

Experimentando con API'S-Aplicación personalizada de consulta a ChatGPT

Braian Felipe Ramirez Ortiz
Liseth Tatiana Quilindo Patiño
Fundación Universitaria Konrad Lorenz

Resumen—El presente proyecto tiene la finalidad de integrar todas las técnicas de programación vistas a lo largo del semestre, donde podemos encontrar el uso del API de ChatGPT, API country, la generación de archivos, conexión a base de datos y el uso y desarrollo de interfaces básicas. .

I. INTRODUCCIÓN

El proyecto tiene como objetivo construir un prototipo que permita proveer el índice de desigualdad de diferentes países. Para ello, se realizó una conexión al API <https://restcountries.com>, donde se consumió el endpoint: <https://restcountries.com/v3.1/name/name>, a través del cual se extrajeron los siguientes objetos de respuesta: país, capital, GINI, población, área, continente y subregión. Estos datos extraídos del API se guardaron en un archivo con extensión txt, text o data.

Por otro lado, la interfaz gráfica permite interactuar con la aplicación. Esta cuenta con dos campos para ingresar los nombres de los países a consultar, un botón para generar la búsqueda en el API country, un botón que muestra una gráfica de barras comparando los índices de desigualdad de los dos países consultados, un botón que procesa la información obtenida del API y la guarda en la base de datos SQLite, y un botón que toma la información encontrada en el API y genera un archivo con el nombre del país consultado, el cual contiene toda la información obtenida.

Adicionalmente, se integra un campo para ingresar una consulta relacionada con los países consultados en ChatGPT, y un botón que genera la consulta en el chat.

II. RESULTADOS

II-A. Arquitectura tres capas

Al implementar la arquitectura de tres capas se logra organizar el código en tres subsecciones, las cuales son: la presentación, que contiene los diseños de interfaces y el código correspondiente a cada una de ellas; en la capa de lógica se encuentran almacenadas las clases que se encargan de desarrollar los procesos de búsqueda y extracción de información de las diferentes API empleadas; y en la última capa, llamada persistencia, se encuentran contenidas las clases que generan la conexión y el guardado de información en la base de datos SQLite.

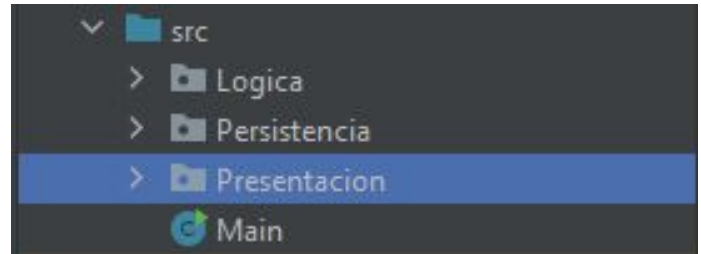


Figura 1. Arquitectura tres capas.

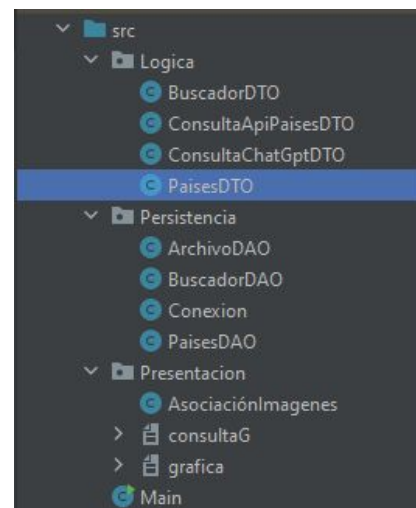


Figura 2. Despliegue arquitectura tres capas.

II-B. Diseño interfaz grafica

Empleando la herramienta de diseño de intelliJ se logra esbozar el diseño de la interfaz principal.

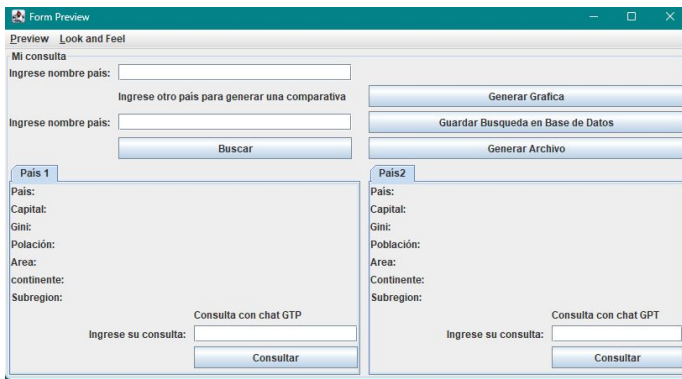


Figura 3. Diseño interfaz.

