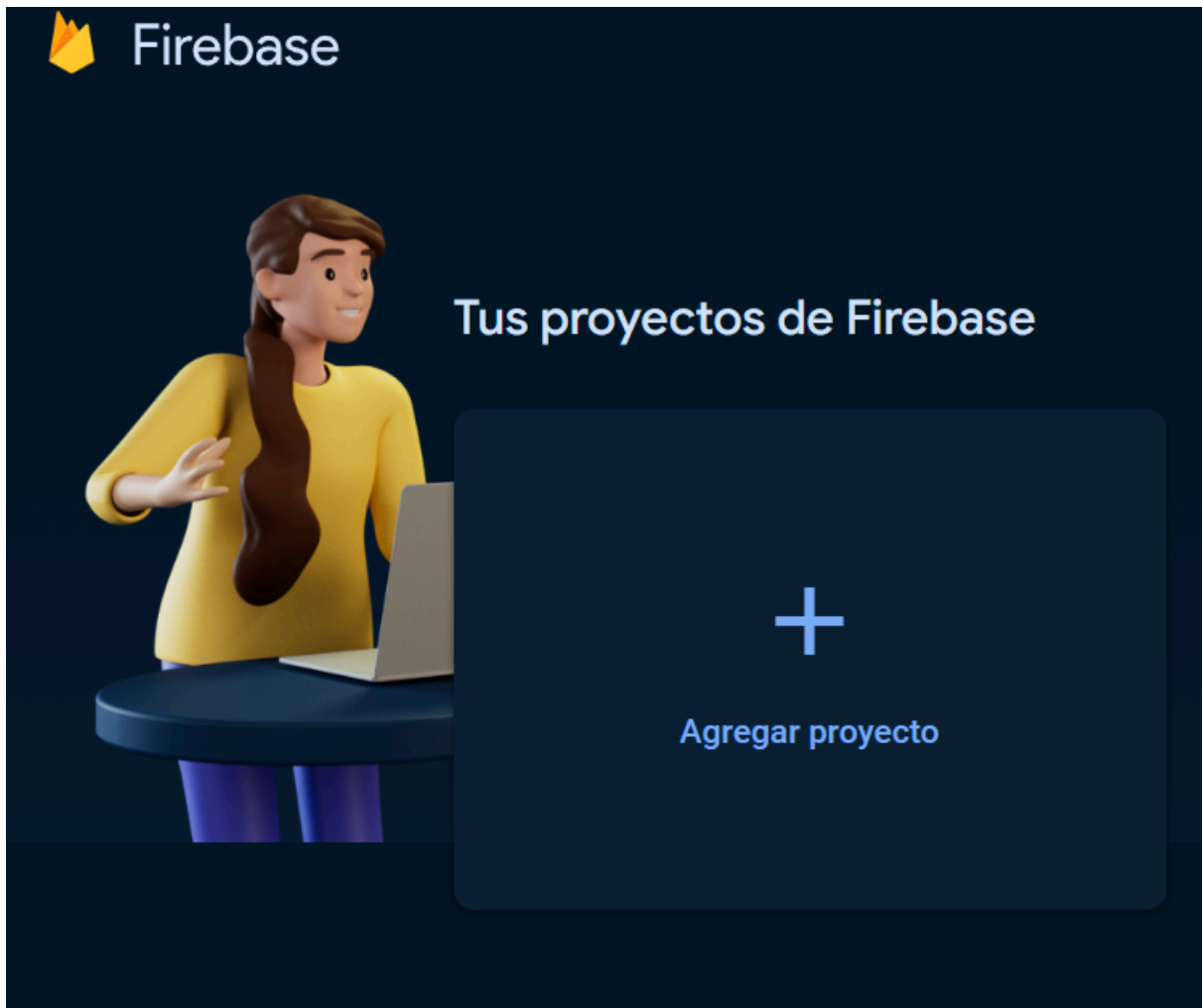
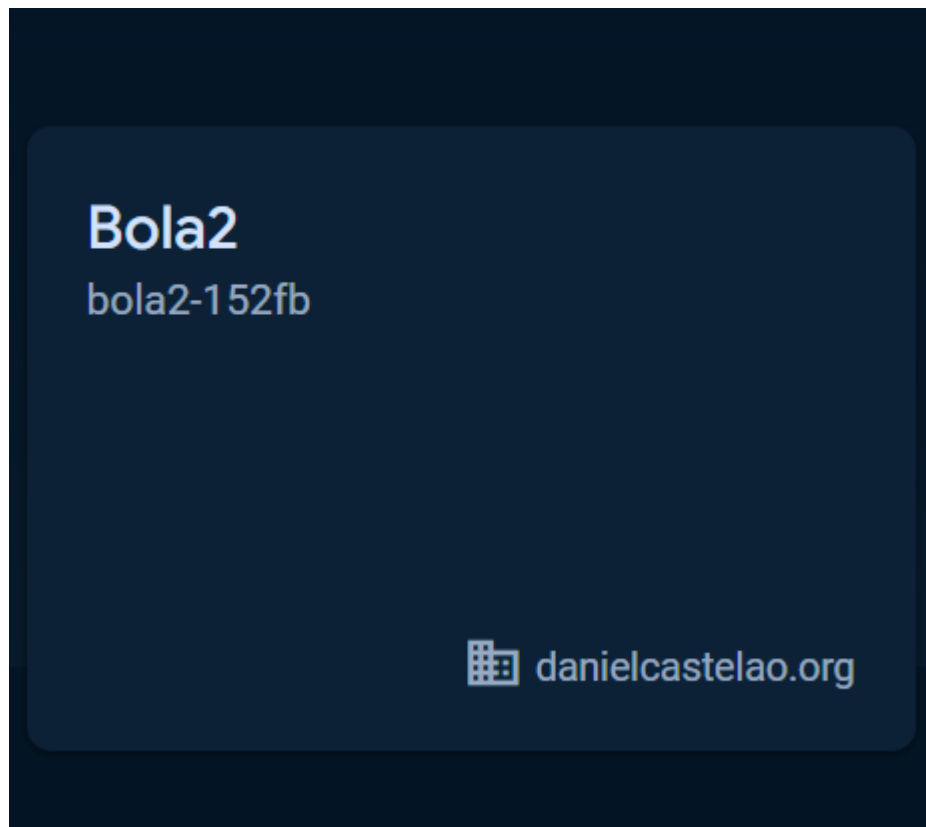


