

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 Auxiliar
Jackeline Alexandra Benitez Benitez



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) 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 MyMusicYourMusic está desarrollando una herramienta que sea capaz de almacenar y reproducir la música que el cliente desee, para escucharla en línea y con ella opcionalmente mandar a hacer dos tipos de CD. Los compactos o los de vinilo si así lo desea el cliente.

Sin embargo, por razones de fiestas decembrinas ha tenido mucha demanda esta empresa con otras empresas que quieren darle un obsequio a sus empleados para estas fechas. Por lo que han decidido optar por usar una plataforma para manejar todos los pedidos que le hagan.

Por ello, cada cliente podrá realizar una playlist con la música que desee para la ocasión que desee. La playlist tendrá una categoría predeterminada según el tipo de música que escoja el usuario, por ejemplo: Navideña, de Fiesta, de Estudio, de relajación, de Ejercicio, entre otros. Además, el usuario podrá escoger si desea mandar a hacer un disco de vinil o disco compacto, para el recuerdo o solo la playlist que sera de reproducción ilimitada y sin anuncios entre canciones. El costo por añadir una canción a la playlist varía según ciertos criterios:

- Si la canción es de los años 1960 o antes costará Q25
 - Si la canción es de los años de antes 1990 y después de 1960 costara Q15
 - Si la canción es de los años de antes o durante 2022 y después de 1990 costara Q5
- además si el usuario escoge mandar a hacer un disco compacto son Q100 y si el usuario escoge mandar a hacer un disco de vinyl son Q500 por playlist.

Los clientes tendrán la opción de crear un máximo de 3 playlists diferentes.

Actualmente, los clientes de YourMusicMyMusic se registran ingresando su nombre, su NIT, su dirección física, su dirección de correo electrónico y la empresa a la que pertenecen y una vez registrados reciben en su correo electrónico un usuario y una clave con la cual pueden ingresar a una consola donde pueden crear, modificar y eliminar playlists (Los usuarios pueden crear un maximo de 3 playlists por cortesía de la empresa a la que laboran)

Usted ha sido contratado para crear un software que se ejecutará en el backend de la infraestructura de YourMusicMyMusic, este software registrará, utilizando API's, todos los datos relacionados a playlists, clientes y empresas para finalmente, calcular la facturación de cada empresa según los pedidos de sus empleados.

MENSAJE DE ENTRADA PARA CONFIGURAR LAS PLAYLISTS DE UN GRUPO DE CLIENTES PERTENECIENTES A VARIAS EMPRESAS

Para poder crear la información necesaria para brindar y cobrar por los servicios que presta YourMusicMyMusic. se utilizará un mensaje XML con la siguiente estructura:

```
<?xml version="1.0"?>
<playlistClientes>
```

```

    <playlist id='$idPlaylist'>
      <nitCliente>$nitCliente</nitCliente>
      <vinyl>valorBooleano</vinyl>
      <compacto>valorBooleano</compacto>
      <categoria>$tipoCategoria</categoria>
      <canciones>
        <cancion id='$idCancion'>
          <nombre>$nombreCancion</nombre>
          <anio>$anioCancion</anio>
          <artista>$nombreArtista</artista>
          <genero>$generoCancion</genero>
        </cancion>
        <cancion id='$idCancion'>
          <nombre>$nombreCancion</nombre>
          <anio>$anioCancion</anio>
          <artista>$nombreArtista</artista>
          <genero>$generoCancion</genero>
        </cancion>
        <cancion id='$idCancion'>
          <nombre>$nombreCancion</nombre>
          <anio>$anioCancion</anio>
          <artista>$nombreArtista</artista>
          <genero>$generoCancion</genero>
        </cancion>
        <cancion id='$idCancion'>
          <nombre>$nombreCancion</nombre>
          <anio>$anioCancion</anio>
          <artista>$nombreArtista</artista>
          <genero>$generoCancion</genero>
        </cancion>
        ...
      </canciones>
    </playlist>
  ...
</playlistClientes>
<listaClientes>
  <cliente nit="$nitCliente">
    <nombre> $nombreCliente </nombre>
    <usuario> $nombreUsuario </usuario>
    <clave> $claveUsuario </clave>
    <direccion> $direccionCliente </direccion>
    <correoElectronico> $eMailCliente </correoElectronico>
    <empresa> $idEmpresa </empresa>
    <playlistsAsociadas>
      <playlist>$idPlaylist</playlist>
      <playlist>$idPlaylist</playlist>
      <playlist>$idPlaylist</playlist>
    </playlistsAsociadas>
  </cliente>
  ...
</listaClientes>
<listaEmpresas>
  <empresa id='$idEmpresa'>
    <nombre>$nombreEmpresa</nombre>
  </empresa>
  ...
</listaEmpresas>

