UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

CÓMPUTO MÓVIL

Trabajo Final.

EQUIPO:

- Duo Inclusivo

NOMBRE INTEGRANTES:

- Gonzalez Villalba Bryan Jesus
- Sánchez Alvirde Andrés Iván

PROFESOR: Ing Marduk Pérez de Lara Domínguez

GRUPO: 03

SEMESTRE 2026-21

FECHA DE ENTREGA: 28/Noviembre/2025

ÍNDICE

Requerimientos de la aplicación....2
Reglas de negocio....2
Requerimientos funcionales....3
Requerimientos no funcionales....3
Justificación de la viabilidad técnica....3
Alcance y Producto Mínimo Viable (MVP)....4
Explicación del flujo de navegación de la app....4
Pantallas del Wireframe....4
Pantalla 1:Splash Screen y pantalla de carga....4
Pantalla 2: Onboarding - Selección de idioma....5
Pantalla 3: Solicitud de permisos....6
Pantalla 4: Home (Dashboard Principal)....7
Pantalla 5: Modo conversación (En ejecución)....8
Pantalla 6: Modo conversación (Modo pausado/Edición)....9
Pantalla 7: Modo visual (Cámara - Viewfinder)....10
Pantalla 8: Modo visual (Resultado OCR)....11
Pantalla 9: Historial de traducciones realizadas....12
Pantalla 10: Ajustes y accesibilidad....13
Dispositivos y sensores....14
Demo / Maqueta de flujo....15
Stack tecnológico....16
Requerimientos para publicación en la AppStore....16
Estimaciones de tiempo y costos....16
Bibliografía....17

Requerimientos de la aplicación

Reglas de negocio

1. **Gratuidad Parcial:** La aplicación debe ofrecer funciones básicas (traducción simple) de forma gratuita para fomentar la adopción masiva.
2. **Modelo de Licenciamiento B2B:** Las funcionalidades avanzadas en estadios o eventos masivos dependen de una licencia activa pagada por el organizador del evento.
3. **Privacidad por Diseño:** Ningún audio o imagen procesada debe almacenarse permanentemente en los servidores; el procesamiento es efímero.
4. **Prioridad de Accesibilidad:** La interfaz debe cumplir obligatoriamente con normas de alto contraste y compatibilidad con lectores de pantalla (VoiceOver) [1,6]

Requerimientos funcionales

- **RF-01 Traducción de Voz Bidireccional:** El sistema debe capturar audio, transcribir y traducir entre dos idiomas seleccionados en tiempo real [5].
- **RF-02 Traducción Visual (OCR):** El sistema debe detectar texto en imágenes capturadas por la cámara y superponer la traducción.
- **RF-03 Detección de Idioma:** La app debe identificar automáticamente el idioma de entrada en el modo conversación.
- **RF-04 Historial Local:** El usuario debe poder consultar las últimas 10 traducciones de texto sin conexión a internet.
- **RF-05 Configuración de Accesibilidad:** Permitir al usuario ajustar el tamaño de fuente y el contraste independientemente de la configuración del sistema.

Requerimientos no funcionales

- **RNF-01 Latencia:** El tiempo de respuesta entre la voz y la traducción visualizada no debe exceder los 2000ms en condiciones de red 4G/5G.
- **RNF-02 Disponibilidad:** El sistema debe operar con una disponibilidad del 99.5% durante eventos programados.
- **RNF-03 Compatibilidad:** Exclusivo para dispositivos iOS con versión 16.0 o superior.
- **RNF-04 Interfaz:** Diseño compatible con Dynamic Type para escalar textos hasta un 200%.

Justificación de la viabilidad técnica

Funcionalidad Evaluada: Procesamiento Híbrido (On-Device + Cloud).

Justificación: La propuesta inicial sugiere el uso de APIs en la nube (Gemini/OpenAI) para la traducción. Sin embargo, en un estadio con 80,000 personas, la red celular suele saturarse. Es **técnicamente viable** implementar un modelo híbrido. Los dispositivos iPhone modernos (con chips A12 Bionic en adelante) poseen un Neural Engine capaz de ejecutar modelos de CoreML para transcripción (Speech-to-Text) y traducción ligera **directamente en el dispositivo** (Offline).

