



UNIVERSIDAD DE MÁLAGA

Aplicación móvil para la monitorización de actividades

Mobile app for activity tracking

Realizado por: Ignacio Martínez Gallardo

Tutorizado por : Francisco José Jaime Rodríguez

26/09/2025

Motivación



Objetivos



Tecnologías



Flutter



Firebase



Dart

Herramientas

Desarrollo



Documentación

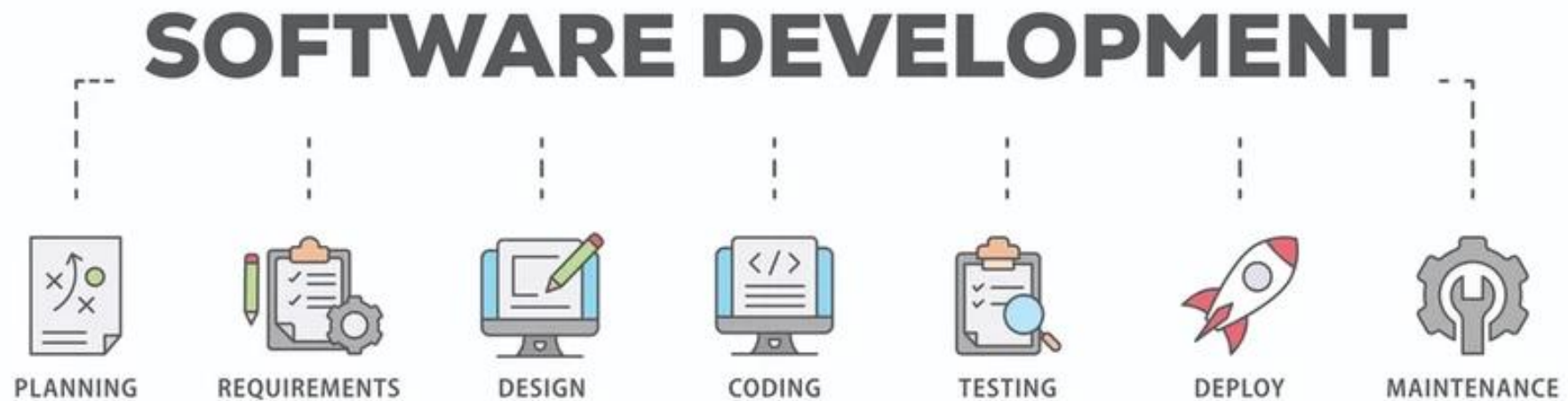
L^AT_EX

Overleaf

Control
de versiones

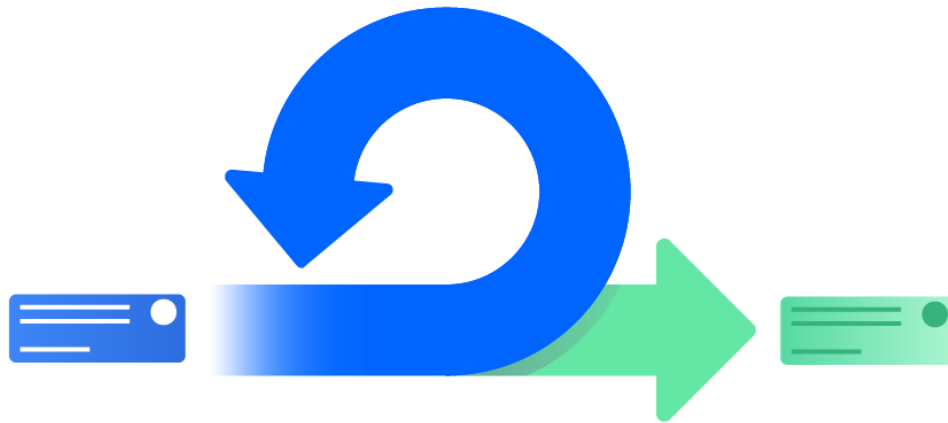


Proyecto

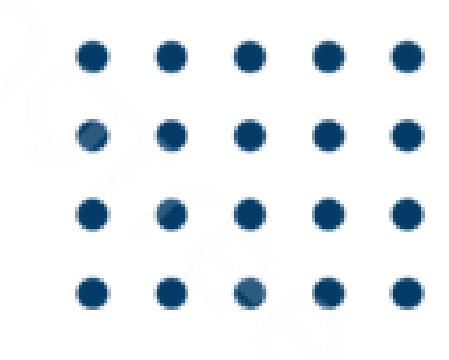


Metodología ágil

- Similar a Scrum
- Sprints de dos semanas
- Reuniones



Planificación



Completado: Sprint 1

Sprint 1
Completado

1

Requisitos Sprint 1

✓ Especificación de requisitos

3 feb - 10 feb

Tecnologías Sprint 1

✓ Estudio de tecnologías: Flutter

3 feb - 10 feb

+ Añade una tarjeta

Completado: Sprint 2

Sprint 2
Completado

1

Requisitos Sprint 2

✓ Descripción de los requisitos

11 feb - 24 feb

Tecnologías Sprint 2

✓ Estudio de tecnologías: Firebase

11 feb - 24 feb

+ Añade una tarjeta

Completado: Sprint 3

Sprint 3
Completado

1

Requisitos Sprint 2

✓ Elaboración de los casos de uso

25 feb - 16 mar

Requisitos Sprint 2

✓ Elaboración del diagrama de casos de uso

25 feb - 16 mar

Interfaz de Usuario Sprint 3

✓ Desarrollo de los mockups de la UI

25 feb - 16 mar

+ Añade una tarjeta

Completado: Sprint 4

Sprint 4
Completado

1

Modelado y Diseño Sprint 4

✓ Diagrama de clases

17 mar - 30 mar

+ Añade una tarjeta

Completado: Sprint 5

Sprint 5
Completado

1

Modelado y Diseño Sprint 5

✓ Diagramas de secuencia

12 may - 25 may

Sprint 5 Código

✓ RF 1. Registro del usuario

12 may - 25 may

Sprint 5 Código

✓ RF 2. Confirmar registro

12 may - 25 may

Sprint 5 Código

✓ RF 3. Inicio de sesión del usuario

12 may - 25 may

Sprint 5 Código

✓ RF 4. Cierre de sesión del usuario

12 may - 25 may

Sprint 5 Código

✓ RF 5. Cambio de contraseña

+ Añade una tarjeta

Completado: Sprint 6

Sprint 6
Completado

1

Sprint 6 Código

✓ RF 23. Seleccionar día

26 may - 8 jun

Sprint 6 Código

✓ RF 17. Crear actividad

26 may - 8 jun

Sprint 6 Código

✓ RF 18. Ver listado de actividades

26 may - 8 jun

Sprint 6 Código

✓ RF 22. Eliminar actividad

26 may - 8 jun

Sprint 6 Código

✓ RF 21. Marcar completación de la actividad

26 may - 8 jun

+ Añade una tarjeta

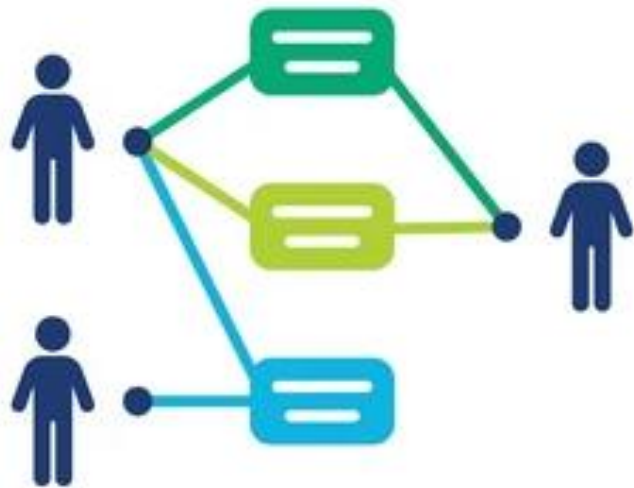
9/31

Análisis y especificación de requisitos

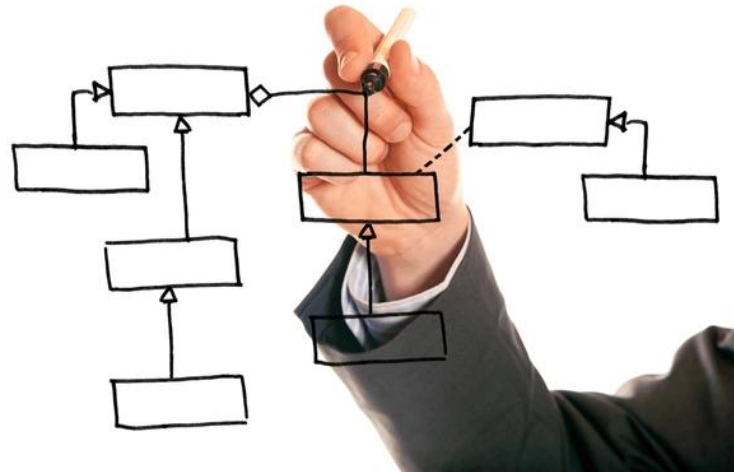


Modelado y diseño del sistema

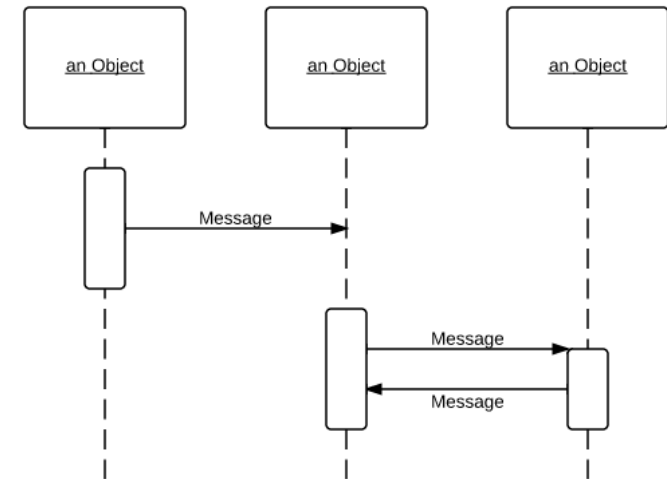
Casos
de uso



Modelado
estructural



Modelado
comportamental



Casos de uso

ID	CU-9
Título	Ver red de amistades
Descripción	El usuario podrá ver un listado con los usuarios que ha añadido a su red de amistades, mostrando para cada amigo su foto de perfil, nombre de usuario y apodo junto a tres botones: enviar mensaje, mandar desafío y eliminar amigo.
Precondición	El usuario ha iniciado sesión. El usuario tiene el menú lateral abierto. El usuario tiene al menos un amigo añadido.
Postcondición	El usuario ve su red de amistades.
Escenario principal	
1. El usuario pulsa el botón “Amigos”. 2. El sistema muestra el listado de amigos del usuario. 3. El usuario ve su listado de amigos.	

