Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2021.

PROYECTO 3

201344708 – Jose Manuel Lara Elias

Resumen

Sin duda una de las mejores opciones a tener en cuenta es usar un lenguaje de alto nivel como Python, del que podemos aprovechar muchas de sus ventajas ya que es un lenguaje interpretado y de tipado dinámico.

Python es ampliamente utilizado, por lo que tiene una gran base de programadores. Combina múltiples paradigmas de programación como el orientado a objetos y el funcional.
Además, su sintaxis está orientada escribir menos líneas de código pero más.

claras que otros lenguajes, lo que facilita enormemente su mantenimiento.

Por otro lado, existen excelentes frameworks y bibliotecas muy asentados y orientados al desarrollo de APIs REST y sistemas de comunicación basados en HTTP, lo que es perfecto para la construcción de microservicios

Palabras Claves:
Modelo, Vista, Router, Serializador,test.

Abstrac

Undoubtedly one of the best options to consider is to use a high-level language such as Python, from which we can take advantage of many of its advantages since it is an interpreted and dynamically typed language.

Python is widely used, so it has a large programmer base. It combines multiple programming paradigms such as object-oriented and functional.
Also, its syntax is oriented to write fewer lines of code but more
clearer than other languages, which greatly facilitates its maintenance.
On the other hand, there are excellent frameworks and libraries that are well established and oriented to the development of REST APIs and communication systems based on HTTP, which is perfect for building microservices.

Keywords:
Model, View, Router, Serializer, test.

Introducción

Los lenguajes de programación son la herramienta básica de construcción de programas, como lo son el machete y el azadón para un campesino, el pico y la pala para un constructor. Python ha ido ganando en adeptos en comunidades como la de software libre, científica y educacional, por su sencillez y posibilidad de concentrarse en los problemas actuales. Este artículo hace referencia a las principales características y los diferentes usos de este lenguaje de programación, por lo que se ha tratado de simplificar la parte técnica. No obstante, para su completa comprensión se necesita un nivel básico de conocimientos acerca de programación.

Python cuenta con facilidades para la programación orientada a objetos, imperativa y funcional, por lo que se considera un lenguaje multi-paradigmas.

Desarrollo del tema

El término API es una abreviatura de Application Programming Interfaces, que en español significa interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

Así pues, podemos hablar de una API como una especificación formal que establece cómo un módulo de un software se comunica o interactúa con otro para cumplir una o muchas funciones. Todo dependiendo de las aplicaciones que las vayan a utilizar, y de los permisos que les dé el propietario de la API a los desarrolladores de terceros.

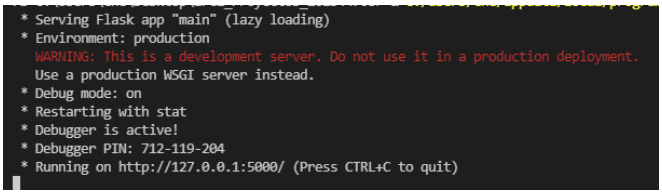
Para qué sirve una API

Una de las principales funciones de las API es poder facilitarles el trabajo a los desarrolladores y ahorrarles tiempo y dinero. Por ejemplo, si estás creando una aplicación que es una tienda online, no necesitarás crear desde cero un sistema de pagos u otro para verificar si hay stock disponible de un producto. Podrás utilizar la API de un servicio de pago ya existente, por ejemplo, PayPal, y pedirle a tu distribuidor una API que te permita saber el stock que ellos tienen.
Con ello, no será necesario tener que reinventar la rueda con cada servicio que se crea, ya que podrás utilizar piezas o funciones que otros ya han creado. Imagínate que cada tienda online debiese tener su propio sistema de pago, para los usuarios normales es mucho más cómodo poder hacerlo con los principales servicios que casi todos utilizan.

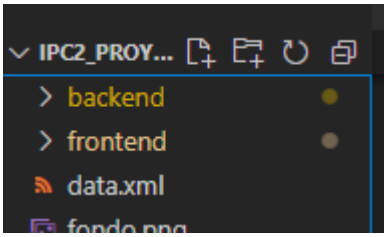
Como programe mis API;

```
14  #Generacion de los endpoints
15  @app.route('/')
16  def home():
17      return "SERVER funciona correctamente."
18
```

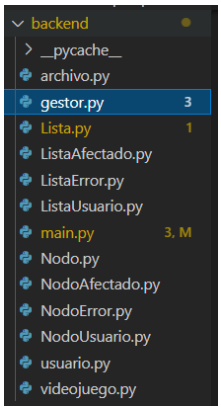
Este es un ejemplo de una API cencía que tiene como nombre route luego definimos home, y esta la consumimos a través del puerto 5000 dónde está flask.



