
PROYECTO 2 – DESARROLLO WEB – SUPERINTENDENCIA DE ADMINISTRACIÓN TRIBUTARIA

201701548 – Arnoldo Luis Antonio González Camey

Resumen

El proyecto consiste en el desarrollo de una solución integral que implemente un API la cual brinde servicios utilizando el protocolo HTTP. Para ello el software debe ser consumido desde internet, este recibe un mensaje con los datos para la autorización de un Documento Tributario Electrónico (DTE) emitido por un contribuyente y como respuesta emite un número único de autorización.

El software debe ser planteado de forma lógica y aplicando los conceptos de la programación orientada a objetos y el uso de una simulación de base de datos.

Para el desarrollo del software se hizo uso del framework Flask el cual se utiliza para recibir y enviar la información y mediante métodos y validaciones ir denegando o aceptando las distintas solicitudes. La página web fue desarrollada a través de Django y los servicios que este proporciona. A su vez se realiza el reporte de la salida de las simulaciones donde se detalla lo obtenido y la simulación de la base de datos se guarda en un archivo XML.

Palabras clave

Software – HTTP – Framework – POO – XML

Abstract

The project consists of the development of an integral solution that implements an API which provides services using the HTTP protocol. For this, the software must be consumed from the Internet, it receives a message with the data for the authorization of an Electronic Tax Document (DTE) issued by a taxpayer and in response issues a unique authorization number.

The software has to be developed logically and applying the concepts of object oriented programming and the use of a database simulation.

For the development of the software we made use of the Flask framework which is used to receive and send the information and by means of methods and validations to deny or accept the different requests. The web page was developed through Django and the services it provides. At the same time, a report of the output of the simulations is made, detailing what is obtained and the simulation of the database is saved in an XML file.

Keywords

Software – HTTP – Framework – POO – XML

Introducción

El siguiente proyecto consta del desarrollo de un API en el lenguaje de programación Python la cual brinda los servicios para ser consumidos a través de una página web, implementando los conceptos de programación orientada a objetos, implementación del protocolo HTTP, utilización de bases de datos y el uso de XML. El software recibe un mensaje con los datos para solicitar la autorización de un Documento Tributario Electrónico (DTE) emitido por un contribuyente a través de una página web.

Para realizar la comunicación Cliente – Servidor se hizo uso del framework Flask el cual implementa servicios y métodos los cuales permiten enviar y recibir información. La página web se desarrolló en Django el cual consta con servicios que permiten la manipulación de la información; este fue elaborado en HTML y se hizo uso del lenguaje de programación Python y JavaScript.

Se realiza la simulación de una base de datos a través de un archivo XML en el cual se guardan las solicitudes aceptadas previamente y se escribe un archivo de salida en el que se almacenan los datos obtenidos.

Desarrollo del tema

Se desarrolló para La Superintendencia de Administración Tributaria un software que pueda ser consumido desde internet como un servicio. Este software recibe un mensaje y como respuesta emite un número único de autorización, este número es un correlativo que inicia con el valor 1 cada día y no se repite de nuevo en ese día. El número consta de una estructura de autorización la cual es el año, mes, día

y por último una sucesión de ocho números en el orden correspondiente.

Al ingresar a la página web se le muestra al usuario una página principal en la cual debe de seleccionar la opción deseada entre cargar el archivo XML de las solicitudes, realizar las peticiones para los reportes deseados tanto dentro de una gráfica como reportes escritos o visualizar información adicional que se encuentra disponible en la página web.

a. Cargar Solicitudes

La información de las solicitudes se debe realizar a través de la carga de un archivo XML, en este se encuentra la hora, fecha, lugar, el NIT del contribuyente emisor y receptor, la referencia de la factura, el valor, el IVA y el total del monto. Si el usuario elige esta opción se procede a leer el archivo por medio de la librería requests y se envía al API el contenido recabado, luego dentro del API se recibe la información y por medio de ElementTree el cual es una librería de Python que permite analizar y navegar por un documento XML y mediante la sintaxis adecuada y un ciclo for se almacenan los datos en variables temporales y se realizan la instancia de los distintos métodos para validar si la información que viene a través de la página es correcta y así almacenarlos en la clase correspondiente. La lectura de la fecha, hora y lugar se realiza a través de la clase Analizador en la cual se van analizando el estado del token actual y mediante validaciones se guarda el lexema correspondiente.

b. Validaciones

Para almacenar la autorización del documento tributario electrónico se deben realizar previamente

una serie de validaciones las cuales permiten verificar si la información es correcta.