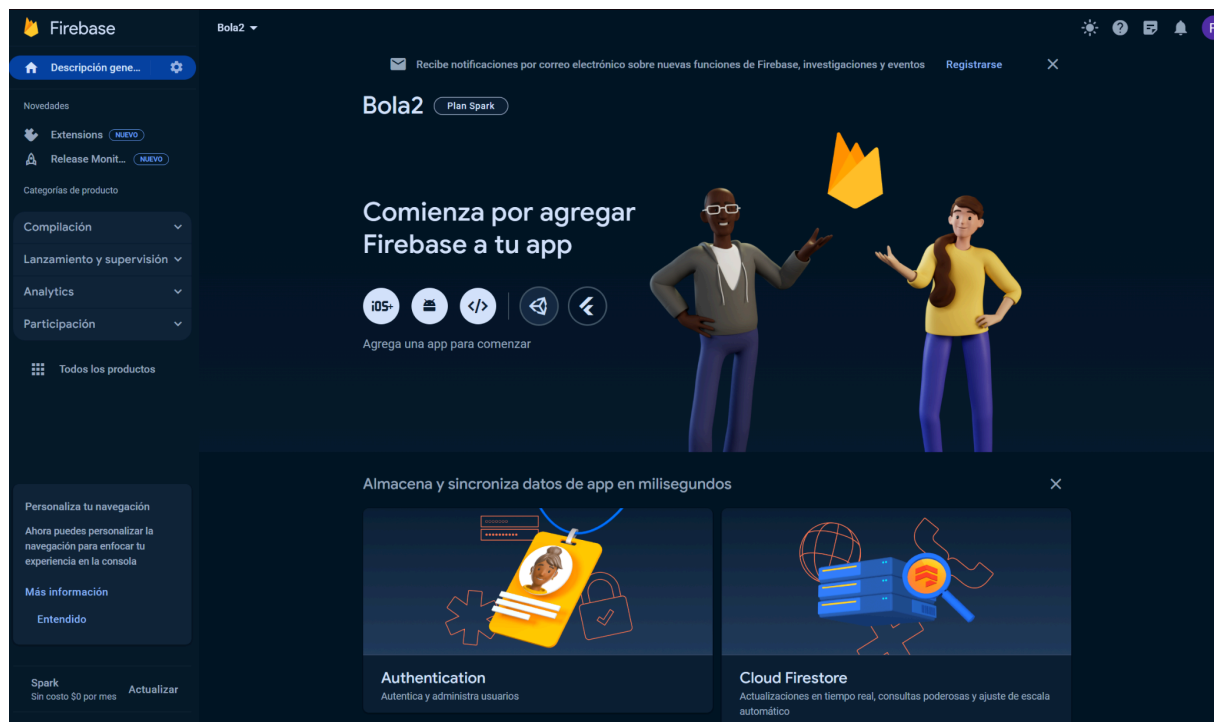
# TAREA FIREBASE



Añadimos un proyecto nuevo en la pagina de firebase



Le llamamos igual que nuestro proyecto de Unity



Una vez hecho entramos dentro de nuestro proyecto

# Comienza por agregar Firebase a tu app



Agrega una app para comenzar

Ahora elegimos Unity en la seccion de agregar firebase a atu app

## × Agrega Firebase a tu app de Unity

### 1 Registrar app

¿Lanzarás tu juego en Apple y Android? Es importante que registres los dos destinos de compilación de la app de Unity en el mismo proyecto de Firebase. Tras registrarlos, la app de Unity aparecerá en tu proyecto de Firebase como una app para Android, Apple o ambos servicios.

☐ Registrar como app para Apple

ID del paquete de Apple ?

com.company.appname

Sobrenombre de la app (opcional) ?

Mi app para Apple

☒ Registrar como app para Android

Nombre del paquete de Android ?

com.company.Bola2

Sobrenombre de la app (opcional) ?

