
PROYECTO 3

202110509 – Mario Ernesto Marroquín Pérez

Resumen

Se ha creado una nueva red social llamada ChapinChat. Para analizar el contenido de la nueva red social y establecer perfiles de los usuarios se contrató a la empresa Tecnologías Chapinas S.A.

La empresa Tecnologías Chapinas creó un servicio web que será consumido por la red social ChapinChat. La estrategia consiste en construir un sistema de información que reciba mensajes, los analice y establezca por cada usuario el porcentaje de contenido que está asociado a cada perfil.

Para la solución a esta problemática, se desarrolló un programa que permite recibir el mensaje, almacenar la información necesaria en formato XML, calcular el peso de los perfiles definidos en el sistema por los usuarios y posteriormente emitir reportes y realizar consultas. Se desarrolló un programa 1 (frontend utilizando Django) y un programa 2 (backend utilizando Flask).

Palabras clave

API: Abreviatura de Application Programming Interfaces. Conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones.

XML: Extensible Markup Language por sus siglas en inglés, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.

Abstract

A new social network called ChapinChat has been created. The company Tecnologías Chapinas S.A. was hired to analyze the content of the new social network and to establish user profiles.

The company Tecnologías Chapinas created a web service that will be consumed by the ChapinChat social network. The strategy consists of building an information system that receives messages, analyzes them and establishes for each user the percentage of content that is associated with each profile.

To solve this problem, a program was developed to receive the message, store the necessary information in XML format, calculate the weight of the profiles defined in the system by the users and subsequently issue reports and perform queries. A program 1 (frontend using Django) and a program 2 (backend using Flask) were developed.

Keywords

API: Abbreviation for Application Programming Interfaces. A set of definitions and protocols used to develop and integrate application software..

XML: Extensible Markup Language is a markup language that defines a set of rules for document encoding.

Introducción

Utilizado la potencia de Python como lenguaje de programación, se desarrolló una aplicación web que se comunica con un servidor mediante el framework Flask, el cual nos ha permitido manejar las solicitudes HTTP y la gestión de los recursos de nuestra aplicación.

Flask es un framework web ligero que permite construir aplicaciones web en Python de manera sencilla y con poca complejidad. Flask es especialmente adecuado para proyectos pequeños y medianos, gracias a su simplicidad y facilidad de uso.

Por otro lado, Django es un framework web completo y robusto que se utiliza ampliamente para el desarrollo de aplicaciones web complejas y escalables. Django cuenta con un conjunto completo de características que facilitan el desarrollo web, como la administración de bases de datos, la autenticación de usuarios, la creación de formularios, la gestión de sesiones y más.

Utilizando Flask para el backend y Django para el frontend, hemos combinado la simplicidad y la facilidad de uso de Flask con la potencia y la funcionalidad de Django para crear una aplicación web que cumple con los requisitos de escalabilidad, seguridad y mantenimiento.

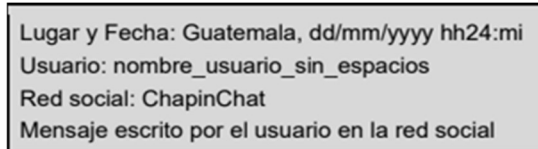
Desarrollo del tema

a. Descripción del Problema

La empresa Tecnologías Chapinas, S.A. ha sido contratada por la nueva red social ChapinChat para desarrollar una herramienta que sea capaz de analizar el contenido de dicha red social y establecer perfiles

de sus usuarios que favorezcan conocer mejor a dichos usuarios y brindar contenido más personalizado a éstos.

Para lograr este fin, la empresa ChapinChat generará los mensajes de su red social con una estructura de texto con el siguiente formato:



```
Lugar y Fecha: Guatemala, dd/mm/yyyy hh24:mi
Usuario: nombre_usuario_sin_espacios
Red social: ChapinChat
Mensaje escrito por el usuario en la red social
```

Figura 1. Mensaje generado por ChapinChat.

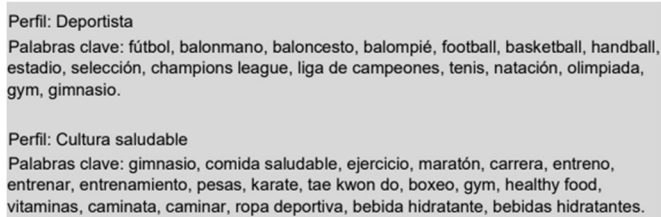
Fuente: Enunciado Proyecto 3 IPC2, 2023.

