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: PROGRAMACIÓN WEB

202000166 – Gerson Rubén Quiroa del Cid

Resumen

En el proyecto presente, se realizó un servicio web para la Superintendencia de Administración Tributaria.

Este software recibirá un mensaje con los datos para solicitar la autorización de un Documento Tributario Electrónico (DTE) emitido por un contribuyente y como respuesta emitirá un número único de autorización, este número será un correlativo que iniciará con el valor 1 cada día y no deberá repetirse de nuevo en ese día.

Para crear este servicio, se utilizaron diferentes Frameworks como lo son Django para el frontend y Flask para el backend. Como base de datos se utilizó un archivo de tipo xml y para las plantillas se utilizó html5 y css3.

La lógica del sistema fue implementada en el lenguaje de programación Python donde se desarrolló todo el proyecto.

Palabras clave

Base de datos, Flask, Django, api, frontend, backend.

Abstract

In the present project, a web service was created for the Superintendency of Tax Administration.

This software will receive a message with the data to request the authorization of an Electronic Tax Document (DTE) issued by a taxpayer and in response it will issue a unique authorization number, this number will be a correlative that will start with the value 1 every day and should not be repeated again on that day.

To create this service, different Frameworks were used such as Django for the frontend and Flask for the backend. An xml file was used as the database and html5 and css3 were used for the templates.

The system logic was implemented in the Python programming language where the entire project was developed.

Keywords

Database, Flask, Django, api, frontend, backend.

Introducción

El presente proyecto tiene como finalidad proveer un servicio web a la Superintendencia de Administración Tributaria (SAT). Para ello se creo una página web donde el usuario podrá pedir solicitudes para un Documento Tributario Electrónico (DTE).

Básicamente el software, pide al usuario una solicitud de DTE al usuario, mediante un archivo xml con una estructura específica y si las facturas están correctas se autorizan las mismas, de lo contrario se sumarán a las facturas rechazadas.

Luego de procesar las solicitudes, el usuario puede elegir diferentes opciones como, por ejemplo, mostrar las autorizaciones, crear resúmenes que se representaran por medio de gráficas, crear reportes y un apartado de ayuda.

Desarrollo del proyecto

El análisis del presente proyecto se empezó primero que nada investigando acerca de las herramientas que se utilizarían en la misma como pueden ser la utilización de Django, Flask, la creación de apis, algunas librerías como matplotlib para el manejo de gráficas, archivos xml, la creación de apps por medio de Django, etc.

Después, se comenzó a elaborar el diagrama de clases de nuestro proyecto, como una visión general de todo el programa, creando las clases que usaremos en lo largo del proyecto. Dicho diagrama se puede visualizar completo en la sección de Apéndices.

Pasaremos a describir brevemente las clases que se utilizaron:

Clase views:

Esta clase la crea automáticamente Django para el manejo de las diferentes views o páginas que tengamos en nuestra aplicación.

Tiene una única variable/constante que se llama endpoint de tipo String ya que nos servirá para conectar la parte del frontend con el backend. Cabe resaltar que lo que contiene el endpoint es la url en donde se encuentra alojado nuestro backend.

Los métodos que tiene esta clase son muy parecidos todos, pero cada función tiene su función particular.

La función index es la encarga de renderizar la página principal de nuestra aplicación, principalmente manejará las peticiones ‘get’ y ‘post’ de la página principal, dependiendo de qué petición quiera el usuario hará diferentes cosas.

La función carga es la encargada de que cuando el usuario cargue una solicitud, la procese y la envíe a una función especial en el backend.

La función enviar procesa la solicitud (previamente cargada) y crea la base de datos.

La función reset es la encargada de limpiar la base de datos cada vez que se presione el botón de reset.

La función iva es la encargada de crear las gráficas con las opciones que elija el usuario, cuando se procese las gráficas renderizará una nueva página donde se mostrarán las gráficas.

Por último, la función regresar, esta función solo se encargará de regresar a la página principal cuando el usuario esté en la página de gráficas.

Clase api:

En esta clase se maneja nuestro backend mediante Flask, aquí estarán los servicios que el frontend necesitará y nos pedirá mediane peticiones http.

Tiene dos variables, una es ‘app’ que es la instancia para utilizar la librería de Flask, y la otra ‘cors’ para que reconozca las peticiones mandadas desde el frontend.