Diagrama de casos de uso

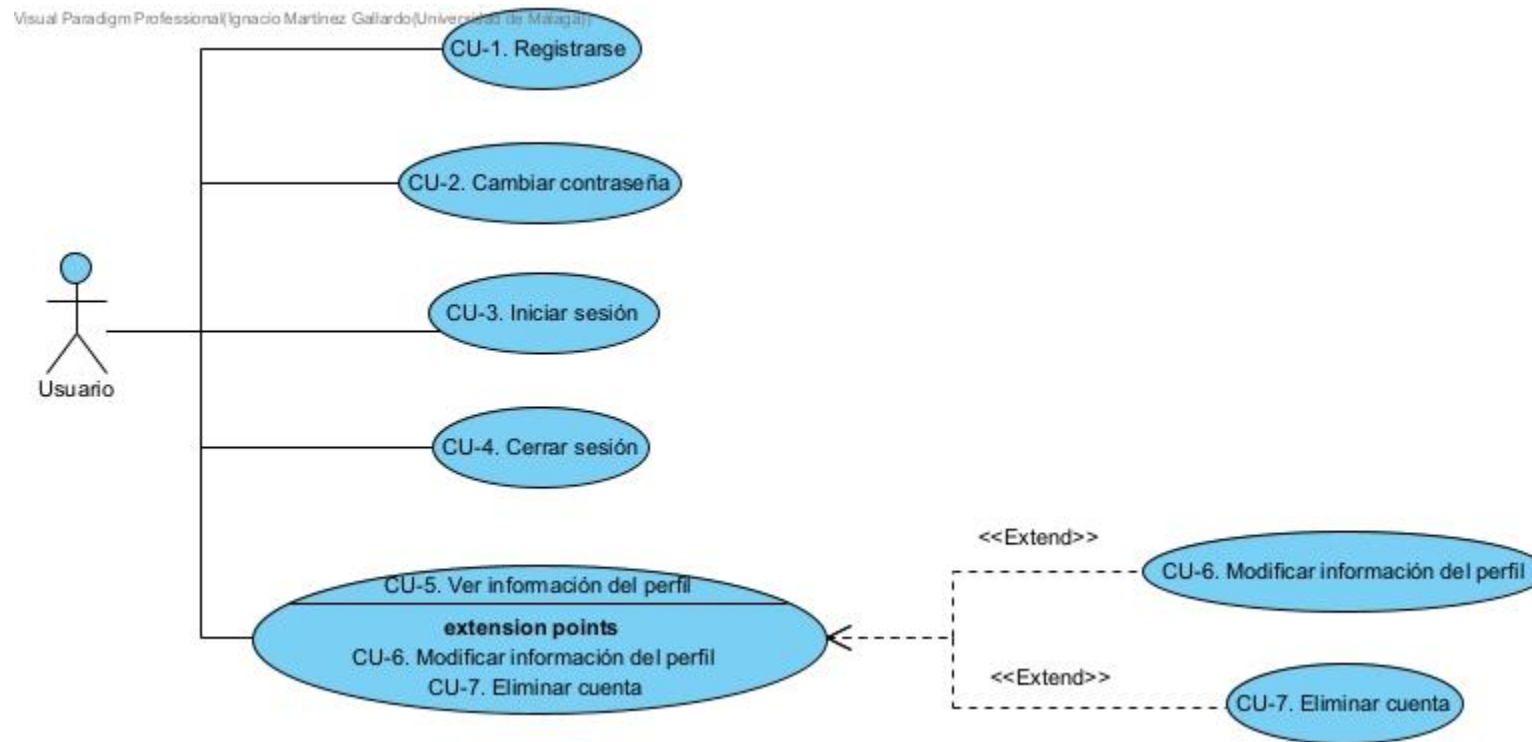
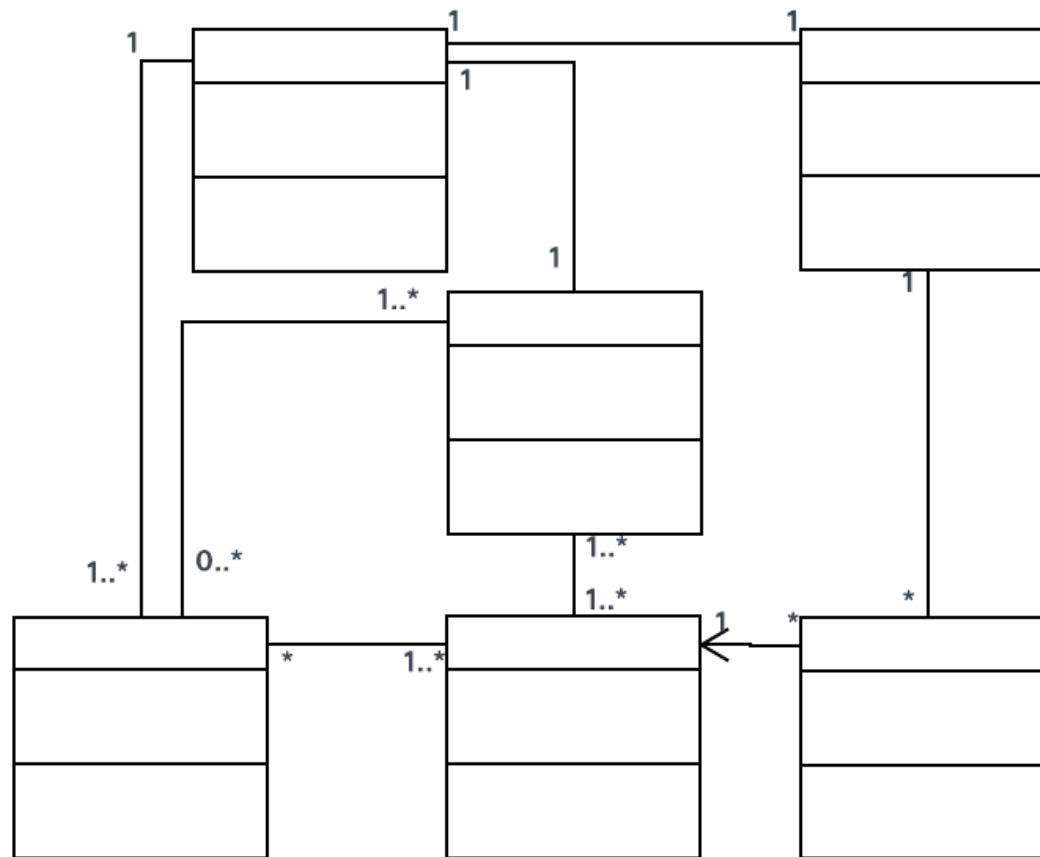
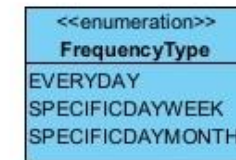
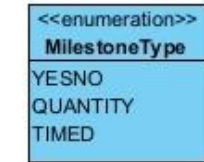
