Microservicios y contenedores.

He optado por usar el framwork Hug de Python para la creación de la API Rest, primero instalé Hug

```
Administrador: Símbolo del sistema
                                                                         Microsoft Windows [Versión 10.0.19041.450]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.
C:\Windows\system32>pip install hug -U
Collecting hug
 Downloading hug-2.6.1-py2.py3-none-any.whl (75 kB)
                                      75 kB 225 kB/s
Collecting falcon==2.0.0
 Downloading falcon-2.0.0-py2.py3-none-any.whl (163 kB)
                                      163 kB 1.3 MB/s
Collecting requests
 Downloading requests-2.24.0-py2.py3-none-any.whl (61 kB)
                                      61 kB 150 kB/s
Collecting certifi>=2017.4.17
 Downloading certifi-2020.6.20-py2.py3-none-any.whl (156 kB)
                                      | 156 kB 1.3 MB/s
Collecting idna<3,>=2.5
 Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
                                      | 58 kB 975 kB/s
Collecting chardet<4,>=3.0.2
 Downloading chardet-3.0.4-py2.py3-none-any.whl (133 kB)
                                      133 kB 2.2 MB/s
Collecting urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
 Downloading urllib3-1.25.10-py2.py3-none-any.whl (127 kB)
```

Después cree el método local.

```
#Importamos lo modulos necesarios
import datetime
import hug

#Indica que será de solo acceso local
@hug.local()

#Definimos la función say_hello
def say_hello(name: hug.types.text, age: hug.types.number, hug_timer=3):#hug.typ
    year_of_birth = datetime.datetime.now().year - age
    return {#Decimos hola al usuario y calculamos su año de nacimiento
        'message': "Hola {0}, naciste el año {1}".format(name, year_of_birth),
         'took': float(hug_timer)
    }

if __name__ == '__main__':
    print(say_hello("panchito", 50))
```

Después cree el método HTTP

```
import datetime
import hug

@hug.get(examples="name=Jhon Doe&age=30")
@hug.local()
def say_hello(name: hug.types.text, age: hug.types.number, hug_timer=3):
    """Decimos hola al usuario y calculamos su año de nacimiento"""
    year_of_birth = datetime.datetime.now().year - age
    return {
        'message': "Hola {0}, naciste el año {1}".format(name, year_of_birth),
        'took': float(hug_timer)
    }

if __name__ == '__main__':
    print(say_hello("Juanito", 23))
```

El método HTTP, lo guarde en la dirección api\sps\helloworld\v1 y ejecute el método.

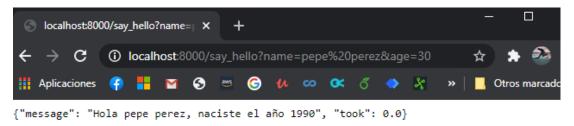
```
C:\Users\KzyCk\Documents\Python\api\sps\helloworld\v1>hug -f AccesoHTTP.py
```

En este momento la API ya esta expuesta, por medio del puerto 8000.

Visualizando desde el navegador se ve lo siguiente.

```
→ C
                 (i) localhost:8000/hola
Aplicaciones
                      H 🗹 🚱 💩
                                                                              » Otros marcadores
    "404": "The API call you tried to make was not defined. Here's a definition of the API
to help you get going :)",
    "documentation": {
        "handlers": {
             "/say_hello": {
                  "GET": {
                      "usage": "Decimos hola al usuario y calculamos su año de nacimiento",
                      "examples": [
                           "http://localhost/say_hello?name=Jhon Doe&age=30"
                      "outputs": {
    "format": "JSON (Javascript Serialized Object Notation)",
    "content_type": "application/json; charset=utf-8"
                     },
"inputs": {
                          "name": {
                               "type": "Basic text / string value"
                           'age": {
                               "type": "A whole number"
                     }
                }
           }
       }
    }
}
```

Y marcando los valores de nombre y edad, observamos que se despliega en la ventana.



Después cree el método Cli.

```
import datetime
import hug

@hug.cli() #Se usa como decorador
@hug.get(examples="name=Jhon Doe&age=30")
@hug.local()

#Definimos la función say_hello
def say_hello(name: hug.types.text, age: hug.types.number, hug_timer=3):#hug.typ
    year_of_birth = datetime.datetime.now().year - age
    return {#Decimos hola al usuario y calculamos su año de nacimiento
        'message': "Hola {0}, naciste el año {1}".format(name, year_of_birth),
        'took': float(hug_timer)
    }

if __name__ == '__main__':
    print(say_hello("panchito", 50))
```

Y ejecute desde la línea de comandos dicho método, se muestra la existencia del endpoint say_hello.

```
C:\Users\KzyCk\Documents\Python\api\sps\helloworld\v1>hug -f AccesoCli.py -c say_Hello "Panchito" 50
AccesoCli
Available Commands:
- say_hello
```

Haciendo pruebas en el cliente Rest de Postman, y haciendo una solicitud tipo GET, obtenemos, lo siguiente.

