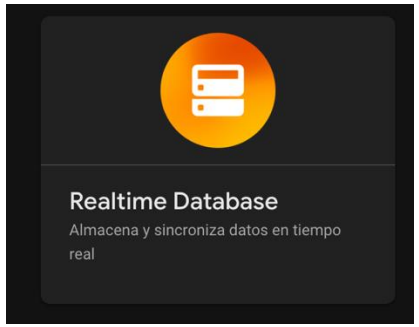
CRUD FIRESTORE

Implementación de un CRUD en Android con Firebase

Conexión a Firebase







Organización del Proyecto

- Estructura de carpetas:
  - `ui`: Contiene los fragmentos (`BuscarFragment`, `CrearFragment`, `ListarFragment`).
  - `models`: Contiene la clase `Contacto`.
  - `activities`: Contiene la `MainActivity` para la navegación.
- Clase `Contacto`:
  - Atributos: `id`, `nombre`, `edad`.
  - Métodos para obtener cada atributo (`getId()`, `getNombre()`, `getEdad()`).

Guardar Contacto en Firebase

```
databaseReference.child(id).setValue(contacto)
        .addOnSuccessListener(aVoid -> {
            Toast.makeText(getContext(), text: "Contacto guardado correctamente", Toast.LENGTH_SHORT).show();
            limpiarFormulario();
        })
        .addOnFailureListener(e -> {
            Toast.makeText(getContext(), text: "Error al guardar el contacto", Toast.LENGTH_SHORT).show();
        });
```

SANTO TOMÁS
INSTITUTO PROFESIONAL

Listar Contactos desde Firebase

```java
public class ListarFragment extends Fragment {

    private ListView listViewContactos;  2 usages
    private ArrayAdapter<String> adapter;  3 usages
    private ArrayList<String> listaContactos;  5 usages
    private DatabaseReference databaseReference;  2 usages

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
                             @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_listar, container,  attachToRoot: false);

        // Inicializar la vista y la lista
        listViewContactos = view.findViewById(R.id.list_view_contactos);
        listaContactos = new ArrayList<>();
        adapter = new ArrayAdapter<>(getContext(), android.R.layout.simple_list_item_1, listaContactos);
        listViewContactos.setAdapter(adapter);

        // Inicializar la referencia a la base de datos
        databaseReference = FirebaseDatabase.getInstance().getReference( path: "contacto");

        // Obtener los contactos desde Firebase
        obtenerContactos();

        return view;
    }

    private void obtenerContactos() {  1 usage
        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override  2 usages
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                listaContactos.clear();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
                    Contacto contacto = dataSnapshot.getValue(Contacto.class);
                    if (contacto != null) {
                        // Agregar el contacto a la lista
                        listaContactos.add(contacto.getNombre() + " - Edad: " + contacto.getEdad());
                        Log.d( tag: "ListarFragment",  msg: "Contacto encontrado: " + contacto.getNombre());
                    }
                }
                adapter.notifyDataSetChanged();
                Log.d( tag: "ListarFragment",  msg: "Número de contactos: " + listaContactos.size());
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(getContext(),  text: "Error al cargar los contactos: " + error.getMessage(), Toast.LENGTH_SHORT).show();
                Log.e( tag: "ListarFragment",  msg: "Error al cargar los contactos", error.toException());
            }
        });
```

SANTO TOMÁS
INSTITUTO PROFESIONAL

Buscar y Actualizar Contacto

```
private void buscarContacto() {  1 usage
    String nombre = etBuscarNombre.getText().toString();

    if (TextUtils.isEmpty(nombre)) {
        Toast.makeText(getContext(),  text: "Por favor, ingrese un nombre", Toast.LENGTH_SHORT).show();
        return;
    }

    databaseReference.orderByChild( path: "nombre").equalTo(nombre)
            .addListenerForSingleValueEvent(new ValueEventListener() {
                @Override  2 usages
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    if (snapshot.exists()) {
                        for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
                            Contacto contacto = dataSnapshot.getValue(Contacto.class);
                            if (contacto != null) {
                                contactoId = contacto.getId();
                                etNuevoNombre.setText(contacto.getNombre());
                                etNuevaEdad.setText(String.valueOf(contacto.getEdad()));
                                tvResultado.setText("Contacto encontrado: " + contacto.getNombre());
                            }
                        }
                    } else {
                        tvResultado.setText("No se encontró el contacto.");
                    }
                }

                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                    Toast.makeText(getContext(),  text: "Error al buscar el contacto", Toast.LENGTH_SHORT).show();
                }
            });
}
```

```
private void actualizarContacto() {  1 usage
    String nuevoNombre = etNuevoNombre.getText().toString();
    String nuevaEdadStr = etNuevaEdad.getText().toString();

    if (TextUtils.isEmpty(nuevoNombre) || TextUtils.isEmpty(nuevaEdadStr)) {
        Toast.makeText(getContext(),  text: "Por favor, complete todos los campos", Toast.LENGTH_SHORT).show();
        return;
    }

    int nuevaEdad = Integer.parseInt(nuevaEdadStr);

    if (contactoId != null) {
        Contacto contactoActualizado = new Contacto(contactoId, nuevoNombre, nuevaEdad);
        databaseReference.child(contactoId).setValue(contactoActualizado)
                .addOnSuccessListener(aVoid ->
                        Toast.makeText(getContext(),  text: "Contacto actualizado", Toast.LENGTH_SHORT).show())
                .addOnFailureListener(e ->
                        Toast.makeText(getContext(),  text: "Error al actualizar el contacto", Toast.LENGTH_SHORT).show());
    } else {
        Toast.makeText(getContext(),  text: "Primero busque un contacto para actualizar", Toast.LENGTH_SHORT).show();
    }
}
```

Eliminar Contacto

```
private void eliminarContacto() {  1 usage
    if (contactoId != null) {
        databaseReference.child(contactoId).removeValue()
                .addOnSuccessListener(aVoid ->
                        Toast.makeText(getContext(),  text: "Contacto eliminado", Toast.LENGTH_SHORT).show())
                .addOnFailureListener(e ->
                        Toast.makeText(getContext(),  text: "Error al eliminar el contacto", Toast.LENGTH_SHORT).show());

        // Limpiar los campos después de eliminar
        etNuevoNombre.setText("");
        etNuevaEdad.setText("");
        etBuscarNombre.setText("");
        tvResultado.setText("Contacto eliminado.");
        contactoId = null;
    } else {
        Toast.makeText(getContext(),  text: "Primero busque un contacto para eliminar", Toast.LENGTH_SHORT).show();
    }
}
```

SANTO TOMÁS
INSTITUTO PROFESIONAL

SANTO TOMÁS
INSTITUTO PROFESIONAL

Fin