

Proyecto 3 – Superintendencia de administración tributaria

Carnet: 201603232 – Mynor Francisco Morán García

Resumen

El desarrollo de la aplicación se planifico de acuerdo a los requerimientos dados por la superintendencia de administración tributaria, en lo cual se consumirá un servicio desde internet. Este software recibirá un mensaje con los datos para solicitar la autorización de un Documento tributario electrónico (DTE) emitido por el contribuyente y como respuesta emitirá una respuesta de autorización, la respuesta que dará el sistema un número correlativo que iniciara con el valor de uno y no deberá de repetirse. También se implementará reportes PDF y gráficas para visualizar la cantidad de documentos tributarios electrónicos autorizados y no autorizados. La conexión del front-end y back-end es a través del protocolo HTTPS.

Este software está construido con la tecnología de Django para el front-end y Flask para el back-end. El lenguaje de programación que manejan cada uno de estos frameworks es Python. Para transmitir los mensajes entre los diferentes frameworks se utilizar los archivos XML.

Palabras clave

- Django.
- Flask.
- XML.
- HTTPS.

Abstract

The development of the application was planned according to the requirements given by the tax administration superintendency, in which a service will be consumed from the internet. This software received a message with the data to request the authorization of an electronic tax document (DTE) issued by the taxpayer and in response it will issue an authorization response, the response that the system will give a correlative number that begins with the value of one and it should not be repeated. PDF reports and graphs will also be implemented to view the amount of authorized and unauthorized electronic tax documents. The connection of the front-end and back-end is through the HTTPS protocol.

This software is built with Django technology for the front-end and Flask for the back-end. The programming language used by each of these frameworks is Python. XML files will be used to transmit the messages between the different frameworks.

Keywords.

- Django.
- Flask
- XML
- HTTPS.

Introducción

La solución de los requerimientos propuestos, es de gran importancia ya que se utilizaron herramientas que actualmente están siendo muy utilizadas a nivel comercial, entonces se entiende que este software está catalogado con la última tecnología que se encuentra en el mercado. Las herramientas son las siguientes: para el lenguaje de programación se utilizó Python y los frameworks que se utilizaron para poder tener servicios en internet. Django se utilizó para el front-end ya que nos permite tener templates (paginas HTML), utilizar url's personalizados y vistas para poder darle la dinámica a los templates. Flask se utilizó para manejarla como API, esto consiste en que Django solicite información a Flask por medio del protocolo de comunicación HTTPS. La API va a manejar todo el proceso de validación de los DTE y brindarle la respuesta a Django.

Desarrollo del tema

Se expondrá las herramientas utilizadas para las distintas funciones del programa.

1. Python.

Este es un lenguaje de programación se usará en todo el programa a desarrollar, desde la lectura de datos hasta los reportes que se mandarán al robot.

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad del código. Se trata de un lenguaje de programación multi-paradigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y en menor medida la programación funcional. Es un lenguaje interpretado, dinámica y multiplataforma.

La programación orientada a objetos si puede lograr hacer todas las funcionalidades que corresponden al programa pero para ello si se necesita una buena abstracción porque si no se logra hacer de buena manera puede llevar un mayor tiempo de producción y no obtener los tiempos esperados.

El diagrama de clases a utilizar en el programa es el siguiente:

2. XML.

Las siglas significan lenguaje de marcado extensible, es un conjunto de códigos que se pueden aplicar en el análisis de datos o la lectura de textos creados por computadoras o personas.

La estructura de un fichero XML se asemeja a un árbol de información. Cada rama o parte del fichero se llama elemento y se distingue por la utilización de una etiqueta. Una etiqueta consiste en una marca hecha en el documento, que señala una porción de éste como un elemento.

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Libro">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Titulo" type="xsd:string"/>
        <xsd:element name="Autores" type="xsd:string" maxOccurs="10"/>
        <xsd:element name="Editorial" type="xsd:string"/>
      </xsd:sequence>
      <xsd:attribute name="precio" type="xsd:double"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Figura3: Lenguaje XML.

Este lenguaje es el que se utiliza para enviar los datos al robot r2e2 para luego procesar el camino corto del terreno. Dentro de Python hay una clase que nos permite leer y analizar el archivo, esta clase se llama elementtree y escribir un archivo XML con la clase llamada cElementTree.

Se da un ejemplo de cómo quedaría el terreno utilizando esta herramienta.

3. 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 te ayuda a escribir software que es:

Completo

Django sigue la filosofía "Baterías incluidas" y provee casi todo lo que los desarrolladores quisieran que tenga "de fábrica". Porque todo lo que necesitas es parte de un único "producto", todo funciona a la perfección, sigue principios de diseño consistentes y tiene una amplia y actualizada documentación.

