Texto, Logotipo

El contenido generado por IA puede ser incorrecto.

**SignBridge — Traductor de Lengua de Señas on‑device**

**Alumnos** :

* Francisco Egenau Navia
* Matías Machuca
* Sebastián Medina

**Carrera:** Ingeniería Informática

**Docente**: Arturo Guerra

# Índice

## Abstract

## 1. Descripción breve del proyecto y relevancia

## 2. Desarrollo de ingeniería

## 2.1 Objetivos

## 2.2 Alcance del MVP

## 2.3 Arquitectura técnica

## 2.4 Metodología y gestión

## 2.5 Plan de trabajo e hitos

## 2.6 Riesgos y mitigaciones

## 2.7 Pruebas y métricas de calidad

## 2.8 Consideraciones éticas y de accesibilidad

## 3. Relación del proyecto con las competencias del perfil de egreso

## 4. Relación con mis intereses profesionales

## 5. Factibilidad del proyecto en la asignatura

## 6. Conclusiones

## 7. Reflexiones

# Abstract

SignBridge es una aplicación móvil que reconoce en tiempo real señas de la Lengua de Señas usando la cámara del dispositivo y modelos de IA ejecutados *on‑device* (sin requerir conexión permanente). El MVP contempla alfabeto, números y 50 palabras frecuentes. La solución prioriza baja latencia, privacidad (procesamiento local) y accesibilidad, utilizando *tooling* de visión por computadora (p. ej., MediaPipe) y modelos convertidos a TensorFlow Lite. Se adopta una arquitectura ligera con backend *free tier* solo para telemetría, actualización de modelos y contenidos. El proyecto alinea competencias de ingeniería (análisis, construcción de software, datos, pruebas y gestión ágil) y es factible en el tiempo de la asignatura mediante un enfoque incremental.

## 1. Descripción breve del proyecto y relevancia

**Descripción.** Desarrollar una app móvil capaz de detectar gestos y señas a partir de puntos clave (mano/cuerpo) e inferir su significado mediante un modelo de ML optimizado para *on‑device inference*. El resultado se presenta en texto y/o voz, promoviendo comunicación inclusiva.  
**Problema que resuelve.** Las personas sordas enfrentan barreras de comunicación en contextos cotidianos (salud, educación, servicios). No existe una herramienta gratuita, en español, que funcione bien en celulares de gama media y sin conexión constante.  
**Usuarios objetivo.** Personas sordas y oyentes, familiares, funcionarios de atención, docentes, estudiantes.  
**Relevancia social y tecnológica.** Promueve accesibilidad e inclusión; valida el uso de IA responsable (privacidad por diseño, *edge computing*), y fortalece competencias de ingeniería de software y datos.

## 2. Desarrollo de ingeniería

### 2.1 ObjetivosGeneral:

* Entregar un MVP funcional que traduzca en tiempo real alfabeto, números y ~50 señas frecuentes.
* Específicos:
* Diseñar e implementar la captura y normalización de landmarks (mano/cuerpo).
* Entrenar/ajustar un clasificador ligero (TFLite) con latencia < 60 ms en equipos de gama media.
* Integrar TTS (texto a voz) y UI accesible (contraste, tamaño, subtítulos).
* Definir telemetría y actualización de contenidos vía free tier (opcional/no bloqueante).

### 2.2 Alcance del MVP

* Incluye: Alfabeto, números, ~50 palabras; detección en tiempo real; UI bilingüe (ES/EN); guía de uso; pruebas en 2–3 modelos de teléfono.
* Excluye (versiones futuras): Composición libre de frases, entrenamiento on‑device, sincronización multiusuario, analítica avanzada.

### 2.3 Arquitectura técnica

* Cliente móvil (Android/iOS): App híbrida o nativa.
* Visión por computadora: MediaPipe (mano/cuerpo) para extracción de landmarks.
* Modelo ML: Clasificador entrenado y convertido a TensorFlow Lite, cuantizado (int8/float16) para mejorar rendimiento.
* Inferencia: 100% on‑device; sin envío de video a servidores (privacidad).
* Backend opcional (free tier): Firebase/Supabase para: (a) feature flags y in‑app updates de modelos/contenidos; (b) telemetría anónima y crash reporting; (c) almacenamiento de recursos.
* Seguridad y privacidad: Procesamiento local; anonimización; permisos granulares; política de privacidad.

### 2.4 Metodología y gestión

* Enfoque: SCRUM. Backlog en Jira con épicas (Captura, Modelo, App, QA/Accesibilidad, Release).
* Sprints: 2–3 semanas. Ceremonias: planificación, dailies, revisión, retrospectiva.
* Entregables por sprint: prototipo de captura; primer modelo; UI accesible; beta cerrada; release MVP.