```

ARQUITECTURA

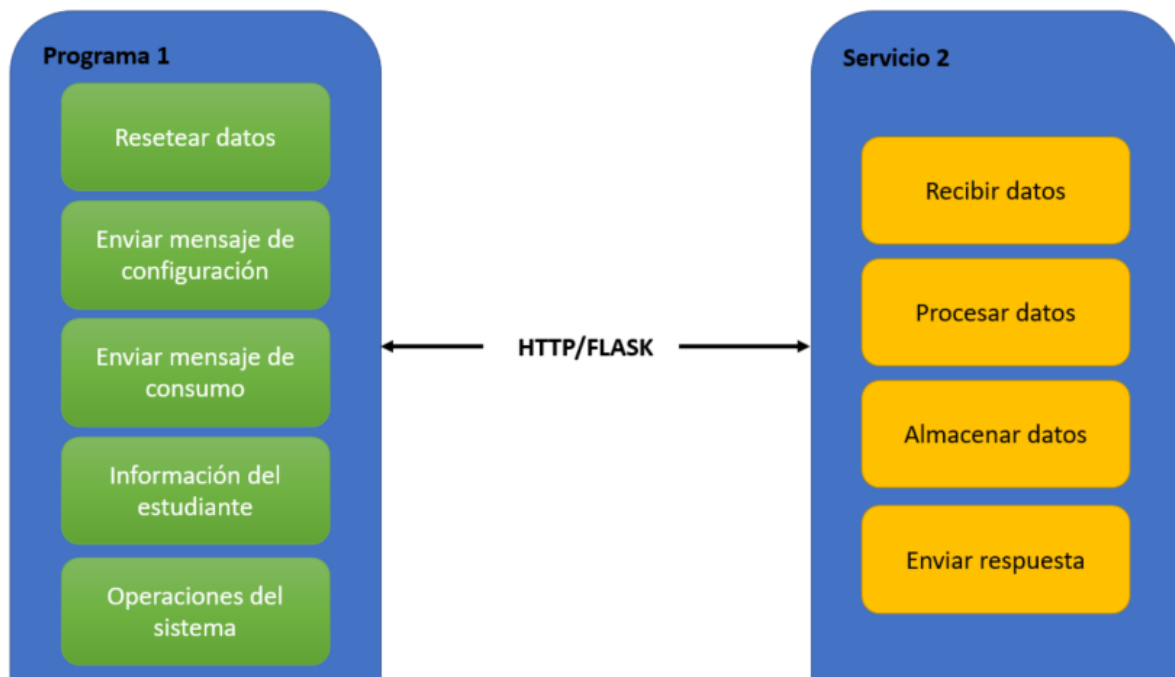


Fig. 1 – Arquitectura general de la aplicación

Programa 1 - Frontend

Este programa consiste en una aplicación Web y consistirá en un simulador de la aplicación principal, contendrá las funcionalidades necesarias para testear el buen funcionamiento de la API (Servicio 2) y la correcta funcionalidad de los procesos para gestionar el servicio de streaming y de pedidos de YourMusicMyMusic; 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:

- **Enviar mensaje de configuración:** Se desplegará una pantalla para gestionar el envío del mensaje de entrada con extensión .xml con una o varias solicitudes de creación de elementos del sistema. Deberá mostrar un mensaje indicando si el mensaje fue enviado exitosamente y el resultado de este, por ejemplo, 3 nuevos clientes creados, 7 nuevas playlists creadas, 2 empresas nuevas creadas.
- **Operaciones del sistema:** En este apartado se debe de tener las siguientes opciones:
 - ❖ **Consultar Datos:** Al seleccionar esta opción se podrá chequear la estructura de información que actualmente maneja el sistema, es decir, podrá ver las playlists con sus canciones, los clientes y las empresas existentes

- ❖ **Creación de nuevos datos:** Al seleccionar esta opción se podrá crear nueva información, es decir, podrá crear nuevas playlists, nuevos clientes o ingresar una nueva Empresa.
- ❖ **Eliminar datos:** Al seleccionar esta opción se podrán eliminar canciones de una playlist o eliminar a un cliente.
- ❖ **Proceso de facturación:** Al seleccionar esta opción se podrá elegir una empresa para poder generar la factura según los clientes que tiene y las playlists que ha creado, dependiendo también si ha mandado a hacer algún disco físico. La factura incluye la siguiente información: Número de factura (debe ser único), Nombre de la empresa a la que se le emite la factura, clientes que pidieron playlists pertenecientes a la empresa, fecha de la factura (último día incluido en el rango seleccionado), monto a pagar.
- ❖ **Reporte en PDF:**
 - **Detalle de factura:** Este reporte permitirá seleccionar una factura y presentará los datos de la factura y el detalle de lo que incluye la factura. Debe ser posible ver el monto a pagar por cada playlist, las canciones en cada uno de ellas y el costo por cada una, así si pidió disco físico al cobro de la factura.
- **Ayuda:** desplegará 2 opciones, una para visualizar información del estudiante y otra para visualizar la documentación del programa.

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, este servicio también tiene la funcionalidad de devolver los datos que fueron almacenados para que sean mostrados como respuesta a las solicitudes realizadas por el “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.

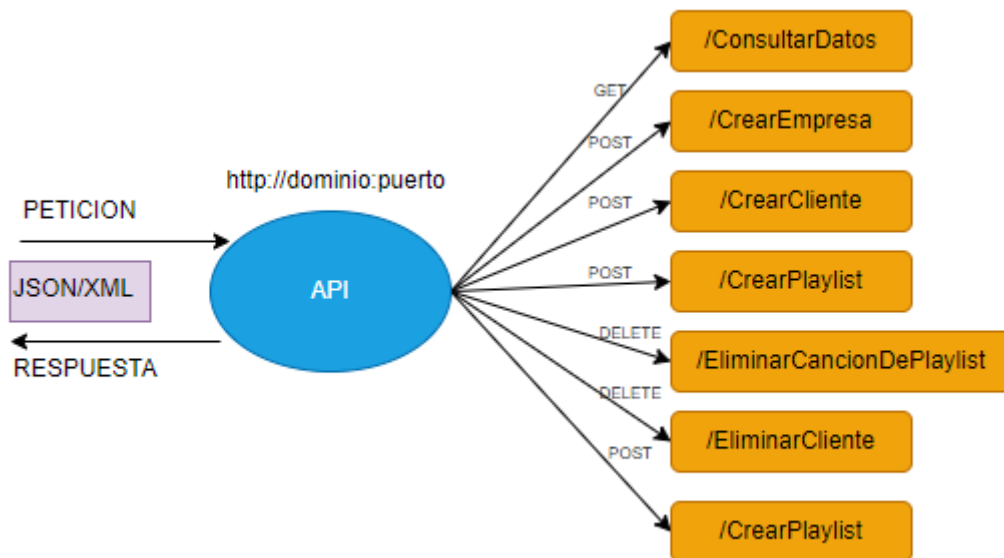


Fig. 4 - Ejemplo de la estructura de un API

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 deberá agregar a su respectivo auxiliar como colaborador del repositorio. El ultimo commit será el que se usará para la calificación 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, deberá adjuntar también su diagrama de clases y la traducción a ingles

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_Proyecto2_#Carnet**.
- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el **30 de diciembre** 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.