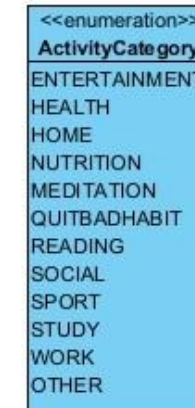
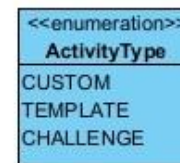
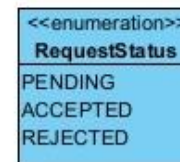
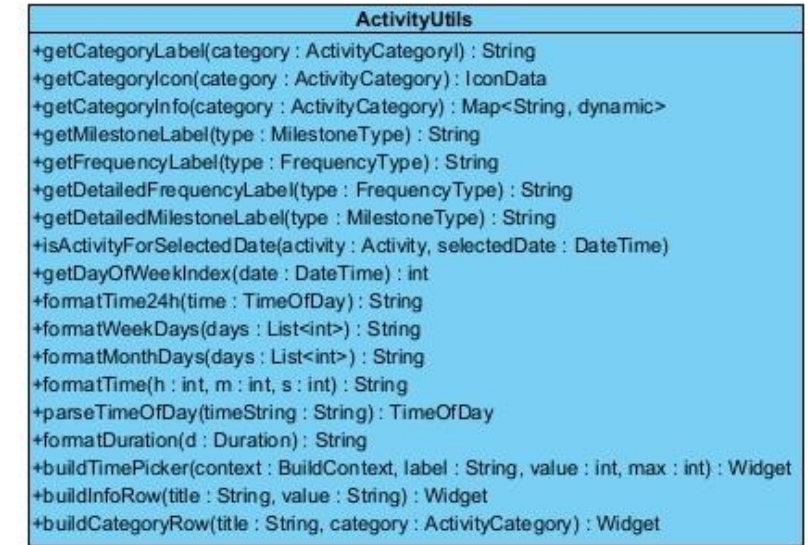
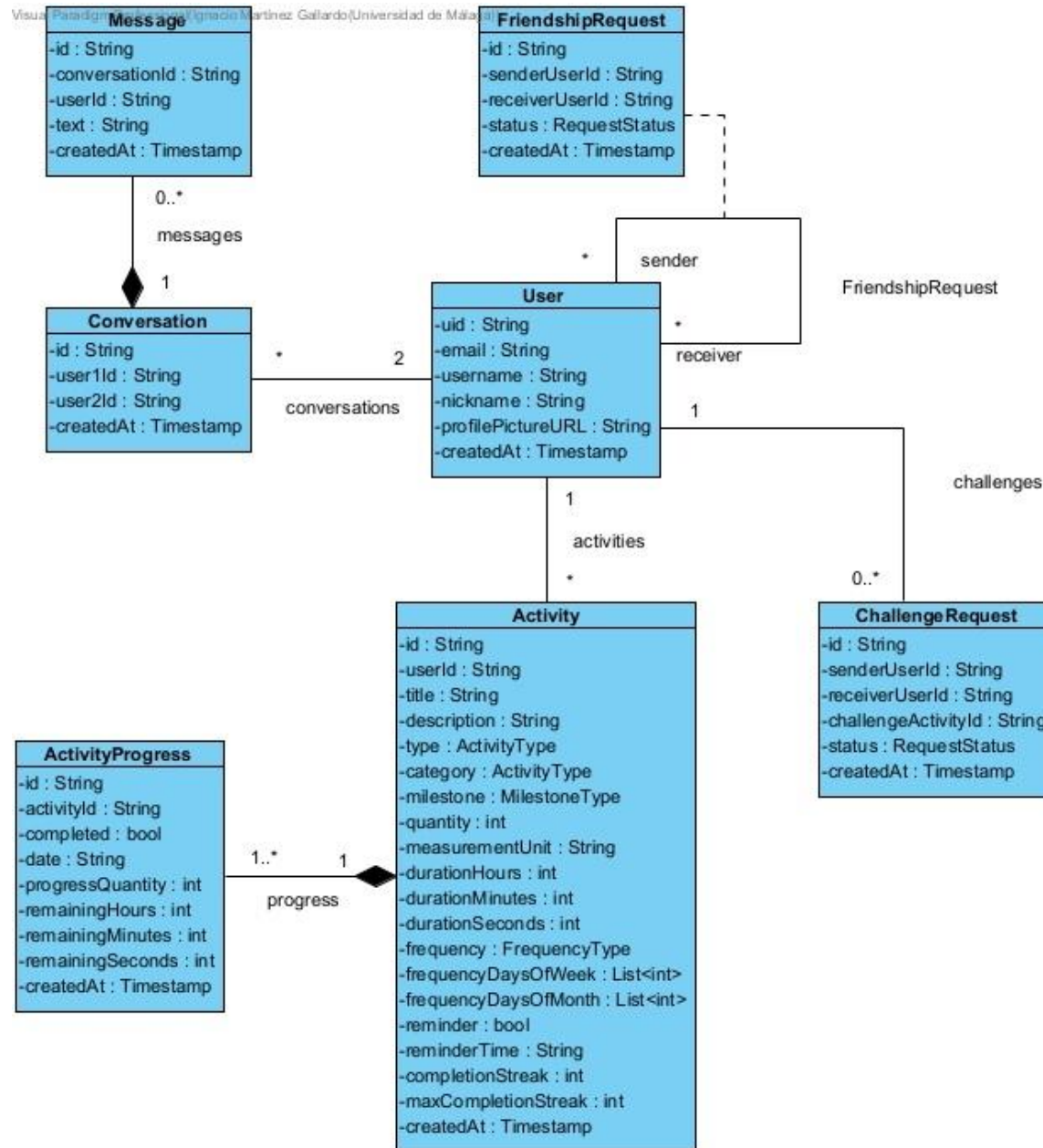


