INFORME FINAL DEL PROYECTO

“IncluQuery”

CASTRO WACHS TRINIDAD

Get Talent 2024

PI DATA STRATEGY & CONSULTING

INDICE

Contenido

INTRODUCCION

Este documento tiene como objetivo presentar la solución desarrollada en el marco del proyecto final del Get Talent, centrado en la implementación de una arquitectura RAG (Retrieval-Augmented Generation). A lo largo de este proyecto, se busca demostrar la aplicación efectiva de los conceptos y técnicas aprendidos, abordando un problema real mediante la integración de diversas tecnologías. La solución propuesta tiene como fin proporcionar un sistema eficiente y funcional para procesar consultas en lenguaje natural, apoyado en una base de datos vectorial y un modelo de lenguaje de última generación.

La solución está orientada a resolver una problemática específica dentro del contexto elegido, para lo cual se ha seguido un enfoque estructurado que incluye la recopilación y preparación de datos, la creación de módulos de recuperación y augmentación de información, así como la integración de un modelo de generación de texto para ofrecer respuestas precisas y coherentes. En todo momento, se ha puesto especial énfasis en la originalidad, pertinencia y efectividad de la solución, con el objetivo de demostrar no solo el conocimiento técnico adquirido, sino también la capacidad para innovar y optimizar procesos mediante el uso de herramientas avanzadas.

En las próximas paginas se documentan detalladamente las decisiones técnicas tomadas durante el desarrollo del proyecto, así como las justificaciones que respaldan cada elección, con el fin de garantizar una ejecución coherente y profesional. A su vez se añade una pequeña conclusión destacando los aprendizajes y lo que fue la experiencia de transitar el Get Talent en Pi Consulting.

ESTUDIO INICIAL

DESCRIPCION DEL CONTEXTO

En Argentina, las personas con discapacidad enfrentan diversas barreras para acceder a información sobre sus derechos, beneficios y servicios disponibles. A pesar de contar con una legislación progresiva, como la Ley Nacional de Discapacidad, y programas de apoyo, la dispersión de recursos, la falta de centralización y la complejidad del lenguaje legal dificultan que este colectivo y sus familias comprendan plenamente sus derechos. Muchas personas con discapacidad y sus familias desconocen los derechos y beneficios que les corresponden debido a la complejidad de las leyes y la falta de recursos accesibles. Este sistema se presenta como una herramienta clave para responder preguntas frecuentes como:

- ¿Qué es el CUD?
- ¿Qué es la curatela?
- ¿Qué se entiende por accesibilidad según la ley, y cómo contribuye a la integración de personas con movilidad reducida?
- ¿Dónde se realiza la solicitud de una Pensión No Contributiva por Invalidez Laboral?

PROBLEMAS

Complejidad y dispersión de la información: Las leyes y los servicios disponibles para las personas con discapacidad son difíciles de entender debido a su lenguaje técnico y la falta de una centralización de recursos. Esto crea confusión entre las personas con discapacidad y sus familias, quienes no siempre saben a dónde acudir para obtener ayuda.

Exclusión social: La falta de acceso a información comprensible y adaptada a las necesidades de este grupo perpetúa la exclusión social. Las personas con discapacidad se ven limitadas en su participación en la vida social y económica, lo que afecta su calidad de vida.

Desigualdad en el acceso a beneficios y servicios: La falta de herramientas accesibles para la consulta de derechos y beneficios impide que muchas personas con discapacidad puedan acceder a los servicios y programas a los que tienen derecho, perpetuando las desigualdades en su calidad de vida.

OPORTUNIDADES

Desarrollo de plataformas accesibles y centralizadas: Existe una oportunidad significativa para crear plataformas digitales que centralicen y presenten la información relevante sobre derechos, beneficios y servicios de manera clara, accesible y comprensible. Estas plataformas podrían ser

adaptadas a diversos formatos y necesidades, como lenguaje sencillo, lectura fácil y accesibilidad web.

