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

Ing. Jaime Francisco Yuman Ramirez
Tutor de curso: Monica Raquel Calderon Muñoz

PROYECTO 2

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)

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 archivos XML como insumos para la comunicación con el API desarrollado.
- Utilizar expresiones regulares para extraer contenido de texto.

DESCRIPCIÓN GENERAL

Debido al éxito generado a través de la aplicación de escritorio de IPCMusic se ha deseado que esta deje de ser manejada a manera local si no que pueda ser consumida por medio de una página web para con ello expandir su alcance y poder mostrar variedad de estadísticas con relación tanto de las canciones, los artistas como los álbumes a través de gráficas entendibles y dinámicas.

IMPLEMENTACIÓN

La arquitectura cliente - servidor habla acerca de una relación entre un programa (cliente) solicita un servicio o recurso de otro programa (el servidor).
Se ha solicitado que construya un software que pueda ser consumido desde internet como un servicio. Dicho software será capaz de transformar los archivos XML actuales hacía un formato JSON, el JSON debe de ser almacenado previo al envío hacia el servidor ya que al pretender la estandarización del almacenaje se espera migrar la aplicación de música hacía un entorno virtual.

Así mismo, se espera poder realizar la migración de listas de reproducción de cualquier tipo de aplicación de música, sin embargo, para migrar estas listas de reproducción desde cualquier otro reproductor la aplicación debe de ser capaz de leer un archivo CSV que se va a transformar a un archivo XML y luego pasará el proceso previamente mencionado.

ARCHIVOS DE ENTRADA Y SALIDA

ARCHIVO CSV

El archivo CSV que se va a manejar es el que almacenará las listas de reproducción de cualquier otra aplicación de reproducción de música diferente a IPCMusic. A continuación se desglosa la información que contendrá.

- **Lista de reproducción**
  - **Nombre**: Conjunto de letras, de encontrarse una variante de este debe ser reportado como error.
  - **Canción**: Conjunto de letras, de encontrarse una variante de este debe ser reportado como error.
  - **Artista**: Conjunto de letras, de encontrarse una variante de este debe ser reportado como error.
  - **Albúm**: Conjunto de letras, de encontrarse una variante de este debe ser reportado como error
  - **Veces reproducida**: Conjunto de números enteros, dicha columna no debe incluir nada más que números positivos, dado sea el caso se encuentre algo diferente debe de ser reportado como error.
  - **Ruta:** Únicamente se aceptan en la ruta documentos con formato .mp3, de encontrarse una variante de este debe ser reportado como error.
  - **Imagen:** Únicamente se aceptan en la ruta documentos con formato .jpg, de encontrarse una variante de este debe ser reportado como error.

ARCHIVO XML

Debido a la naturaleza de la información que se almacena en los archivos CSV es importante que dicha información no posea erratas, por lo que se le solicita que previo a la transformación a XML se realice un análisis y de existir algún error debe ser reportado al usuario.

Si el archivo CSV presenta un error, este no es transformado a un archivo XML si no que se muestra al usuario un listado con los posibles errores encontrados e invitando al usuario para que sean corregidos.

Caso contrario, los archivos CSV no posean ningún error, se espera un archivo XML de la forma