Diagrama de clases





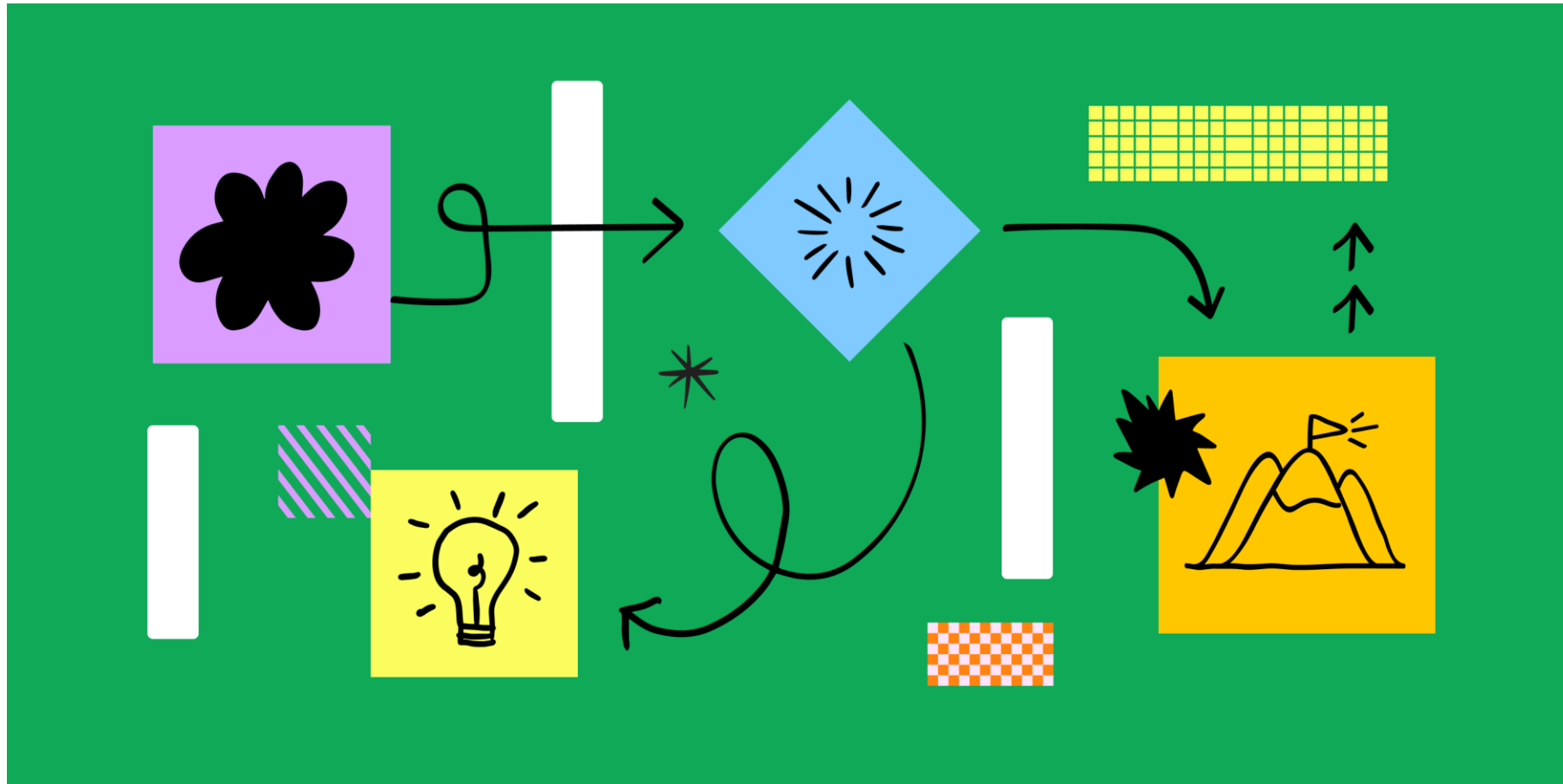
AuthService
-firebaseAuth : FirebaseAuth -userService : UserService
+signUp(email : String, username : String, password : String, passwordConfirmation : String, context : BuildContext) : Future<void> +signIn(email : String, password : String, context : BuildContext) : Future<void> +logout(context : BuildContext) : Future<void> +resetPassword(email : String, context : BuildContext) : Future<void> +deleteAccount(context : BuildContext, password : String) : Future<void>