Fortalecimiento de la inclusión social: Brindar acceso a información relevante puede mejorar la participación social y económica de las personas con discapacidad, permitiéndoles acceder a los beneficios que les corresponden, lo que contribuiría a una mayor autonomía y calidad de vida.

Empoderamiento de las personas con discapacidad y sus familias: Proveer herramientas claras y accesibles para comprender los derechos y servicios disponibles empoderaría a las personas con discapacidad y a sus cuidadores, permitiéndoles tomar decisiones informadas y acceder a recursos que fomenten su independencia y bienestar.

PLAN DE PROYECTO

OBJETIVO DEL SISTEMA

El objetivo de este sistema es proporcionar información clara, precisa y accesible sobre los derechos, beneficios y servicios para personas con discapacidad en Argentina, respondiendo de manera rápida y efectiva a las consultas de los usuarios. Utilizando un lenguaje sencillo y una terminología fácil de entender, el sistema permitirá a los usuarios obtener respuestas específicas, limitadas al contexto geográfico y legal de Argentina. Las personas con discapacidad, sus familias o cuidadores serán los principales usuarios, quienes podrán hacer consultas sobre temas como beneficios sociales, derechos laborales o leyes de accesibilidad vigentes en el país, obteniendo información relevante, comprensible y adaptada a sus necesidades

USUARIOS PRINCIPALES

El sistema está diseñado para ser utilizado por los siguientes grupos:

- **Personas con discapacidad:** Usuarios directos que buscan información sobre sus derechos, beneficios y servicios disponibles.
- **Familiares y cuidadores:** Personas que apoyan a las personas con discapacidad y necesitan acceso a información clara para brindar asistencia adecuada.
- **Profesionales o entidades que apoyan a este grupo:** Organizaciones, instituciones o profesionales que trabajan con personas con discapacidad y requieren información precisa y actualizada para orientar a sus usuarios.

ALCANCE DEL SISTEMA

El sistema se enfocará en los siguientes aspectos:

- **Tema principal:** El sistema abordará principalmente los derechos, leyes y servicios relacionados con la discapacidad en Argentina, brindando acceso a información sobre beneficios sociales, derechos laborales, leyes de accesibilidad, entre otros.
- **Región inicial:** El sistema se centrará en **Argentina**, considerando la legislación, normativas y recursos disponibles en el país. La información será contextualizada a las leyes y servicios específicos de Argentina, aunque se puede expandir en el futuro a otras regiones si es necesario.
- **Documentos base:** El sistema utilizará como base una variedad de fuentes confiables, como **leyes nacionales**, **reglamentaciones gubernamentales**, **guías oficiales** y **recursos de organizaciones no gubernamentales (ONGs)** especializadas en discapacidad.

MODULOS NECESARIOS

El sistema debe incluir los siguientes módulos para asegurar que las respuestas sean completas y precisas:

- **Recuperación:** Este módulo se encargará de buscar en la base de datos vectorial las leyes, fragmentos de textos y documentos relevantes que se correspondan con la consulta del usuario. Utilizará algoritmos de búsqueda semántica para obtener la información más adecuada.
- **Augmentación:** Este módulo enriquecerá las respuestas obtenidas, proporcionando explicaciones claras, contexto adicional o ejemplos relevantes que ayuden a comprender mejor la información solicitada.
- **Generación:** Este módulo será responsable de convertir los fragmentos de información y el contexto enriquecido en una **respuesta comprensible** y coherente, que se adapte al lenguaje natural del usuario y esté claramente estructurada.