Costa de diferentes funciones, dependiendo si son peticiones get o post se ejecutará una función u otra. Las funciones son las siguientes:

Función get_datos es la encargada de leer las solicitudes cuando ya esté cargada previamente. Luego se devolverá al frontend mediante un objeto Response.

La función post_datos se ejecutará cuando la petición sea de tipo post. Es la encargada de crear un archivo llamados ‘solicitud.xml’ y en él escribirá el archivo que le manden como entrada el usuario.

La función post_procesos llama a la función incioProceso que se encuentra en la clase procesos.py.

La función get_procesos leerá un archivo llamado ‘autorizacion.xml’ que contendrá la base de datos y la mostrará en la página principal.

La función reset solamente leerá el archivo ‘autorización.xml’ y escribirá una cadena vacía para borrar todos los datos.

La función get_fechas es la encargada de mostrar las fechas de las autorizaciones para que el usuario pueda filtrar cuando desee graficar los datos de la base de datos.

Por último, tenemos la función get_graficas que es la encargada de recibir los parámetros que el usuario eligió para realizar las gráficas y mandarlos a la clase procesos.py.

Clase urls:

Esta clase es muy sencilla, Django la crea automáticamente cuando se genera un proyecto para poder manejar las diferentes páginas que podamos tener, mas bien dicho para manejar todas las views que tengamos en la clase views.py.

Clase procesos:

Probablemente este módulo sea el más importante del programa, también el más extenso. Aquí se maneja toda la funcionalidad del proyecto.

Son muchas las funciones las que se encuentran en este archivo, las iremos mencionando brevemente:

Función inicioProceso: es función es la que crea la base de datos, también inicializa una lista que será donde almacenemos todas las autorizaciones.

Función xmlF y databaseCarga: estas dos funciones son muy parecidas, ambas tienen como función leer los archivos xml que genere el programa. Con la diferencia de que la función xmlF leerá el archivo de entrada, mientras que datbaseCarga leerá la base de datos ya tengamos almacenado.

Función verificaciones: esta función es la encargada de verificar si el archivo de entrada contiene facturas correctas, si en alguno de los campos detecta un error, esa factura será rechazada, de lo contrario se aceptará la factura.

Función verificacionNIT: esta función es especialmente para que verifique los nits de la solicitud son correctos.

Función autorización: aquí se filtran las facturas que son correctas de las que no, también se crean las fechas que se utilizaran para las autorizaciones.

La función facRechazado: esta función rechaza las facturas donde hubo campos incorrectos, si se

rechaza una factura, solamente se activa una bandera de rechazo.

Función contadores: esta función es la encargada de aumentar los contadores de las facturas que estén erróneas, dependiendo de error que tengan se irá aumentando el contador en diferentes campos.

Función fechasF: esta función es la encargada de generar las fechas para las autorizaciones crear contadores dependiendo de la fecha.

Función salida: esta función es la que crea la lista de autorizaciones que se mostrará al usuario cuando éste lo requiera.

Función databaseSalida: esta función es la encargada de crear la base de datos que nos ayudará a manejar las diferentes peticiones que el usuario desee. Esta función prácticamente es la misma que la función salida, pero con otras etiquetas extras que nos ayudarán al manejo del software.

Función fechasHTML: esta función es la que devuelve una lista de fechas en orden para poder mostrarlos al usuario y elija la filtración por fechas para generar las gráficas.

Función graficas: esta función es el encargado de leer la base de datos y de generar las dos gráficas que de los resúmenes.

Función resumen1 y resumen2: estas funciones son las encargadas de generar las gráficas que el usuario necesita visualizar, cada función se encargará de hacer una gráfica con la ayuda de la librería matplotlib.

Por último, una función llamada ordenar: solo se encargará de ordenar las autorizaciones por fechas de menor a mayor.

Clase Factura, Autorización y Error:

Estas clases se usarán como objeto para implementar en el proyecto.

Conclusiones

Se aprendió a usar diferentes Frameworks que nos ayudarán para realizar aplicaciones web, tanto en la parte del frontend como en la parte del backend.

Se modeló una página web estética mediante html y css y Bootstrap, por lo que se aprendió a usar dichas tecnologías.

Se aprendió a procesar archivos tipo xml, a guardar la información del archivo en objetos y listas.
Así como a escribir archivos del mismo tipo xml, con su estructura característica.

Se comprendió el uso de la herramienta matplotlib para crear gráficas dinámicamente y así tener una visualización mejor de la base de datos.

