

**Universidad San Carlos de Guatemala**  
**Facultad de Ingeniería**  
**Escuela de Ciencias y Sistemas**  
**Introducción a la Programación y Computación 2**

**Inga. Claudia Liceth Rojas Morales**  
**Ing. Marlon Antonio Pérez Türk**  
**Ing. Byron Rodolfo Zepeda Arevalo**  
**Ing. José Manuel Ruiz Juárez**  
**Ing. Edwin Estuardo Zapeta Gómez**

**Tutores de curso:**  
**Diego Pérez Alvarez**  
**Marvin Daniel Rodríguez**  
**Oscar Velásquez Leon**  
**Daniel Arturo Alfaro**  
**Sergio Felipe Zapeta**



## **PROYECTO 3**

### **OBJETIVO GENERAL**

Desarrollar una solución integral que implemente un API que brinde servicios utilizando el Protocolo HTTP bajo el concepto de programación orientada a objetos (POO) y el uso de bases de datos.

### **OBJETIVOS ESPECÍFICOS**

- Implementar un API a través de lenguaje Python que pueda ser consumida utilizando el protocolo HTTP.
- Utilizar el paradigma de programación orientada a objetos para construir software.
- Utilizar bases de datos para almacenar información de forma persistente.
- Utilizar archivos XML como insumos para la comunicación con el API desarrollado.
- Utilizar expresiones regulares para extraer contenido de texto.

## ENUNCIADO

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

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.

La figura 2 muestra ejemplos de Perfiles, un perfil determina una lista con “N” palabras clave que identifican personas con este perfil de interés. Las palabras claves pueden ser una palabra simple, o bien, estar formadas por varias palabras (Ej. comida saludable). Las palabras pueden ser compartidas por uno o más perfiles.

#### Palabras Excluidas

Palabras: a, un, una, en, para, por, que, qué, la, las, los, el, unas, si, no, sino, entre, otro, otra, otros, otras, de, del, nos, sus, su, am, pm.

Figura 3 – Ejemplo de palabras excluidas definidas por Tecnologías Chapinas, S.A.

La figura 3 muestra las palabras excluidas del algoritmo que desarrollará Tecnologías Chapinas, S.A. El objetivo de estas palabras es que no sean consideradas para establecer el porcentaje de palabras que se asocian a un perfil. Adicionalmente, todos los valores numéricos (1, 40, 33, 50.33, etc.) son excluidos por default.

#### Ejemplo de cálculo de porcentaje de probabilidad por perfil en un mensaje:

Dadas las definiciones de las figuras 2 y 3, y el siguiente mensaje:

Hola amigos, **nos** vemos hoy **en el** gym... recuerden **que** después vamos **a** entrenar **para la** carrera 2K **del** próximo sábado. **No** olviden **su** Ropa Deportiva y **sus** bebidas Hidratantes. Recuerden **que** hoy **por la** noche juega **la** selección **de** fútbol, **nos** vemos **en** Taco Bell **a las 7 pm**.

El párrafo anterior tiene en total 50 palabras, sin embargo, resaltado en amarillo hay palabras excluidas, resaltado en verde hay valores numéricos que por default están excluidos. Por lo tanto, para hacer el análisis de perfilado del mensaje para el usuario que lo generó, hay 28 palabras. (OJO: todos los signos de puntuación no se consideran palabras).

Ahora, se debe validar la cantidad de palabras dentro del mensaje que se asocian a los perfiles, si una misma palabra aparece más de una vez, siempre cuenta para analizar el número de palabras del perfil:

Perfil “Deportista”: gym, selección, fútbol: Total 3 palabras,  $3/28 = 10.71\%$

Perfil “Cultura saludable”: gym, entrenar, carrera, ropa deportiva, bebidas hidratantes: Total 7 palabras,  $7/28 = 25\%$

Por lo tanto, este mensaje tiene una probabilidad de 25% de estar relacionado al perfil “Cultura saludable” y un 10.71% de probabilidad de estar relacionado al perfil “Deportista”.

**OJO:** Las mayúsculas y minúsculas no son importantes para determinar si pertenecen a las palabras clave. Cuando una palabra clave está formada por varias palabras suma esa cantidad de palabras para obtener el porcentaje correspondiente al perfil, es decir, “ropa deportiva” adiciona 2 palabras para calcular el porcentaje. En el caso de las palabras formadas por varias palabras, no importa si están separadas por espacios, tabulaciones o cambios de línea.