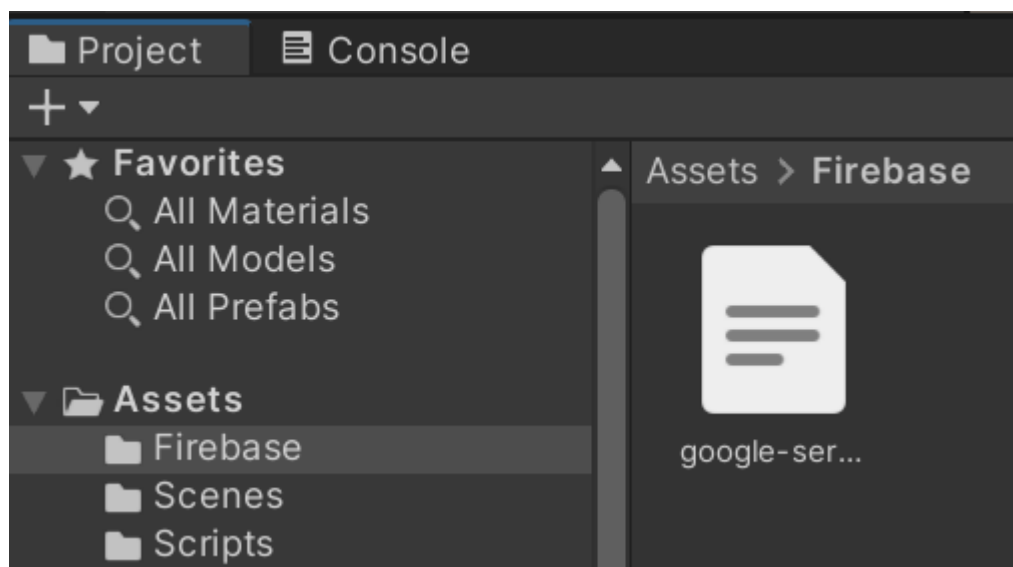
Bola2

Registrar app

Aquí elegimos si la aplicación será para Android o Apple o ambos servicios



Ahora nos descargamos el archivo de configuración de Firebase



Dentro de nuestra carpeta Assets metemos nuestro archivo de configuración

## × Agrega Firebase a tu app de Unity

✓ Registrar app

✎ Descargar archivo de configuración

3 Agregar el SDK de Firebase

↓ Descargar SDK de Firebase Unity (ZIP) [🔗](#)

