
PROYECTO 3

202010910 – Singrid Cristabel Chicoj Martinez

Resumen

Una API es aquella que nos permite que dos sistemas informáticos interactúen entre sí. Por ejemplo, en nuestra aplicación debemos de mostrar las menciones y hashtags que hay al interactuar con nuestro documento xml además de mostrar las interacciones positivas y negativas.

En este caso los sistemas informáticos que interactúan entre si se les conoce como Frontend y Backend con los cuales obtenemos nuestra API.

El Backend es parte o rama del desarrollo web encargada de que toda la lógica de una página funcione y el Frontend es la parte del desarrollo web que se dedica a la parte frontal de un sitio web, en pocas palabras del diseño de un sitio web.

Nuestra aplicación web está compuesta por varias ramas importantes para darle un funcionamiento clave y así amigable al usuario.

Palabras clave

Api

Backend

Fronend

Abstract

Traducir al idioma inglés, el resumen redactado en la columna de la izquierda.

La traducción debe ser revisada con un profesional en ingeniería con amplios conocimientos del idioma inglés, en caso que en forma personal no se posean.

Evitar la utilización del traductor de google u otra similar.

El abstract y las keywords deben abarcar solamente esta columna.

Keywords

Traducción al idioma inglés de las palabras clave.

Introducción

En el mundo digital, la sinergia entre API, frontend y backend es esencial para la creación de aplicaciones y sitios web eficientes.

Una API (Interfaz de Programación de Aplicaciones) actúa como un intermediario que permite que las aplicaciones se comuniquen y compartan datos. El frontend representa la interfaz visible que los usuarios interactúan, mientras que el backend gestiona los procesos y datos en segundo plano.

La API facilita la comunicación entre el frontend y el backend, permitiendo que los usuarios interactúen de manera intuitiva con una aplicación mientras los procesos y la lógica empresarial se gestionan detrás de escena. Esta colaboración fluida entre los tres elementos es fundamental para proporcionar experiencias digitales cohesivas y efectivas.

Desarrollo del tema

API (Interfaz de Programación de Aplicaciones), a menudo considerada el puente que conecta sistemas informáticos, es la clave para permitir la comunicación y la interacción entre diferentes aplicaciones y servicios. Funciona como un traductor de lenguajes, permitiendo que aplicaciones independientes compartan datos y funcionalidades de manera segura y eficiente. Las API son los cimientos sobre los cuales se construyen soluciones tecnológicas más complejas, facilitando la integración de servicios, desde redes sociales hasta sistemas de pago y servicios de mapas.

Frontend, por otro lado, es la cara visible de la experiencia del usuario. Es la parte de la aplicación o el sitio web con la que interactuamos directamente. Los botones, formularios, gráficos y otros elementos visuales que hacen que la aplicación sea atractiva y fácil de usar. Los desarrolladores frontend trabajan

para dar vida a estos elementos en el navegador web o en aplicaciones móviles, utilizando tecnologías como HTML, CSS y JavaScript.

Backend, el tercero en este trío se ocupa del funcionamiento interno de la aplicación. Trabaja en segundo plano para procesar datos, gestionar la lógica de negocio, interactuar con bases de datos y proporcionar los servicios necesarios para que el frontend funcione sin problemas. Los desarrolladores backend utilizan lenguajes de programación y frameworks específicos para crear la infraestructura que respalda la experiencia del usuario.

Entonces la API, frontend y backend forman un equipo indispensable para la creación y el funcionamiento de aplicaciones y sitios web modernos. Esta simbiosis permite que los usuarios naveguen, interactúen y disfruten de experiencias digitales de manera eficiente y atractiva, aprovechando al máximo el potencial de la tecnología.

a. Manejo de archivos Xml

El manejo de un archivo xml puede ser un poco complicado y compleja al principio, pero a través de distintas librerías del lenguaje de Python será más sencilla de manejar.

Con estas librerías podemos leer y escribir el mismo con un código y ejecutarlo, hay que dejar enunciado que hay varias maneras de interpretar un archivo xml, pero en esta solución se ocuparon las siguientes:

Tabla I.

Librería para lectura y escritura XML.

FROM	IMPORT
xml.etree.ElementTree	ET

Fuente: elaboración propia, 2023.

La lectura de nuestro archivo podemos continuar con la solución ya que nuestros datos deberán estar integrados en nuestra lista. En estas listas deberán contener los siguientes datos que son las fechas y el texto además de las palabras positivas y negativas.