### **Reglas y consideraciones del formato para cada mensaje recibido:**

- **LUGAR Y FECHA:** Nombre de una ciudad, siempre se encuentra después de la palabra Lugar y Fecha: y antes de una coma (,). Luego, viene la fecha en formato dd/mm/yyyy hh24:mi (horas en formato 24 horas – 0 a 23 horas).
- **Usuario:** Nombre de un usuario, puede contener cualquier carácter excepto espacios, signos de tabulación o cambios de línea. Ej. de usuario: map001, [map001@usac.edu.gt](mailto:map001@usac.edu.gt). (Ojo: el uso de un correo electrónico como usuario es posible, pero no es obligatorio que los usuarios siempre sigan las reglas de estructura de un correo electrónico).
- **Red social:** ChapinChat (luego de la palabra ChapinChat viene el mensaje enviado por la red social).
- El mensaje es un texto libre que puede contener cualquier contenido, ya que el usuario puede escribir acerca de cualquier tema de su interés.

Finalmente, Tecnologías Chapinas, S.A. ha creado un algoritmo para establecer el peso de un perfil en un usuario de la red social ChapinChat. Este algoritmo se basa en los porcentajes de probabilidad de los perfiles en los mensajes que el usuario a generado en la red social, de tal manera que, se promedien por cada perfil los porcentajes de todos los mensajes cuyo porcentaje de probabilidad es mayor que cero, y este promedio corresponderá al peso de este perfil en el usuario.

### **Ejemplo de cálculo del peso de cada perfil para un usuario:**

Si tenemos al usuario map 001, y este usuario ha generado varios mensajes en la red social según la siguiente tabla:

Usuario: map001			
Mensaje	% probabilidad perfil "Deportista"	% de probabilidad perfil "Cultura Saludable"	
1/4/2023 09:31	10.71%	25%	
1/4/2023 12:07	5.65%	40.33%	
1/4/2023 15:21	0%	21.5%	
1/4/2023 15:33	0%	0	
1/4/2023 18:04	7.3%	23.33%	
1/4/2023 18:58	21.23%	0%	

Tabla 1 – Mensajes enviados por usuario y porcentajes de probabilidad de que los mensajes asocien al usuario en los perfiles definidos

Entonces, para calcular el peso del perfil "Deportista" para el usuario map001 se utilizaría la siguiente fórmula:  $(10.71 + 5.65 + 7.3 + 21.23) / 4 = 11.22$ ; mientras que, el perfil "Cultura Saludable" tendría un peso de:  $(25 + 40.33 + 21.5 + 23.33) / 4 = 27.54$ .

Usted ha sido contratado para desarrollar un programa que luego de recibir el mensaje descrito en la figura 1, deberá almacenar la información necesaria en formato XML, para calcular los pesos de los perfiles definidos en el sistema por los usuarios y posteriormente emitir reportes y realizar consultas.

## MENSAJES DE ENTRADA Y SALIDA

### Solicitud No. 1 – Definición de perfiles y palabras descartadas (Entrada)

El archivo para solicitar crear la definición de perfiles y palabras descartadas estará desarrollado utilizando XML, a continuación, se muestra su estructura:

```
<?xml version="1.0"?>
<configuracion>
  <perfiles>
    <perfil>
      <nombre> [hilera de caracteres] </nombre>
      <palabrasClave>
        <palabra> [hilera de caracteres] </palabra>
        ...
      </palabrasClave>
    </perfil>
    ...
  </perfiles>
  <descartadas>
    <palabra> [hilera de caracteres] </palabra>
    ...
  </descartadas>
</configuracion>
```

**NOTA:** La solicitud de definición de perfiles y palabras descartadas será acumulativa, es decir, cada vez que se haga una solicitud de definición de perfiles y palabras clave deberá agregar las palabras de la solicitud a perfiles existentes y al listado de palabras descartadas existentes. Si no existe algún perfil deberá crearlo.

### Respuesta a Solicitud No. 1

La respuesta, a la solicitud No. 1, debe ser presentada en formato XML, a continuación, se presenta un ejemplo:

```
<?xml version="1.0"?>
<respuesta>
  <perfilesNuevos>
    Se han creado 3 perfiles nuevos
  </perfilesNuevos>
  <perfilesExistentes>
    Se han actualizado 2 perfiles existentes
  </perfilesExistentes>
  <descartadas>
    Se han creado 20 nuevas palabras a descartar
  </descartadas>
</respuesta>
```

### Solicitud No. 2 – Creación de nuevos mensajes (Entrada)

El archivo para agregar nuevos mensajes de la red social al sistema estará desarrollado utilizando XML, a continuación, se muestra su estructura:

```
<?xml version="1.0"?>
<listaMensajes>
  <mensaje>
    Lugar y Fecha: Guatemala, 01/04/2023 09:31
    Usuario: map001
    Red social: ChapinChat
```

```

    Hola amigos, nos vemos hoy en el gym... recuerden que después vamos a entrenar para la carrera
    2K del próximo sábado. No olviden su Ropa Deportiva y sus bebidas Hidratantes. Recuerden
    que hoy por la noche juega la selección de fútbol, nos vemos en Taco Bell a las 7 pm.
  </mensaje>
  ...
</listaMensajes>