- Fecha: El formato de la fecha debe ser el siguiente: dd/mm/yyyy hh24:min, el cual consta de la fecha seguido de la hora en un formato de 24 horas.
- Referencia: La referencia es un código interno enviado por el emisor el cual debe ser un texto único de máximo 40 posiciones.
- Valor: Este consta de un número real positivo de 2 posiciones decimales.
- IVA: Debe ser el resultado de multiplicar el valor por 0.12 y redondear dicha cifra a dos posiciones decimales.
- Total: Este debe ser el resultado de sumar el valor y el IVA.
- NIT emisor y receptor: Estos constan de una serie de dígitos, en el cual el último número debe ser la validación del NIT; la cantidad de dígitos debe ser entre 1 y 20.

Para la validación del NIT emisor y receptor se deben realizar un algoritmo verificar la veracidad de este.

1. Multiplicar cada carácter por su respectiva posición, siendo la posición 1 el carácter más a la derecha, exceptuando el primero el cual es el número de la validación del NIT ingresado.
2. Sumar todos los resultados obtenidos de la multiplicación.
3. Obtener el módulo 11 de la sumatoria, es decir dividir la sumatoria entre 11 y el residuo de la división ese es el dato deseado.
4. Al número 11 restarle el resultado obtenido en el paso anterior.

5. Calcular el módulo 11 del resultado obtenido en el punto 4, si este resultado de dicha operación es menor a 10 y el resultado es igual al número de la validación del NIT ingresado (el último dígito del NIT) entonces el NIT es válido y se procede con el resto de las validaciones.

c. Peticiones

Este apartado es para que el usuario pueda realizar las peticiones sobre distintos reportes y la información recopilada.

- Consultar datos: Al seleccionar esta opción se realiza la consulta de los datos de salida donde se encuentran los resultados de las solicitudes. Para ello se realiza un método POST en el cual se hace una petición para acceder a la información que tiene el API en cierto url, esta petición se recibe en el software, luego dentro de este se accede a un método el cual guarda en una lista las fechas que se encuentran registradas en el API, retorna dicha lista, para seguido de este recorrerla e ir obteniendo la información almacenada tanto en la clase DTE como en la clase Error las cuales contienen la información de los resultados de las validaciones realizadas con anterioridad. Al finalizar toda la información se almacena en una cadena la cual se retorna por medio de un jsonify a la página web para ser mostrada en un área de texto para que el usuario pueda visualizarla en la página web.
- Resumen de IVA: El usuario al seleccionar esta opción accederá a otra ventana en la cual debe ingresar la fecha en la cual desea obtener

el resumen de movimientos por NIT en esa fecha, esta información se mostrará al usuario en una gráfica.

Para realizar dicha petición se le envía al API la fecha en la cual se desea filtrar la información, luego esta recibe la información en formato Json, procede a obtener la fecha de este y hace la instancia de un método el cual se encarga de guardar en una lista un Json con la fecha, el IVA, el NIT emisor y receptor para posteriormente devolverle al Cliente el Json y por medio de la librería Chart.js realizar y visualizar la gráfica en pantalla.

- **Resumen por rango de fechas:** El usuario al seleccionar esta opción accederá a otra ventana en la cual debe ingresar un rango de fechas en la cual desea obtener los valores totales o los valores sin IVA, esta información se mostrará al usuario en una gráfica.

Para realizar dicha petición se le envía al API el rango de fechas en la cual se desea filtrar y el tipo de datos que desea obtener, luego esta recibe la información en formato Json, procede a obtener la información de este y hace la instancia de un método el cual se encarga de guardar en una lista un Json con la fecha y el valor deseado; para posteriormente devolverle al Cliente el Json y por medio de la librería Chart.js realizar y visualizar la gráfica en pantalla.