ESTRUCTURA DEL CODIGO DEL PROYECTO

```
|- codigo
  |- app
        |- routers
              |- consulta.py
        |- services
              |- buscar.py
             |- generacion.py
```

```
        |- utils
                |- extraccion.pdf
                |- procesamiento.py
|- data
        |- chromadb
        |- INFORMACION_LIMPIO.txt
        |- INFORMACION.pdf

|- frontend
        |- index.html
        |- logo.png
        |- script.js
        |- styles.css
|- venv
|- .env
|- main.py
|- requirements.txt
```

DESCRIPCION DE CADA CARPETA Y ARCHIVO

- **/app**: Esta carpeta contiene toda la lógica del proyecto.
    - **/routers**: Aquí está el archivo, *consulta.py,* donde se define el endpoint de la API, que es para manejar las consultas de los usuarios.
    - **/services**: Aquí está la lógica más compleja de negocio, como la búsqueda en la base de datos para recuperar los datos de la misma y enviársela junto con la query al LLM a través del módulo de generación.
    - **/utils**: Carpeta con funciones auxiliares que se usan en varias partes del sistema, como la limpieza de datos o la transformación de textos de pdf a un formato 'txt'. También aquí esta la lógica para realizar la división de texto en chunks; la generación de embeddings y la lógica para realizar el almacenamiento en la base de datos vectorial haciéndola persistente.
- **/data**: En esta carpeta se guarda los archivos de datos importantes que contiene las leyes y derechos relacionados con la discapacidad. Aquí también se ubica la base de datos.

- **/frontend**: Esta carpeta contiene los archivos esenciales para la interfaz de usuario: index.html para estructurar la página, logo.png como imagen del logo, script.js para la lógica y las interacciones, y styles.css para definir el diseño visual de la página.
- **main.py**: Aquí se procesa un archivo de texto para generar embeddings, realiza una búsqueda en una base de datos vectorial y configura una API con FastAPI para gestionar consultas sobre leyes y derechos.
- **requirements.txt**: Archivo donde están listadas todas las dependencias necesarias para ejecutar el proyecto.

NOMBRE DEL SISTEMA: “IncluQuery”

IncluQuery es una herramienta que mezcla tecnología con un toque de empatía, pensada para hacer la vida más fácil a las personas con discapacidad, sus familias y cuidadores. Su diseño está enfocado en darles acceso a información que realmente puede marcar la diferencia. El nombre refleja lo que busca: ser una herramienta accesible y útil, ayudando a crear un mundo más inclusivo. Con IncluQuery, la información se convierte en un puente que facilita la igualdad de oportunidades y un entorno más equitativo para todos.

"*IncluQuery: Tu consulta, nuestra misión. Tu derecho, nuestra prioridad.*"

Este slogan fue pensado y elegido porque refleja el compromiso de brindar información accesible y clara a todas las personas, especialmente en el ámbito de la discapacidad. **"Tu consulta, nuestra misión"** destaca la dedicación de responder a cada pregunta de manera eficiente, asegurando que todos los usuarios reciban la atención que necesitan. **"Tu derecho, nuestra prioridad"** resalta el enfoque en hacer que la información sobre derechos y servicios esté disponible para todos, sin barreras, y de la mejor forma posible. Este slogan refleja la misión que me propuse de garantizar que la información llegue de manera clara, inclusiva y accesible, empoderando a las personas para que puedan hacer valer sus derechos de forma efectiva.

DECISIONES TECNICAS Y SUS JUSTIFICACIONES

FAST-API

Decidí utilizar FastAPI debido a sus ventajas clave en el desarrollo de APIs. Es un framework rápido y eficiente, basado en Python, que permite crear aplicaciones web y APIs con un rendimiento sobresaliente, gracias a su soporte para programación asíncrona (async). Esto lo hace ideal para manejar grandes volúmenes de peticiones de manera rápida y sin complicaciones. Además, FastAPI genera automáticamente documentación interactiva de la API (con Swagger y ReDoc), lo que facilita tanto el desarrollo como las pruebas, ya que no es necesario escribir documentación adicional.

