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. José Manuel Ruiz Juárez
Ing. Dennis Stanley Barrios Gonzalez
Ing. Edwin Estuardo Zapeta Gómez
Ing. Fernando José Paz González

Tutores de curso:
Dayana Alejandra Reyes Rodríguez
Andrea María Cabrera Rosito
Paula Gabriela García Reinoso
Piter Angel Esaú Valiente de León
Mario César Morán Porras
Denilson Florentín de León Aguilar

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. está desarrollando una herramienta que sea capaz de analizar contenido de redes sociales y establecer el sentimiento de los usuarios respecto a una empresa y los servicios que provee.

Para lograr este fin, la empresa Tecnologías Chapinas, S.A. lee mensajes en redes sociales y construye estructuras de texto con el siguiente formado:

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

La empresa Tecnologías Chapinas, S.A. ha creado una estrategia para establecer si un mensaje tiene un sentimiento positivo, negativo o neutro a través de la creación de un diccionario de datos que determine palabras que puedan calificar un mensaje como positivo o negativo, en caso de no tener palabras del diccionario de datos específico, o bien, que la cantidad de palabras con sentimientos positivos y negativos sean iguales, entonces, se considera que el mensaje es neutro. En este mismo diccionario de datos, es posible determinar los nombres de las empresas y sus servicios que se están analizando para determinar si en un momento dado, las redes sociales están mostrando un sentimiento positivo o negativo del mismo.

Reglas y consideraciones del formato para cada mensaje recibido:

- **FECHA:** Fecha en formato dd/mm/yyyy hh24:mi (horas en formato 24 horas – 0 a 23 horas)
- **LUGAR:** Nombre de una ciudad, siempre se encuentra después de la palabra Lugar y Fecha: y antes de una coma (,).
- **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. (Ojo: el uso de un correo como usuario es posible, pero no es obligatorio que los usuarios siempre sigan las reglas de estructura de un correo electrónico).
- **Red social:** Nombre de una red social, puede contener cualquier carácter excepto espacios, signos de tabulación o cambios de línea.

El programa por desarrollar, luego de recibir el mensaje antes mencionado, **deberá almacenar la información necesaria en formato XML, que constituirá la base de datos de la aplicación**, para posteriormente emitir reportes y realizar consultas. El archivo de respuesta o salida deberá presentar la siguiente información:

FECHA: dd/mm/yyyy
Cantidad total de mensajes recibidos: Valor entero que representa el total de mensajes recibidos
Cantidad total de mensajes positivos: Valor entero que representa el total de mensajes positivos recibidos
Cantidad total de mensajes negativos: Valor entero que representa el total de mensajes negativos recibidos
Cantidad total de mensajes neutros: Valor entero que representa el total de mensajes neutros recibidos
Empresa:
Número total de mensajes que mencionan a Empresa: Valor entero
Mensajes positivos: Valor entero
Mensajes negativos: Valor entero
Mensajes neutros: Valor entero
Servicio #1
Número total de mensajes que mencionan al servicio: Valor entero
Mensajes positivos: Valor entero
Mensajes negativos: Valor entero
Mensajes neutros: Valor entero
…
…
Empresa #n:
Número total de mensajes que mencionan a Empresa o servicio #n: Valor entero
Mensajes positivos: Valor entero
Mensajes negativos: Valor entero
Mensajes neutros: Valor entero
Servicio #1
Número total de mensajes que mencionan al servicio: Valor entero
Mensajes positivos: Valor entero
Mensajes negativos: Valor entero
Mensajes neutros: Valor entero
…
…
FECHA: dd/mm/yyyy
…

La estructura completa del archivo de respuesta o salida está descrita en la siguiente sección. Debe considerar que cada mensaje solamente puede estar clasificado como positivo, negativo o neutro, pero que en un mismo mensaje se puede mencionar cero, una o muchas empresas o servicios que se encuentren bajo análisis.