- **Guardar gráfica:** El usuario puede guardar la gráfica obtenida tanto en formato PDF como en una imagen, para esto se hace uso de un método en JavaScript el cual consta en guardar el lienzo donde se encuentra la gráfica

y convertirlo en PDF y lo mismo para convertirlo en una imagen PNG.

d. Ayuda

En esta opción el usuario puede escoger entre ver la información del estudiante quien realizó el proyecto o ver la documentación de este. Si se desea ver la información del estudiante se accede a esta opción y aparecerá una alerta en la parte superior con la información solicitada. Para ver la documentación se selecciona la opción, se da clic y se abrirá una nueva ventana con el pdf de la documentación.

e. Enviar

El usuario al seleccionar esta opción se enviará las solicitudes de autorización del archivo cargado con anterioridad, para ello se realiza un método POST el cual envía la petición, el API la recibe y procede a realizar las validaciones correspondientes.

f. Reset

El usuario en cualquier momento puede eliminar los datos almacenados dentro de la API, para ello se realiza un método POST el cual envía la petición, el API la recibe y procede a eliminar el contenido de la lista de autorizaciones.

g. Base de Datos

El software tiene la capacidad de almacenar la información de una forma persistente para ello simula una base de datos utilizando un archivo XML para este propósito, este funciona de la siguiente forma: por medio de un método se accede a otro el cual se van guardando en una lista las solicitudes autorizadas con anterioridad, las cuales se encuentran en formato Json, luego se recorre dicha lista por medio de un ciclo for en el cual se va concatenando la información

pertinente e ir formando así la estructura del archivo XML; al finalizar se procede a retornar la cadena obtenida y por medio de otro método el cual hace uso de los métodos open y write los cuales son utilizadas para escribir un archivo. El archivo se sobrescribe cada vez que una autorización es aprobada.

g. Archivo de Salida

El archivo de salida el cual contiene la información recaba a lo largo del proceso se procede a escribir al momento de finalizar con la ejecución del programa. Para guardarlo se recorren los métodos necesarios y se guarda la estructura básica de un archivo XML en variables temporales. Posteriormente se hace la instancia de un método encargado de recorrer el proceso de elaboración y guardar el nombre del producto en una variable que se retorna al finalizar dicho método. Al finalizar se unen las variables en una sola y se hace la instancia al método open y write los cuales son utilizadas para escribir un archivo.

Conclusiones

- La utilización de un framework como Flask para el envío y recepción de peticiones por parte de la página web facilita mucho la forma y el manejo de la información que se envía y se recibe,
- El manejo de archivos en formato Json es una gran ventaja para llevar un control optimo de la información y datos que se desean enviar y recibir desde el API ya que este formato cuenta con un nombre identificador lo que facilita la interpretación de la información
- El uso de la librería requests y el uso de JavaScript para realizar los métodos POST y GET otorga facilidad al manejo de

información debido a la simpleza que estos manejan.

- La utilización de la librería Chart.js para la visualización gráfica de la información es una gran ayuda para visualizar los datos obtenidos a lo largo de la ejecución de la página web.