Así se miraría el servidor ejecutándose exitosamente.



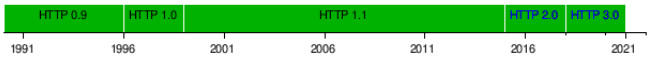
Hablemos un poco de lo que es el proyecto en general, aquí podemos apreciar en esta fotografía cómo está guardado el proyecto. Este contiene dos carpetas principales la primera es back end en este contenemos todo lo que es del servidor de flask que como lo había mencionado éste se encarga de todo lo que es el funcionamiento del programa qué se le llama en programación back end. La otra carpeta principal es la de front end eso es lo que contiene es toda la parte que el cliente mira o visualiza también llamada como habíamos indicado en programación frontend.



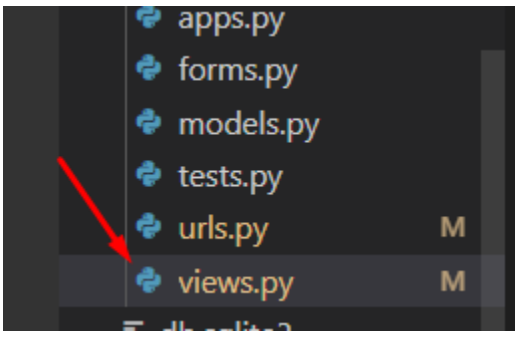
en la carpeta de backend podemos observar varias clases punto.py en las cuales incluye nuestra memoria que utilizamos para esto use TDA. también contamos con la clase gestor.py que la podría considerar como la clase principal.

Hablaremos un poco del protocolo de transferencia de hipertexto que utilizamos en nuestro proyecto.

El Protocolo de transferencia de hipertexto es el protocolo de comunicación que permite las transferencias de información a través de archivos (XHML, HTML . . .) en la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, siendo el más importante de ellos el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de sesión, y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.



Como lo utilizo en mi proyecto;
Con la clase views



```
def RecibirTexto(request):
    contexto ={}
    if request.method == 'GET':
        form = RetornoForm(request.GET)
        if form.is_valid():
            response = requests.get(endpoint+'/stats');

            contexto= {
                'texto': response.text,
            }
        else:
            print('F')
    return render(request, 'signup.html', contexto)
```

Esta se podría decir que es la clase principal en nuestro servidor de django. Esta clase prácticamente es la que nos conecta con nuestro servidor de flask, en esta clase utilizamos el puerto 5000 en la variable en points que se puede apreciar en la imagen, el servidor de Django en este proyecto está corriendo en el puerto 8000.

en este método lo que realizamos es creamos una variable llamada contexto luego de eso entra con if diversificamos si es valida si es así creamos una variable llamada form de tipo retorno form que eso utilizamos para poder acceder a la información de nuestro HTML luego de eso con una variable llamada response con él el método get mandamos a traer la información que deseamos y la guardamos en la variable contexto con él con el nombre de texto después de eso la página HTML con la información que deseamos.

Conclusiones

La programación por objetos es la agregación de varias técnicas de programación que se han desarrollado con el correr del tiempo. Su uso se está difundiendo mucho por la importancia que tiene el desarrollo de interfaces para programas, las que de una manera muy natural pueden implementarse usando técnicas de abstracción y programación por objetos.

Sin embargo, OOP no es una tecnología que vaya a resolver todos nuestros problemas de computación. Las panaceas no existen, y los computólogos nos vemos obligados a conocer cada vez más técnicas para resolver los desafiantes problemas del mundo actual.

Pero tal vez el efecto colateral más positivo que se obtenga de OOP sea el entrenamiento a que fuerza a los programadores, quienes poco a poco comprenden mejor como hacer programas muy bien modularizados. En estos nuevos programas cada módulo se acerca cada vez más a ser realmente un "componente de software". Esta nueva disciplina de programación dará muy buenos réditos, pues de seguro mejorará mucho la calidad de los sistemas disponibles para los usuarios finales.

Referencias bibliográficas

Di Mare, Adolfo: *Abstracción de Datos en Pascal*, Reporte técnico PIBDC-02-90, ECCI-UCR, 1991.

Goldberg, Adele & Robson, David: *Smalltalk-80, the Language and its Implementation*, Addison-Wesley, 1983.

What is Object-Oriented Programming, IEEE Software, pp [10-20], Mayo 1988, http:// www.research.att.com /~bs/ papers.html.

Wirth, Niklaus: *Programming in Modula-2, Second Edition***, R.R. Donnelley & Sons, Harrisonburg, Virginia, 1982.**