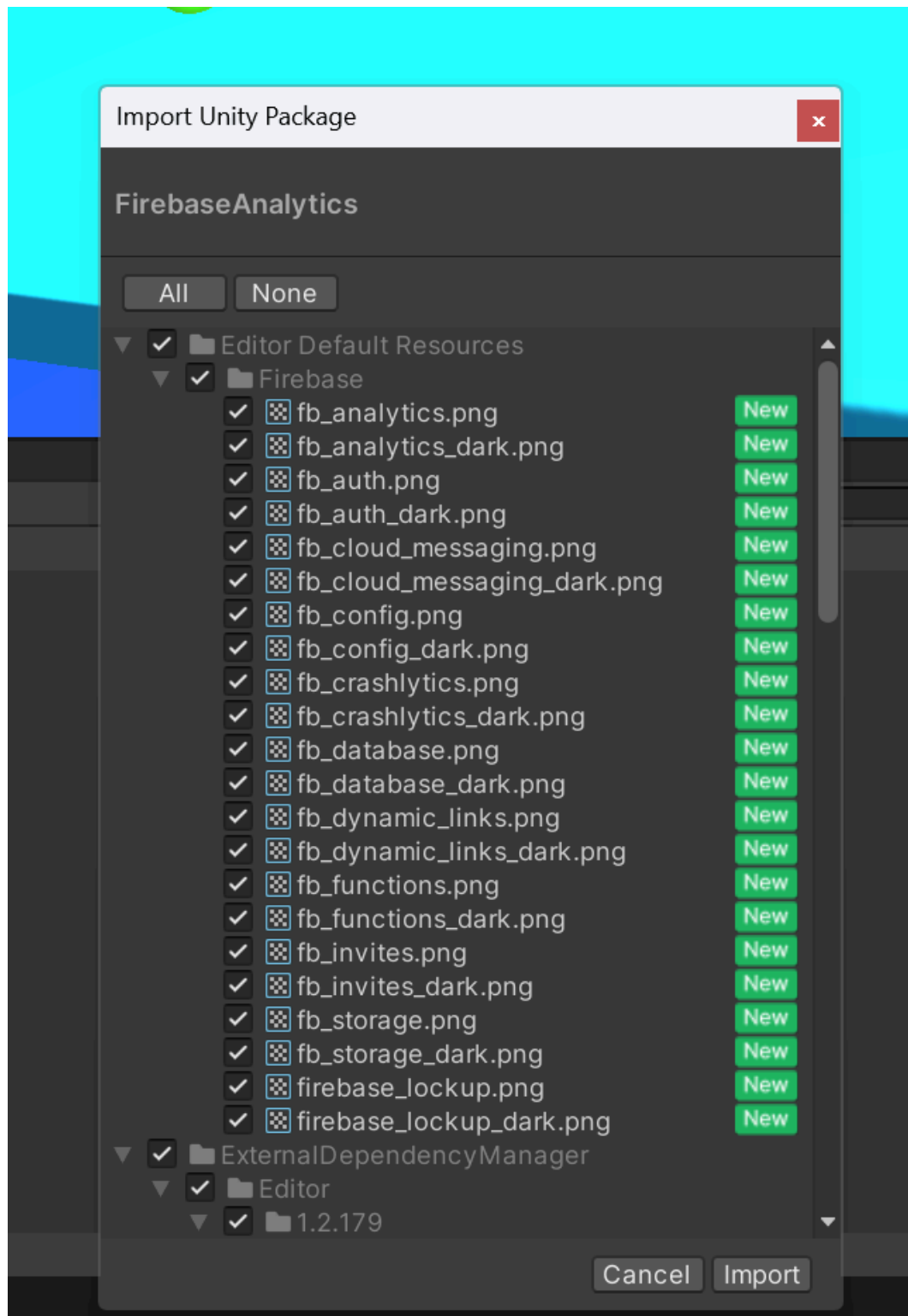
1. Descomprime el SDK que descargaste en la ubicación que prefieras.
2. Abre tu proyecto de Unity, ve a Elementos > Importar paquete > Paquete personalizado.
3. Desde el SDK que descomprimiste, selecciona importar el SDK de Analytics (`FirebaseAnalytics.unitypackage`). Además, agrega los SDK de los otros [productos de Firebase](#) [🔗](#) que desees.
4. En la ventana *Importar paquete de Unity*, haz clic en **Importar**.