Anexos

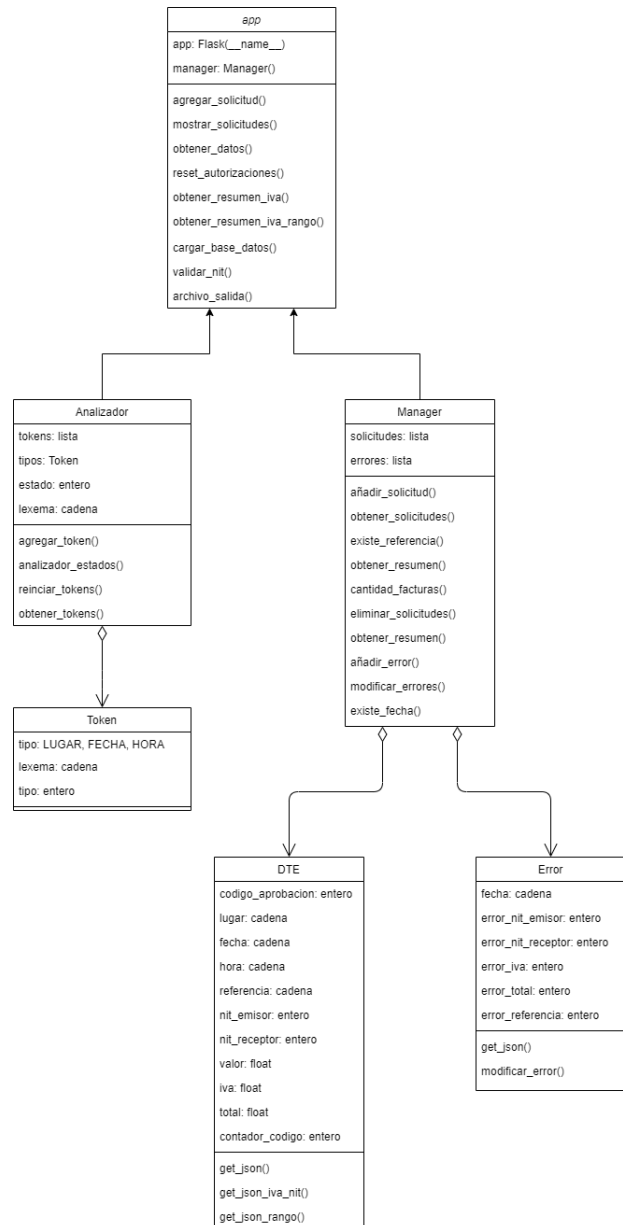


Figura 1. Diagrama de Clases

Fuente: elaboración propia

SAT

Home

Superintendencia de Administración Tributaria

Autorización de Documento Tributario Electrónico (DTE)

Cargar Autorización:

Seleccionar Archivo:

Choose File

No file chosen

Enviar

Consultar Datos

Peticiones

Información Estudiante

Ayuda

Reset

Arnoldo Luis Antonio González Camey

© 2021

Figura 2. Página principal

Fuente: elaboración propia

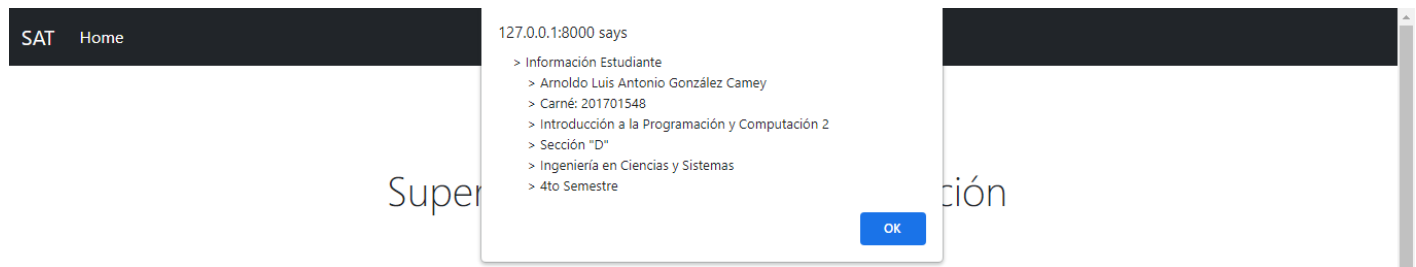


Figura 3. Información estudiante

Fuente: elaboración propia

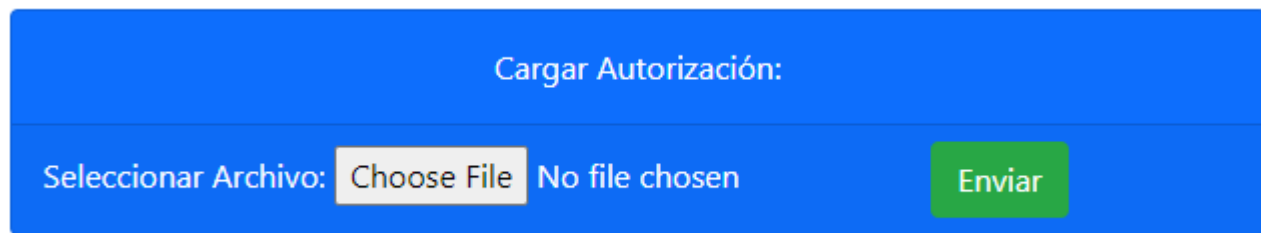


Figura 4. Carga Autorización

Fuente: elaboración propia



Figura 5. Página petición resumen IVA

Fuente: elaboración propia

SAT Home

Petición Rango Fechas

Autorización de Documento Tributario Electrónico (DTE)

