Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2021.

PROYECTO 3 – IPC2

202000194 – Alvaro Emmanuel Socop Pérez

Resumen

El presente proyecto consiste en la creación de un servicio web desarrollada en el lenguaje Python que ha sido solicitado por La Superintendencia de Administración Tributaria, el cual recibe datos de un Documento Tributario Electrónico (DTE) emitido por un contribuyente y se devuelve un número único de autorización, se ha utilizado el Framework Django en el área del Front-End y Flask en el área del Back-End.

La aplicación web funciona con la lectura de archivos XML para posteriormente por medio de expresiones regulares validarlos y procesar un archivo de salida (autorizaciones.xml) el cual posee los resultados de las transacciones que se han realizado en cada una de las fechas.

El servicio cuenta con filtros de los cuales se obtiene cierto tipo de información por ejemplo, “Iva por fecha y NIT” y “valor autorizado por rango de fechas” de los cuales se muestran por medio de gráficas para su mejor entendimiento.

.

Palabras clave
API, Frontend, Backend, Peticiones, Framework.

Abstract

The present project consists of the creation of a web service developed in Python language that has been requested by the Superintendence of Tax Administration, which receives data from an Electronic Tax Document (DTE) issued by a taxpayer and returns a unique authorization number, the Django Framework has been used in the Front-End area and Flask in the Back-End area.

The web application works by reading XML files and then using regular expressions to validate them and process an output file (autorizaciones.xml) which has the results of the transactions that have been made on each of the dates.

The service has filters from which certain types of information are obtained, for example, "VAT by date and NIT" and "authorized value by date range", which are shown in graphs for better understanding.

Keywords
API, Frontend, Backend, Request, Framework.

Introducción

El proyecto tiene como enfoque la implementación de un Framework para trabajar el Frontend, y el uso de una API para familiarizarnos con las bases de Datos, se utilizan estructuras y algoritmos implementados para mostrar los filtros, utilizando el lenguaje de programación Python, para la creación de la interfaz se ha utilizado el modelo vista controlador creado con el framework Django siendo un marco web de Python de alto nivel que fomenta el rápido desarrollo y el diseño limpio. En el Frontend se hace posible la conexión o consumo de las API's realizadas en el área del Backend, en este caso elaborado en Flask, para que sea funcional la parte del Frontend.

Se utilizaron librerías ajenas a Python para la creación de gráficos en donde se resumen los filtros que se pueden aplicar a los datos ingresados por medio de los archivos de entrada.

Se utilizo el formato XML para la lectura de los datos, y luego se enviaron a la API, para que sean procesados y podamos obtener respuestas, se diseñó la interfaz utilizando views de django y varias funciones y métodos que este framework nos ofrece como la herencia el cual permite ahorrar código, se manejaron los datos desde django hacia la api escrito en Flask.

Programación en Python

Python es un lenguaje de scripting que no depende de plataforma y es orientado a objetos, realiza cualquier tipo de programa, aplicaciones para Windows hasta servidores de red o, páginas web. Es un lenguaje interpretado, por lo cual no es necesario compilar el código fuente para su ejecución.

Una de las ventajas que se presentan es la rapidez de desarrollo, y su pequeña desventaja o inconveniente es que ofrece una menor velocidad.

Se considera es un lenguaje de programación que admite el uso de varios paradigmas de programación (modelos de desarrollo), por lo que no exige a los programadores un estilo único para programar.

Estructuras de programación secuenciales

La estructura secuencial es aquella en la que una acción (instrucción) sigue a otra en secuencia. Las tareas se suceden de tal modo que la salida de una es la entrada de la siguiente y así sucesivamente hasta el fin del proceso.

La asignación consiste, en el paso de valores o resultados a una zona de la memoria. Dicha zona será reconocida con el nombre de la variable que recibe el valor. La asignación se puede clasificar de la siguiente forma:

a. Simples: Consiste en pasar un valor constante a una variable.
b. Contador: Consiste en usarla como un verificador del número de veces que se realiza un proceso.
c. Acumulador: Consiste en usarla como un sumador en un proceso.

d. De trabajo: Donde puede recibir el resultado de una operación matemática que involucre muchas variables.

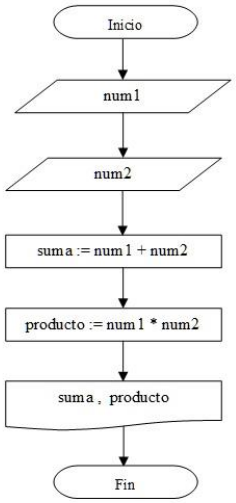

Figura 1. Diagrama de flujo como estructura secuencial.

Fuente: elaboración propia, o citar al autor, año y página.

Estructuras cíclicas

Estas estructuras también llamadas bucles, se utilizan para realizar o ejecutar operaciones o acciones repetitivas, las cuales tienen un fin o limite

Las estructuras cíclicas se utilizan para ejecutar fragmentos de código un número limitado de veces. Existen tres estructuras cíclicas generales, las cuales son:

a. Desde Hasta (For)
b. Hacer Mientras (While)

Las estructuras tienen el mismo objetivo, ejecutar un fragmento de código un número limitado de veces, su principal diferencia se encuentra en la forma en la cual limitan el número de ocasiones que se ejecutará el código.

Estructuras condicionales

Las estructuras condicionales existen en todos los lenguajes de programación y nos dejan realizar algo según una condición.

La estructura if permite que un programa ejecute unas instrucciones cuando se cumplan una condición.

La estructura de control if else permite que un programa ejecute unas instrucciones cuando se cumple una condición y otras instrucciones cuando no se cumple esa condición.

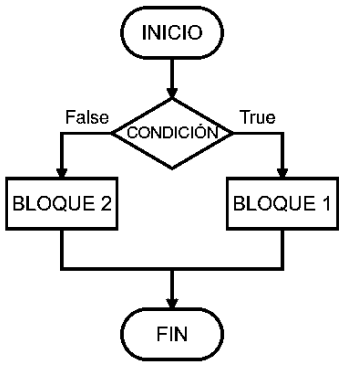

Figura 2. Diagrama de flujo sentencia if-else.

Fuente: elaboración propia.

API

La expresión Application Programming Interface. Hace referencia a un traductor, cuya función es conectar sistemas, software y aplicaciones. Con las API es posible ofrecerle una experiencia de uso más familiar a las personas.

Las API le permiten al usuario final utilizar una aplicación, software o incluso una simple hoja de cálculo, consultando, cambiando y almacenando datos de diferentes sistemas, sin que el usuario tenga que ingresar a ellos, directamente.

El propósito de una API es intercambiar datos entre diferentes sistemas, la mayoría de las veces estos intercambios de datos tienen como objetivo

automatizar procesos manuales y / o permitir la creación de nuevas funcionalidades.

Flask

Flask es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.

Figura 3. Flask.

Fuente: Wikipedia, 2015, web.

API Rest

la API utiliza requisiciones HTTP responsables de las operaciones básicas necesarias para la manipulación de datos.
Siendo un conjunto de restricciones que se utilizan para que las solicitudes HTTP cumplan con las directrices definidas en la arquitectura.
Básicamente, las restricciones determinadas por la arquitectura Rest son:

•**Cliente-servidor:** las aplicaciones existentes en el servidor y el cliente deben estar separadas.

• **Sin estado:** las requisiciones se realizan de forma independiente, es decir, cada una ejecuta solo una determinada acción

• **Caché:** la API debe utilizar la caché para evitar llamadas recurrentes al servidor.

• **Interfaz uniforme:** agrupa otros cuatro conceptos en los que se determina que los recursos deben ser identificados, la manipulación de los recursos debe ser a través de la representación.

Las principales solicitudes son:

- POST: crea datos en el servidor;
- GET: lectura de datos en el host;
- DELETE: borra la información;
- PUT: registro de actualizaciones.

Protocolo HTTP

El Protocolo de Transferencia de HiperTexto (Hypertext transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP.
La especificación completa fue propuesta por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como el World Wide Web.
Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del cliente Web.

- El cliente Web descodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.

- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente. Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD,…), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (casi siempre HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las

capacidades del browser, datos opcionales para el servidor.

- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información.

- Se cierra la conexión TCP.

Expresiones Regulares

Las expresiones regulares son patrones que se utilizan para hacer coincidir combinaciones de caracteres en cadenas. Las expresiones regulares son patrones utilizados para encontrar una determinada combinación de caracteres dentro de una cadena de texto. Las expresiones regulares proporcionan una manera muy flexible de buscar o reconocer cadenas de texto.

Para cada regex existe un denominado autómata finito (también conocido como máquina de estado finito) que acepta el lenguaje especificado por la expresión y que, con ayuda de la construcción de Thompson, se desarrolla a partir de una expresión regular. Por otro lado, para cada autómata finito también hay una expresión regular que describe el lenguaje aceptado por el autómata. Este puede generarse bien con el algoritmo de Kleene o bien con la eliminación de estados.

Archivos XML

XML consiste en un lenguaje de marcado que tiene la finalidad de definir una sintaxis para la codificación de documentos, que tanto los usuarios como las propias máquinas en sí puedan ser capaces de leer.

Lo hace mediante la utilización de una serie de etiquetas que definen la estructura que posee el documento en cuestión, además de cómo debe ser transportado y almacenado.

Se parece mucho al lenguaje de marcado de hipertexto (HTML), usado esencialmente para la codificación de las páginas web, y que utiliza un conjunto de símbolos de marcado predefinidos que describen el formato que posee el contenido de una página web.

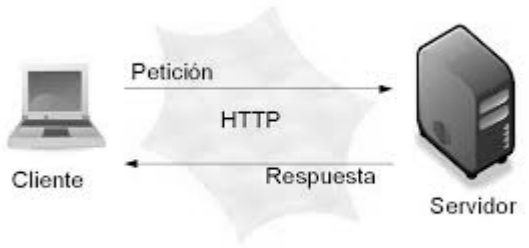

Figura 5. Servidor Cliente.

Fuente: Unypython, 2015, web.

DJANGO

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

Django usa un componente basado en la arquitectura "shared-nothing" (cada parte de la arquitectura es independiente de las otras, y por lo tanto puede ser reemplazado o cambiado si es necesario).

Django está escrito en Python, el cual se ejecuta en muchas plataformas. Lo que significa que no está sujeto a ninguna plataforma en particular, y puede ejecutar sus aplicaciones en muchas distribuciones de Linux, Windows y Mac OS X.

Figura 6. Estructura del l Proyecto.

Fuente: Elaboración propia.

Algoritmo calculo de NIT´s

1. Se multiplica cada carácter por su posición respectiva (siendo la posición 1 el carácter más a la derecha), excepto el primero.
2. Se suma todos los resultados.
3. Se obtiene el módulo 11 de la sumatoria.
4. A 11 se resta el resultado obtenido en el punto 3.
5. Se calcula el módulo 11 del resultado obtenido en el punto No. 4, si este resultado es 10, entonces el dígito verificador del NIT deberá ser K.

Conclusiones

Se ha reconocido que existen varios métodos para levantar un servidor y el uso de Flask simplifica varias cosas puesto que al conectarlo con el frontend creado en Django no se notó mucha complejidad, siendo práctico su uso.

En este tipo de aplicaciones web no se necesita recargar la página para obtener datos momentáneos, aunque no se considera una ventaja específica del desarrollo basado en REST, con el que podemos conseguir aplicaciones web que se asemejan más a aplicaciones de escritorio.

Python es un lenguaje de programación fácil de entender y programar con variedad de librerías de las cuales nos ha servido para la resolución del problema y construcción de clases necesarios para el funcionamiento del algoritmo.

La información viaja por medio de un archivo XML y varias respuestas parseadas JSON convirtiendo el proyecto en escalable.

Al hacer peticiones al servidor se obtienen datos planos, considerando menor tiempo de transferencia comparándolos al recibir los datos en conjunto con Html, css y Javascript.

Referencias bibliográficas

C. Clase, (2013, 20 octubre). *C Con Clase | Estructuras de datos (cap4)*. Estructura de Datos (TDA). http://conclase.net/c/edd/cap4?cap=004

C. S. Severance, (2009). *Python para informaticos*. Explorando la informacion, version 2.7.2, Inc.

C. Larman, (2003). *Introducción al análisis y diseño orientado a objetos*, Hall Prentice.

J. A. Luis, (2008). *Fundamentos de Programación*. McGraw-Hill, Inc.

R. Hernandez, (2001). *Estructuras de datos y Algoritmos*, Hall Prentice.

ANEXOS

Interfaz Gráfica (Frontend)

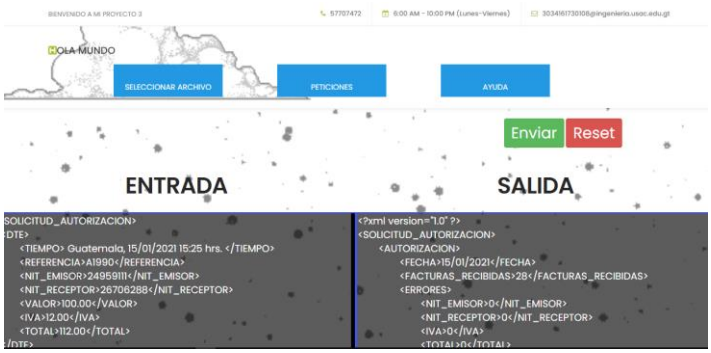

Figura 6. Interfaz Gráfica del Proyecto.

Fuente: Elaboración propia.