Con nuestras listas enlazadas la información obtenida del archivo xml y eso es fundamental para realizar las validaciones del problema propuesto, en este hay que manejar nuestros datos como si fuera una matriz. Al momento de realizar nuestras listas y obtener toda aquella información necesaria que son las fechas y palabras de nuestro archivo de entrada tenemos que mostrar un resultado de nuestros datos en un archivo xml de salida.

b. Frontend

El estilo de nuestra aplicación se debe al frontend en este debemos de tener botones además de áreas de texto para las respuestas obtenidas. En el frontend hay opciones con las cuales vamos a identificar lo que el usuario necesite en este programa hay tres opciones estas son “cargar archivo de mensajes”, “cargar archivo de configuraciones” y “peticiones”. En la opción de peticiones tenemos otras sub - opciones en las cuales el backend se encarga ejecutarlo y el frontend de devolverlo.

c. Backend

El backend es el encargado del algoritmo y la lógica de dicho programa además del servicio del API que brindara los servicios utilizados en HTTP, la funcionalidad del backend es procesar los datos recibidos en el frontend para luego almacenar los datos en listas y así utilizar los métodos necesarios para consumir nuestra API.

Para consultar los hashtags en nuestro backend se utilizo un algoritmo sencillo utilizando una lógica básica para nuestras listas y al extraer el texto de nuestra entrada se debió de comparar cada carácter

para encontrar el correcto “#” ya que este esta en medio por lo que es sencillo reconocer que al momento de leer un # y al momento de terminar con el mismo “#” el texto en medio de ambos es un hashtag y se puede concluir con la cadena de texto.

Ahora bien, con las menciones debemos de saber que una mención inicia con el carácter “@” y al momento de encontrar un espacio termina ya que este con la diferencia del hashtag no esta en medio del carácter por lo que cada vez que encuentre un “@” eso significa que es una mención a un usuario.

Y finalizamos con la consulta de nuestros sentimientos en los mensajes recibidos estos tienen tres opciones que son “positivos”, “negativos” y “neutros”, esto significa que al interactuar el mensaje con nuestra lista de sentimientos y al momento de encontrar una palabra que está dentro de la lista de positivos significa que esta es una palabra positiva usando la misma lógica para las palabras negativas, y si en dado caso la palabra se encuentra tanto en negativa como positiva significa que es una palabra neutra además de tener palabras positivas y negativas rechazadas por el archivo de entrada.

d. Graphviz

Graphviz es una manera de representar de forma gráfica una estructura de datos, para que el usuario pueda tener una mejor vista del resultado luego de cargar el archivo de entrada al igualmente al procesar el archivo se genera una gráfica.

Este punto es un poco complicado ya que durante la búsqueda no hay demasiada información para generar una gráfica específica, pero si hay maneras de encontrar diseños y estilos para conseguir una gráfica.

e. Datos de estudiante

Esta opción de nuestro menú principal muestra un pequeño botón de “Ayuda” donde devuelve en el área de texto los datos del estudiante que soluciono el problema propuesto durante el proyecto, indica tanto nombre, carnet, curso, sección y semestre al que pertenece además de mostrar una dirección web el cual muestra la documentación del mismo algoritmo.

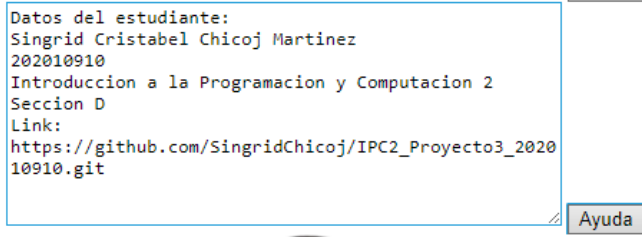


Figura 1. Datos de estudiante y dirección web.

Fuente: elaboración propia, 2023.

f. Inicializar Listas

Para inicializar nuestras listas puede parecer una tarea bastante sencilla y así es, pero al mismo tiempo se debe saber desde que clase inicializar esta misma esto se refiere que las listas deben de tener el método clear para poder limpiar todas las listas utilizadas.

g. EndPoints

Los EndPoints son URLs específicas que representan recursos o servicios dentro de una API. Cada EndPoint está asociado con una operación específica, como GET, PUT, POST o DELETE. Los EndPoints son utilizados para acceder a recursos o realizar acciones en nuestra API. Cada EndPoint debe ser único y se utiliza para interactuar con una parte específica de la API.

h. PUT

El método POST es uno de los principales métodos utilizados en las API y se utiliza para enviar datos al servidor con el fin de crear un nuevo recurso o realizar una acción que modifique el estado del

servidor. Cuando realizas una solicitud POST a través de una API, generalmente envías datos en el cuerpo de la solicitud en lugar de adjuntarlos a la URL. Esto permite enviar información más extensa y compleja, como formularios web o datos de creación de nuevos registros.