Rango Menor

Rango Mayor

Seleccione el Tipo de Valores

Valores Totales ▾

Obtener Descargar Gráfica

Figura 6. Página petición rango fechas

Fuente: elaboración propia

```
<SOLICITUD_AUTORIZACION>
  <DTE>
    <TIEMPO>Guatemala, 15/11/1992 10:55 hrs</TIEMPO>
    <REFERENCIA>AZ1232</REFERENCIA>
    <NIT_EMITOR>39888320</NIT_EMITOR>
    <NIT_RECEPTOR>24284955</NIT_RECEPTOR>
    <VALOR>100.25</VALOR>
    <IVA>12.03</IVA>
    <TOTAL>112.28</TOTAL>
  </DTE>
</DTE>
```

Figura 7. Archivo de Entrada Solicitudes

Archivo XML con las solicitudes

Fuente: elaboración propia


```
<LISTAAUTORIZACIONES>
  <AUTORIZACION>
    <FECHA>15/11/1992</FECHA>
    <FACTURAS_RECIBIDAS>3</FACTURAS_RECIBIDAS>
    <ERRORES>
      <NIT_EMITOR>2</NIT_EMITOR>
      <NIT_RECEPTOR>2</NIT_RECEPTOR>
      <IVA>2</IVA>
      <TOTAL>2</TOTAL>
      <REFERENCIA_DUPLICADA>2</REFERENCIA_DUPLICADA>
    </ERRORES>
    <FACTURAS_CORRECTAS>2</FACTURAS_CORRECTAS>
    <CANTIDAD_EMITORES>1</CANTIDAD_EMITORES>
    <CANTIDAD_RECEPTORES>1</CANTIDAD_RECEPTORES>
    <LISTADO_AUTORIZACIONES>
      <APROBACION>
        <NIT_EMITOR ref="AZ1232">39888320</NIT_EMITOR>
        <CODIGO_APROBACION>1992111500000000</CODIGO_APROBACION>
      </APROBACION>
      <APROBACION>
        <NIT_EMITOR ref="DZ1232">39888320</NIT_EMITOR>
        <CODIGO_APROBACION>1992111500000001</CODIGO_APROBACION>
      </APROBACION>
    </LISTADO_AUTORIZACIONES>
  </AUTORIZACION>
</AUTORIZACIONES>
```

Figura 8. Archivo de Salida

Archivo XML con la información de las facturas solicitadas

Fuente: elaboración propia

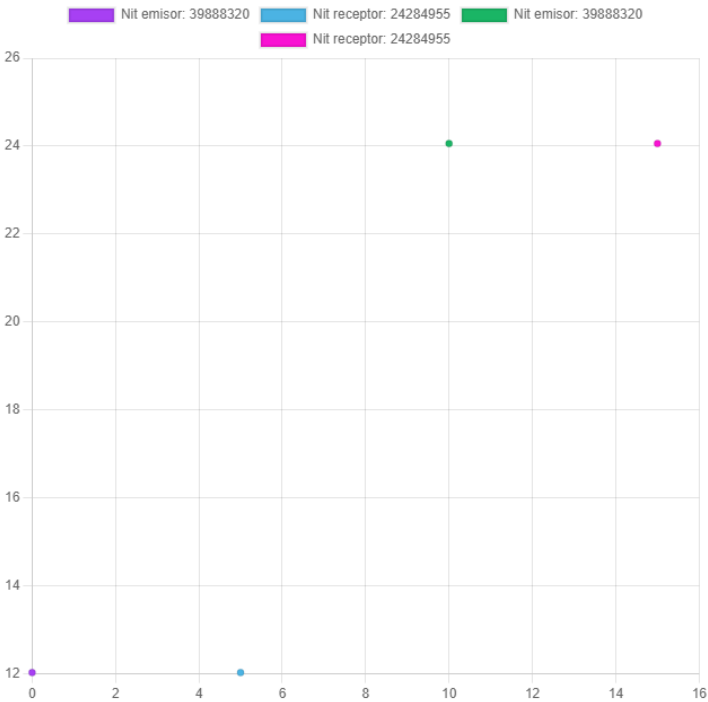


Figura 9. Gráfica Petición Resumen IVA

Fuente: elaboración propia