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

202010406 – Dennis Mauricio Corado Muñóz

Resumen

Se realizo el desarrollo de un proyecto de desarrollo web utilizando las tecnologías de Django, JavaScript y Flask para su implementación. El objetivo del proyecto era construir un software que pueda ser consumido desde internet como un servicio, el software recibe un archivo que contiene datos para solicitar las autorizaciones de documentos tributarios electrónicos, que emite un contribuyente.
Para el desarrollo de este proyecto se hizo uso de programación orientada a objetos en el área del servidor y para el control de las vistas se hizo uso de Django, creando url's que permitan el acceso a diferentes vistas en el proyecto y así poder realizar diferentes actividades.

Palabras clave
HTTP, API, XML, JSON

Abstract

The development of a web development project was carried out using Django, JavaScript and Flask technologies for its implementation. The objective of the project was to build software that can be consumed from the internet as a service. The software receives a file containing data to request authorizations for electronic tax documents, which are issued by a taxpayer.
For the development of this project, object-oriented programming was used in the server area and for the control of the views, Django was used, creating urls that allow access to different views in the project and thus be able to carry out different activities .

Keywords
HTTP, API, XML, JSON

Introducción

Durante el desarrollo del proyecto se hizo uso de paradigmas de programación así como buscar una correcta comunicación entre nuestro frontend y backend, se utilizo el microservicio de Flask para poder proveer un backend mas adecuado a la situación y al manejo de datos en la aplicación, haciendo uso del protocolo HTTP se realizaron consultad al backend para poder, desde procesar la información del archivo enviado hasta poder generar gráficas, pdfs y ver un mejor desarrollo de la información, durante el desarrollo del proyecto se busca dejar de una forma clara tanto para un usuario final como para un desarrollador conocer la forma en la que se implementó la solución.

Desarrollo del tema

El proyecto No 3 del curso de introducción a la programación y computación 2, tenia como objetivo general el desarrollo de una solución integral que implemente un API que brinde servicios utilizando el Protocolo HTTP bajo el concepto de programación orientada a objetos y haciendo uso de una simulación de base de datos por medio de un archivo creado en base al archivo que se sube al inicio de la aplicación, este archivo recaba la información mas importante y realiza verificaciones para los diferentes campos del archivo inicial, se busco implementar una API a través de Python que pueda ser consumida utilizando el protocolo HTTP, construir software haciendo uso de OOP, utilización e implementación de archivos con extensión XML para el manejo de la información y hacer uso de expresiones regulares para extraer el contenido de textos como lo son la fecha.

El proyecto al ser iniciado debe recibir un archivo de entrada con extensión XML que contiene información como, lugar y fecha, referencia, nit emisor, nit receptor, valor, IVA y total. Estos campos deben ir incluidos en el archivo de entrada, todos ellos en conjunto corresponden a un documento electrónico tributario (DTE) y el archivo de entrada puede incluir de un DTE hasta muchos, según se necesite.

Una vez teniendo clara una descripción de los archivos de entrada a ser utilizados en el proyecto, se puede mencionar un poco de la estructura del frontend, la estructura de este es un poco simple buscando así proporcionar de forma más entendible y rápida lo que el usuario necesite, en la sección de anexos se podrá observar por medio de imágenes la forma en la que este luce en sus diferentes opciones, la pagina principal de la aplicación, cuenta con un NAV que contiene 3 diferentes opciones, cargar archivo, peticiones y ayuda. La opción cargar archivo como su nombre lo indica permite la subida de un archivo para poder ser procesado en el área del backend, posteriormente se tiene la opción de peticiones, este es un combobox que contiene 3 opciones:

1. Consultar datos, permite mostrar los datos ya procesados del archivo en una estructura XML, este texto contiene la información recibida por diferentes DTE’s en una determinada fecha, se realiza un análisis para verificar que todo venga correcto, sin embargo esto se vera mas a detalle

cuando se realice la descripción del backend de la aplicación.

2. Graficar, esta opción redirige a una nueva página que contiene información que puede ser seleccionada por el usuario para generar 2 tipos de gráficas, una gráfica contiene lo que es el resumen de IVA por fecha y NIT, y la otra contiene lo que es un Resumen por rango de fechas, la información que van a contener estas va a ser decidida por el usuario al seleccionar que fecha desea ver.

3. Reporte, esta opción genera un reporte de las graficas seleccionadas por el usuario dicho reporte es archivo pdf que contiene dichas graficas.

Siguiendo con el desarrollo del frontend, se tienen en la pagina principal 2 áreas de texto, el área que tiene como titulo entrada despliega en forma de texto el contenido del archivo cargado por el usuario, la segunda área que tiene como titulo salida muestra la información del archivo de salida al ser seleccionada la opción de consultar datos, también se tiene 2 botones uno azul que dice “enviar” y uno rojo que dice “reset”, el botón azul envía la información de la caja de texto entrada a ser procesada por el backend y el botón reset borra información que exista, por ejemplo el archivo que se crea al enviar información cuya función en ser una simulación de base de datos de donde se puedan obtener datos más adecuados.