Versátil

Django puede ser (y ha sido) usado para construir casi cualquier tipo de sitio web — desde sistemas manejadores de contenidos y wikis, hasta redes sociales y sitios de noticias. Puede funcionar con cualquier framework en el lado del cliente, y puede devolver contenido en casi cualquier formato (incluyendo HTML, RSS feeds, JSON, XML, etc). ¡El sitio que estás leyendo actualmente está basado en Django!

Internamente, mientras ofrece opciones para casi cualquier funcionalidad que desees (distintos motores de base de datos, motores de plantillas, etc.), también puede ser extendido para usar otros componentes si es necesario.

Seguro

Django ayuda a los desarrolladores evitar varios errores comunes de seguridad al proveer un framework que ha sido diseñado para "hacer lo correcto" para proteger el sitio web automáticamente. Por ejemplo, Django, proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como colocar informaciones de sesión en cookies donde es vulnerable (en lugar de eso las cookies solo contienen una clave y los datos se almacenan en la base de datos) o se almacenan directamente las contraseñas en un hash de contraseñas.

Un hash de contraseña es un valor de longitud fija creado al enviar la contraseña a una *cryptographic hash function*. Django puede validar si la contraseña ingresada es correcta enviándola a través de una función hash y comparando la salida con el valor hash almacenado. Sin embargo debido a la naturaleza "unidireccional" de la función, incluso si un valor hash almacenado se ve comprometido es difícil para un atacante resolver la contraseña original.

Django permite protección contra algunas vulnerabilidades de forma predeterminada, incluida la inyección SQL, scripts entre sitios, falsificación de solicitudes entre sitios y clickjacking (consulte Seguridad de sitios web para obtener más detalles sobre dichos ataques).

Escalable

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). Teniendo en cuenta una clara separación entre las diferentes partes significa que puede escalar para aumentar el tráfico al agregar hardware en cualquier nivel:

servidores de cache, servidores de bases de datos o servidores de aplicación. Algunos de los sitios más concurridos han escalado a Django para satisfacer sus demandas (por ejemplo, Instagram y Disqus, por nombrar solo dos).

Mantenible

El código de Django está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable. En particular, utiliza el principio No te repitas "Don't Repeat Yourself" (DRY) para que no exista una duplicación innecesaria, reduciendo la cantidad de código. Django también promueve la agrupación de la funcionalidad relacionada en "aplicaciones" reutilizables y en un nivel más bajo, agrupa código relacionado en módulos (siguiendo el patrón Model View Controller (MVC)).

Portable

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. Además, Django cuenta con el respaldo de muchos proveedores de alojamiento web, y que a menudo proporcionan una infraestructura específica y documentación para el alojamiento de sitios de Django.

a. Flask.

Actualmente en el desarrollo moderno de aplicaciones web se utilizan distintos Frameworks que son herramientas que nos dan un esquema de trabajo y una serie de utilidades y funciones que nos facilita y nos abstrae de la construcción de páginas web dinámicas. En general los Frameworks están asociado a lenguajes de programación (Ruby on Rails (Ruby), Symphony (PHP)), en el mundo de Python el más conocido es Django pero Flask es una opción que quizás no

tenga una curva de aprendizaje tan elevada pero nos posibilita la creación de aplicaciones web igual de complejas de las que se pueden crear en Django.
Ventajas de utilizar Flask:

- Proporciona una estructura del proyecto, es decir, todas las apps que estén construidas con Flask van a tener los mismos elementos y ficheros.
- Facilita la colaboración.
- Es fácil encontrar bibliotecas adaptadas al Framework.

Conclusiones

El algoritmo para optimizar el proceso de analizar los archivos xml es fácil de manejar ya que Python trae librerías capaz de resolver dicho problema con la lectura, con lo cual se facilita el proceso con la lectura de los DTE.

Django y Flask realmente son compatible ya que se puede transferir datos por medio de los protocolos HTTPS, al mismo tiempo puede procesar o almacenar los datos, de acuerdo a lo que se requiera utilizar.

Referencias bibliográficas

1. Green, Bert F. Jr. (1961). Computer Languages for Symbol Manipulation. *IRE Transactions on Human Factors in Electronics*.
2. <https://flask.palletsprojects.com/en/2.0.x/>.
3. Espacios de nombres en XML 1.1. Traducción de la recomendación del W3C sobre los espacios de nombres en XML 1.1.
4. Conjunto de información XML. Traducción de la recomendación del W3C sobre conjunto de información XML (segunda edición)

Anexos.