ARCHIVOS DE ENTRADA Y SALIDA

Solicitud - Entrada

El archivo para solicitar la clasificación de mensajes estará desarrollado utilizando XML, a continuación, se muestra un ejemplo:

```
<?xml version="1.0"?>
<solicitud_clasificacion>
  <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>
    <empresas_analizar>
      <empresa>
        <nombre> USAC </nombre>
        <servicios>
          <servicio nombre="inscripción">
                <alias> inscribí </alias>
                <alias> inscrito </alias>
                ...
          </servicio>
          <servicio nombre="asignación">
                <alias> asignado </alias>
           </servicio>
           <servicio nombre="graduación"> </servicio>
         ...
        </servicios>
      </empresa>
          ...
    </empresas_analizar>
  </diccionario>
  <lista_mensajes>
    <mensaje>
      Lugar y fecha: Guatemala, 01/04/2022 15:01 Usuario:
          map0001@usac.edu   Red social: Twitter
          El servicio en la USAC para inscripción fue muy bueno y me siento muy satisfecho.
    </mensaje>
    <mensaje>
      Lugar y fecha: Guatemala,
          01/04/2022 15:20 Usuario: map0002@usac.edu   Red social: Facebook Hoy me inscribí
          en la USAC, no encontré parqueo e inicié molesto mi gestión, luego tuve que hacer
          cola y me indicaron que era la cola incorrecta, esto me enojó mucho y mejor me
          fui.
    </mensaje>
    <mensaje>
      Lugar y fecha: Guatemala, 01/04/2022 15:43 Usuario: map0003@usac.edu
          Red social: WhatsApp
          Ya estoy inscrito en la USAC, el proceso fue muy bueno, pero al salir,
          mi carro tenía pinchadas las llantas, esto me preocupó
          y me fui decepcionado.
    </mensaje>
    ...
  </lista_mensajes>
</solicitud_clasificacion>
```

Nota: Para el análisis de los archivos de entrada no debe tomarse en consideración mayúsculas o minúsculas, es decir, asignación = Asignación = AsigNAciÓN; y tampoco debe considerar las tildes, es decir, asignación = Asignacion = ASIGNACION. Esto aplica para todos los conceptos (empresas, servicios, sentimientos positivos, sentimientos negativos, usuarios, redes sociales, etc.).

Respuesta - salida

La respuesta, luego de consumir el servicio antes mencionado, debe ser presentada en formato XML, a continuación, se presenta un ejemplo:

```
<?xml version="1.0"?>
<lista_respuestas>
  <respuesta>
    <fecha> 01/04/2022 </fecha>
      <mensajes>
        <total> 20 </total>
        <positivos> 8 </positivos>
        <negativos> 7 </negativos>
        <neutros> 5 </neutros>
      </mensajes>
      <analisis>
       <empresa nombre="USAC">
          <mensajes>
            <total> 3 </total>
            <positivos> 1 </positivos>
            <negativos> 1 </negativos>
            <neutros> 1 </neutros>
          </mensajes>
          <servicios>
              <servicio nombre="inscripción">
               <mensajes>
                  <total> 3 </total>
                  <positivos> 1 </positivos>
                  <negativos> 1 </negativos>
                  <neutros> 1 </neutros>
                </mensajes>
              </servicio>
               ...
          </servicios>
       </empresa>
          ...
      </analisis>
  </respuesta>
  ...
</lista_respuestas>
```

ARQUITECTURA

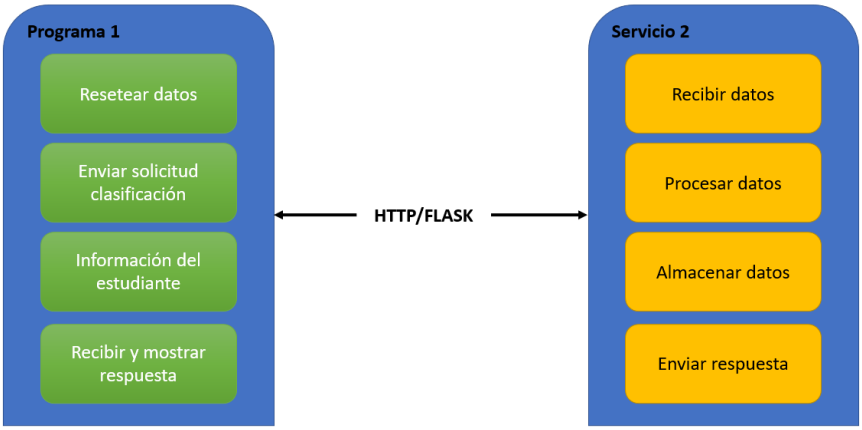

Fig. 1 – Arquitectura general de la aplicación

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 la API (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:

- **Cargar Archivo:** Se desplegará una pantalla para gestionar la carga de los archivos de entrada con extensión .xml con una o varias solicitudes de clasificación. Se especifica en la sección de archivos de entrada y salida.
- **Peticiones:** En este apartado se debe de tener las siguientes opciones:

    - **Consultar Datos:** Al seleccionar esta opción se deben de consultar los datos almacenados en el último archivo de respuesta y se mostrarán los datos en un recuadro de texto de salida.
    - **Resumen de clasificación por fecha:** Al seleccionar esta opción se podrá elegir la fecha por la cual se requiere filtrar y si se desea incluir una empresa o todas las empresas, entonces se debe de **mostrar gráficamente** un resumen de mensajes clasificados. Debe presentar el total de mensajes, el total de mensajes positivos, el total de mensajes negativos y el total de mensajes neutros para la fecha y empresa seleccionados.
    - **Resumen por rango de fechas:** Al seleccionar esta opción se podrá elegir un rango de fechas por la cual se requiere filtrar, luego podrá elegir si desea ver una empresa o todas las empresas. Se deberá **presentar gráficamente** por cada fecha y empresa el total de mensajes, el total de mensajes positivos, el total de mensajes negativos y el total de mensajes neutros.

- **Reporte en PDF:** Deberá ser posible realizar un reporte de todas las peticiones anteriormente descritas de forma detallada, es decir, incluir los datos de mensajes totales, mensajes positivos, mensajes negativos y mensajes neutros por empresa y servicio.
- **Prueba de mensaje:** Esta opción permitirá ingresar un mensaje con la siguiente estructura:

```
<?xml version="1.0"?>
<mensaje>
Lugar y fecha: Guatemala,
  01/04/2022 15:20 Usuario: map0002@usac.edu  Red social: Facebook Hoy me inscribí en la USAC, no encontré parqueo e inicié molesto mi gestión, luego tuve que hacer cola y me indicaron que era la cola incorrecta, esto me enojó mucho y mejor me fui.
</mensaje>
```

Y deberá presentar una respuesta como la siguiente:

```
<respuesta>
  <fecha> 01/04/2022 </fecha>
  <red_social> Facebook </red_social>
  <usuario> map0002@usac.edu </usuario>
  <empresas>
    <empresa nombre=USAC>
      <servicio> inscripción </servicio>
      …
    </empresas>
    …
  </empresas>
  <palabras_positivas> 0 </palabras_positivas>
  <palabras_negativas> 2 </palabras_negativas>
  <sentimiento_positivo> 0% </sentimiento_positivo>
  <sentimiento_negativo> 100% </sentimiento_negativo>
  <sentimiento_analizado> negativo </sentimiento_analizado>
</respuesta>
```

Debe tomar en cuenta que estos mensajes para prueba no se almacenan en su base de datos

- **Ayuda:** desplegará 2 opciones, una para visualizar información del estudiante y otra para visualizar la documentación del programa.
- **Botón Enviar:** Enviará las solicitudes de clasificación del recuadro de texto a la Api para su procesamiento.
- **Botón Reset:** Este botón mandará la instrucción a la Api para devolver al estado inicial la Base de Datos, es decir, sin datos.

Fig. 3 - Sugerencia de interfaz.

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 archivos de entrada y salida[1], 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 / Componentes”.

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.

[1] El estudiante puede definir otros archivos que sean útiles para mostrar los resultados de las solicitudes de autorización atendidas.