Referencias bibliográficas

Auxiliares de IPC2, (2021). *Proyecto 3.* Disponible en: https://bit.ly/3Gc8GN8

Flask, (2021). *Quickstart*. Flask.com. Disponible en: https://bit.ly/3jseetg

Django, (2021). *Documentation.* Django.com. Disponible en: https://bit.ly/3BivUO5

Bootstrap, (2021). *Documentation*. Bootstrap.com. Disponible en: https://bit.ly/3jvz76D

Apéndices

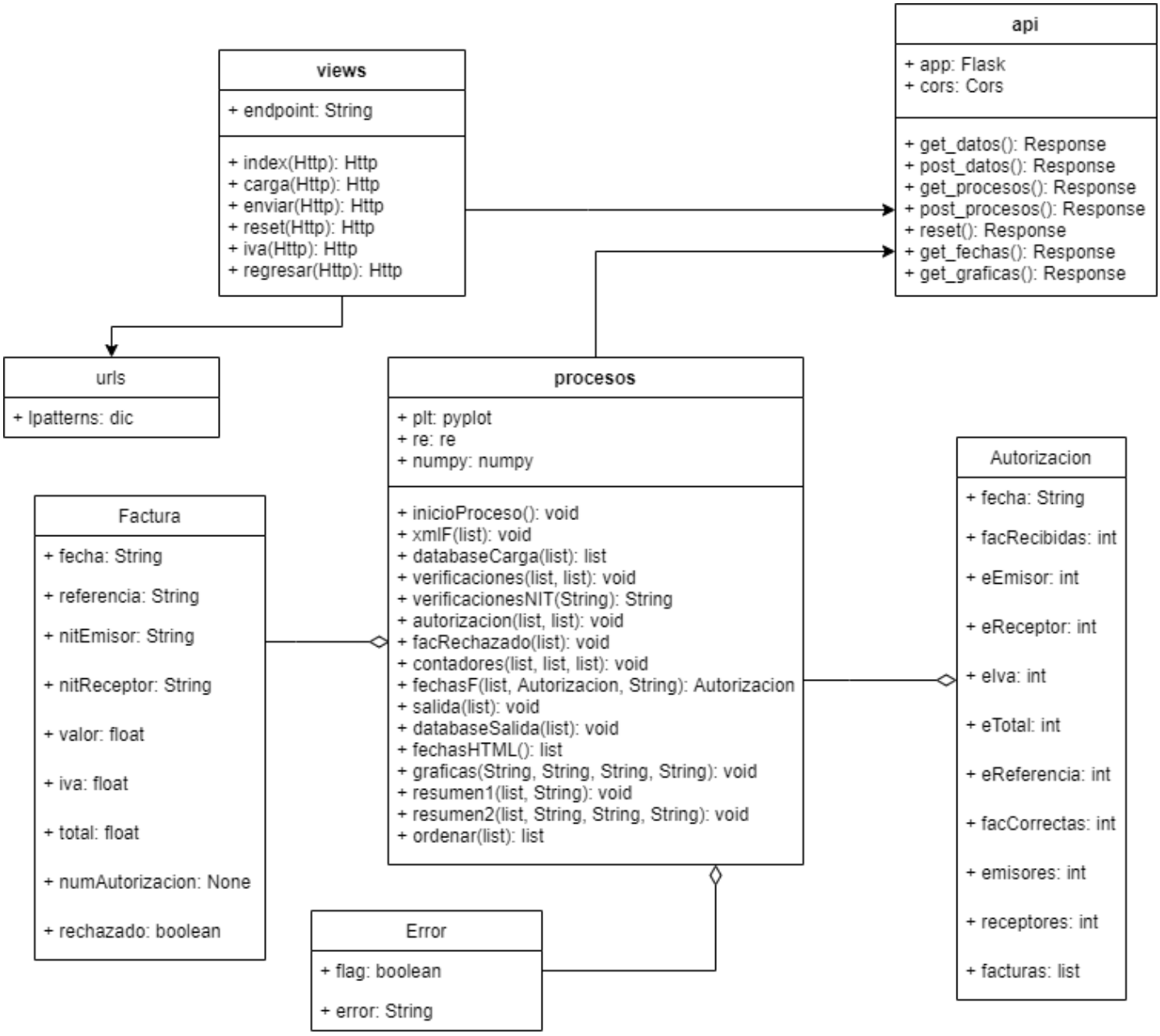

Figura 1. Diagrama de clases proyecto 3

Fuente: elaboración propia.