El botón de ayuda es un combobox que contiene dos opciones,

1. Información del estudiante, esta opción redirige al usuario a otra pagina dentro de la aplicación donde se muestra en forma de tabla información pertinente al estudiante que desarrollo la aplicación.

2. Ver documentación, esta opción como su nombre lo indica permite el despliegue de la documentación de la aplicación, redirige al usuario a otra pagina donde se puede ver el PDF de la documentación desplegado.

La parte de las vistas y los redireccionamientos son funciones del software de Django que permite este tipo de implementaciones en el área del frontend.

Por la parte del frontend estas son todas las funciones con las que cuenta la aplicación buscando ser simple y entendible para el usuario.

Todos los datos recopilados en el frontend se envían al backend haciendo uso de un archivo de JavaScript, este archivo contiene la url que utiliza Django para desplegar la aplicación y hace uso del DOM para poder obtener los datos necesario, cuando se hace uso del botón cargar archivo este detecta la acción y llena la primer caja de texto con la información del archivo, luego al presionar el botón enviar hace uso de fetch para enviar una petición de tipo POST al backend y así obtener como respuesta el texto del archivo de salida, al igual que cuando se presiona el botón reset, se detecta cuando fue presionado y se envía una petición al backend de tipo DELETE esta petición borra el contenido de las cajas de texto y elimina el archivo que funciona como base de datos, así mismo cuando se desea consultad los datos del archivo de entrada este realiza la petición y llena la segunda caja de texto con la información del archivo de salida.

Por otro lado se tiene el backend, este se encarga de todo lo que es el procesamiento de la información enviada desde el frontend.

Para el desarrollo del backend se hizo uso de Flask para poder manipular las distintas entradas y envíos de información de una manera más simple y entendible, todo inicia por el archivo app.py, este archivo contiene métodos definidos por medio de rutas esperadas dentro de la aplicación.

Tenemos dos listas al inicio de la aplicación cuya información va a ser manejada como dos variables globales, estas listas van a almacenar la información ya procesada y convertida en objetos tanto del archivo de entrada como del archivo de salida, la primer ruta hace uso del método post y sirve para manejar el archivo de entrada, haciendo uso de la programación orientada a objetos se crean dos objetos, uno de tipo Reader y otro de tipo Analizador, el primero objeto se encarga de extraer la información del archivo de entrada y convertirlo en diferentes objetos para un análisis mas simple, el segundo objeto se encarga de hacerle un análisis a cada parte de los diferentes objetos creados en la aplicación, este análisis incluye, extraer la fecha adecuada, sin el lugar o la hora, las referencias, nits, valores, iva y totales de todo el archivo inicial.

Otra ruta contenida en el archivo es la de delete file, esta ruta como su nombre lo indica se encarga de eliminar el archivo generado al analizar el archivo de entrada, elimina el archivo y por ende estaría borrando la base de datos.

La ruta get info, proporciona la información contenida en el archivo autorizaciones.xml envía el contenido del archivo en forma de texto para poder ser mostrado por el frontend.

Y por ultimo se tiene la ruta get dict, devuelve en formato JSON la información del archivo de salida, esta para ser utilizada en las gráficas.

Dejando de lado las rutas que maneja la aplicación de flask para el envío y análisis de la información, se puede dar un vistazo mas de cerca a los procesos internos que son llevados a cabo en la aplicación.

Uno de los archivos principales y mas importantes de las aplicación es el archivo lector, este archivo tiene una única clase nombrada ‘Reader’ esta clase tiene distintos métodos el principal de todos es el método ‘read_file’ este método hace uso de la librería Element Tree para extraer la información del archivo XML y dentro de este método se manda a llamar a los demás, tenemos el método get_numeros, este método se encarga de descartar los espacios en blanco que posee cualquier secuencia de caracteres, posteriormente se tiene el método ‘get_tiempo’ el cual hace uso de expresiones regulares para poder obviar el texto como el lugar y la hora, estos elementos se encuentran en la parte de fecha del documento y este método retorno lo que es la fecha en forma dd/mm/yyyy, una vez descritos el funcionamiento de estos métodos vamos a continuar describiendo el primer método, pues al pasar por cierto análisis el texto obtenido del documento se almacena en variables dicho contenido y se almacena en una lista objetos de tipo factura.

Para los objetos de tipo factura se tiene otra clase dentro de este mismo archivo denominada

'Factura', dicha clase se encarga de obtener como parámetros los campos que puede contener cada factura así como un método 'to_dict' que se encarga de transforma dicha en información en formato JSON para su procesamiento en el lado del frontend.