### 2.5 Plan de trabajo e hitos

|  |  |  |
| --- | --- | --- |
| **Semana** | **Hito** | **Entregable** |
| 1–2 | Prototipo de captura + dataset inicial | Demo de *landmarks* y *pipeline* de datos |
| 3–4 | Modelo TFLite (alfabeto/números) | Modelo cuantizado + *benchmark* |
| 5–6 | Integración UI + TTS + accesibilidad | *Build* funcional |
| 7–8 | Ampliación a ~50 señas + pruebas en 2–3 equipos | Reporte de latencia y precisión |
| 9 | *Beta* y corrección de defectos | Lista de *issues* resueltos |
| 10 | *Release* MVP + documentación | Guía de usuario + informe final |
|  |  |  |

### 2.6 Riesgos y mitigaciones

* **Datos insuficientes / variabilidad gestual.** Usar *data augmentation* y capturas controladas; foco en un vocabulario acotado.
* **Rendimiento en gama media.** Cuantización y *profiling*; límites de FPS/entrada.
* **Ambientes de luz/oclusiones.** Recomendaciones de uso y detección de baja confianza con *fallback* a texto manual.
* **Brecha con comunidad sorda.** Consulta temprana (validación de vocabulario y UX); respeto a LSCh.

### 2.7 Pruebas y métricas de calidad

* **Modelo:** Accuracy y F1 macro por clase; matriz de confusión; *hold‑out* por usuario.
* **App:** Latencia *end‑to‑end* (< 300 ms), *jank* < 5%, *crash‑free sessions* > 99%.
* **Accesibilidad:** Contrast ratio ≥ 4.5:1; tamaños de fuente escalables; *screen reader*; *haptic feedback*.

### 2.8 Consideraciones éticas y de accesibilidad

* **Privacidad por diseño:** sin envío de video; *opt‑in* para telemetría.
* **Inclusión:** co‑diseño con usuarios sordos; lenguaje respetuoso; documentación clara.
* **Licencias/datos:** uso de datasets permitidos; atribución; *model cards*.

## 3. Relación del proyecto con las competencias del perfil de egreso

* Análisis y diseño de soluciones de software. Modelado de pipeline de visión y arquitectura móvil.
* Desarrollo e integración de aplicaciones. Implementación de app móvil, integración de MediaPipe/TFLite/TTS.
* Gestión de datos y modelos. Curación de dataset, feature engineering, entrenamiento y evaluación.
* Aseguramiento de la calidad. Unit/UI tests, métricas de rendimiento, accesibilidad y estabilidad.
* Gestión de proyectos y metodologías ágiles. SCRUM, planificación por sprints, trazabilidad en Jira.
* Ética, profesionalismo y comunicación. Accesibilidad, privacidad, documentación y divulgación responsable.

## 4. Relación con mis intereses profesionales

Me especializo en desarrollo web/móvil e inteligencia artificial aplicada. Este proyecto consolida habilidades en visión por computador y edge ML, alineadas con mi interés por soluciones con impacto social y despliegue real en producción. Fortalece mi portafolio técnico (arquitecturas modernas, CI/CD, accesibilidad) y mi proyección hacia roles de Desarrollador/a Móvil, Ingeniero/a de ML aplicado o Full‑Stack con enfoque en IA.

## 5. Factibilidad del proyecto en la asignatura

* **Alcance acotado y medible:** vocabulario inicial (alfabeto, números y ~50 señas).
* **Tecnologías maduras y *free tier*:** MediaPipe y TFLite permiten *on‑device*; Firebase/Supabase brindan servicios opcionales sin costo inicial.
* **Plan por sprints con entregables:** hitos definidos (ver §2.5) y riesgo contenido.
* **Recursos disponibles:** repositorio, guías y *tooling*; pruebas en teléfonos propios; *benchmarking* del modelo.  
  **Conclusión:** es **viable** entregar el MVP dentro del periodo de la asignatura con calidad suficiente para evaluación y *feedback* de usuarios.

## 6. Conclusiones

SignBridge aborda un problema real de inclusión mediante una arquitectura ligera, privada y eficiente. El enfoque incremental asegura valor temprano y aprendizaje continuo. La alineación con competencias del perfil de egreso y la viabilidad técnica/temporal respaldan su pertinencia como proyecto APT.

## 7. Reflexiones

Este proyecto me reta a equilibrar precisión del modelo con rendimiento en dispositivos reales, y a diseñar con accesibilidad y comunidad en el centro. Más allá del MVP, la evolución natural es ampliar vocabulario, mejorar robustez en escenarios reales y explorar composición de frases, manteniendo principios de privacidad y ética en IA.