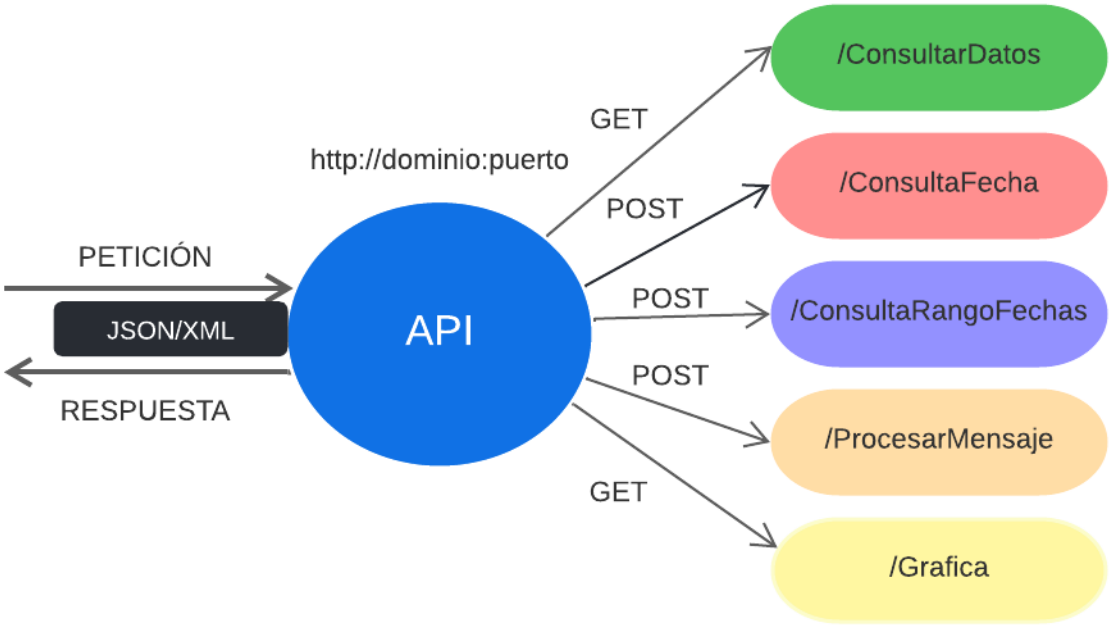

Fig. 4 - Ejemplo de la estructura de un API

CONSIDERACIONES

Se deberá realizar **la implementación utilizando programación orientada a objetos en lenguaje Python**, algoritmos desarrollados por el estudiante e implementación de estructuras a través de Tipos de Dato Abstracto (TDA) propios del estudiante; que permita almacenar la información de los archivos de entrada y poder interactuar con dicha información. El estudiante deberá abstraer la información y definir qué estructuras implementar que le faciliten la solución. Por lo que puede implementar pilas, colas, listas simples, dobles, circulares, listas ortogonales, listas n-enlazadas o listas de listas para poder solventar el proyecto. En este proyecto SI está permitido el uso de estructuras propias de Python (list, dict, tuple, set). **Para el desarrollo de gráficas en el frontend (Programa 1) deberá utilizar la librería chartjs**.

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) mínimo. **Se deberá agregar a su respectivo auxiliar como colaborador del repositorio**. El último release será el release final y se deberá de realizar antes de entregar el proyecto en la fecha estipulada.

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. Es obligatorio incluir el diagrama de clases que modela la solución de software presentada por el estudiante y los diagramas de actividades con los principales algoritmos implementados en la solución.

RESTRICCIONES

- Solo se permitirá la utilización de los IDEs y frameworks indicados en este enunciado.
- Uso obligatorio de programación orientada a objetos (POO) en lenguaje Python. De no cumplir con la restricción, no se tendrá derecho a calificación.
- El nombre del repositorio debe de ser **IPC2_Proyecto3_#Carnet.**
- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- Los archivos de entrada no podrán modificarse.
- Los archivos de salida deben llevar la estructura mostrada en el enunciado obligatoriamente.
- Deben existir 4 releases uno por cada semana, de esta manera se corrobora el avance continuo del proyecto. **Se definirá una penalización por cada release faltante**.
- 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.
- Cualquier caso de copia parcial o total tendrá una nota de 0 y será reportada a Escuela de Ciencias y Sistemas.
- **NO HABRÁ PRÓRROGA.**

ENTREGA

● La entrega será el 28 de octubre 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.