La empresa Tecnologías Chapinas, S.A. creará un servicio web que será consumido por la red social ChapinChat que enviará el mensaje descrito en la Figura No. 1.

La estrategia de Tecnologías Chapinas, S.A. consiste en construir un sistema de información que reciba estos mensajes, analice el contenido del mensaje y establezca por cada usuario el porcentaje de contenido que está asociado a cada perfil que haya sido configurado en este sistema. Los perfiles serán conjuntos de palabras clave, que, al aparecer en el mensaje, permitirán calcular el porcentaje de palabras en relación al total de palabras que corresponden a dicho perfil. Para lograr una mayor exactitud, Tecnologías Chapinas S.A., también define un listado de palabras que no será considerado como palabra clave a analizar.

La empresa Tecnologías Chapinas define un perfil como un conjunto de palabras que pueden representar intereses del usuario que pueden ser de utilidad para

ChapinChat, estos perfiles serán dinámicos y podrán cambiar en el tiempo, según los intereses de ChapinChat.



Perfil: Deportista
Palabras clave: fútbol, balonmano, baloncesto, balompié, football, basketball, handball, estadio, selección, champions league, liga de campeones, tenis, natación, olimpiada, gym, gimnasio.

Perfil: Cultura saludable
Palabras clave: gimnasio, comida saludable, ejercicio, maratón, carrera, entreno, entrenar, entrenamiento, pesas, karate, tae kwon do, boxeo, gym, healthy food, vitaminas, caminata, caminar, ropa deportiva, bebida hidratante, bebidas hidratantes.

Figura 2. – Ejemplo de perfiles definidos por Tecnologías Chapinas, S.A.

Fuente: Enunciado Proyecto 3 IPC2, 2023.

b. Conceptos Efectuados en la Creación del Programa.

API:

Una API (Application Programming Interface) es un conjunto de reglas, protocolos y herramientas que se utilizan para construir software y aplicaciones. Básicamente, una API permite que diferentes aplicaciones interactúen entre sí de forma programática, lo que significa que los desarrolladores pueden utilizar la API para enviar y recibir datos entre sus propias aplicaciones y otras aplicaciones que también utilicen la misma API.

Una API define los puntos de entrada y salida de un sistema, lo que permite que los desarrolladores utilicen los servicios de una aplicación o plataforma específica sin tener que conocer todos los detalles técnicos de su funcionamiento interno. Por ejemplo, las APIs pueden utilizarse para integrar la funcionalidad de un servicio de redes sociales en una aplicación móvil, o para acceder a información de una

plataforma de comercio electrónico desde un sistema de gestión de inventarios.

Las APIs permiten que diferentes aplicaciones se comuniquen entre sí y compartan información de manera eficiente, lo que resulta en una mejor experiencia de usuario y en una mayor eficiencia en el desarrollo de software.

Framework:

Un framework, en el contexto de la programación, es un conjunto de herramientas, bibliotecas, convenciones y patrones de diseño que se utilizan para desarrollar software de manera más rápida y eficiente.

Un framework proporciona una estructura para el desarrollo de aplicaciones, que permite a los desarrolladores centrarse en la lógica de negocio en lugar de tener que preocuparse por los detalles técnicos de la infraestructura. Al utilizar un framework, los desarrolladores pueden aprovechar las funcionalidades preexistentes, como la gestión de la base de datos, el enrutamiento de solicitudes, la seguridad, entre otros, lo que les permite desarrollar aplicaciones de manera más rápida y con mayor calidad.

Cada framework tiene su propia arquitectura y características, y se utilizan en diferentes áreas de la programación, desde aplicaciones web hasta aplicaciones móviles y de escritorio.

Frontend:

El frontend (o también conocido como "front-end") se refiere a la parte de una aplicación o sitio web que interactúa directamente con los usuarios finales. Se trata de la interfaz de usuario que los usuarios ven y con la que interactúan.

Su objetivo es proporcionar una experiencia de usuario atractiva e interactiva. Los desarrolladores frontend se encargan de diseñar y desarrollar la interfaz de usuario y de asegurarse de que sea fácil de usar y accesible para todos los usuarios.

Backend:

El backend (o también conocido como "back-end") es la parte de una aplicación o sitio web que se encarga del procesamiento y almacenamiento de datos. Se trata de la capa de la aplicación que está oculta al usuario final y se encarga de manejar las solicitudes y la lógica del negocio detrás de una aplicación.