- **Viabilidad:** Alta. Apple provee el framework `Speech` y `Translation` (en iOS 17+) que permiten estas tareas sin internet.
- **Estrategia:** La app intentará conectar a la API en la nube para máxima precisión; si la latencia supera los 500ms o no hay red, cambiará automáticamente al modelo local de CoreML [3], garantizando el servicio aunque baje ligeramente la calidad gramatical.

Alcance y Producto Mínimo Viable (MVP)

Alcance del Proyecto: Desarrollar una aplicación nativa iOS que elimine barreras idiomáticas y auditivas en eventos deportivos y turísticos mediante IA.

Definición del MVP: El MVP incluirá:

1. Onboarding con selección de idioma materno.
2. Modo Conversación (Split Screen) funcional con API en la nube.
3. Modo Visual (Cámara) con detección de texto simple (sin AR compleja, solo foto estática y traducción).[3]
4. Ajustes básicos de tamaño de texto. *Se excluyen del MVP:* Modo offline robusto, historial en la nube, perfiles de usuario y monetización integrada.

Explicación del flujo de navegación de la app

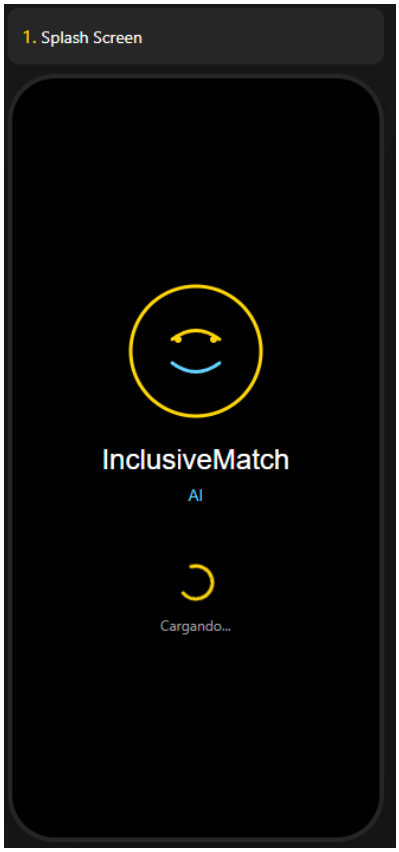
La navegación es lineal y jerárquica para evitar la carga cognitiva:

1. **Inicio:** El usuario abre la app. Si es la primera vez, pasa por el **Onboarding**.
2. **Home:** Llega al menú principal donde elige **Voz** o **Cámara**.
3. **Acción:** Al entrar a un modo, la interacción es inmediata.
4. **Retorno:** Siempre existe un botón "Atrás" o gesto de "Swipe back" para volver al Home.
5. **Ajustes:** Accesibles desde el Home para configuración rápida.

Pantallas del Wireframe

Pantalla 1:Splash Screen y pantalla de carga

- **Funcionalidad:** Muestra el logo y carga configuraciones iniciales.
- **Datos:** Ninguno visible. Internamente carga tokens de sesión.
- **Servicios:** Verificación de estado del servidor (Health Check).
- **Almacenamiento Local:** Lectura de `UserDefaults` para saber si es "First Launch".



Pantalla 2: Onboarding - Selección de idioma

- **Funcionalidad:** El usuario elige su lengua materna y la lengua objetivo (o "Detectar").
- **Datos:** Lista de idiomas (JSON). Selección del usuario (String).
- **Origen:** Catálogo local de idiomas soportados.
- **Operaciones:** Selección y guardado.
- **Almacenamiento Local:** Se guarda la preferencia en `UserDefaults`.