El cual, tras almacenarlo, será mostrado en pantalla. El cliente debe de ser capaz de poder realizar modificaciones en él a través de un editor de texto previo a su envío al servidor y guardar las modificaciones. También es importante considerar que el cliente debe de tener la capacidad de agregar o eliminar la cantidad de etiquetas que considere necesarias.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ListasReproduccion>
    <Lista nombre="MiLista">
    <cancion nombre="Attack on bangtan">
        <artista>"BTS"</artista>
        <album>"O!RUL8,2?"</album>
        <vecesReproducida>2</vecesReproducida>
        <imagen>"C:\Users\Monica Calderon\Desktop\Diciembre 2021\IPC2\AOB_BTS.jpg"</imagen>
        <ruta>"C:\Users\Monica Calderon\Desktop\Diciembre 2021\IPC2\AOB_BTS.mp3"</ruta>
    </cancion>
    <cancion nombre="the middle">
        <artista>"Jimmy Eat World"</artista>
        <album>"Bleed American"</album>
        <vecesReproducida>3</vecesReproducida>
        <imagen>"C:\Users\Monica Calderon\Desktop\Diciembre 2021\IPC2\theMiddleJEW.jpg"</imagen>
        <ruta>"C:\Users\Monica Calderon\Desktop\Diciembre 2021\IPC2\theMiddleJEW.mp3"</ruta>
    </cancion>
    <cancion nombre="Perfectly Perfect">
        <artista>"Simple plan"</artista>
        <album>"Taking One for the Team"</album>
        <vecesReproducida>7</vecesReproducida>
        <imagen>"C:\Users\Monica Calderon\Desktop\Diciembre 2021\IPC2\SimplePlan_PP.jpg"</imagen>
        <ruta>"C:\Users\Monica Calderon\Desktop\Diciembre 2021\IPC2\SimplePlan_PP.mp3"</ruta>
    </cancion>
    </Lista>
</ListasReproduccion>
```

Fuente: Elaboración propia

ARQUITECTURA

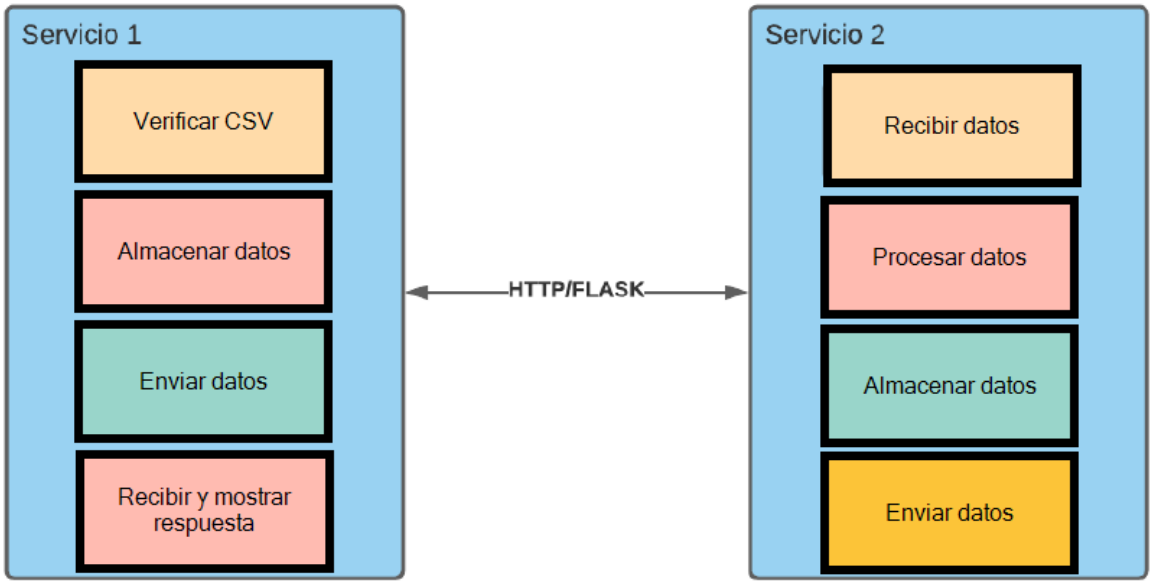

Figura 1: Arquitectura de la aplicación

SERVICIO FRONTEND

Consiste en una aplicación web, que contará con las funcionalidades tales como:

- Lectura archivos XML.
- Conversión archivos CSV a XML.
- Reproducción de música.
- Calificar canciones.

Se espera que en la aplicación se pueda visualizar el archivo XML previó al envío así como el XML correspondiente con los datos que el servidor retorna.

Los componentes mínimos a considerar son:

- **Carga archivo:** Ventana a través la cual se seleccionan los archivos CSV y XML, de los archivos CSV se realizará la transformación hacía un archivo XML. De existir errores en el archivo CSV se debe de mostrar en pantalla que el archivo no se ha podido transformar.
- **Reproducción de canciones:** La aplicación web podrá ser capaz de reproducir las canciones que se envíen desde el servidor.
- **Clasificación de canciones:** El reproductor de canciones permitirá al usuario a calificar las canciones entre uno a cinco estrellas.
- **Peticiones:** Apartado con las siguientes características:
    - **Canciones más escuchadas:** Se observa a través de una gráfica de barras las canciones que más se han reproducido.
    - **Artistas más reproducidos:** Gráfica en forma de pie que mostrará los artistas más reproducidos.
    - **Clasificación:** Gráfica en forma de pie que muestra la cantidad de artistas distribuído por las estrellas obtenidas.

    - **Listas más escuchadas:** Gráfico a discreción del estudiante en donde se presenta el top cinco de las listas de reproducción más escuchadas, no hay necesidad que las listas sean reproducidas completas para ser tomadas en cuenta.