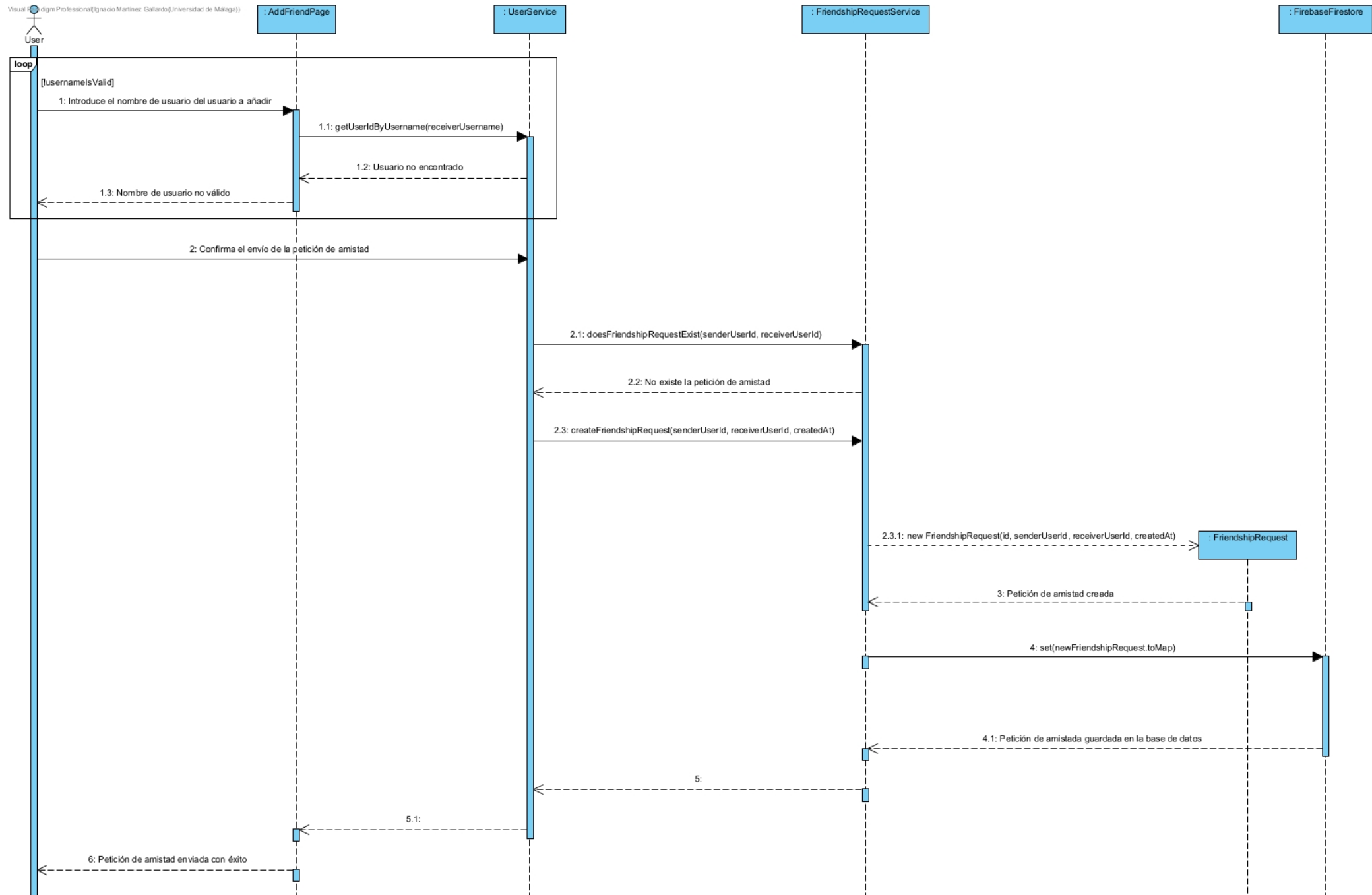
ActivityService
-collection : CollectionReference<Map<String, dynamic>> -progressService : ActivityProgressService
+createActivity(userId : String, title : String, description : String, type : ActivityType, category : ActivityCategory, milestone : MilestoneType, quantity : ... +getUserActivitiesStream(userId : String) : Stream<List<Activity>> +getUserCustomActivitiesStream(userId : String) : Stream<List<Activity>> +getTemplateActivitiesStream() : Stream<List<Activity>> +getActivityById(activityId : String) : Stream<Activity> +getActivityByIdOnce(activityId : String) : Future<Activity> +updateActivity(id : String, title : String, description : String, category : ActivityCategory, reminder : bool, reminderTime : String) : Future<void> +deleteActivityById(id : String) : Future<void> +deleteAllActivitiesForUser(userId : String) : Future<void> +updateStreak(activity : Activity) : Future<void> +checkAndResetBrokenStream(activity : Activity) : Future<void> +saveActivitiesSummaryForWidget(activities : List<Activity>) : Future<void>

FriendshipRequestService
-collection : CollectionReference<Map<String, dynamic>>
+doesFriendshipRequestExist(userId : String, user2Id : String) : Future<bool> +createFriendshipRequest(senderUserId : String, receiverUserId : String, createdAt : Timestamp) +getUserSentFriendshipRequests(userId : String) : Stream<List<FriendshipRequest>> +getUserReceivedFriendshipRequests(userId : String) : Stream<List<FriendshipRequest>> +getFriendshipRequestById(id : String) : Stream<FriendshipRequest> +getUserFriends(userId : String) : Stream<List<FriendshipRequest>> +updateFriendshipRequest(id : String, status : RequestStatus) : Future<void> +deleteFriendshipRequestById(id : String) : Future<void> +deleteFriend(userId1Id : String, user2Id : String) : Future<void> +deleteAllFriendshipsForUser(userId : String) : Future<void>