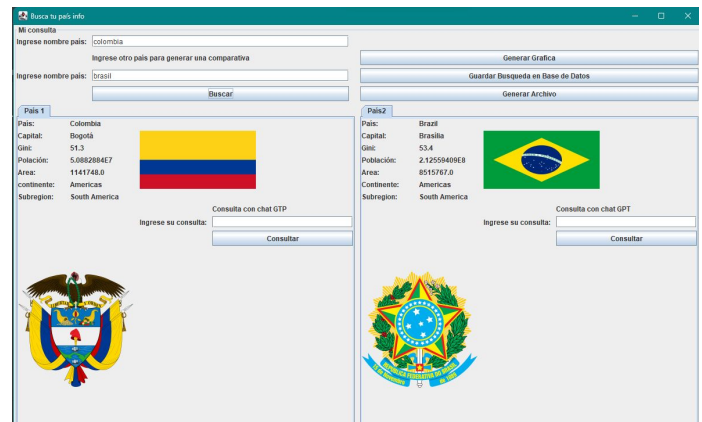


Figura 5. Resultados de la búsqueda.

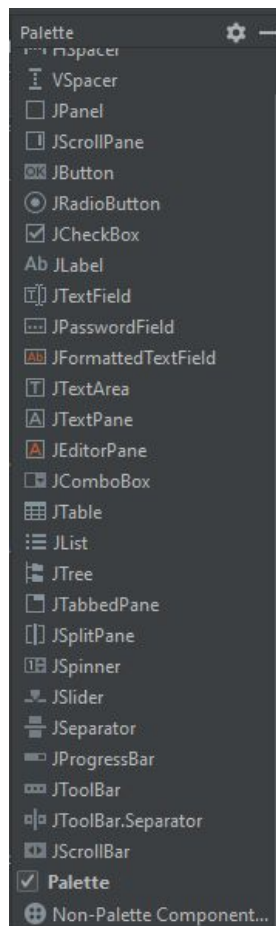


Figura 4. Diseño interfaz.

II-D. Generación de grafica

Con la información extraída del API country se seleccionan dos datos: el nombre del país y el GINI (índice de desigualdad), para generar un gráfico de barras comparativo entre los dos países consultados.

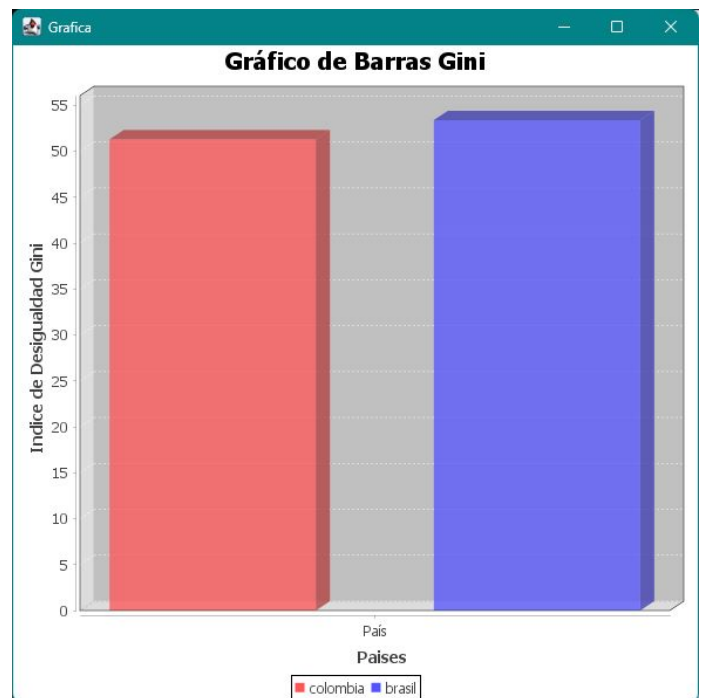


Figura 6. Grafica comparación Gini.

II-C. Obtención de datos de API country

Una vez establecidas las conexiones al API country, se logra obtener la información necesaria del array de objetos de los países consultados. Empleando objetos pertenecientes a cada una de las clases, es posible pasar la información entre ellas mediante métodos que reciben parámetros y la función "get". Con el uso de etiquetas (labels), se facilita la impresión de la información en la interfaz previamente desarrollada.

