
PROYECTO VALIDACIONES SAT

202010223 – Luis Angel Barrera Velásquez

Resumen

Se realizó este proyecto con el objetivo de brindar una solución a la problemática de validar los datos tributarios de la SAT (llamado así por fines de ejemplo) donde se tiene un archivo en formato XML con varios datos tributarios el cual se ingresan por medio de la página principal web de HTML y luego se mandan al Backend para procesar todo el documento.

La página principal del Frontend se desarrolló a través de una plantilla de html la cual fue adaptada para brindar una interfaz gráfica bastante agradable para el usuario y fácil de entender para esto se debe tener en cuenta que se requiere un servidor en este caso fue utilizado Django para hacer posible una fácil interacción con el backend desarrollado en flask.

Básicamente el funcionamiento de la app web es validar los datos tributarios ingresados y devolver resúmenes que sean fáciles de interpretar.

Palabras clave

Formato
Web
Validar
Interfaz
Datos

Abstract

This project was carried out with the objective of providing a solution to the problem of validating the tax data of the SAT (so called for example purposes) where there is a file in XML format with various tax data which are entered through the HTML main web page and then sent to the Backend to process the entire document.

The main page of the Frontend was developed through an html template which was adapted to provide a very pleasant graphical interface for the user and easy to understand. For this, it must be taken into account that a server is required in this case, Django was used. to make possible an easy interaction with the backend developed in flask.

Basically the operation of the web app is to validate the tax data entered and return summaries that are easy to interpret.

Keywords

*Format
Web
Validate
Interface
Data*

Introducción

La realización de este proyecto se tomó en cuenta diferentes conceptos de programación para poder dar una solución acertada y efectiva del problema inicial para esto fue muy importante la correcta interpretación de los datos de archivo de entrada y la correcta estructuración de los datos obtenidos.

Este ensayo tiene el propósito de proporcionar al lector un panorama claro de los conceptos utilizados para la elaboración de la solución esto para entender el funcionamiento del programa y si se necesita realizar alguna actualización saber que conceptos son necesarios emplear para modificar la lógica de programación.

Desarrollo del tema

a. Archivo XML

XML (Extensible Mark up Language) es un lenguaje de marcado que codifica los datos en texto plano. Permitiendo así que estos puedan ser legibles tanto por parte de máquinas como por personas, de manera análoga a los JSON. Actualmente es un lenguaje que es usado por múltiples programas para almacenar y transmitir datos estructurados. A diferencia de los archivos JSON o Excel no existe una forma fácil de importar los XML en Python, por lo que requiere algo más de trabajo.

Los archivos XML se componen de etiquetas que nos aportan datos e información que queremos procesar. Estas etiquetas pueden estar de forma individual o anidadas. Habitualmente un fichero XML incluye mucha información y debe de ser procesada correctamente por el usuario, en este caso el desarrollador. Cuanto más grande sea un fichero XML nos estará indicando que más información trae. Es importante mencionar que en este proyecto fue implementado un case insensitive para la lectura del archivo XML esto quiere decir que no importa si las

etiquetas están escritas con mayúsculas, minúsculas o mixtas estas igualmente van a ser reconocidas por el programa y guardado sus datos para un futuro procesamiento de estos según los gustos del usuario

b. Estructura dinámica de datos

son estructuras que en este caso para Python pueden ser de distinto, organizadas de alguna manera que se necesite y estas a su vez a medida que se van introduciendo más datos se van agregando de manera automática y sin ocupar mucha memoria ya que a medida de que vayamos utilizando más elementos en la estructura así aumentara la memoria utilizada sin reservar espacios en memoria que no se estén utilizados, esto ayuda a que el programa funcione de una manera optimizada y fluida ya que no consumirá recursos en exceso del equipo donde se esté ejecutando el software.

c. Liberia Minidom

xml.dom.minidom es una implementación mínima de la interfaz Document Object Model (Modelo de objetos del documento), con una API similar a la de otros lenguajes. Está destinada a ser más simple que una implementación completa del DOM y también significativamente más pequeña. Aquellos usuarios que aún no dominen el DOM deberían considerar usar el módulo xml.etree.ElementTree en su lugar para su procesamiento XML.

d. HTML

HTML es un lenguaje de marcación que sirve para definir el contenido de las páginas web. Se compone en base a etiquetas, también llamadas marcas o tags, con las cuales conseguimos expresar las partes de un documento, cabecera, cuerpo, encabezados, párrafos, etc. En definitiva, el contenido de una página web.