Pantalla 3: Solicitud de permisos

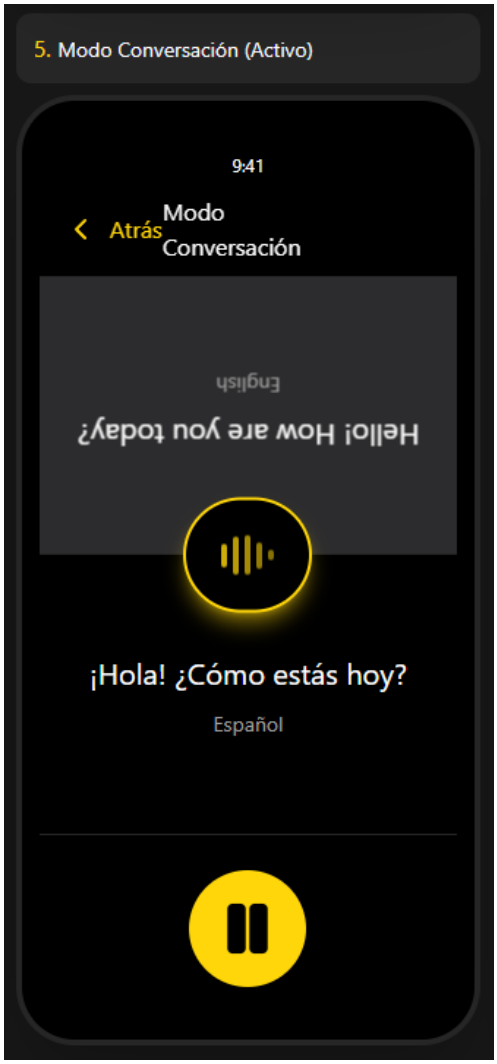
- **Funcionalidad:** Explica y solicita acceso a Micrófono y Cámara.
- **Datos:** Estado de permisos (Booleano).
- **Servicios:** Framework `AVFoundation` para solicitar acceso al hardware.
- **Gestos:** Tap en botones grandes "Permitir".

Pantalla 4: Home (Dashboard Principal)

- **Funcionalidad:** Centro de navegación. Botones gigantes para "Conversación" y "Cámara". Acceso a Historial y Ajustes.
- **Datos:** Iconos estáticos y etiquetas de texto dinámico.
- **Almacenamiento:** No aplica.
- **Orientación:** Vertical (Portrait).

Pantalla 5: Modo conversación (En ejecución)

- **Funcionalidad:** Pantalla dividida. Escucha audio, transcribe y traduce en tiempo real.
- **Datos:** Stream de Audio (Input), Texto Transcrito (String), Texto Traducido (String).
- **Vigencia:** Efímera (se borra al salir, salvo que se guarde).
- **Servicios:** Conexión WebSocket a API de IA o uso de `SFSpeechRecognizer` (Local)[5].
- **Operaciones:** `POST` audio -> `GET` text.



Pantalla 6: Modo conversación (Modo pausado/Edición)

- **Funcionalidad:** El usuario pausa la conversación para leer con calma o corregir una palabra mal escuchada.
- **Datos:** Texto estático de la última interacción.
- **Gestos:** Tap largo en el texto para copiar al portapapeles o reproducir en voz alta (TTS).
- **Servicios:** API Text-to-Speech (AVSpeechSynthesizer).

Pantalla 7: Modo visual (Cámara - Viewfinder)

- **Funcionalidad:** Muestra la vista de la cámara con un recuadro de enfoque. Botón de captura.
- **Datos:** Buffer de video en tiempo real.
- **Servicios:** `AVCaptureSession`.
- **Uso de Sensor:** Acelerómetro para detectar si el teléfono está estable antes de sugerir la captura.

Pantalla 8: Modo visual (Resultado OCR)