II-E. Consulta con chatGPT

Generando una conexión con el API de chatGPT, se procedió a crear una contraseña y el pago por el uso del servicio de consulta por token empleado. De este modo de logro traer los resultados de la consulta ingresada.



Figura 7. Consleta con chatGPT.

III. CONCLUSIONES

- La arquitectura de tres capas permite organizar las clases del código según su función, haciendo la lectura del código más limpia y sencilla.
- La integración de las diferentes APIs en el prototipo enriqueció la interacción con el usuario.
- La implementación de librerías externas permite expandir las funciones implementadas en el prototipo y nos permite ofrecer una mejor y más completa experiencia para el usuario.

IV. ANEXOS

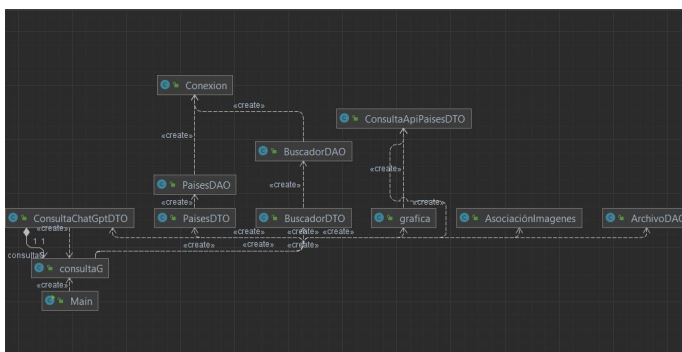


Figura 8. Diagrama de clases.

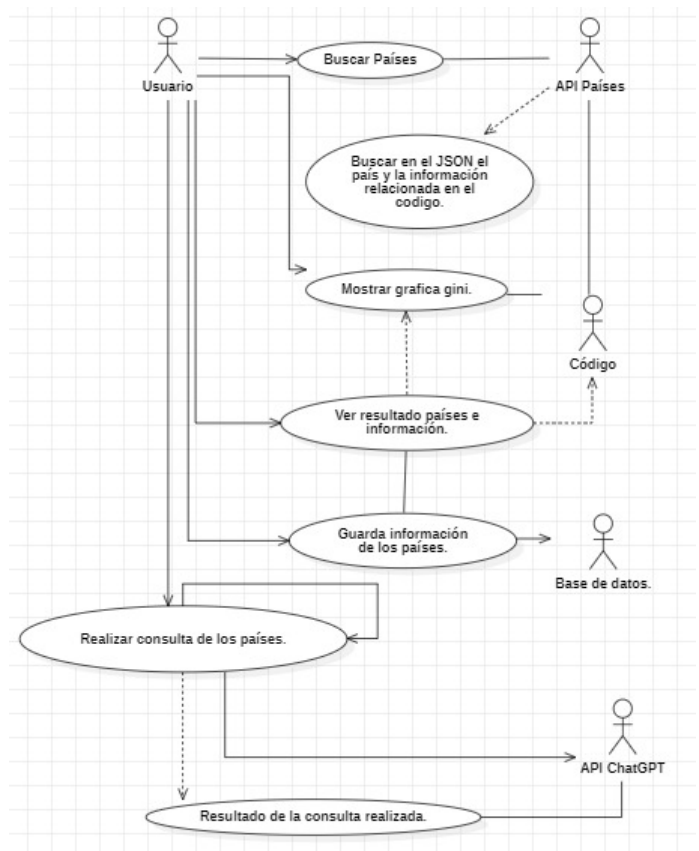


Figura 9. Diagrama de casos de uso.

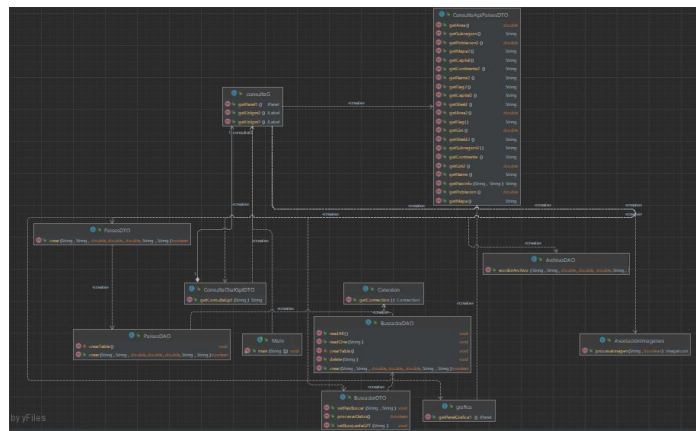


Figura 10. Diagrama de flujo.