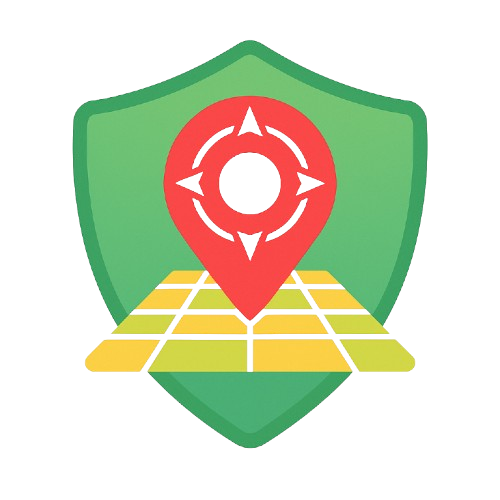
**"Año de la recuperación y consolidación de la economía peruana"**

**FACULTAD DE INGENIERÍA**

**TEMA:**

**TP2.E8 Entrega de sprint 01**

**Curso:**

**Taller de proyectos 2**

**NRC: 17185**

**Docente:**

**Americo Estrada Sanchez**

**Alumno:**

* Joan Nicole Cordova Bustamante
* Mijhael Amilkar Mejia Ballona
* Alexander Villa Andia
* Deyvid Brayan Vargaya Coaquira

**CUSCO -  PERÚ**

**2025**

Sprint 01 - WaykiSafe

**Fecha:** 28/09/2025

**Versión:** 1.0

# 1. Introducción

Este documento detalla el desarrollo del Sprint 01 del proyecto WaykiSafe, una aplicación móvil de seguridad ciudadana orientada a facilitar el reporte de incidentes mediante Inteligencia Artificial y la participación activa de los usuarios a través de reportes comunitarios.

En este sprint se trabajó en las funcionalidades fundamentales de acceso, que permiten establecer la base del sistema: registro de usuario, inicio de sesión

y autenticación mediante Google. Asimismo, se incluyen los avances técnicos alcanzados con evidencias de código y el seguimiento de las tareas a través del burndown chart.

# 3. Lineamientos de Diseño

Durante el Sprint 01, se desarrolló una primera versión funcional de la aplicación móvil WaykiSafe, enfocada en la gestión básica de usuarios. Esta versión tiene como objetivo permitir que los ciudadanos comiencen a interactuar con el sistema de manera segura y confiable.

Las funcionalidades planificadas para este entregable son:

* **Registro de usuario**
* **Inicio de sesión con credenciales propias**
* **Inicio de sesión con Google (Google Login)**
* **Diseño de interfaces iniciales**

Estas funcionalidades constituyen la base sobre la cual se implementarán en los siguientes sprints los módulos avanzados de WaykiSafe, tales como:

* Reporte de incidentes en tiempo real
* Integración con Inteligencia Artificial para clasificación automática de reportes
* Visualización de mapas de seguridad ciudadana
* Notificaciones y alertas preventivas

# 2. Objetivo del Sprint

Implementar funcionalidades básicas de acceso del usuario: registro e inicio de sesión, incluyendo persistencia de datos local y validaciones.

# 3. Historias de Usuario y Tareas

### Historia de Usuario 1.1: Registro de usuario

**Descripción:** Como turista, quiero poder registrarme en la aplicación con mi correo y contraseña, para acceder a mis datos y configuraciones desde cualquier dispositivo.

**Tareas:**

| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| --- | --- | --- | --- | --- |
| 1.1.1 | Diseñar la interfaz del formulario de registro | UX/UI Designer | 4 | Pendiente |
| 1.1.2 | Implementar el formulario (nombre, correo, contraseña, teléfono) | Frontend Dev | 6 | Pendiente |
| 1.1.3 | Configurar Firebase Database para almacenar usuarios | Backend Dev | 6 | Pendiente |
| 1.1.4 | Implementar validaciones (contraseña segura, campos obligatorios) | Backend Dev | 5 | Pendiente |
| 1.1.5 | Guardar datos en modo offline (SQLite/local) | Backend Engineer | 6 | Pendiente |
| 1.1.6 | Pruebas unitarias e integración con login | QA Engineer | 6 | Pendiente |

### Historia de Usuario 1.2: Iniciar sesión

**Descripción:** Como turista, quiero poder iniciar sesión con mi correo y contraseña, para acceder a mis datos en la aplicación.

**Tareas:**

| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| --- | --- | --- | --- | --- |
| 1.2.1 | Diseñar la interfaz de inicio de sesión | UX/UI Designer | 4 | Pendiente |
| 1.2.2 | Implementar formulario de login | Frontend Dev | 6 | Pendiente |
| 1.2.3 | Configurar Firebase Authentication | Backend Dev | 6 | Pendiente |
| 1.2.4 | Implementar validaciones (correo válido, contraseña correcta) | Frontend Dev | 5 | Pendiente |
| 1.2.5 | Implementar inicio de sesión sin internet | Backend Engineer | 6 | Pendiente |
| 1.2.6 | Pruebas de integración | QA Engineer | 5 | Pendiente |

### Historia de Usuario 1.3: Registro e inicio con Google

**Descripción:** Como turista, quiero poder registrarme e iniciar sesión usando mi cuenta de Google, para facilitar el acceso rápido a la aplicación.