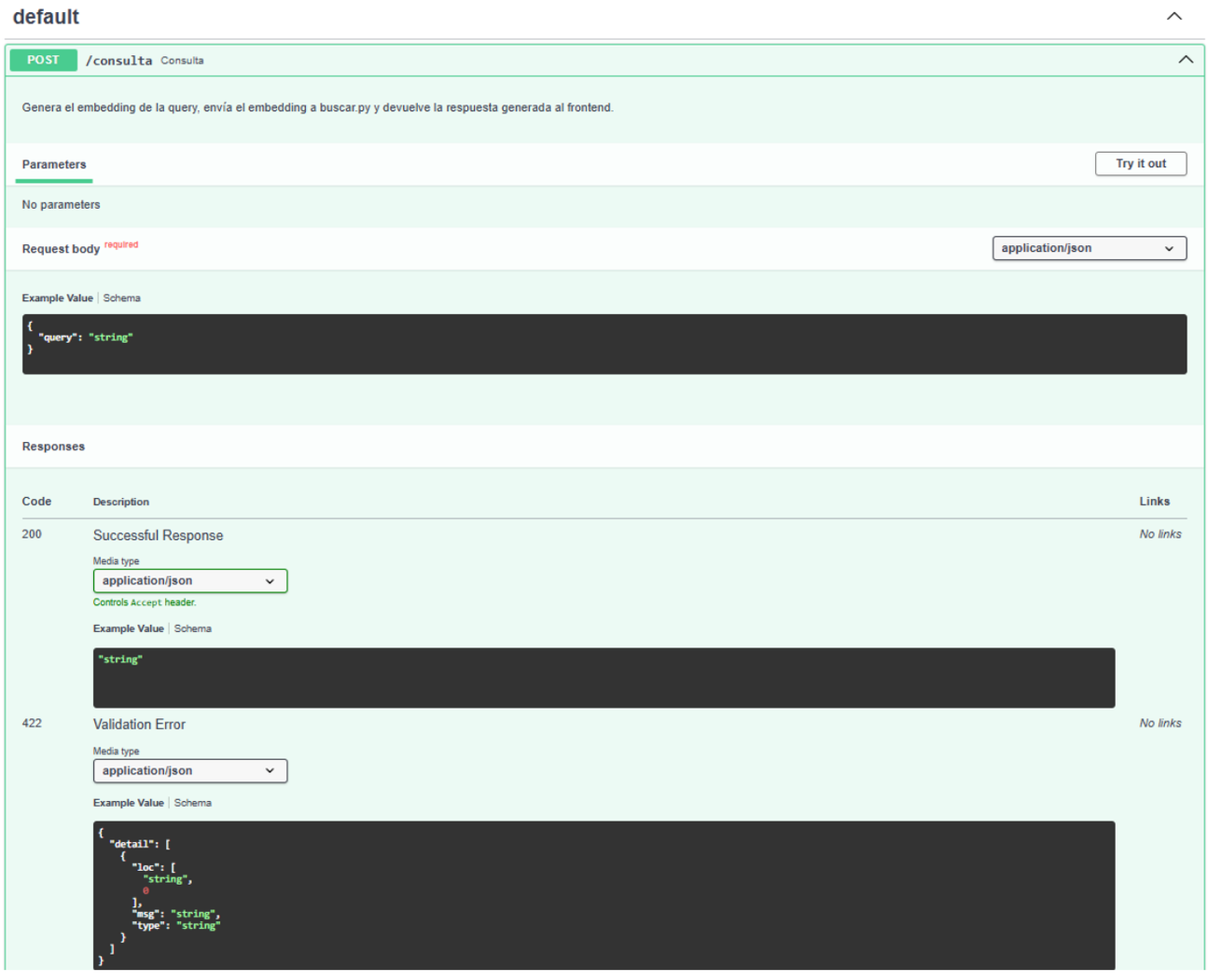
Otra razón importante es que FastAPI es fácil de usar y entender, lo cual me permitió aprovechar los conocimientos que adquirí durante la cursada, donde trabajamos con este framework. Al usar FastAPI, pude aplicar directamente lo aprendido en el curso, lo que hizo el proceso de desarrollo más fluido y eficiente. Además, su diseño modular y flexible me permitió organizar el proyecto de manera clara, facilitando su mantenimiento y escalabilidad a medida que el proyecto avanzaba.