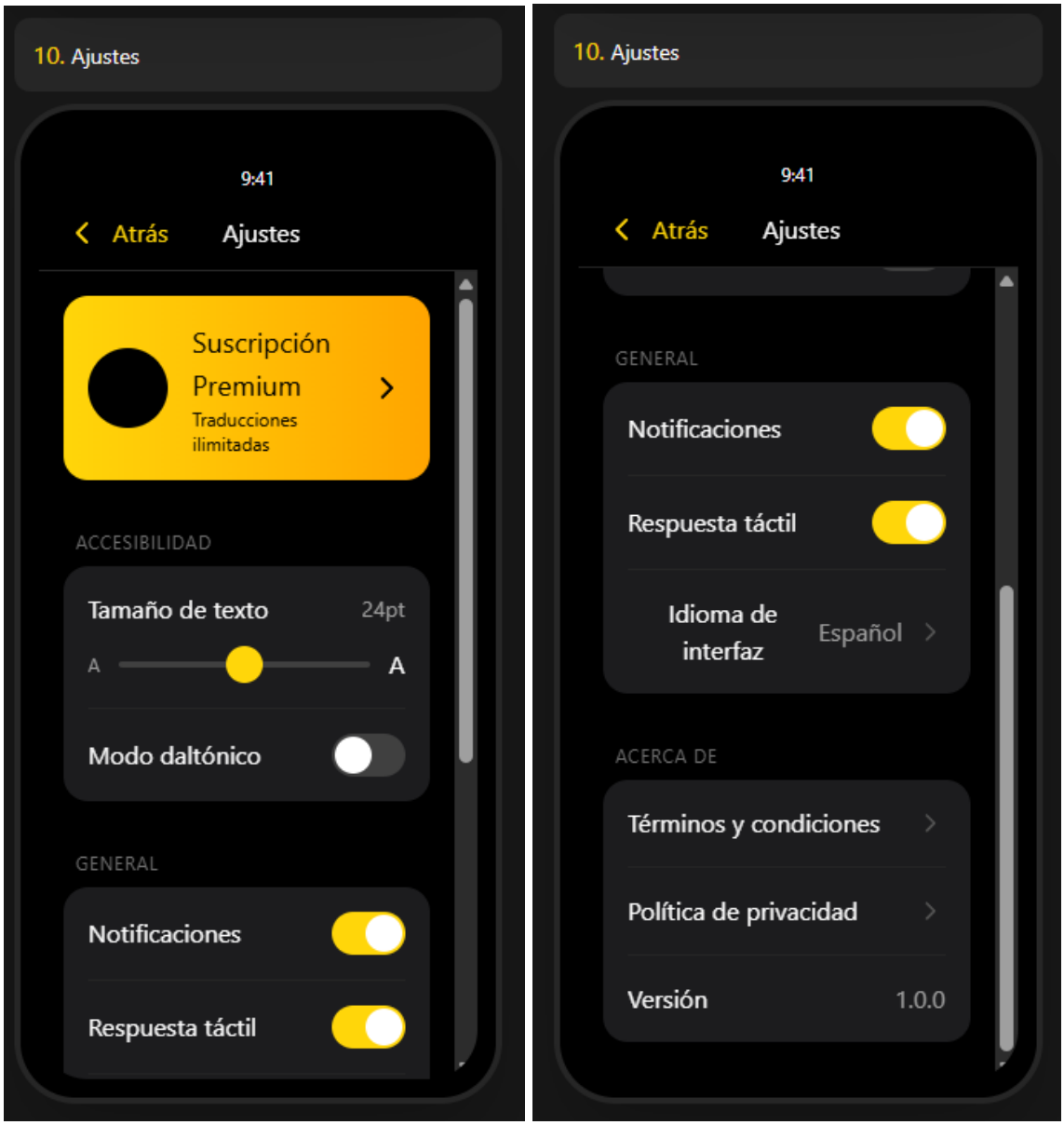
- **Funcionalidad:** Muestra la imagen congelada con el texto traducido superpuesto sobre el original. [4]
- **Datos:** Imagen (Bitmap), Coordenadas del texto (Rect), Texto traducido (String).
- **Servicios:** Vision Framework (OCR) + API Traducción.
- **Almacenamiento Local:** Opción de guardar la imagen en la galería ("Camera Roll").

Pantalla 9: Historial de traducciones realizadas

- **Funcionalidad:** Lista cronológica de las últimas interacciones guardadas.
- **Datos:** Array de Objetos {Fecha, TextoOriginal, TextoTraducido, Tipo}.
- **Origen:** Base de datos local.
- **Operaciones:** Lectura (Read) y Borrado (Delete).
- **Almacenamiento Local:** Uso de **CoreData** o **SwiftData** para persistencia de datos de texto (ligero).

Pantalla 10: Ajustes y accesibilidad

- **Funcionalidad:** Configuración de tamaño de letra, modo oscuro/alto contraste, y gestión de suscripción [1,6].
- **Datos:** Variables de configuración (Float para tamaño, Bool para contraste).
- **Servicios:** Conexión a StoreKit para verificar estatus de suscripción (Premium vs Free).