    - **Listas más populares:** La popularidad de una lista de reproducción se basa en la cantidad de canciones que posee, se debe de mostrar un Top 10 de las listas de reproducción más populares
- **Ayuda**: desplegará 2 opciones, una para visualizar información del estudiante y otra para visualizar la documentación del programa.
- **Botón Enviar**: Enviará los eventos del recuadro de texto a la Api para su posterior procesamiento.
- **Reporte errores:** Desplegará un reporte en donde indique los errores encontrados en el análisis del archivo CSV indicando el motivo por el cual no fue posible realizar la transformación a XML.

SERVICIO BACKEND

Este servicio consiste en una API que brindara servicios utilizando el protocolo HTTP, su funcionalidad principal es procesar los datos recibidos del servicio de frontend, luego de procesar los datos es necesario que estos sean almacenados en un archivo xml, este servicio también tiene la funcionalidad de devolver los datos que fueron almacenados para que sean mostrados como anteriormente se indica.

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

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 mínimo 4 releases o versiones del proyecto. 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.
Para la realización de la interfaz gráfica queda a discreción del estudiante que librería utilizar.

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 puede tener un mínimo de 4 y un máximo de 7 páginas de contenido, este máximo no incluye los apéndices o anexos donde se pueden mostrar modelos y diseños utilizados para construir la solución. Este informe debe expresar con claridad el diseño de objetos ideado para resolver este proyecto por lo que debe expresar el diagrama de clases y los diagramas de actividades de los algoritmos más importantes.

Debe de tomar en cuenta que es un ensayo formal, por lo que se calificará tanto la redacción, como ortografía y presentación.

RESTRICCIONES

- Solo se permitirá la utilización de los IDEs discutidos en el laboratorio.
- Uso obligatorio de programación orientada a objetos (POO).
- El nombre del repositorio debe de ser IPC2_Proyecto2Diciembre_#Carnet.
- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- Los archivos de entrada no podrán modificarse.
- Se calificará la versión del cuarto release o del último release realizado previo a la fecha de entrega. No se calificará dado que se dé el caso que existan modificaciones de código en fechas posteriores a la entrega.
- 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.
- No es permitido el uso de JavaScript para el envío ni para la recepción de datos
- Para el backend debe de utilizarse el framework Flask mientras que para el frontend debe utilizarse Django
- No se calificará frontend sin backend ni viceversa.
-
- **COPIAS TOTALES O PARCIALES SERÁN REPORTADOS A LA ESCUELA Y OBTENDRÁN NOTA DE 0 PUNTOS.**
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el día sábado 30 de diciembre antes de las 23:59.
- La entrega será por medio de la UEDI.
- Se deberá crear un repositorio con nombre IPC2_Proyecto2Diciembre_#carnet.