ENDPOINT /CONSULTA

La decisión de realizar una única API, en este caso la de consulta, se basó en la necesidad de priorizar la funcionalidad principal que el usuario utilizaría: realizar consultas y recibir respuestas de manera eficiente. Esta API está diseñada específicamente para cumplir con el objetivo de permitir a los usuarios hacer preguntas y obtener respuestas claras y relevantes. Al centralizar todas las operaciones relacionadas con la consulta en un solo endpoint (/consulta), se simplifica el proceso, tanto desde el lado del desarrollo como en la experiencia del usuario.

Además, al manejar todo en una única API, se optimiza el flujo de trabajo al procesar el embedding de la consulta, buscar la información relevante y generar la respuesta de manera directa y rápida. Esto permite mantener un enfoque más ágil y claro en el desarrollo, sin sobrecomplicar la arquitectura con múltiples endpoints que, en este caso, no son necesarios. La decisión también fue influenciada por la estructura del proyecto, donde el objetivo principal era centrarse en la funcionalidad de consulta para ofrecer al usuario una experiencia fluida y efectiva.

DOCUMENTACION DE LA API



COHERE

Decidí usar **Cohere** para generar los embeddings de la consulta y para que un modelo de LLM genere las respuestas, ya que fue la herramienta que nos enseñaron en clases y sobre la cual hemos estado trabajando. A pesar de tener la limitación de un número limitado de llamadas mensuales, me resultó más práctico crear nuevas cuentas según lo necesitaba, en lugar de optar por soluciones como **Sentence-Transformers**, que requería modificar la versión de Python a una mucho más antigua, lo cual no era ideal para el entorno que estoy utilizando. Además, al usar Cohere, evité posibles problemas de incompatibilidad, ya que diferentes herramientas de embeddings como **Sentence-Transformers** pueden tener arquitecturas distintas, lo que podría generar embeddings diferentes y, en consecuencia, afectar la precisión al recuperar la información relevante.

MODELO LLM UTILIZADO: “command-r-08-2024”

Decidí usar el modelo **"command-r-08-2024"** de **Cohere LLM** debido a que es un modelo optimizado para tareas de generación de texto y consultas en lenguaje natural, lo que se ajustaba perfectamente a los requerimientos de mi proyecto. Este modelo tiene una excelente capacidad para comprender y generar respuestas coherentes y relevantes a partir de las consultas, lo que lo hace ideal para un sistema de preguntas y respuestas como el mío. Además, al haberlo utilizado en proyectos anteriores, ya estaba familiarizado con su funcionamiento, lo que facilitó su integración y aceleró el desarrollo.

Aunque Cohere lanzó una actualización en diciembre, no pude instalarla debido a que generaba errores, por lo que decidí mantener la versión **"command-r-08-2024"**, ya que había demostrado ser estable y efectiva en el contexto de mi proyecto

BASE DE DATOS VECTORIAL: CHROMADB

Decidí usar **ChromaDB** como base de datos vectorial debido a su excelente rendimiento en la gestión y búsqueda de embeddings, lo que la hace ideal para sistemas que requieren búsquedas semánticas rápidas y precisas, como el mío. **ChromaDB** permite indexar, almacenar y recuperar vectores de alta dimensionalidad de manera eficiente, lo que es fundamental para manejar grandes volúmenes de datos de embeddings generados por el modelo de Cohere. Además, su integración sencilla con Python y su capacidad para escalar bien en proyectos de esta naturaleza fueron factores clave. Al ser la base de datos vectorial que nos enseñaron en clase y con la que ya habíamos trabajado en proyectos anteriores, contaba con el beneficio adicional de la familiaridad y la experiencia previa, lo que facilitó la implementación y redujo el tiempo necesario para la puesta en marcha del proyecto. Esta combinación de rendimiento y familiaridad técnica hizo que **ChromaDB** fuera la opción más adecuada para mi sistema.