```

## Respuesta a Solicitud No. 2

La respuesta, a la solicitud No. 2, debe ser presentada en formato XML, a continuación, se presenta un ejemplo:

```

<?xml version="1.0"?>
<respuesta>
  <usuarios>
    Se procesaron mensajes para 15 usuarios distintos
  </usuarios>
  <mensajes>
    Se procesaron 40 mensajes en total
  </mensajes>
</respuesta>

```

## ARQUITECTURA

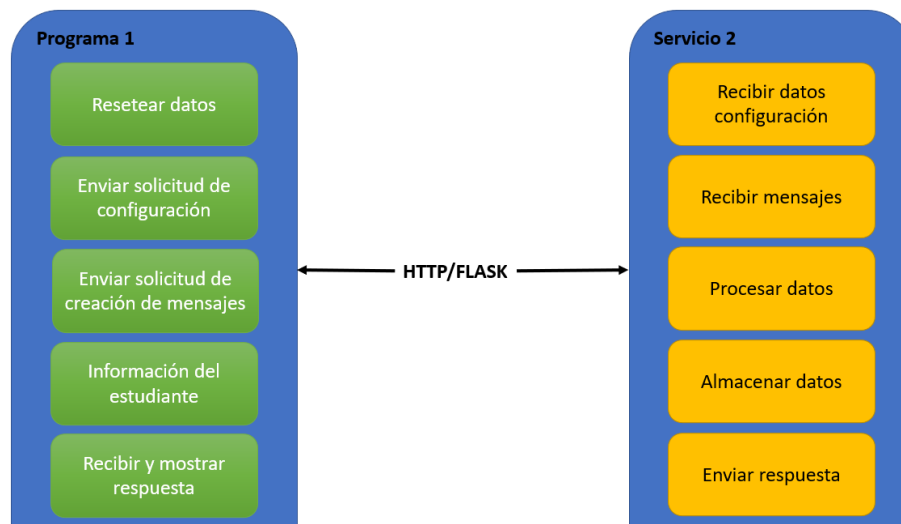


Fig. 4 – Arquitectura general de la aplicación

## Programa 1 - Frontend

Este programa consiste en una aplicación Web y consistirá en un simulador de la aplicación principal, contendrá únicamente las funcionalidades necesarias para testear el buen funcionamiento de las APIs (Servicio 2), 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.

Para realizar el frontend deberá utilizarse el framework **Django**, el cual trabaja con el patrón MVT (Modelo-Vista-Template).

Componentes:

- **Solicitar servicio:** Se desplegará 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 con el formato establecido en la sección de mensajes de entrada y salida.
- **Peticiones:** En este apartado se debe de tener las siguientes opciones:

❖ **Detalle de mensajes por usuario:** Al seleccionar esta opción se podrá elegir una fecha por la cual se requiere filtrar y si se desea incluir un usuario o todos los usuarios, entonces se debe de **mostrar** un resumen de mensajes por usuario identificando los perfiles y el porcentaje de probabilidad de que el perfil aplique al usuario. En la tabla 1 de este documento hay un ejemplo de la información a mostrar por 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 deberá **presentar una tabla** por cada usuario mostrando el peso correspondiente a cada perfil configurado en el sistema, este cálculo se puede ver en la sección “Ejemplo de cálculo del peso de cada perfil para un usuario” de este documento. Este reporte podrá ser descargado en formato .pdf si el usuario lo desea.

❖ **Prueba de solicitudes de creación de mensajes:** Esta opción permitirá enviar una solicitud con un mensaje con la siguiente estructura:

```
<?xml version="1.0"?>
<mensaje>
  Lugar y fecha: Guatemala,
  01/04/2023 15:21 Usuario: map0002@usac.edu Red social: ChapinChat
  Hola amigos, nos vemos hoy en el gym... recuerden que después vamos a
  entrenar para la carrera 2K del próximo sábado. No olviden su Ropa
  Deportiva y sus bebidas Hidratantes. Recuerden que hoy por la
  noche juega la selección de fútbol, nos vemos en Taco Bell a las 7 pm.
</mensaje>
```

Y deberá presentar una respuesta como la siguiente:

```
<?xml version="1.0"?>
<respuesta>
  <fechaHora> 01/04/2023 15:21 </fechaHora>
  <usuario> map0002@usac.edu </usuario>
  <perfiles>
    <perfil nombre=Deportista>
      <porcentajeProbabilidad> 10.71% </porcentajeProbabilidad>
      <pesoActual> 11.22 </pesoActual>
    </perfil>
    ...
  </perfiles>
</respuesta>
```

Debe tomar en cuenta que estos mensajes para prueba también se almacenan en su base de datos y afectan los cálculos de peso del usuario.