Se compone de lenguajes de programación, frameworks y bibliotecas, y su objetivo es proporcionar una funcionalidad eficiente y segura para la aplicación. Los desarrolladores de backend se encargan de diseñar y desarrollar la lógica del negocio detrás de una aplicación y de asegurarse de que la aplicación sea escalable, segura y fácil de mantener.

XML:

XML (eXtensible Markup Language) es un lenguaje de marcado de texto que permite representar y almacenar información estructurada de manera legible para las personas y las máquinas. XML se utiliza para definir y describir los datos, y su estructura se basa en etiquetas que describen los diferentes elementos de la información.

XML es un estándar abierto que permite a los desarrolladores crear sus propios formatos de archivo personalizados y es ampliamente utilizado para el intercambio de datos entre sistemas. Por ejemplo, se utiliza a menudo para representar documentos, datos

de configuración, mensajes en aplicaciones de mensajería, entre otros.

XML es una herramienta versátil y valiosa para la representación y el almacenamiento de información estructurada en la programación y la tecnología de la información.

Programación Orientada a Objetos:

La programación orientada a objetos (POO) es un paradigma de programación que se basa en el concepto de "objetos". Un objeto es una entidad que combina datos y comportamientos, y se puede pensar en un objeto como una instancia de una clase.

En POO, se modelan los problemas y las soluciones en términos de objetos y clases. Una clase es una plantilla o definición que describe los datos y los comportamientos comunes de un conjunto de objetos. Cada objeto se crea a partir de una clase y hereda todos los datos y comportamientos definidos en la clase.

La POO tiene varias ventajas, incluyendo una mayor legibilidad y mantenibilidad del código, una mejor organización y reutilización de código, y una mayor capacidad para trabajar con soluciones complejas y modulares.

Algunos de los conceptos clave de la POO incluyen la encapsulación, la herencia y el polimorfismo. La encapsulación permite proteger los datos y los comportamientos de un objeto, mientras que la herencia permite reutilizar y ampliar código existente. El polimorfismo permite que los objetos de diferentes clases respondan de manera diferente al mismo mensaje o acción.

c. Construcción del Programa

El programa cuenta con dos partes importantes, el frontend y el backend.

Frontend:

En esta aplicación se podrán mostrar los eventos que se procesarán y los datos estadísticos que fueron almacenados en la base de datos XML de salida.

Se utilizó el framework Django, el cual trabaja con el patrón MVT (Modelo-Vista-Template).

Entre los componentes y opciones que existen en el frontend están:

- Solicitar Servicio:

Se despliega una pantalla para gestionar el envío de solicitudes a las APIs de configuración y creación de mensajes, así como sus respectivas respuestas; dependiendo del servicio elegido se deberá preparar un mensaje XML.

- Peticiones:

Dentro de este apartado se encuentran las siguientes opciones:

- Detalle de mensajes por usuario:

Se puede elegir una fecha por la cual se requiere filtrar y si se desea incluir un usuario o todos los usuarios, entonces muestra un resumen de mensajes por usuario identificando los perfiles y el porcentaje de probabilidad de que el perfil aplique al usuario. Este reporte podrá ser descargado en formato PDF si el usuario lo desea.

- Resumen de pesos por usuario:

Al seleccionar esta opción se podrá elegir si desea ver un usuario o todos los usuarios. Se presenta una tabla por cada usuario mostrando el peso

correspondiente a cada perfil configurado en el sistema. Este reporte podrá ser descargado en formato PDF si el usuario lo desea.

- Prueba de solicitudes de creación de mensajes:

Esta opción permite enviar una solicitud con un mensaje y obtener una respuesta.

- Ayuda:

Se despliegan dos opciones, en una se podrá visualizar la información del estudiante y en la otra opción se podrá visualizar la presente documentación.

- Reset:

Esta opción mandará la instrucción correspondiente a la Api para devolver al estado inicial la Base de Datos, es decir, sin datos de configuración ni mensajes.

Backend:

Este servicio consiste en una API que brindará servicios utilizando el protocolo HTTP, su funcionalidad principal es procesar los datos recibidos, luego de procesar los datos estos son almacenados en uno o varios archivos xml. Este servicio también tiene la funcionalidad de devolver los datos que fueron almacenados para que sean mostrados en el Frontend.

