

ANEXO V: Manual del programador.

*Chatbot aplicado a la resolución de consultas en asignaturas
de Bases de Datos*

Trabajo de Fin de Grado

INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

Septiembre de 2023

Autor:

Manuel Santa Isabel Mayo

Tutoras:

Ana Belén Gil González

Ana De Luis Reboredo

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN	1
2. CÓDIGO FUENTE	2

1. INTRODUCCIÓN

En este anexo se detallan los archivos de programa que se entregan junto a nuestra documentación. Vamos a dividir dichos archivos en diferentes ficheros en los que se dividen. Además, explicaremos los conceptos más importantes y a destacar de los archivos. Además, explicaremos de forma sencilla como activar nuestro servicio de forma local.

2. CÓDIGO FUENTE

Vamos a explicar la arquitectura de nuestro código fuente, destacando las características de los archivos y ficheros más significativos. Nuestro código fuente se divide en dos grandes ficheros:

- **“back\DJANGO_BACKEND\mysite”**, contiene todos los ficheros relacionados con el Backend de Django. Destacaremos los ficheros y archivos más importantes:
 - **“build”**, fichero que contiene el montaje de nuestro Frontend de ReactJ. Con este fichero los usuarios podrán interactuar con la web.
 - **“bot_architecture”**, fichero que contiene todas las funciones relacionadas con nuestra red neuronal
 - **Intents_en.json**, set de datos con el que entrenaremos nuestra red neuronal
 - **RN_NL.py**, código de Python con el que crearemos nuestro modelo entrenado de Red Neuronal.
 - **chatbot_model.h5**, modelo de red neuronal entrenado
 - **basic_chatbot.py**, código de Python que usaremos para poder resolver las distintas llamadas entre el usuario y el modelo entrenado.
 - **“usuarios”**, fichero que contiene las funciones para registrar usuarios. El resto de las funciones están dentro de la biblioteca de Djoser.
 - **“message”**, fichero que contiene las funciones para guardar y cargar los mensajes de los usuarios con ChatGPT.
 - **“gestion_gpt”**, fichero que contiene las funciones relacionadas tanto con el registro de peticiones para hacer llamadas a ChatGPT, como el control que se lleva a cabo para validar dichas peticiones. Destacamos que dentro de este fichero, más en concreto en el archivo **views.py**, tenemos los límites establecidos para cada usuario además de la herramienta tiktoken, usada para poder medir la longitud de los mensajes.
 - **“mysite”**, fichero principal y donde tendremos toda la configuración de nuestro servicio Django.
- **“\fron\dbcapy”**, fichero que contiene todo lo relacionado con el Frontend de nuestra página web. Al tratarse de una aplicación de ReactJS, viene incluida de varios ficheros, pero nosotros nos centramos solo en el fichero **“\src”**.
 - **“\containers”**, fichero que contiene todas las páginas de nuestra web.
 - **“ConfirmarContraseña.js”**
 - **“ConfirmarUsuario.js”**
 - **“Home.js”**
 - **“OlvidarContraseña.js”**
 - **“Registro.js”**
 - **“RevisarMail.js”**
 - **“Teoría.js”** y **“Práctica.js”**, estos dos archivos son los que contienen las funciones de llamada a la API de ChatGPT.
 - **“\actions”**, este fichero se utilizará para poder comunicar nuestro Frontend con Backend. Este compuesto por diferentes archivos:

Manuel Santa Isabel Mayo

- **“chat_bot.js”**, archivo que gestiona las llamadas a nuestra red neuronal
- **“chat_messages.js”**, archivo que gestiona las peticiones y validaciones de los usuarios cuando queremos hacer una llamada a ChatGPT.

Para poder activar nuestro servidor, simplemente debemos ir al fichero de Backend, **“DBCapyDevProject\DBCapy\back\DJANGO_BACKEND\mysite”**, y escribir el siguiente comando:

```
Python -m manage runserver
```

De esta forma sencilla activamos nuestro servidor.