- **Ayuda:** desplegará 2 opciones, una para visualizar información del estudiante y otra para visualizar la documentación del proyecto.
- **Opción 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.

## Servicio 2 - Backend

Este servicio consiste en una API que brindará servicios utilizando el protocolo HTTP, su funcionalidad principal es procesar los datos recibidos del programa 1, luego de procesar los datos es necesario que estos sean almacenados en uno o varios archivos xml, algunos de estos archivos están especificados en la sección de mensajes de entrada y salida<sup>1</sup>, este servicio también tiene la funcionalidad de devolver los datos que fueron almacenados para que sean mostrados como se indica en la sección “Programa 1 – Frontend”.

Para la realización de este servicio debe utilizarse el framework **Flask**. El estudiante deberá definir por su propia cuenta los métodos que necesitará para la realización de este servicio. Esto significa que debe implementar tantos métodos como necesite para consumir la API.

**NOTA:** Durante la calificación de este proyecto, el Servicio 2 podrá ser consumido desde otro cliente, por ejemplo, Postman.

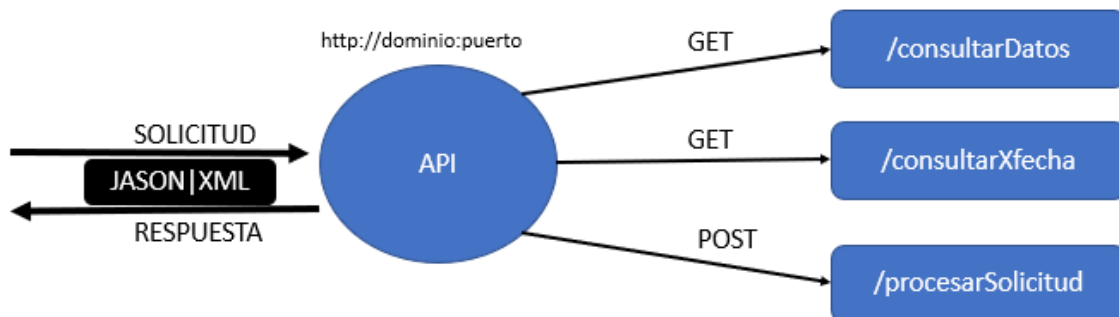


Fig. 5 - Ejemplo de la estructura de un API

## CONSIDERACIONES

Debe utilizarse versionamiento para el desarrollo del proyecto. Se utilizará la plataforma Github en la cual se debe crear un repositorio en el que se gestionará el proyecto. Se deben realizar 4 releases o versiones del proyecto (se recomienda realizar una por semana del tiempo disponible). Se deberá agregar a su respectivo auxiliar como colaborador del repositorio. El cuarto release será el release final y se deberá de realizar antes de entregar el proyecto en la fecha estipulada.

<sup>1</sup> El estudiante puede definir otros archivos que sean útiles para mostrar los resultados de las solicitudes atendidas.



Es obligatorio el desarrollo de la aplicación web (frontdend) y el servicio backend que atiende las solicitudes realizadas por la aplicación web a través del consumo de APIs.

## DOCUMENTACIÓN

Para que el proyecto sea calificado, el estudiante deberá entregar la documentación utilizando el formato de ensayo definido para el curso. En el caso del proyecto, el ensayo debe tener entre 4 y 7 páginas de contenido, este máximo no incluye los apéndices o anexos donde se pueden mostrar modelos y diseños utilizados para construir la solución.

## RESTRICCIONES

- Solo se permitirá la utilización de los IDEs y frameworks discutidos en el laboratorio.
- Uso obligatorio de programación orientada a objetos (POO).
- El nombre del repositorio debe de ser **IPC2\_Proyecto3\_#Carnet**.
- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- Se calificará el cuarto release almacenado en el repositorio Github. Los cambios realizados después de ese release no se tomarán en cuenta.
- Para dudas concernientes al proyecto se utilizarán los foros en UEDI de manera que todos los estudiantes puedan ver las preguntas y las posteriores respuestas.
- Para que el proyecto sea calificado deberá realizar una aplicación web (frontend) y el backend que expondrá los APIs que serán consumidos por la aplicación web.
- **NO HABRÁ PRÓRROGA.**

## ENTREGA

- La entrega será el **5 de mayo** a las 23:59 como máximo.
- La entrega será por medio de la UEDI.
- La documentación debe estar subida en el repositorio en una carpeta separada.
- Para entregar el proyecto en UEDI se deberá subir un archivo de texto con el link del repositorio.