Significado del Escudo | Universidad Loyola Universidad Loyola

Asistente Universitario

Aplicación con redes neuronales densas

Autor:

Marcelo Bryan Ramirez Yapura (15351-5)

Este informe detalla la definicio y el desarrollo de un asistente virtual web con python

CONTENIDO

[I. EVALUACION DE NECESIDADES 1](#_Toc165968417)

[1.1. Infraestructura, artefactos y otros 1](#_Toc165968418)

[II. OBJETIVO 1](#_Toc165968419)

[2.1. Reacondicionar: 1](#_Toc165968420)

[2.2. Renovar: 1](#_Toc165968421)

[2.3. Limpiar: 1](#_Toc165968422)

[III. TAREAS DE MANTENIMIENTO 2](#_Toc165968423)

[3.1. Reacondicionar 2](#_Toc165968424)

[3.2. Renovar 2](#_Toc165968425)

[3.3. Limpiar 2](#_Toc165968426)

[IV. FRECUENCIAS EN EL MANTENIMIENTO 3](#_Toc165968427)

[4.1. Frecuencia de Reacondicionamiento 3](#_Toc165968428)

[V. ASIGNACION DE RECURSOS 3](#_Toc165968429)

[VI. CALENDARIO DE MANTENIMIENTO 5](#_Toc165968430)

[6.1. Crognograma 5](#_Toc165968431)

[VII. PROCEDIMIENTOS DE SEGURIDAD 6](#_Toc165968432)

[VIII. REGISTRO Y MONITOREO 7](#_Toc165968433)

[8.1. Registro 7](#_Toc165968434)

[8.2. Monitoreo 7](#_Toc165968435)

[IX. EVALUAR Y AJUSTAR 8](#_Toc165968436)

[9.1. Evaluación 8](#_Toc165968437)

[9.2. Ajuste 8](#_Toc165968438)

# INTRODUCCIÓN

Este proyecto se centra en la creación de un chatbot académico para una universidad, utilizando una red neuronal densa para inferir y responder a una variedad de preguntas de manera coherente y precisa. El objetivo principal es proporcionar un asistente virtual eficiente y útil que pueda ayudar a los estudiantes y visitantes de la universidad respondiendo sus consultas de manera automática. La implementación abarca tecnologías tanto de frontend como de backend, con una infraestructura de despliegue y gestión robusta.

# TECNOLOGÍAS UTILIZADAS

## Pythonxºxº

* **Flask**: Un microframework para crear aplicaciones web ligeras y rápidas en el lado del servidor.
* **SQLite**: Base de datos ligera utilizada para almacenar mensajes y datos de entrenamiento del chatbot.

## Frontend

* **React**: Biblioteca de JavaScript para construir interfaces de usuario. React se utiliza para manejar el estado y los componentes del chatbot.
* **HTML y CSS**: Lenguajes de marcado y estilo para estructurar y diseñar la interfaz del chatbot. El diseño está optimizado para pantallas móviles.
* **React Switch**: Componente utilizado para alternar entre diferentes instancias del chatbot.

# DESCRIPCION DEL PROYECTO

## Backend: Python y Flask

El backend de la aplicación está desarrollado en Python utilizando el framework Flask. Las principales funciones del backend incluyen:

Gestión de Rutas y Endpoints:

* GET /: Renderiza la página principal del chatbot.
* POST /get: Recibe preguntas del usuario y devuelve respuestas generadas..
* POST /save: Guarda las preguntas del usuario en la base de datos para el entrenamiento futuro del chatbot.

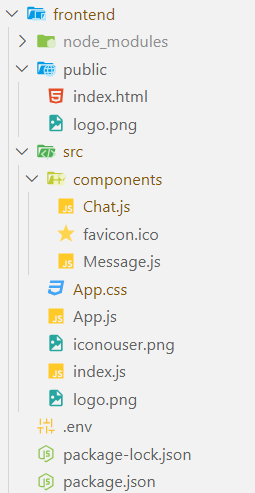
## Base de Datos SQLite:

Se utiliza SQLite para almacenar los mensajes de los usuarios y los datos de entrenamiento. Esta base de datos es ligera y se integra fácilmente con Flask.

## Frontend: React

El frontend está desarrollado utilizando React, HTML y CSS, con un enfoque en el diseño responsivo para dispositivos móviles.

* **Componente Chat**: Maneja la interacción principal del usuario con el chatbot. Permite enviar preguntas y recibir respuestas.
* **Componente Message**: Renderiza los mensajes individuales en la interfaz del usuario, incluyendo íconos personalizados para el usuario y el chatbot.
* **Componente Switch**: Permite al usuario alternar entre dos instancias del chatbot, una para respuestas generales y otra para guardar mensajes de entrenamiento.
* **Gestión de Estado**: Utiliza **useState** y **useEffect** para gestionar el estado y los efectos secundarios en los componentes.



## Infraestructura: Fly.io y Hostinger

* **Fly.io**: Se utiliza para desplegar la aplicación Flask en un entorno de producción. Fly.io proporciona IPs públicas y capacidades de escalado automático.
* **Hostinger**: Utilizado para registrar y gestionar el dominio personalizado del chatbot. La configuración DNS permite redirigir loyobot.com al servidor de Fly.io, asegurando una experiencia de usuario fluida.
* **Docker**: Utilizado para contenedizar la aplicación. El Dockerfile define el entorno y las dependencias necesarias, lo que facilita la replicación del entorno de desarrollo en producción.