
PROYECTO 3

201900952 – Gerardo Jose Cifuentes Luna

Resumen

Se nos presenta la oportunidad de hacer un programa que pueda ser consumido por el internet para poder lograr facilitar las consultas que los usuarios hacen con la SAT (Superintendencia de Administración Tributaria).

Este software tiene el fin de poder permitir a la SAT hacer una sistematización de las peticiones de los consumidores sobre la Autorización de un Documento Tributario Electrónico y el software emitirá como respuesta un numero único de autorización si es aceptada la petición, para cada petición, este tiene que dar un análisis a las peticiones y hacer unas operaciones antes de dar las respuestas.

Palabras clave

- *HTTP*
- *XML*
- *API*

Abstract

We are presented with the opportunity to make a program that can be consumed by the internet in order to facilitate the queries that users make with the SAT (Superintendency of Tax Administration).

This software is intended to allow the SAT to systematize consumer requests for the Authorization of an Electronic Tax Document and the software will issue a unique authorization number in response if the request is accepted, for each request, this You have to give an analysis to the requests and do some operations before giving the answers.

Keywords

- *HTTP*
- *XML*
- *API*

Introducción

Tenemos que recibir una petición para la autorización de un documento tributario electrónico, este será manejado desde el lenguaje XML, donde tendrá un formato que traerá los datos necesarios para poder validar dicha petición y dar la autorización según como el software analice lo ingresado en el XML. El software es solicitado tener su solución como una pagina web, esto requiere un uso de API para que logremos hacer un software desarrollado con Python por lo cual utilizamos Django, dentro de esta pagina web, tendremos varias solicitudes que tendremos que manejar sobre la utilización de todo el software, por medio de manejo de operaciones, TDA y desarrollo web.

.

Desarrollo del tema

Para el desarrollo del software se requiere un manejo de las entradas que vienen en un archivo XML, luego de tomar un manejo y obtención de los datos de de los contribuyentes.

Se necesita el manejo de TDA's por lo cual el uso de Objetos donde almacenaremos y manipularemos los datos obtenidos, la solución también tiene su interfaz en una pagina web y lleva ciertas operaciones, reportes y respuestas durante su uso por lo cual existen ciertos temas importantes de conocimiento para el desarrollo.

a. Paradigma Orientado a Objetos (POO):

es un paradigma de programación, un modelo de programación que nos da guías sobre cómo trabajar con objetos. Se basa en el concepto de clases y objetos.

Existen “Clases” que son la estructura base o

plantilla para un objeto y los “Objetos” son las instancias dentro de una clase que disponen atributos y métodos específicos para cada objeto pero con la plantilla de la clase. Existen 4 pilares de la Programación Orientada a Objetos:

- Encapsulamiento: Privatizar funciones y atributos de un objeto
- Herencias: Heredar métodos y atributos a clases hijas
- Abstracción: Con los atributos y métodos se utilizan con los objetos para manipularlos y cambiarlos
- Polimorfismo: Es Diseñar objetos para compartir comportamientos.

b. Tipos de estructura de Datos Abstractos (TDA):

Son Estructuras creadas para el manejo de Objetos como datos Abstractos por lo que este permite un manejo de información dinámica que tenga varios datos enlazados a un solo objeto o en este caso se les llaman “Nodos”: Un nodo es un espacio de memoria que permite almacenar la información del dato abstracto y para manejo de su memoria en una estructura también tiene los “punteros” que permiten marcar a los datos enlazados y su orientación por medio de esto se crean las siguientes

Implementaciones:

- Conjuntos o Listas.
- Pilas y Colas.
- Grafos.
- Arboles Binarios de Búsqueda.

c. XML: Del ingles Extensible Markup Language, El lenguaje de marcado es un conjunto de códigos

que se pueden aplicar en el análisis de datos o la lectura de textos creados por computadoras o personas. El lenguaje XML proporciona una plataforma para definir elementos para crear un formato y generar un lenguaje personalizado.

Un archivo XML se divide en dos partes: "Prolog" que consiste en metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios. Y el "Body" que se compone de dos partes: estructural y de contenido siendo la información que se comparte y maneja.

- d. Desarrollo Web: Como parte de la importante en el desarrollo del problema se solicita que tenga una interfaz web, "Implementar un API a través de lenguaje Python que pueda ser consumida utilizando el protocolo HTTP." Conocemos el desarrollo web como la creación de sitios web donde además de la apariencias tiene funcionalidad manejada desde el servidor. Para conseguirlo se hace uso de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos de base de datos con el uso de un navegador web a fin de realizar determinadas tareas o mostrar información. Los aspectos como diseño web, publicación web, programación web.

- Diseño web: este requiere del trabajo de la Interfaz que el usuario vera desde la pagina web, aquí se hace manejo del lenguaje HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) para la estructura de la pagina y CSS para el estilo de la pagina

como dicen sus siglas en inglés de cascading style sheets.

- Publicacion web: Aquí refiere a como utilizamos HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML, para poder manejar y poder utilizar la pagina web y manejar los recursos y peticiones para la programación web
- Programacion web: Se necesita poder tener una Web API para el desarrollo web, esto refiere a su programación donde aquí se maneja las API's (Interfaz de Programación de Aplicaciones, en inglés) esta API tendrá tanto como el diseño web, como la funcionalidad de los eventos y respuestas que daremos tanto en Algoritmos que manejen la información y lo que refiere para que pueda ser consumida utilizando protocolos HTTP.
- e. Algoritmo: Es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades. Del cual se usara para manejo de las instrucciones y simulación.
- f. Django: Es un framework para Python el cual utilizamos para desarrollo como Web API, ofrece un conjunto de herramientas y bibliotecas para ayudar a los programadores a lograr un objetivo. Por ejemplo, si el objetivo es desarrollar una aplicación web, entonces el marco proporciona bibliotecas que facilitan el esfuerzo de codificar el frontend web, el backend o ambos. En este caso, el marco abstrae las complejidades de lograr

las operaciones individuales relacionadas con la aplicación web y ofrece un código repetitivo para simplificar la lógica de programación. Además, algunos frameworks avanzados también definen su sintaxis y reglas semánticas para estructurar el programa de una manera específica, para definir un patrón de desarrollo de software. Tenemos lo que sería:

- * el Desarrollo FrontEnd: que dispone del :UI,Control de Acceso “si es necesario”,Aplicaciones web “que son donde el usuario tiene la navegación y control con las funciones de las aplicaciones” y protocolo HTTP “Utilizado como el componente que envía las peticiones web al backend ”. Todo es donde interactúan los usuario.

- * el Desarrollo BackEnd: donde el servidor trabaja y hace respuestas a peticiones y almacena datos , unos componente importantes son: Los datos que recibe y manipula, El control de acceso que devuelve una respuesta al Front End, Las utilidades que se tienen para manejar las localizaciones,archivos y errores dentro los modulos de la aplicación web, Seguridad para tener una pagina responsable siguiendo las directivas para soporte HTTPS, y claro el componente HTTP que sirve como la conexión entre el Front End y el BackEnd.

Para Django es importante saber como se maneja su desarrollo por lo cual se presenta la estructura que presenta al momento de trabajar con el:

URLs: Aunque es posible procesar peticiones de cada URL individual vía una función individual, es mucho más sostenible escribir una función de

visualización separada para cada recurso. Se usa un mapeador URL para redirigir las peticiones HTTP a la vista apropiada basándose en la URL de la petición. El mapeador URL se usa para redirigir las peticiones HTTP a la vista apropiada basándose en la URL de la petición. El mapeador URL puede también emparejar patrones de cadenas o dígitos específicos que aparecen en una URL y los pasan a la función de visualización como datos.

Vista (View): Una vista es una función de gestión de peticiones que recibe peticiones HTTP y devuelve respuestas HTTP. Las vistas acceden a los datos que necesitan para satisfacer las peticiones por medio de modelos, y delegan el formateo de la respuesta a las plantillas ("templates").

Modelos (Models): Los Modelos son objetos de Python que definen la estructura de los datos de una aplicación y proporcionan mecanismos para gestionar (añadir, modificar y borrar) y consultar registros en la base de datos.

Plantillas (Templates): una plantilla (template) es un fichero de texto que define la estructura o diagrama de otro fichero (tal como una página HTML), con marcadores de posición que se utilizan para representar el contenido real. Una vista puede crear dinámicamente una página usando una plantilla, rellenándola con datos de un modelo. Una plantilla se puede usar para definir la estructura de cualquier tipo de fichero; ¡no tiene por qué ser HTML!

Conclusiones

Para el desarrollo web API tomar en cuenta cada uno de los factores a demás de la practica sobre como funciona una API y los frameworks que ayudan con este, poder manejar todo esto conviene al momento de pasar una solución de un software a manera remota que solamente local, un desarrollo muy utilizado últimamente y por lo tanto muy importante de aprender.

El problema requiere una soluciones que pasan por ciertos procesos pero estos solamente requieren del poder tener un sistema apropiado para el reconocimiento de toda la informacion, luego manejar las peticiones de manera correcta y luego por medio de las opciones que se le dan a los usuarios dar las respuestas necesarias del BackEnd al FrontEnd, todo este proceso toma un tiempo de desarrollo que bien estructurado es posible.

Referencias bibliográficas

Anonimo,(2021).Algoritmo de

Anonimo, (2020).Que es la programacion Orientada a Objetos.<https://profile.es/blog/que-es-la-programacion-orientada-a-objetos>.

Ivan de Souza, (2019).Que es XML.
<https://rockcontent.com/es/blog/que-es-xml/>

Anonimo,(2021).Introduccion de Django,
<https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>

Apéndices

