

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Facultad de Ingeniería
Escuela de Ciencias y Sistemas



Proyecto - Fase 1 Inteligencia Artificial

G#7

NOMBRE	CARNET
Vania Argueta Rodríguez	201213487
Henry Ronely Mendoza Aguilar	202004810
David Eduardo López Morales	201907483
Vernik Carlos Alexander Yaxon Ortiz	201712057

Manual Técnico

Introducción

Este manual describe en detalle la arquitectura, herramientas y metodologías utilizadas en el desarrollo del proyecto para el curso de Inteligencia Artificial 1 en la Universidad de San Carlos de Guatemala. El propósito del proyecto es implementar un modelo de inteligencia artificial capaz de interpretar entradas de texto en español y responder de forma coherente. Además, incluye la documentación necesaria para la instalación, configuración, y funcionamiento del sistema.

Objetivos

Desarrollar un modelo que interprete entradas de texto en idioma español y genere respuestas coherentes. Este modelo debe ejecutarse completamente en el navegador, sin depender de servicios externos.

Herramientas utilizadas

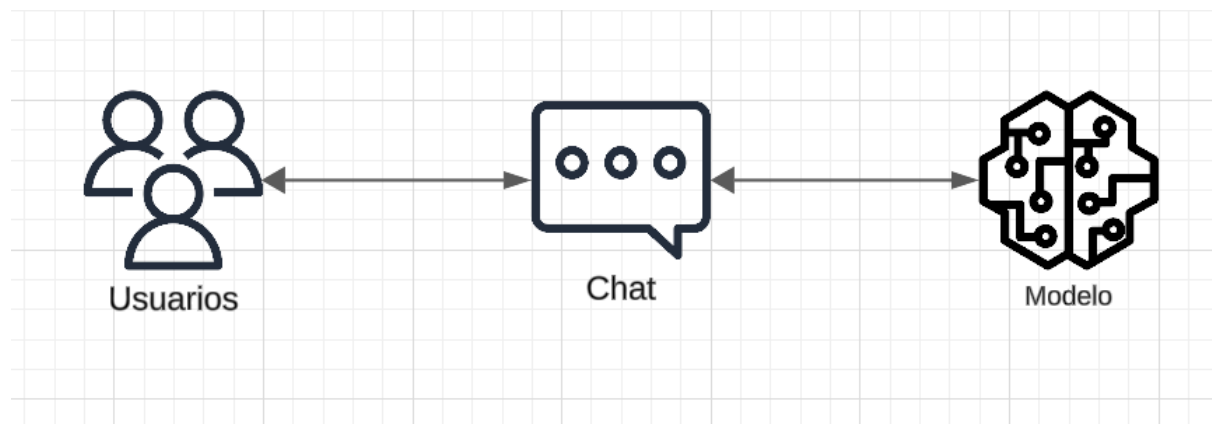
- **React + Vite:** React es una biblioteca de JavaScript para construir interfaces de usuario. En este proyecto, se utilizó junto con Vite, una herramienta de desarrollo que optimiza el tiempo de construcción y mejora el rendimiento del entorno de desarrollo. La combinación de estas herramientas permitió un desarrollo rápido y eficiente de componentes modulares y reutilizables.
- **Bootstrap:** Bootstrap es una biblioteca de diseño CSS que facilita la creación de interfaces responsive y estéticamente atractivas. En el proyecto, se usó para diseñar una interfaz de usuario que sea intuitiva y adaptable a diferentes dispositivos.
- **JavaScript:** El lenguaje principal del proyecto, JavaScript, permitió la implementación de la lógica de negocio y la interacción entre el modelo de inteligencia artificial y la interfaz gráfica. Su uso garantizó una integración fluida con las bibliotecas React y TensorFlow.js.
- **TensorFlow.js:** TensorFlow.js es una biblioteca para el desarrollo y entrenamiento de modelos de machine learning directamente en el navegador. En este proyecto, se utilizó para crear un modelo de red neuronal capaz de procesar texto en español y generar respuestas coherentes.

Arquitectura de la aplicación

El sistema tiene una arquitectura basada en tres capas principales:

1. **Capa de Presentación:** Diseñada con React y estilizada con Bootstrap para proporcionar una experiencia de usuario fluida y atractiva.
2. **Capa de Procesamiento:** Implementada con TensorFlow.js para procesar y generar respuestas basadas en entradas de texto.
3. **Capa de Interacción:** Gestionada por JavaScript para vincular la lógica del modelo con la interfaz gráfica del chat.

Flujo de interacción



Este diagrama representa el flujo de interacción en el sistema de chat basado en el modelo previamente entrenado

1. Los Usuarios envían un mensaje al sistema a través del Chat.
2. El Chat pasa la consulta al Modelo para su procesamiento.
3. El Modelo devuelve la respuesta más relevante al Chat.
4. Finalmente, el Chat presenta la respuesta generada a los usuarios.

Implementación del Modelo

El modelo de inteligencia artificial utilizado en este proyecto se basa en el Universal Sentence Encoder (USE) proporcionado por TensorFlow.js. Este modelo preentrenado permite codificar oraciones en vectores numéricos que luego se pueden utilizar para realizar tareas de procesamiento de lenguaje natural, como la clasificación o búsqueda de similitudes.

Importación de Librerías Necesarias

```
tend-chat > src > utils > JS modelo.js > ...
You, 3 hours ago | 2 authors (You and one other)
import * as tf from '@tensorflow/tfjs';
import * as use from '@tensorflow-models/universal-sentence-encoder';
```

Definición de una Base de Conocimiento: Una lista de respuestas predefinidas se almacena en un array, y estas respuestas se comparan con la entrada del usuario:

```

3
4 // Base de conocimiento
5 const responses = [
6   'I am feeling great!',
7   'The capital of China is Beijing.',
8   'You have five fingers on your hand.',
9   'The sky is blue on a clear day.',
10  'Water boils at 100 degrees Celsius.',
11  'Tengo 27 años',
12  'Soy de Guatemala',
13  'Mi color favorito es el azul',
14  'Mi comida favorita es la pizza',
15  'Mi película favorita es Star Wars',
16  'Mi serie favorita es Breaking Bad',
17  'Mi libro favorito es El Alquimista',
18  'Mi deporte favorito es el fútbol',
19  'Mi equipo favorito es el Barcelona',
20  'Mi animal favorito es el perro',
21  'Mi pasatiempo favorito es leer',
22  'Mi hobby favorito es cocinar',
23  'Mi lugar favorito es la playa',
24  'Mi canción favorita es come as you are'.
```

Carga del Modelo y Generación de Respuestas: El modelo Universal Sentence Encoder se carga de forma asíncrona. Una vez cargado, codifica las oraciones de entrada y las respuestas para comparar similitudes:

```

7 let model = null;
8
9 // modelo universal sentence encoder
10 export async function loadModel() {
11   if (!model) {
12     model = await use.loadQnA();
13   }
14 }
15
16 export async function getResponse(question) {
17   if (!model) {
18     throw new Error("Error al cargar modelo. Err: Se debe");
19   }
20
21   const input = {
22     queries: [question],
23     responses: responses,
24   };
25
26   const embeddings = await model.embed(input);
```

Para realizar operaciones matemáticas y de álgebra lineal, en este caso se utilizó para calcular similitudes entre la consulta y las respuesta a partir de los vectores

obtenidos a partir de los textos de la base de conocimiento, estos vectores son los llamados embeddings.

```
// para calcular similitudes entre la consulta y las respuestas
const scores = tf.matMul(
  embeddings.queryEmbedding,
  embeddings.responseEmbedding,
  false,
  true
).arraySync();
```