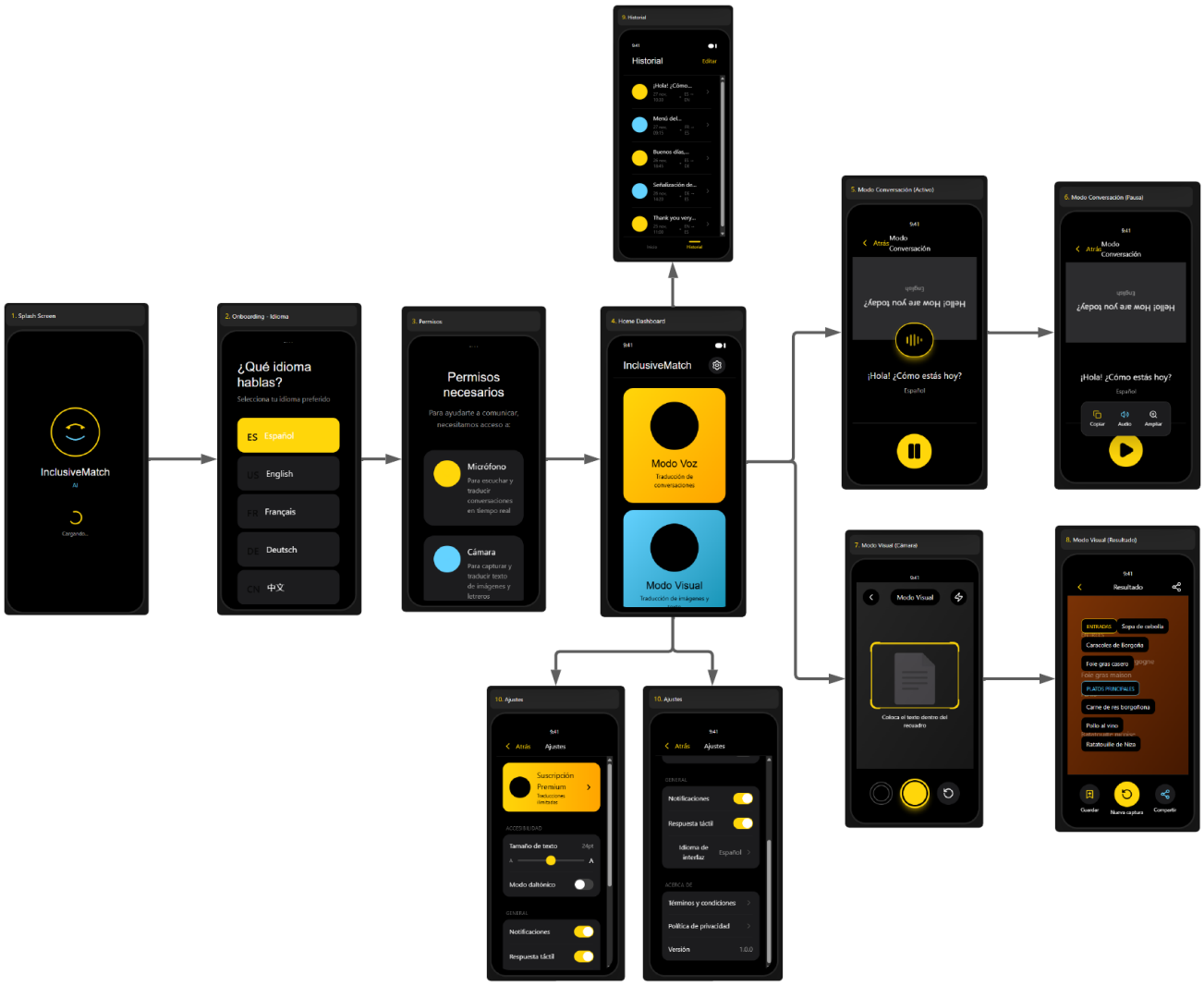
Dispositivos y sensores

- **Dispositivos:** iPhone (modelos XR en adelante).
- **Tamaños:** Optimizado para pantallas de 6.1" y 6.7".
- **Orientaciones:**
    - Modo Conversación: Vertical (Portrait) preferente, u Horizontal (Landscape) para modo "Mesa" (donde cada usuario ve su mitad orientada hacia él).
    - Modo Visual: Vertical.
- **Sensores:**
    - **Micrófono:** Captura de voz (esencial).
    - **Cámara:** Captura de texto (esencial).