El método POST es esencial para la funcionalidad interactiva en aplicaciones y sitios web, ya que permite a los usuarios enviar datos, realizar transacciones y realizar cambios en el servidor. En resumen, el método POST en una API se utiliza para enviar datos al servidor con el propósito de crear nuevos recursos o realizar acciones que afecten el estado del servidor, como la creación de registros, envío de formularios y más.

Por ejemplo, en nuestra API utilizamos la solicitud POST para grabar los mensajes, configuración y limpiar los datos de las listas. Los datos enviados en la solicitud POST son procesados por el servidor, que crea un nuevo recurso o realiza la acción especificada.

i. GET

El método GET es una de las principales operaciones utilizadas en las API. Se utiliza para solicitar información o recursos de un servidor. Cuando realizas una solicitud GET a un servidor a través de una API, estás pidiendo datos específicos, como un artículo de un catálogo, información de usuario o cualquier otro recurso disponible. Los parámetros pueden ser enviados a través de la URL, lo que permite a la API identificar y responder a la solicitud apropiadamente. Por ejemplo, en nuestra aplicación debemos de obtener los hashtags y menciones que hay en nuestro archivo de entrada además de obtener la información de los mensajes utilizados para las palabras claves en los sentimientos recibidos.

En resumen, GET se usa para recuperar información, POST para crear datos y los EndPoints son las rutas específicas que representan los recursos o servicios en una API. Estos conceptos son fundamentales para entender cómo funcionan y cómo interactuar con las API en el desarrollo de aplicaciones.

Promover la reflexión que conduzca a la adopción de una u otra postura. En conjunto, GET y POST son métodos HTTP utilizados para solicitar y enviar datos a través de una API.

Conclusiones

La sinergia entre API, frontend y backend es fundamental en la creación de aplicaciones y sitios web exitosos. Las API permiten la comunicación y el intercambio de datos, el frontend proporciona la experiencia del usuario, y el backend gestiona la funcionalidad subyacente. Estos elementos trabajan en conjunto para brindar experiencias digitales cohesivas y efectivas, abriendo las puertas a una amplia gama de aplicaciones y servicios que hacen que la tecnología sea accesible y útil para usuarios de todo el mundo. La comprensión de esta colaboración es esencial para cualquier desarrollador o equipo que busque crear soluciones tecnológicas efectivas y atractivas.

Las API, junto con los métodos POST, GET y los EndPoints, son componentes esenciales en el mundo de la programación y el desarrollo de aplicaciones. Permiten la comunicación entre sistemas, la transferencia de datos y la interacción con servicios en línea. POST y GET desempeñan roles cruciales al permitir que las aplicaciones envíen y reciban información, y los EndPoints son las puertas de entrada que hacen posible el acceso a recursos específicos en una API.

La comprensión de estos conceptos es fundamental para construir aplicaciones exitosas y para aprovechar el poder de la tecnología en la creación de soluciones digitales efectivas.

Referencias bibliográficas

Agencia Marketing y Desarrollo Web. (2021). *¿Qué es Back End, Front End y Back Office y por qué es importante para tu web?* Nestrategia. <https://nestrategia.com/desarrollo-web-back-end-frontend/#:~:text=En%20otras%20palabras%2C%20el%20Back,la%20comunicación%20con%20el%20servidor.>

Ander Fernández Jauregui. (2023). *Cómo crear una API en Python.* <https://anderfernandez.com/blog/como-crear-api-en-python/>

Iván Jahel Bautista García. (2021). *Backend y Frontend, ¿Qué es y cómo funcionan en la programación?* Servnet. <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>

Apéndices

- Arquitectura general de la aplicación

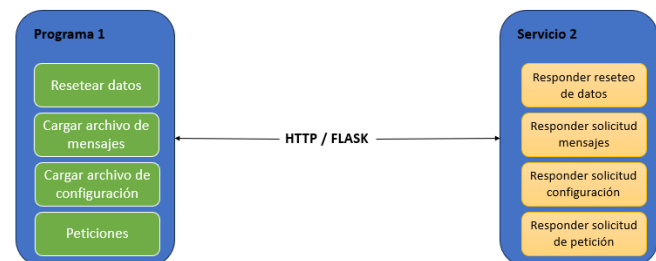


Figura 2. Arquitectura para frontend y backend.

Fuente: Universidad de San Carlos de Guatemala, Enunciado Proyecto 03, página 05, 2023.