Anterior

Siguiente

✎ Próximos pasos

Ahora descargamos los paquetes



Y los importamos a nuestro proyecto



Ahora creamos un Objeto estático llamado Firebase para conectarnos a nuestra base de datos

```
Assets > Scripts > FirebaseInit.cs > ...
1  using Firebase;
2  using Firebase.Database;
3  using UnityEngine;
4
0 references
5  public class FirebaseInit : MonoBehaviour
6  {
0 references
7      void Start()
8      {
9          FirebaseApp.CheckAndFixDependenciesAsync().ContinueWith(task => {
10              var dependencyStatus = task.Result;
11              if (dependencyStatus == DependencyStatus.Available)
12              {
13                  Debug.Log("Firebase is ready");
14              }
15              else
16              {
17                  Debug.LogError($"Could not resolve all Firebase dependencies: {dependencyStatus}");
18              }
19          });
20      }
21  }
22
23
24
```

En este script nos conectamos a la base de datos



Ahora vamos a crear una nueva base de datos




Bola2 ▾

# Realtime Database

Datos Reglas Copias de seguridad Uso **Extensions**

 Protege tus recursos de Realtime Database contra los abusos, como

 <https://bola2-152fb-default-rtdb.firebaseio.com>

<https://bola2-152fb-default-rtdb.firebaseio.com/>

- ▼ — GameObjects
  - ▼ — Object1
    - x: 1
    - y: 2
    - z: 3

Y creamos nuestra estructura

```
Assets > Scripts > UploadObjectData.cs > ObjectPosition
1  using Firebase.Database;
2  using UnityEngine;
3
0 references
4  public class UploadObjectData : MonoBehaviour
5  {
0 references
6      void Start()
7      {
8          DatabaseReference reference = FirebaseDatabase.DefaultInstance.RootReference;
9
10         // Crear datos de ejemplo
11         ObjectPosition object1Position = new ObjectPosition { x = 1, y = 2, z = 3 };
12         string json = JsonUtility.ToJson(object1Position);
13
14         // Subir datos a Firebase
15         reference.Child("GameObjects").Child("object1").Child("position").SetRawJsonValueAsync(json);
16     }
17 }
18
[System.Serializable]
19
2 references
20 public class ObjectPosition
21 {
1 reference
22     public float x;
1 reference
23     public float y;
1 reference
24     public float z;
25 }
26
```

Con este código mandamos la posición de nuestro prefab del jugador a nuestra base de datos

```
▼ — posicion
  x: -3.773113250732422
  y: 0.5
  z: 3.10884165763855
```