- **Acelerómetro/Giroscopio:** Para orientar el texto en el "Modo Mesa" automáticamente según la inclinación del teléfono hacia uno u otro usuario.
- **Haptic Engine:** Feedback vibratorio al detectar un idioma o finalizar una traducción para usuarios con baja visión.

Demo / Maqueta de flujo

Se hace un diagrama de cómo están conectadas las pantallas pero de igual forma se adjunta un link para poder ver la demo en figma. https://www.figma.com/make/xfxdpT70zjXfz3jhPZ5VXR/Untitled?node-id=0-1&t=vOXOSO8mGATy55YJ-1. De igual manera en la carpeta del repositorio donde se encuentra el pdf estarán las imágenes de la carpeta Wireframe.



Stack tecnológico

- **Lenguaje:** Swift 5.9.
- **Framework UI:** SwiftUI (para interfaz declarativa y adaptable).
- **Arquitectura:** MVVM (Model-View-ViewModel) [7].
- **IA / Machine Learning:**
    - CoreML (Modelos locales) [3].
    - Vision Framework (OCR) [4].
    - Google Gemini API (Procesamiento complejo en nube)[2,8].
- **Backend:** Firebase (Auth y Analytics).

Requerimientos para publicación en la AppStore

1. **Privacy Labels:** Declarar que se recolectan "Datos de Audio" y "Información de Uso", pero que no se vinculan a la identidad del usuario para rastreo [1,6].
2. **Permisos en Info.plist:**
    - `NSMicrophoneUsageDescription`: "InclusiveMatch necesita escucharte para traducir tu voz en tiempo real."
    - `NSCameraUsageDescription`: "Necesitamos la cámara para leer y traducir letreros."
3. **Accesibilidad:** La app debe pasar la validación de Xcode Accessibility Inspector (etiquetas en todos los botones para VoiceOver).

Estimaciones de tiempo y costos

Tiempo de Desarrollo: 14 Semanas (3.5 Meses).

- Semanas 1-2: Diseño y Prototipado.
- Semanas 3-6: Desarrollo del Core (Modo Voz y API).
- Semanas 7-9: Desarrollo Modo Visual (OCR).
- Semanas 10-12: Implementación Offline y Persistencia.
- Semanas 13-14: Pruebas, corrección de bugs y despliegue.

Costos Estimados (MXN):

- Desarrollo (Mano de obra estimada): $120,000 MXN.
- Licencia Apple Developer: $1,700 MXN/año.
- Servicios Cloud (Gemini API + Firebase): $2,000 MXN/mes (escalable según uso).
- Mantenimiento: $5,000 MXN/mes (actualizaciones y servidores).

Bibliografía

1. *Human Interface Guidelines | Apple Developer Documentation*. (s. f.). Apple Developer Documentation. https://developer.apple.com/design/human-interface-guidelines
2. *Gemini API | Google AI for Developers*. (s. f.). Google AI For Developers. https://ai.google.dev/gemini-api/docs?hl=es-419
3. Apple Inc., *"Core ML Framework Documentation,"* Apple Developer, 2024.
    a. https://developer.apple.com/documentation/coreml/
4. *Recognizing Text in Images | Apple Developer Documentation*. (s. f.). Apple Developer Documentation. https://developer.apple.com/documentation/vision/recognizing-text-in-images
5. *Speech | Apple Developer Documentation*. (s. f.). Apple Developer Documentation. https://developer.apple.com/documentation/Speech
6. *Typography | Apple Developer Documentation*. (s. f.). Apple Developer Documentation. https://developer.apple.com/design/human-interface-guidelines/typography
7. Hudson, P. (2024, 11 abril). *Introducing MVVM into your SwiftUI project*. Hacking With Swift. https://www.hackingwithswift.com/books/ios-swiftui/introducing-mvvm-into-your-swiftui-project
8. James, M., Toddy, R., Lawson, & Joseph, O. (s. f.). *Performance Tradeoffs in Latency, Accuracy, and Resource Consumption for Edge Security AI*. ResearchGate. https://www.researchgate.net/publication/395030307_Performance_Tradeoffs_in_Latency_Accuracy_and_Resource_Consumption_for_Edge_Security_AI#:~:text=Lightweight%20models:%20Significantly%20lower%20latency,consumption