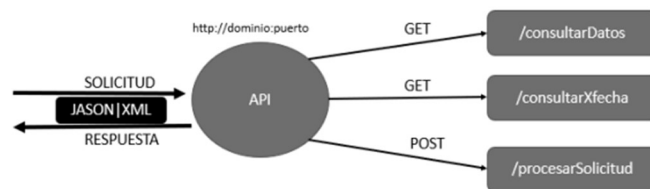


Figura 3. Ejemplo estructura de un API.

Fuente: Enunciado Proyecto 3 IPC2, 2023.

d. Diagrama de Clase del Programa

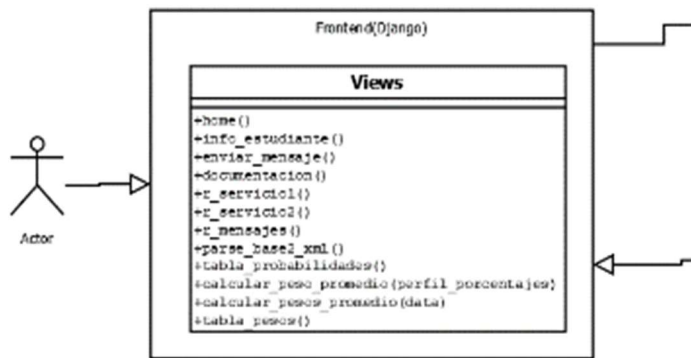


Figura 4. Parte 1 Diagrama de clase.

Fuente: Elaboración propia, 2023.

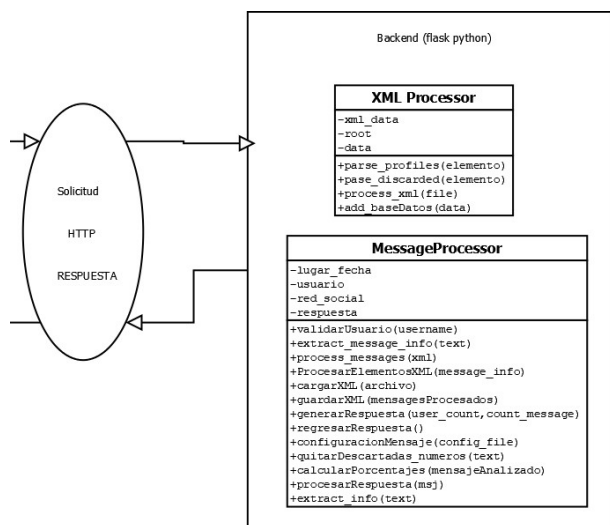


Figura 5. Parte 2 Diagrama de clase.

Fuente: Elaboración propia, 2023.

aplicaciones. Los frameworks como Flask y Django ofrecen una amplia variedad de características y herramientas para el desarrollo web, lo que hace que el proceso de desarrollo sea más rápido, eficiente y escalable.

- El uso de la Programación Orientada a Objetos facilita el almacenamiento de datos en memoria de las listas enlazadas al proporcionar una forma clara y organizada de representar y manipular los datos. Esto puede mejorar la eficiencia y la mantenibilidad del código, lo que a su vez puede aumentar la productividad y la calidad del software.
- El uso de XML es fundamental en muchos aspectos de la industria tecnológica, incluyendo la integración de sistemas, la transmisión de datos y la representación de información estructurada. Su facilidad de uso y flexibilidad lo convierten en una herramienta valiosa para los desarrolladores y los profesionales de la tecnología.

Referencias bibliográficas

Amazon.(2018).*¿Qué es XML? - Explicación de XML - AWS*. Amazon Web Services, Inc.
<https://aws.amazon.com/es/what-is/xml/>

AppMaster. (2022). *¿Qué es un XML?* AppMaster - ultimate all-in no-code platform.
<https://appmaster.io/es/blog/que-es-xml>

Conclusiones

- El uso de frameworks es una práctica fundamental en el desarrollo de software moderno, ya que proporcionan una base sólida y estructurada para el desarrollo de

García, I. J. B. (2021, 30 marzo). *Backend y Frontend, ¿Qué es y cómo funcionan en la programación?*

<https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>

Muñoz, J. D. (2020, 22 diciembre). *Qué es Flask*. OpenWebinars.net.

<https://openwebinars.net/blog/que-es-flask/>

School, T. (2023, 8 marzo). *¿Qué es Django y para qué se utiliza?* Tokio School.