FORMATO DE LA INFORMACION A PROCESAR

Opté por utilizar un archivo .txt en lugar de un formato JSON debido a que, aunque ambos formatos presentan complicaciones, el uso del archivo de texto resultó más sencillo para este caso. Además, aproveché **pdfplumber** como una herramienta nueva en mi flujo de trabajo para extraer el contenido de los archivos PDF de manera eficiente, convirtiéndolo en un archivo .txt. Este enfoque permitió trabajar con el texto de forma más directa, simplificando su división en chunks y posterior conversión en embeddings para almacenarlos en ChromaDB. Mientras que JSON habría requerido una estructura más compleja para organizar los datos, el .txt ofreció una solución más práctica para este proyecto.

LANGCHAING – “RECURSIVE_CHARACTER_TEXT_SPLITTER”

La decisión de usar RecursiveCharacterTextSplitter de **Langchain** para la división de *chunks* se basa en su capacidad para dividir texto de manera eficiente y flexible, adaptándose a diferentes

tamaños de fragmentos sin perder la coherencia del contenido. Esta herramienta permite dividir textos grandes en trozos más pequeños basados en un número máximo de caracteres, respetando las palabras completas, lo que es crucial para mantener la legibilidad y la calidad del texto. Además, al ser parte de **Langchain**, una biblioteca conocida en el ámbito de procesamiento de lenguaje natural, ofrece una integración fluida con otras herramientas y procesos del proyecto, como la generación de embeddings y el almacenamiento en bases de datos vectoriales como **ChromaDB**.

FRONTEND – HTML, CSS Y JS

Decidí crear una interfaz de usuario con HTML, CSS y JavaScript porque son tecnologías ampliamente utilizadas, accesibles y fáciles de integrar con FastAPI. Este enfoque me permitió desarrollar una interfaz simple, funcional y compatible con cualquier navegador, enfocándome en brindar una experiencia amigable para el usuario. Además, estas herramientas me dieron la flexibilidad necesaria para personalizar el diseño y facilitar la interacción con el sistema de consultas de manera clara y eficiente.

CORS

Se decidió implementar CORS (Cross-Origin Resource Sharing) en el proyecto para permitir que el frontend, alojado en una ubicación diferente al backend, pudiera interactuar de manera segura con la API de FastAPI. Esta configuración es esencial en entornos web modernos, ya que garantiza que las solicitudes entre diferentes dominios se realicen de forma controlada y sin bloquear el acceso necesario para el funcionamiento del sistema. Habilitar CORS fue crucial para integrar el frontend desarrollado con HTML, CSS y JavaScript con el backend, asegurando una comunicación fluida y segura.

CONCLUSION FINAL

El desarrollo de este proyecto ha sido una experiencia profundamente enriquecedora, tanto a nivel técnico como personal. Implementar un sistema basado en la arquitectura RAG para abordar una problemática real como la accesibilidad a información sobre derechos y beneficios para personas con discapacidad en Argentina no solo fue un desafío intelectual, sino también una motivación constante para aportar valor a un tema de gran relevancia social.

Este proceso permitió consolidar y aplicar conocimientos en áreas clave como procesamiento del lenguaje natural, bases de datos vectoriales, y desarrollo de sistemas modulares en Python, demostrando la capacidad de integrar conceptos complejos en una solución funcional. A lo largo del proyecto, gestionar de manera individual cada etapa, desde la planificación y toma de decisiones técnicas hasta la implementación y validación, representó un reto significativo que fortaleció habilidades de organización, resolución de problemas y resiliencia frente a la incertidumbre.

La motivación detrás de este proyecto no solo radicó en el cumplimiento de objetivos académicos, sino también en el impacto positivo que esta solución puede tener en su contexto de aplicación. La posibilidad de ofrecer un sistema eficiente y accesible para responder consultas en lenguaje natural refuerza la idea de que la tecnología puede ser una herramienta poderosa para mejorar la calidad de vida y reducir barreras de acceso.

Desde el punto de vista profesional, el aprendizaje obtenido y la experiencia práctica adquirida en el desarrollo de este sistema representan un salto importante hacia un perfil más sólido y competitivo.

Esta experiencia marca un hito en el crecimiento personal y profesional, inspirando a continuar explorando soluciones tecnológicas que generen impacto y contribuyan al desarrollo de la sociedad.