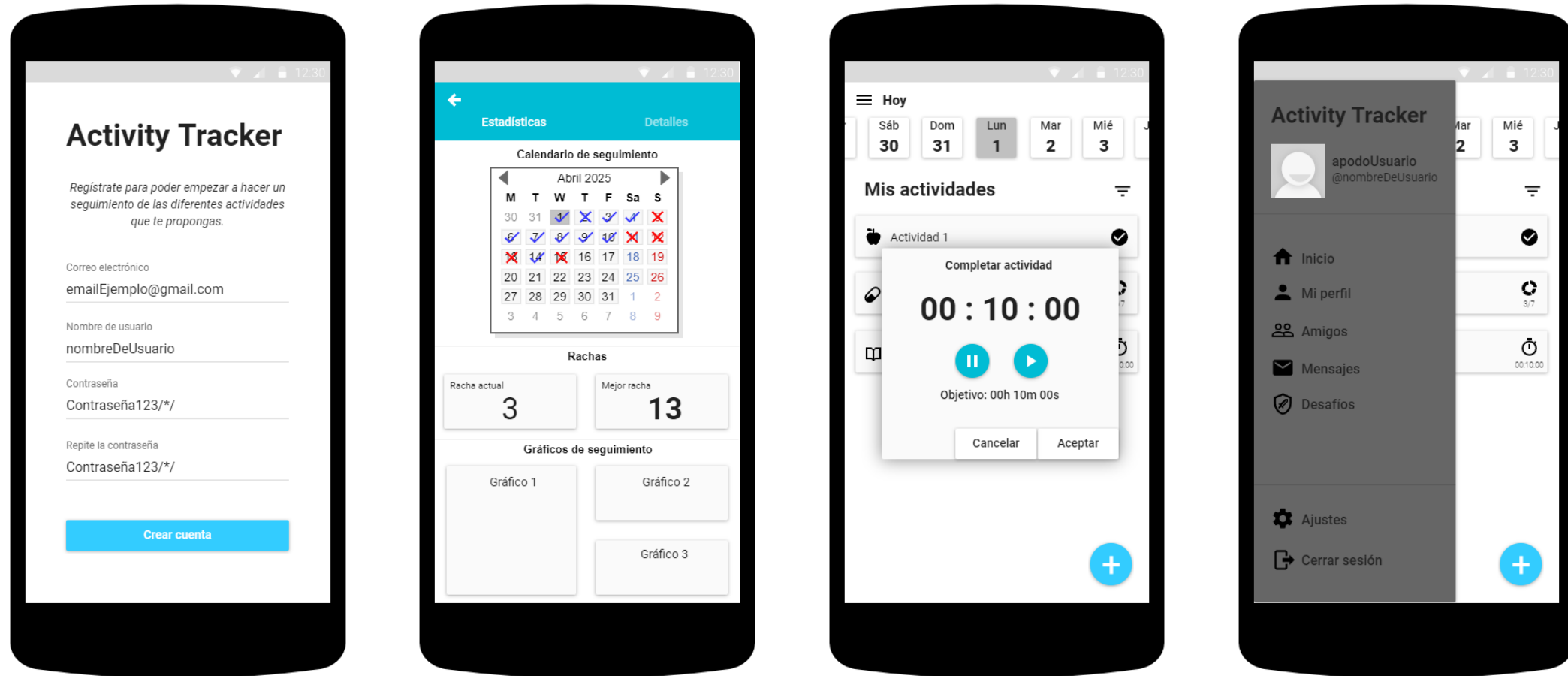
UserService
-collection : CollectionReference<Map<String, dynamic>> -firebaseAuth : FirebaseAuth
+createUser(userCredential : userCredential, username : String) : Future<void> +streamCurrentUserData() : Stream<AppUser> +getUserById(uid : String) : Stream<AppUser> +getUserIdByUsername(username : String) : Future<String> +deleteCurrentUser() : Future<void> +updateCurrentUser(nickname : String, newImageUrl : String) : Future<void>

Diagramas de secuencia





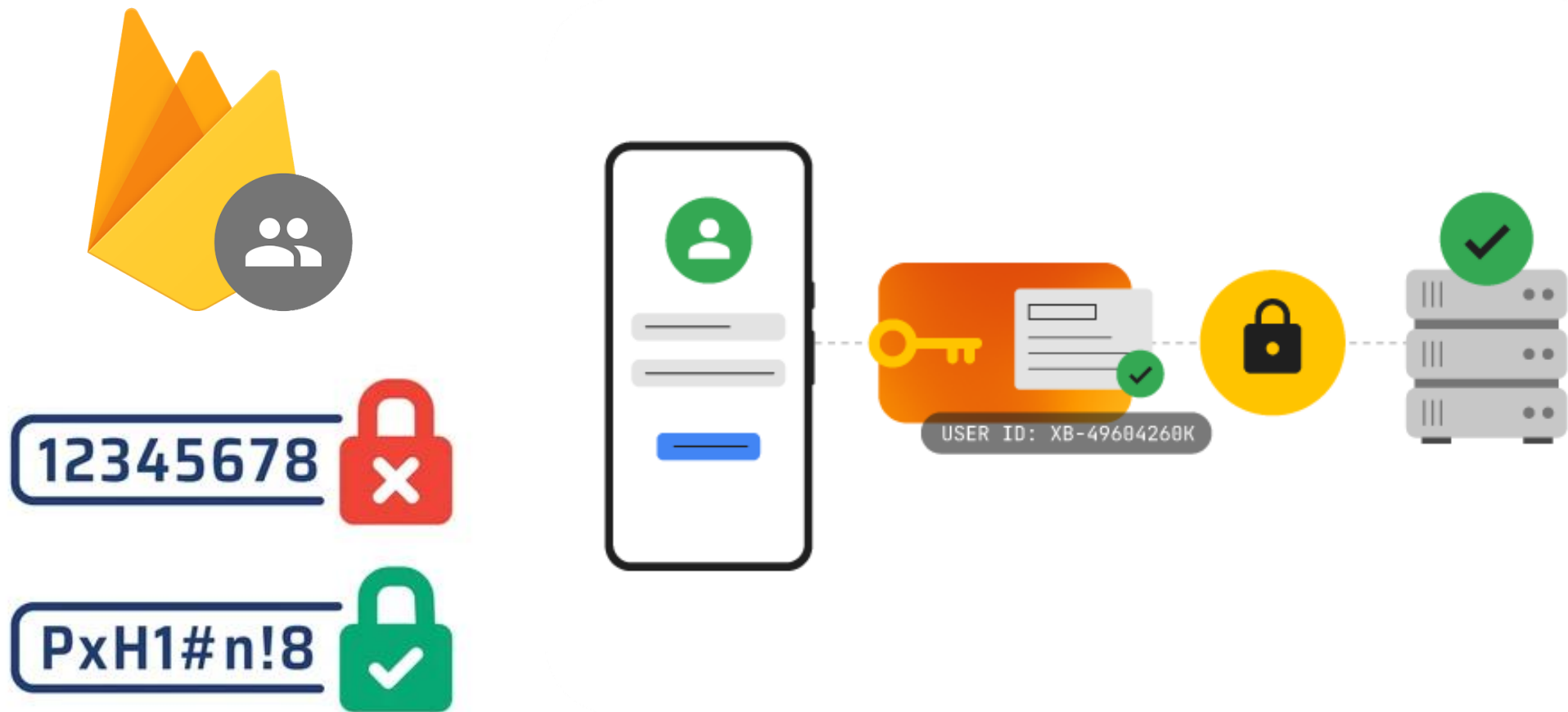
Mockups de la Interfaz de usuario



Desarrollo e implementación

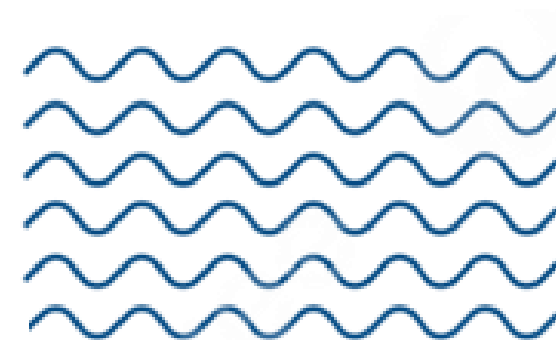
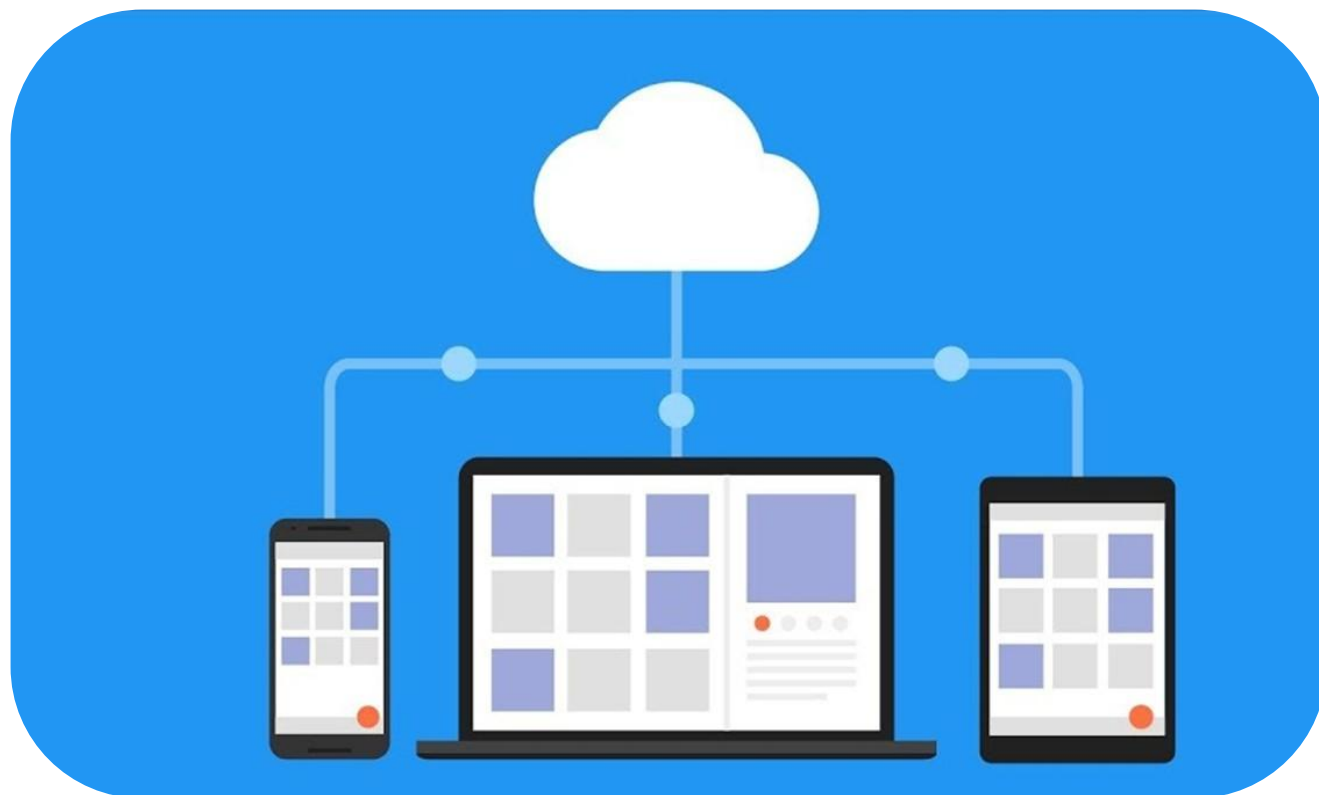


Firebase Authentication





Firebase Cloud Firestore



Notificaciones

17:27 

   96

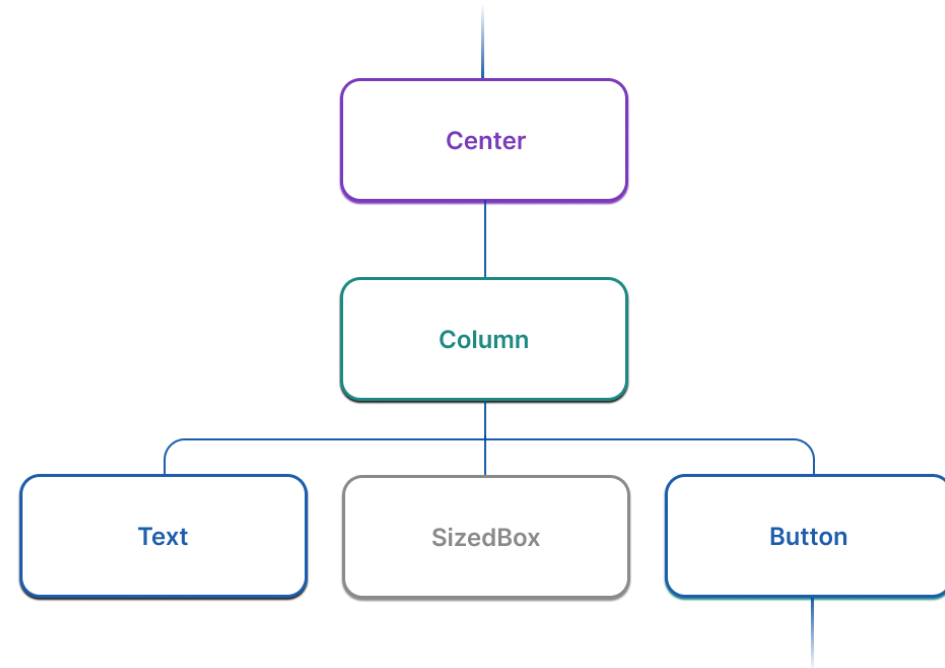
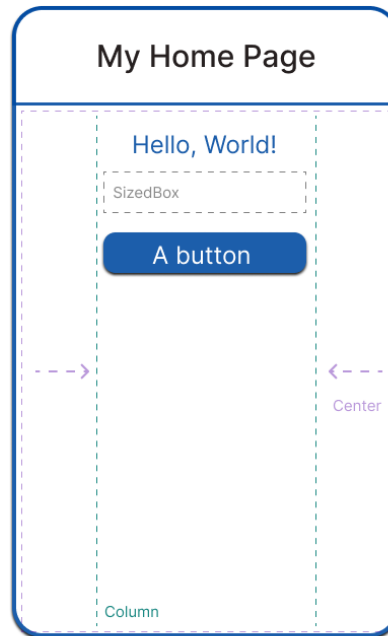


Activity Tracker 10:00

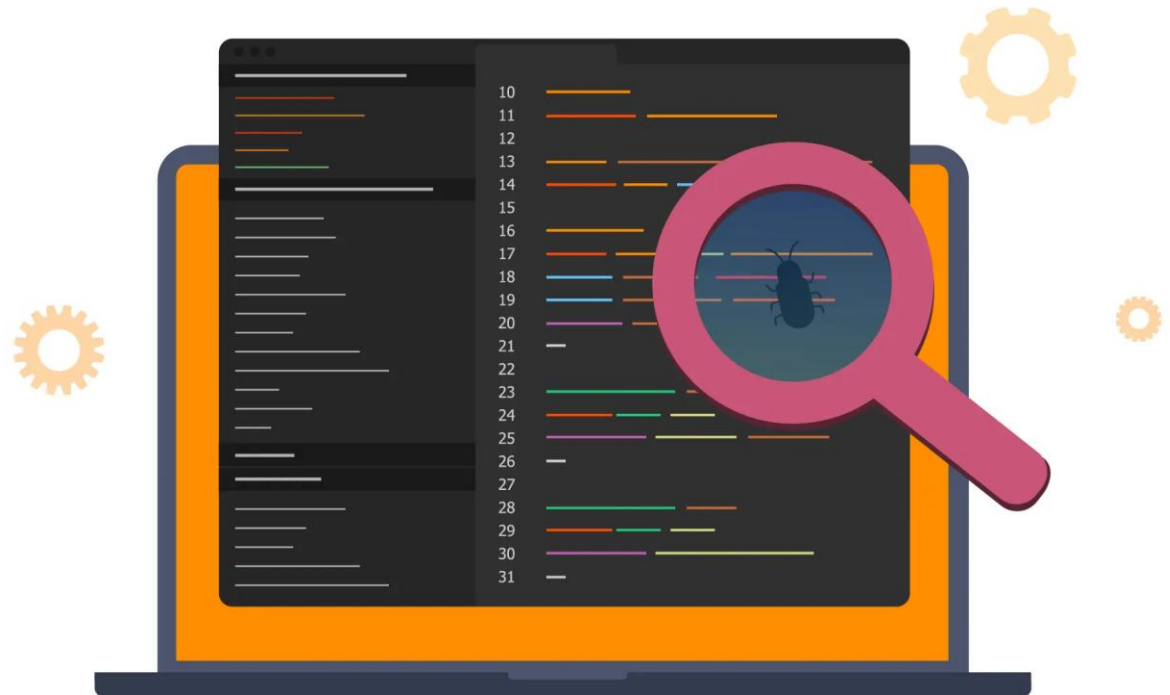
Beber agua

¡Recuerda completar esta actividad antes de que acabe el día!

Interfaz y experiencia de usuario



Validación y verificación



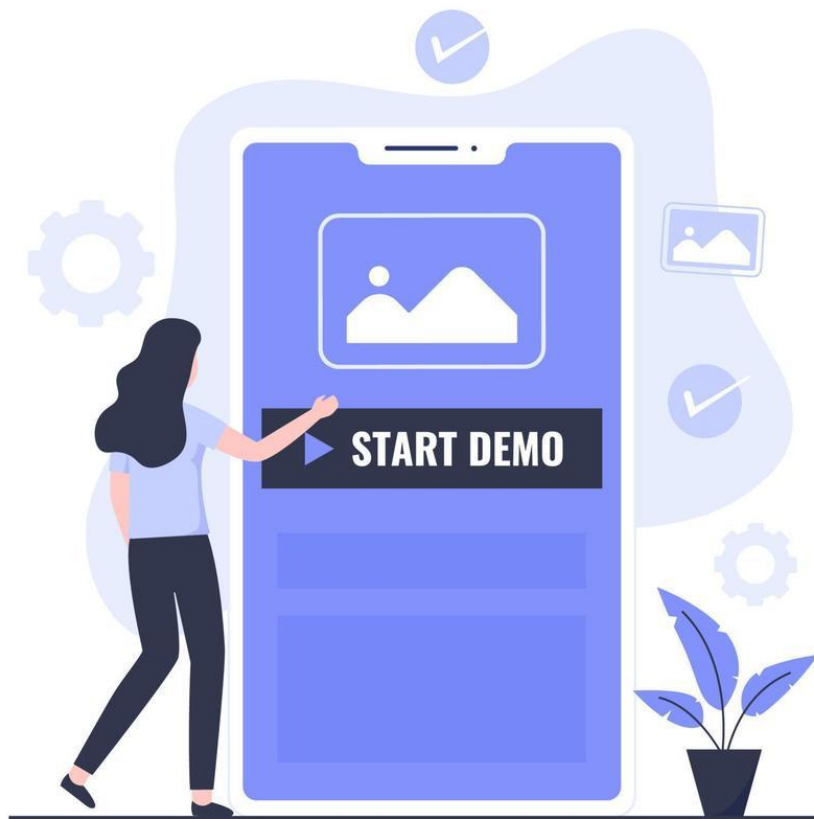
Pruebas unitarias

- ✓ Prueba realizada en 27/8/2025, 20:28:14
 - ✓ (setUpAll) test\activity_service_test.dart
 - ✓ Create a new custom activity ActivityService < test\activity_service_test.dart
 - ✓ Get all the activities of the user ActivityService < test\activity_service_test.dart
 - ✓ Get all the activities created by the user ActivityService < test\activity_service_test.dart
 - ✓ Get all the template activities ActivityService < test\activity_service_test.dart
 - ✓ Get an activity by its ID ActivityService < test\activity_service_test.dart
 - ✓ Get an activity by its ID for an activity that does not exist ActivityService < test\activity_service_test.dart
 - ✓ Update an activity ActivityService < test\activity_service_test.dart
 - ✓ Delete an activity by its ID ActivityService < test\activity_service_test.dart
 - ✓ Deletes all the activities of the user ActivityService < test\activity_service_test.dart

Pruebas de usuarios



Demostración





Conclusiones

- Expectativas
- Objetivos
- Conocimientos aplicados
- Retos superados

✓

✓

✓

✓



Líneas futuras

- Versión iOS
- Mejorar desafíos de actividad
- Archivo de actividades
- Notificaciones para solicitudes y mensajes
- Sistema para proponer plantillas de actividades





Gracias por
vuestra atención

¿Preguntas?