<https://www.tokioschool.com/noticias/que-es-django/>

Anexos

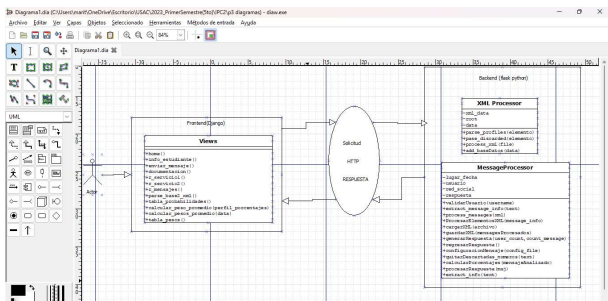


Figura 6. Creación de los diagramas de clases.

Fuente: Elaboración propia, 2023.

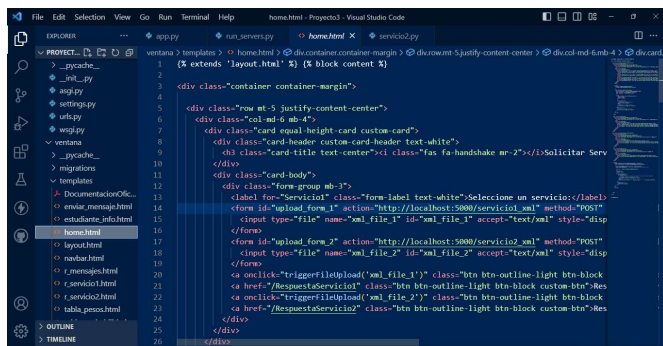


Figura 7. Plantilla HTML home.

Fuente: Elaboración propia, 2023.

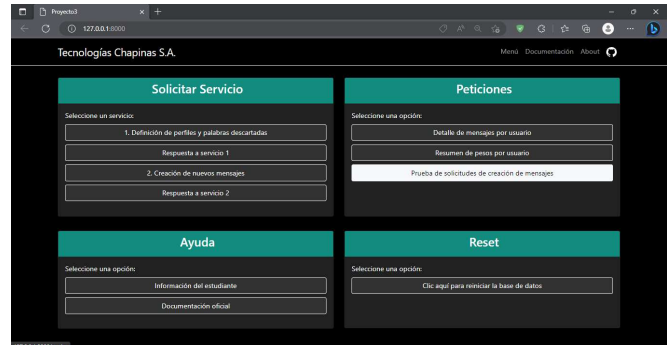


Figura 8. Pantalla principal.

Fuente: Elaboración propia, 2023.

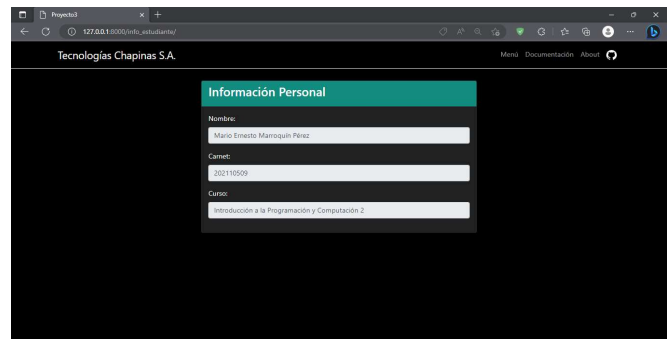


Figura 9. Ventana información estudiante.

Fuente: Elaboración propia, 2023.

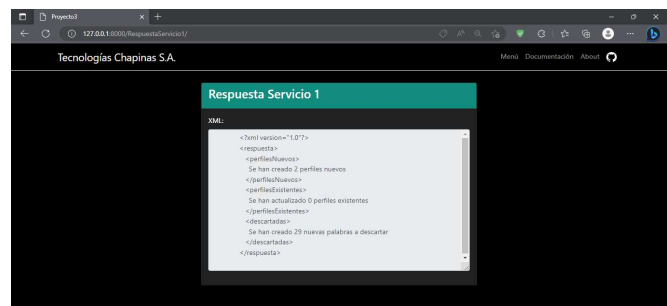


Figura 10. Ventana respuesta servicio 1.

Fuente: Elaboración propia, 2023.

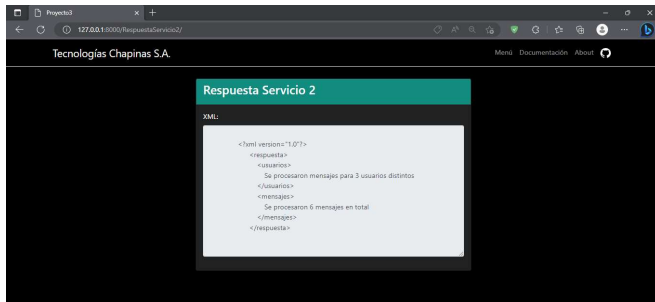


Figura 11. Ventana respuesta servicio 2.