**Tareas:**

| **ID** | **Tarea** | **Responsable** | **Estimación (Horas)** | **Estado** |
| --- | --- | --- | --- | --- |
| 1.3.1 | Configurar Firebase Authentication con Google | Backend Dev | 8 | Pendiente |
| 1.3.2 | Implementar botón “Continuar con Google” | Frontend Dev | 6 | Pendiente |
| 1.3.3 | Manejar respuestas de autenticación (token, correo, nombre) | Backend Dev | 5 | Pendiente |
| 1.3.4 | Vincular cuenta Google con base de datos de usuarios | Backend Engineer | 6 | Pendiente |
| 1.3.5 | Pruebas de login/registro con Google | QA Engineer | 5 | Pendiente |

**4. Evidencias de Código**

A continuación, se incluirán fragmentos del código implementado para las historias de usuario.

**4.1 Historia de Usuario 01**

| FirebaseAuth mAuth = FirebaseAuth.getInstance();  mAuth.createUserWithEmailAndPassword(email, password)  .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  @Override  public void onComplete(@NonNull Task<AuthResult> task) {  if (task.isSuccessful()) {  FirebaseUser user = mAuth.getCurrentUser();  // Guardar en Firestore  FirebaseFirestore db = FirebaseFirestore.getInstance();  Map<String, Object> usuario = new HashMap<>();  usuario.put("nombre", nombre);  usuario.put("correo", email);  usuario.put("telefono", telefono);  db.collection("users").document(user.getUid()).set(usuario);  } else {  // Manejo de error  }  }  });  **VALIDACIÓN**  if (TextUtils.isEmpty(email)) {  emailEditText.setError("Correo requerido");  }  if (!Patterns.EMAIL\_ADDRESS.matcher(email).matches()) {  emailEditText.setError("Correo inválido");  }  if (password.length() < 6) {  passwordEditText.setError("La contraseña debe tener al menos 6 caracteres");  } |
| --- |

**4.2 Historia de Usuario 02**

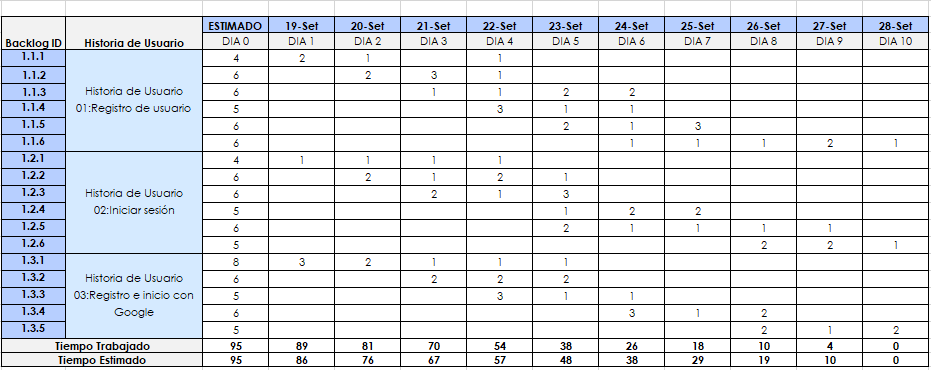
| FirebaseAuth mAuth = FirebaseAuth.getInstance();  mAuth.signInWithEmailAndPassword(email, password)  .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  @Override  public void onComplete(@NonNull Task<AuthResult> task) {  if (task.isSuccessful()) {  FirebaseUser user = mAuth.getCurrentUser();  // Navegar a MainActivity  } else {  // Mostrar error  }  }  }); |
| --- |

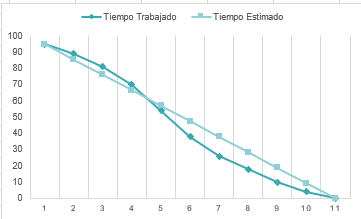
**4.3 Historia de Usuario 02**

| **CONFIGURACIÓN INICIO DE SESION CON GOOGLE**  GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN)  .requestIdToken(getString(R.string.default\_web\_client\_id))  .requestEmail()  .build();  GoogleSignInClient mGoogleSignInClient = GoogleSignIn.getClient(this, gso);  **AUTENTICACIÓN DE FIREBASE CON GOOGLE**  Task<GoogleSignInAccount> task = GoogleSignIn.getSignedInAccountFromIntent(data);  GoogleSignInAccount account = task.getResult(ApiException.class);  AuthCredential credential = GoogleAuthProvider.getCredential(account.getIdToken(), null);  mAuth.signInWithCredential(credential)  .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  @Override  public void onComplete(@NonNull Task<AuthResult> task) {  if (task.isSuccessful()) {  FirebaseUser user = mAuth.getCurrentUser();  // Guardar/actualizar en Firestore  }  }  }); |
| --- |

**5. Burndown Chart**

El siguiente gráfico representa el progreso del sprint, comparando las horas restantes con los días de ejecución.





# 6. Análisis del Sprint

Durante el Sprint 01, el equipo mostró un buen desempeño en el diseño de las interfaces de inicio de sesión y registro dentro de WaykiSafe, destacando la coordinación entre el diseño y el desarrollo frontend. No obstante, surgieron dificultades en la configuración de Firebase y la integración de los servicios de autenticación, lo que generó algunos retrasos. Además, se identificó la necesidad de mejorar la planificación en las estimaciones de tiempo y en la definición de los criterios de aceptación. Para el siguiente sprint, se priorizará una revisión más detallada de dependencias técnicas y pruebas de integración para evitar bloqueos en la continuidad del desarrollo.

# 7. Conclusiones

El Sprint 01 permitió establecer una base sólida para las funcionalidades de acceso de los usuarios en WaykiSafe, logrando avances en el diseño y parte de la implementación del sistema de registro e inicio de sesión. El equipo trabajó de forma colaborativa y organizada, lo que sentó las bases para abordar con mayor eficacia las siguientes funcionalidades relacionadas con la seguridad y el monitoreo en los próximos sprints.