HTML es el primer lenguaje que debe aprender cualquier persona interesada en construir un sitio web. A partir del HTML podemos pasar a muchos otros lenguajes interesantes que sirven para hacer cosas diversas y más avanzadas. Es decir, sea cual sea la tecnología, herramienta o gestor de contenido que nos hayamos propuesto aprender, o que tengamos que usar en nuestro día a día, HTML siempre será el lenguaje en el que toda web se construye y, por tanto, es de obligado conocimiento para todos.

Aprender HTML es sencillo. En pocos días o semanas serás capaz de entender y usar las etiquetas más comunes y componer documentos HTML (páginas web) correctas.

e. Frontend

Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

Para convertirte en Frontend Developer debes saber HTML y CSS, los lenguajes de maquetación que nos permiten definir la estructura y estilos de una página web. Y también JavaScript, un lenguaje de programación para definir la lógica de nuestra aplicación, recibir las solicitudes de los usuarios y enviárselos al backend.

Dominando estas tecnologías puedes usar algunos frameworks, librerías o preeprocesadores que expanden tus capacidades para crear todo tipo de interfaces de usuario. Algunos de ellos son React, Vue, Angular, Svelte, Bootstrap, Foundation, Sass, Less, Stylus y PostCSS.

f. 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 fue desarrollado inicialmente entre 2003 y 2005 por un equipo que era responsable de crear y mantener sitios web de periódicos. Después de crear varios sitios, el equipo empezó a tener en cuenta y reutilizar muchos códigos y patrones de diseño comunes. Este código común se convirtió en un framework web genérico, que fue de código abierto, conocido como proyecto "Django" en julio de 2005. Django ha continuado creciendo y mejorando desde su primer hito, el lanzamiento de la versión (1.0) en septiembre de 2008, hasta el reciente lanzamiento de la versión 1.11 (2017). Cada lanzamiento ha añadido nuevas funcionalidades y solucionado errores, que van desde soporte de nuevos tipos de bases de datos, motores de plantillas, caching, hasta la adición de funciones genéricas y clases de visualización (que reducen la cantidad de código que los desarrolladores tienen que escribir para numerosas tareas de programación).

g. Bootstrap

Bootstrap es un kit de herramientas de código abierto para desarrollos web responsive con HTML, CSS y JavaScript. Con él puedes darle forma a tu sitio web a través del uso de sus librerías CSS y JavaScript. Incluye diferentes componentes: ventanas modales, menús, cuadros, botones, formularios... Es decir, los elementos que necesitas para maquetar tu página.

Bootstrap es una excelente herramienta que te permite crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y

pantallas, sea cual sea su tamaño. Desde Bootstrap 3, el framework se ha vuelto más compatible con el desarrollo web responsive. De esto ya te habló Álvaro en 2015, pero vamos a recordarlo.

h. Flask

Flask es un “micro” Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC.

La palabra “micro” no designa a que sea un proyecto pequeño o que nos permita hacer páginas web pequeñas sino que al instalar Flask tenemos las herramientas necesarias para crear una aplicación web funcional pero si se necesita en algún momento una nueva funcionalidad hay un conjunto muy grande extensiones (plugins) que se pueden instalar con Flask que le van dotando de funcionalidad.

De principio en la instalación no se tienen todas las funcionalidades que se pueden necesitar pero de una manera muy sencilla se pueden extender el proyecto con nuevas funcionalidades por medio de plugins.