Fuente: Elaboración propia, 2023.

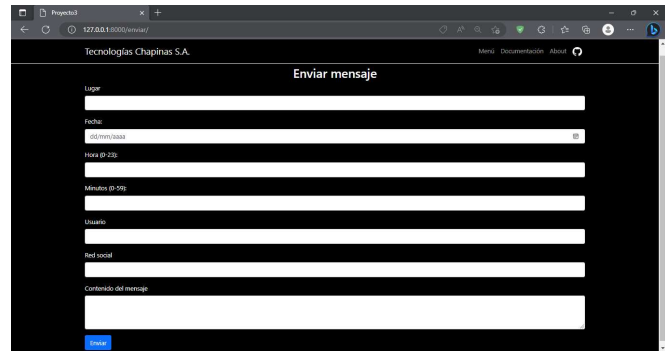


Figura 14. Formulario para mensajes de prueba.

Fuente: Elaboración propia, 2023.

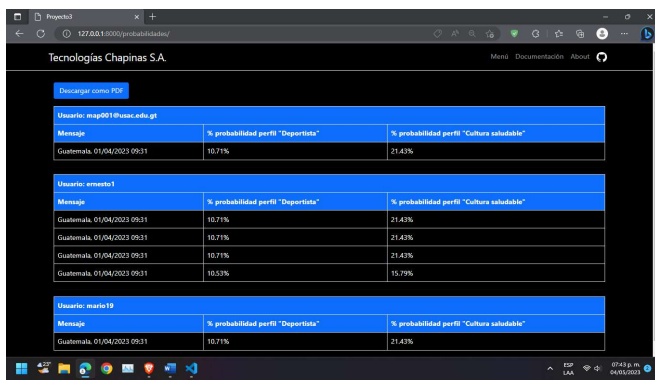


Figura 12. Ventana probabilidad de perfiles.

Fuente: Elaboración propia, 2023.

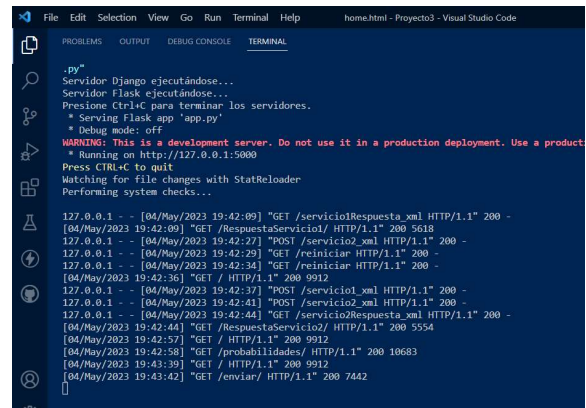


Figura 15. Peticiones HTTP.

Fuente: Elaboración propia, 2023.

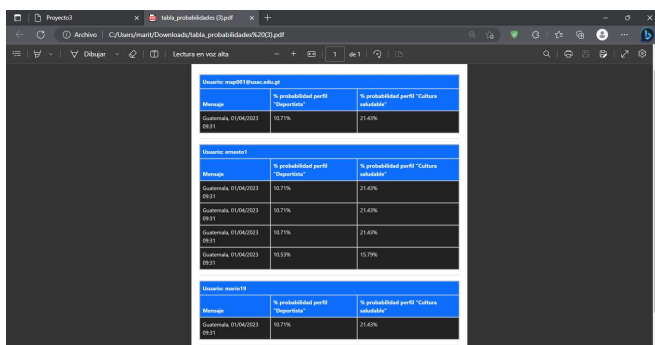


Figura 13. Generación de PDF.

Fuente: Elaboración propia, 2023.

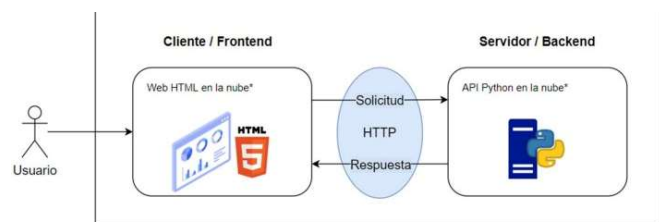


Figura 16. Ejemplo diagrama de clases.

Fuente: Elaboración propia, 2023.