# **Fase de Desarrollo del Proyecto**

## **1. Configuración del Entorno de Desarrollo**

### **Lenguaje de Programación:**

* Python será el lenguaje principal, aprovechando bibliotecas especializadas en inteligencia artificial.

### **Frameworks y Herramientas:**

* **TensorFlow** y **PyTorch**: Para la creación y entrenamiento de modelos de IA.
* **OpenCV**: Procesamiento de imágenes capturadas por las cámaras.
* **Keras**: Desarrollo y despliegue de modelos de machine learning.
* **Google Text-to-Speech API** o **pyttsx3**: Para síntesis de voz.

### **Control de Versiones:**

* Uso de **Git** y **GitHub** para la gestión de versiones.
* Organización de ramas:
  + **Main:** Código estable.
  + **Develop:** Desarrollo en curso.
  + Ramas específicas para nuevas funcionalidades.

## **2. Desarrollo del Backend**

### **Entrenamiento del Modelo de IA:**

* Recopilación y preprocesamiento de datos (imágenes y videos de señas).
* Entrenamiento del modelo con **TensorFlow/PyTorch** y **Keras**.
* Evaluación continua para garantizar una precisión superior al 85%.

### **API para Traducción:**

* Creación de endpoints para:
  + Capturar datos desde gafas y dispositivos móviles.
  + Procesar información en la nube mediante modelos alojados en **AWS** o **Google Cloud**.
  + Devolver resultados traducidos (texto o audio).

### **Almacenamiento:**

* Uso de servicios en la nube para almacenar datos procesados temporalmente.

## **3. Desarrollo del Frontend**

### **Interfaz de Usuario:**

* **Gafas AR:**
  + Diseño minimalista para visualizar texto traducido.
  + Captura de video en tiempo real.
* **Aplicación Móvil:**
  + Configuración avanzada (idiomas, velocidad de traducción, preferencias).
  + Visualización de estadísticas de uso.

### **Flujo de Interacción:**

* Captura y traducción de señas en tiempo real.
* Presentación de resultados como texto en pantalla o audio mediante síntesis de voz.
* Ajustes personalizables desde la aplicación móvil.

## **4. Integración y Pruebas**

### **Pruebas de Integración:**

* Verificación de comunicación eficiente entre gafas, aplicación móvil y servicios en la nube.
* Medición de latencia para traducciones en tiempo real.

### **Pruebas Automatizadas:**

* Implementación de herramientas como **Jenkins** o **GitHub Actions** para integración y despliegue continuo (CI/CD).
* Pruebas de precisión, latencia y usabilidad del sistema.

## **5. Metodología de Desarrollo**

### **Metodología Ágil (Scrum):**

* El desarrollo se organizará en **sprints** para:
  1. Implementar reconocimiento de señas en tiempo real.
  2. Integrar traducción y visualización en texto/audio.
  3. Añadir configuraciones avanzadas en la aplicación móvil.
* Revisión constante al final de cada sprint para realizar mejoras basadas en retroalimentación.

## **6. Documentación**

### **Documentación Técnica:**

* Detalle sobre la configuración de hardware y software.
* Guía sobre dependencias y estructura del código.

### **Manual de Usuario:**

* Instrucciones para el uso del producto.
* Resolución de problemas comunes y pasos de configuración inicial.

## **7. Entrega Final**

* Distribución del software empaquetado para gafas AR y dispositivos móviles.
* Configuraciones predeterminadas para facilitar la instalación.
* Publicación en **Google Play Store** y **Apple App Store**.