El patrón MVC es una manera o una forma de trabajar que permite diferenciar y separar lo que es el modelo de datos (los datos que van a tener la App que normalmente están guardados en BD), la vista (página HTML) y el controlador (donde se gestiona las peticiones de la app web).

i. Framework

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 asociados 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.

En la Wiki de Python se muestra un listado con los distintos Frameworks compatibles, se podrá ver que algunos de los de la lista dicen “full stack” que quiere decir que el Framework cuando se instala por primera vez viene con numerosas funcionalidades o las tiene casi todas para hacer una app web completa, ejemplo: Django. Hay otros que no son “full stack” como por ejemplo Flask.

j. Backend

Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas.

Algunos de los lenguajes de programación para Backend son Python, Node.js, PHP, Go, Ruby y C#. Y así como en el frontend, todos estos lenguajes tienen diferentes frameworks que te permiten trabajar mejor según el proyecto que estás desarrollando, como Django, Flask, Express.js, Laravel, Symphony Framework, Ruby on Rails y ASP.Net. Cada uno lo hemos elegido sobre todo porque tienen una gran comunidad que los respalda.

j. Funcionamiento Backend y Frontend en conjunto

Considera que una web se conforma por una gran variedad de documentos que se relacionan entre ellos por medio de enlaces, lo que significa que si quieres entrar a una web y escribes la dirección URL en el navegador, se traduce como que estás solicitando que se muestre dicha página web.

Lo siguiente que hará el dispositivo es verificar qué servidor de software tiene el sitio. El servidor recibe esta información, verifica la petición que hizo y te permite ejecutar la acción.

En ocasiones, se puede presentar el caso de que no se requiere una conexión a la base de datos, por ejemplo, accedes a una página y cuando inicias sesión, se hace de manera automática una petición que conecta a la base de datos para verificar los accesos y la suscripción que se tiene, es allí donde el backend devuelve la respuesta al servidor.

Después aparece el frontend, que es quien va a recibir la información que transmitió el backend y la va a acomodar en la interfaz del sitio web o perfil del usuario.

k. Automata

Este tipo de autómatas admite su definición de dos maneras bien diferentes: Como autómatas traductores o reconocedores. La definición como autómatas traductores continua a la definición de las máquinas secuenciales, y se los podría definir como una subclase de estas, ya que los autómatas finitos tendrían como limitante no poder iniciar desde cualquier estado como lo hacen en las máquinas secuenciales.

La forma que adoptaremos para la definición de los autómatas finitos deterministas es como autómatas reconocedores, ya que se ajusta con los contenidos de la informática teórica y utilización que se les da dentro del diseño de los analizadores léxicos.

Estos autómatas solo se limitarán a aceptar o no una determinada cadena recibida en la entrada, por lo tanto podemos decir que la salida de los mismos solo tendrá dos valores posibles aceptar o no aceptar a la palabra de entrada.

Conclusiones

- La implementación Frontend-Backend es una forma perfecta de poder realizar diferentes aplicaciones web que dan resolución a problemáticas de la actualidad y necesidades de todo tipo.
- El manejo de archivos XML es una forma muy eficiente de estructurar datos de una manera clara y precisa sin necesidad de pasar horas en estructuras complejas y la lectura a través de Python es bastante sencilla.
- Django es una forma muy sencilla de poder desarrollar una aplicación web de manera rápida, efectiva y segura.

Referencias bibliográficas

- CALDERA VERGARA, Roberto, et al. Estudio del framework de desarrollo web Django. 2017.
- Jones, C. A., & Drake Jr, F. L. (2001). Python & XML: XML Processing with Python. " O'Reilly Media, Inc.".
- Joyanes Aguilar, L. (2003). Fundamentos de programación: algoritmos y estructura de datos y objetos.

- Ojeda, L. R. Tda Programacion Orientado a Objetos en Turbo. Univ. Nacional de Colombia.
- Roldán Blay, C. (2021). Crear e interrumpir bucles en Python.

Anexos

Durante el desarrollo del proyecto fue necesario realizar un automata para el reconocimiento de fechas, para el cual se utilizo un sencillo autómata que se muestra a continuación:

