
PROYECTO 3 - "Tecnologías Chapinas, S.A."

202004745 – Allen Giankarlo Román Vásquez

Resumen

El proyecto 3 "Tecnologías Chapinas, S.A." se realiza para poner en práctica los conocimientos adquiridos en el laboratorio y clase magistral del curso "Introducción a la Programación y Computación" de la carrera de Ingeniería en Ciencias y Sistemas de la Universidad San Carlos de Guatemala creando un proyecto en el lenguaje de programación "Python" implementando las funciones básicas del lenguaje de programación, también aplicando los temas y conceptos teóricos aprendidos en el laboratorio, por ultimo creando otro tipo de conocimiento al usar frameworks y consumiendo APIs por medio del protocolo HTTP, todo a través de páginas creadas con HTML y CSS y distintas librerías que nos permiten la conexión y envío de datos usando archivos de extensión XML y el modelo Frontend-API-Backend con Python, almacenando esta información usando el paradigma de programación Orientada a Objetos, por último el uso de expresiones regulares para una extracción de datos precisa de los archivos de entrada.

Palabras clave

Frameworks, API, Frontend, Backend, HTTP

Abstract

Project "Tecnologías Chapinas, S.A." was performed for performance the knowledge acquired in the laboratory class and the masterly class "Introduction to programming and computation" of the Science and systems Engineering career from the University "San Carlos de Guatemala" creating a project with programming language "Python" implementing the basic functions of the programming language, also applying the theoretical topics and concepts learned in the laboratory class, and lastly, creating another type of knowledge by using frameworks and consuming APIs through the HTTP protocol, all through pages created with HTML and CSS and different libraries that allow us to connect and send data using XML extension files with the Frontend-API-Backend model with Python, storing this information using the programming paradigm "Oriented to Objects", finally the use of regular expressions for a precise data extraction from the entry files.

Keywords

Frameworks, API, Frontend, Backend, HTTP

Introducción

La superintendencia de Administración Tributaria ha solicitado crear un software que pueda ser consumido desde internet como un servicio en la que se solicita al usuario ingresar un documento de extensión XML en el que solicita la autorización de un Documento Tributario Electrónico (DTE) emitido por un contribuyente y como respuesta llega un documento de extensión XML donde incluye un número único de autorización que se usará posteriormente para realizar diferentes tipos de reportes.

Desarrollo del tema Python

Python es un lenguaje de programación interpretado cuya principal filosofía es que sea legible por cualquier persona con conocimientos básicos de programación. Entre las principales características que cuenta este lenguaje de programación están:

- Es totalmente gratuito.
- Está respaldado por una enorme comunidad.
- Es un lenguaje multiparadigma (combina muchos paradigmas de programación).
- Sus aplicaciones no se limitan a solo un área en concreto.
- Es apto para todas las plataformas.

Fue desarrollado por el investigador holandés Guido van Rossum, quien en aquellas épocas trabajaba en el centro de investigación CWI (Centrum Wiskunde & Informatica) de Ámsterdam.

A van Rossum le fue asignado un proyecto en el que tenía que desarrollar un sistema operativo distribuido llamado Amoeba, las condiciones en que se

encontraba el CWI, es que tenía implementado un lenguaje de programación llamado ABC.

Guido decidió crear un nuevo lenguaje para el proyecto Amoeba, para superar las limitaciones y problemas con los que se encontró. Una vez presentado Python, CWI decidió liberar el intérprete del lenguaje bajo una licencia open source propia.

En tanto, para el inicio del siglo XXI y coincidiendo con la publicación de la versión 1.6, se tomó la decisión de cambiar la licencia por una que sea compatible con GPL (General Public Licence). Esta nueva licencia se denominará Python Software Foundation Licence.



Figura 1. Logo de Python

Fuente: Python Software Foundation. (2020)

Flask

Flask es un framework web WSGI ligero. Está diseñado para que la puesta en marcha sea rápida y sencilla, con la capacidad de escalar a aplicaciones complejas. Comenzó como una simple envoltura alrededor de Werkzeug y Jinja y se ha convertido en uno de los marcos de aplicaciones web de Python más populares.

Flask ofrece sugerencias, pero no impone ninguna dependencia o diseño del proyecto. Depende del desarrollador elegir las herramientas y bibliotecas que desea utilizar. Hay muchas extensiones proporcionadas por la comunidad que facilitan la adición de nuevas funciones.



Figura II. Logo de Flask

Fuente: The Pallets Projects. (2020)

Django

Django es un framework de Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de gran parte de la molestia del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto.

Django se basa en el patrón de diseño conocido como modelo–vista–controlador (MVC). Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt.



Figura III. Logo de django

Fuente: The Pallets Projects. (2020)

Modelo Vista Template

Es el modelo que usa Django para el uso de su framework y se compone de la siguiente estructura:

- **Modelo:** Maneja todo lo relacionado con la información, esto incluye como acceder a esta, la validación, relación entre los datos y su comportamiento.

- **Vista:** Es un enlace entre el modelo y el template. Decide que información sera mostrada y por cual template.
- **Template:** Decide como sera mostrada la información.

HTTP

Hypertext Transfer Protocol (HTTP) (o Protocolo de Transferencia de Hipertexto en español) es un protocolo de la capa de aplicación para la transmisión de documentos hipermedia, como HTML. Fue diseñado para la comunicación entre los navegadores y servidores web, aunque puede ser utilizado para otros propósitos también. Sigue el clásico modelo cliente-servidor, en el que un cliente establece una conexión, realizando una petición a un servidor y espera una respuesta del mismo. Se trata de un protocolo sin estado, lo que significa que el servidor no guarda ningún dato (estado) entre dos peticiones.

Control de acceso HTTP (CORS)

Las Solicitudes Inter-Sitio HTTP (Cross-site HTTP requests en inglés), son peticiones HTTP por recursos pertenecientes a un dominio distinto al dominio del recurso que está haciendo la petición. Por ejemplo, una página HTML de un dominio A (<http://dominioa.ejemplo/>) hace una solicitud por una imagen en un dominio B (<http://dominiob.foo/imagen.jpg>) a través del elemento img. Hoy en día, las webs utilizan recursos de otros orígenes muy a menudo, incluyendo hojas de estilo CSS, imágenes, scripts y otros recursos. El Control de Acceso HTTP posibilita a los desarrolladores web a controlar cómo su sitio web responde a solicitudes de otros orígenes.

Métodos HTTP

GET: El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

POST: El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

PUT: El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

DELETE: El método DELETE borra un recurso en específico.

otros. Estas características hacen de JSON un lenguaje de intercambio de datos ideal.

```
{
  "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  },
  "phoneNumbers": [
    "212-732-1234",
    "646-123-4567"
  ]
}
```

Figura I V. Estructura de un archivo JSON

Fuente: IBM Business Automation Workflow. (2018)

App.py

Clase donde se generan las diferentes rutas de la API con Flask.

Objetos.py

Archivo donde están las diferentes clases de los objetos, con su método constructor.

Frontend

Carpeta donde se encuentra todos los archivos que utiliza Django para gestionar la app web, con sus htmls, views y demás métodos que usa Django.

JSON

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. JSON es de fácil lectura y escritura para los usuarios. JSON es fácil de analizar y generar por parte de las máquinas. JSON se basa en un subconjunto del lenguaje de programación JavaScript, Estándar ECMA-262 3a Edición - diciembre de 1999. JSON es un formato de texto completamente independiente del lenguaje, pero que utiliza convenios que resultan familiares a los programadores de lenguajes de la familia C, incluidos C, C++, C#, Java™, JavaScript, Perl, Python y mucho

Javascript

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado. Es la tercera capa del pastel de las tecnologías web estándar, dos de las cuales (HTML y CSS) hemos cubierto con mucho más detalle en otras partes del Área de aprendizaje.



Figura V. Logo de Javascript

Fuente: MDN Web Docs. (2015)

HTML

HTML, siglas en inglés de HyperText Markup Language ('lenguaje de marcado de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. HTML se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW).

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
  <footer>
    <p>CC</p>
  </footer>
</html>
```

Figura VI. Ejemplo de HTML

Fuente: ResearchGate. (2020)

CSS

CSS (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.² Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas.

```
div{
  width: 350px;
  margin: 1em 0 1em 0;
}
#tag p{
  text-align: justify;
  font-size:10px;
  font-weight: bold;
  background-color: #FFFFFF;
}
```

Figura VII. Ejemplo de CSS

Fuente: ResearchGate. (2009)

Rutas Web

Las Rutas de navegación son algo que forma parte del estudio del tráfico de una página web y de todas sus partes. Es la trayectoria que sigue cada usuario desde que entra hasta que abandona una web, registrando la landing page, los clicks, las secciones visitadas, los contenidos vistos y la última página visitada antes de abandonar la web.

Conclusiones

Las paginas Web en su mayoría se construyen con el modelo Frontend-API-Backend, donde en el frontend se usa HTML, CSS y Javascript mientras que la API debe ser una específica para la conexión que se haga con el Backend dependiendo el lenguaje de programación.

La programación web hoy en día es una de las tecnologías mas solicitadas por lo que aprender a manejarla es realmente importante para un ingeniero en sistemas

Referencias bibliográficas

- A. Donwney, (2002). *Aprenda a pensar como un Programador con Python*. Green Tea Press.
- C. Severance, (2009). *Python para todos. Explorando la información con Python 3*. Green Tea Press
- J. Bullinaria, (2019). *Data Structures and Algorithms*. Green T. School of Computer Science, University of Birmingham, UK.
- M. Acera, (2011). *XML Edición 2012*. Anaya Multimedia
- V. Das, (2006) *Principles of Data Structures using C and C++*. New Age International (P) Ltd., Publishers

APÉNDICE A

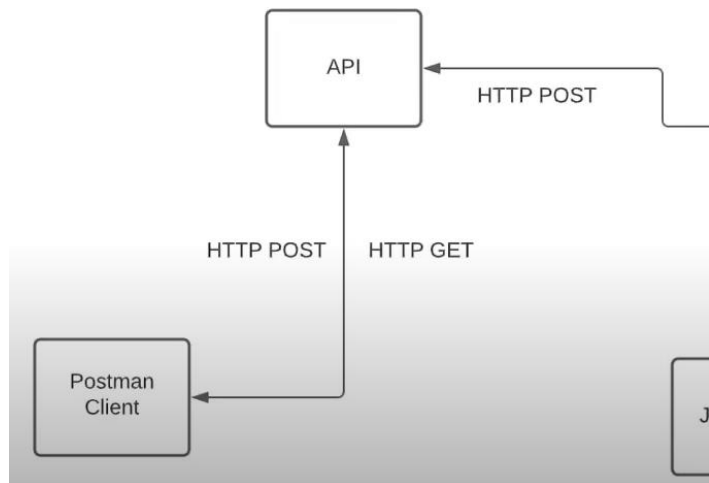


Figura VIII. Esquema de conexión API-Django

Fuente: elaboración propia

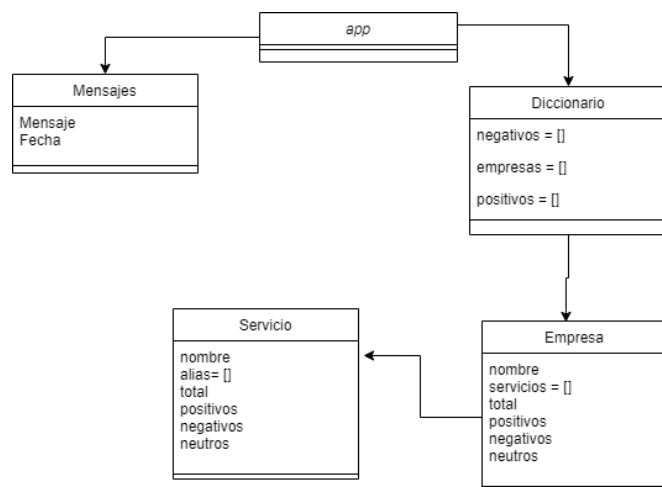


Figura IX. Diagrama de clases de la API

Fuente: elaboración propia