Se tiene también el archivo analizador.py este archivo contiene cierta cantidad de clases, estas clases en su mayoría son para un almacenamiento más entendible y manejable, entre las clases que se encuentran dentro de este archivo esta la clase 'Autorizacion' esta clase maneja la información de cada DTE que fue autorizada, así como la lista de aprobaciones por las facturas que no contengan error, esta clase tiene dos métodos, el método 'to_dict' que como se menciono arriba sirve para convertir la información a formato JSON y tiene uno extra que es 'get_apr' este método se encarga de recorrer la lista que contiene el objeto autorización y convertir dichos objetos también a formato JSON. Posteriormente en este archivo se puede observar también la clase 'Error' que se va a encargar de encapsular la información de los errores detectados por cada autorización, y la clase Aprobacion, esta clase almacena la información pertinente a cada aprobación es decir a cada factura que no contenga errores y también cuenta con el método 'to_dict.

La clase principal del archivo y por la cual se le da el nombre a este, es la clase 'Analizador' dicha clase en encarga de realizar un análisis a las diferentes variables de cada DTE, si existen errores se van aumentando los contadores, primero se almacena la información de las fechas, buscando omitir las fechas repetidas y posteriormente se analiza cada DTE que concuerde con cada fecha para poder crear su respectivo objeto Autorización, se hace uso de ciclos para poder leer cada autorización si alguna factura cumple con uno de los errores como que la suma del iva con el valor no den el resultado del total que exista una referencia duplicada en el mismo día, que el valor del iva no sea el correcto al multiplicar el valor por 0.12 o que el nit no sea valido haciendo uso del algoritmo 'Modulo 11' para determinar si el NIT tanto receptor como emisor son válidos, si la factura cumple con no tener ninguno de estos errores entonces se procede a guardar dicha factura en un arreglo de objetos de tipos aprobación señalando que dichas facturas no contienen ningún error y por ende pueden ser procesadas.

Como ultimo archivo utilizado en nuestra API, tenemos el archivo escribir.py, este archivo contiene la clase 'Escribir' la cual tiene un método 'escribir_salida' que recibe como parámetro la lista de autorizaciones, se hace uso de la librería Element Tree y se hace un recorrido de dicha lista para poder crear el archivo 'autorizaciones.xml' con una correcta estructura y con la información necesaria para ser procesada y mostrada en la interfaz del usuario, esta clase también se apoya en un método denominado 'indent' cuyo objetivo es darle una identacion apropiada al archivo y que cuando este sea abierto en un editor de texto contenga los saltos de línea que faciliten su comprensión.

Esta es la descripción de la lógica utilizada en el desarrollo de esta aplicación esperando dar un mayor entendimiento para la persona que lo lea, se busco una mayor modularidad del código y

segmentarlo de una forma entendible tanto para el usuario como para el programador.

Conclusiones

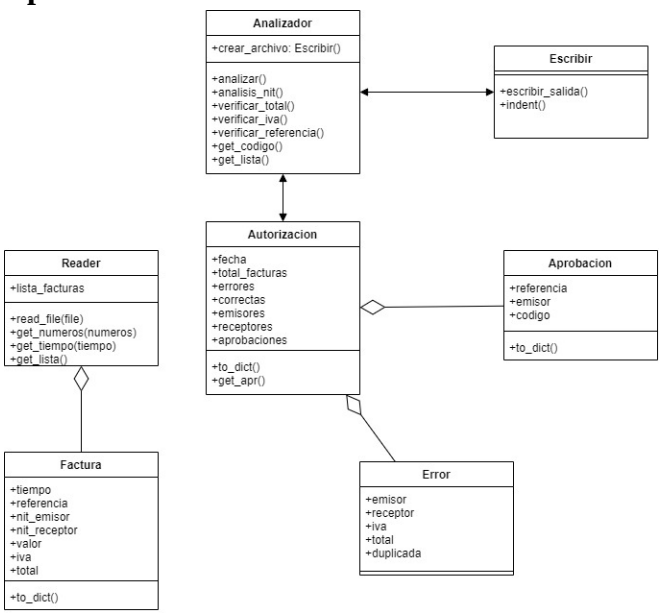
Las aplicaciones web facilitan la transmisión de información desde empresas hasta usuarios finales, el correcto desarrollo de estas aplicaciones e interfaces agradables hacen que la aplicación tenga un mejor recibimiento, haciendo uso de API's se puede tener una forma mas simple de desarrollar estas aplicaciones y así obtener una aplicación mas segmentada y modular siendo más fácil de entender por quien desee mejorar algo de esta.

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

C. J. Date, (1991). *An introduction to Database Systems.* Addison-Wesley Publishing Company, Inc.

Apéndices



Apéndice 1. Diagrama de clases Proyecto 3.