- Sugerencia de la estructura de un API

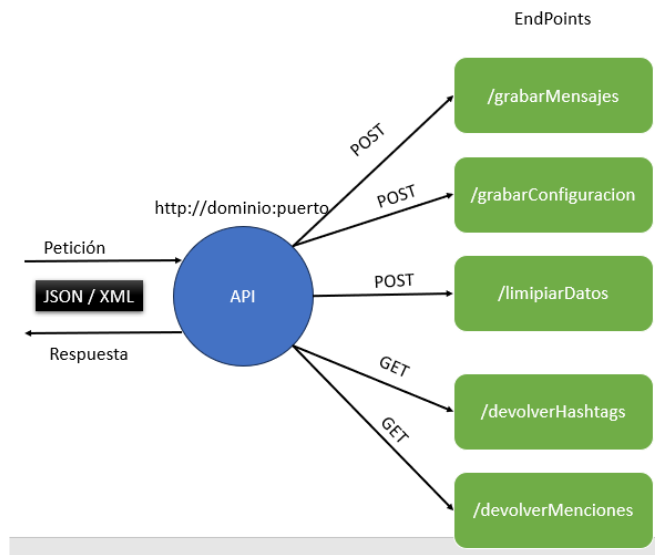


Figura 3. Métodos y EndPoints recomendados.

Fuente: Universidad de San Carlos de Guatemala, Enunciado Proyecto 03, página 07, 2023.

- Estructura xml antes de ejecutar los procesos designados

```
<?xml version="1.0"?>
<MENSAJES>
  <MENSAJE>
    <FECHA> Guatemala, 15/01/2023 15:25 hrs. </FECHA>
    <TEXTO> Bienvenido a USAC @estudiante01 @estudiante02,
            es un gusto que seas parte de esta institución
            #bienvenidaUSAC#
    </TEXTO>
  </MENSAJE>
  ...
</MENSAJES>
```

```
<?xml version="1.0"?>
<diccionario>
  <sentimientos_positivos>
    <palabra> bueno </palabra>
    <palabra> excelente </palabra>
    <palabra> cool </palabra>
    <palabra> satisfecho </palabra>
    ...
  </sentimientos_positivos>
  <sentimientos_negativos>
    <palabra> malo </palabra>
    <palabra> pésimo </palabra>
    <palabra> triste </palabra>
    <palabra> molesto </palabra>
    <palabra> decepcionado </palabra>
    <palabra> enojo </palabra>
    ...
  </sentimientos_negativos>
</diccionario>
```

Figura 4. Estructura xml de entrada.

Fuente: Universidad de San Carlos de Guatemala, Enunciado Proyecto 03, página 03, 2023.

- Diagrama de clases que muestra el flujo de la aplicación

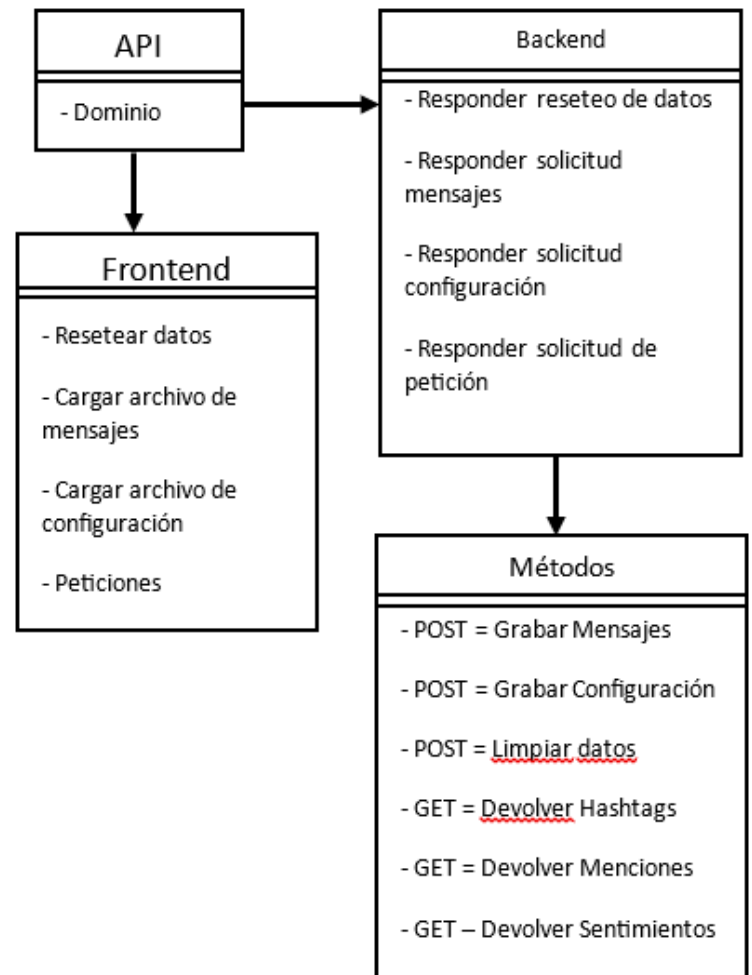


Figura 5. Diagrama de clases
Fuente: elaboración propia, 2023.