Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Escuela de vacaciones junio 2023
Laboratorio Software Avanzado
Ing. Marco Tulio Aldana Prillwitz
Aux. Fernando José Paz González

Practica 1

Jairo Sebastian Ramírez Palacios
201800712

Índice

Link de repositorio

Para mayor referencia de librerías y frameworks utilizados
[Zeebaz/soap-rest-request (github.com)]

REST

Para el consumo de servicios REST se utilizó el lenguaje JavaScript, con la librería React para el desarrollo del cliente web y Axios para soporte de peticiones http y https.

Se implementó unas interfaces que permiten el consumo de endpoints post y get, con soporte de la librería axios, mediante el URL y el cuerpo de la petición en el caso del método post.

```
import axios from "axios";

export const getRestResquest = async (url) => {
  try {
    const response = await axios.get(url);
    return response.data;
  } catch (error) {
    console.log("error on get rest", error);
  }
};

export const postRestRequest = async (url, data) => {
  try {
    const response = await axios.post(url, data);
    return response.data;
  } catch (error) {
    console.log("error on post rest", error);
  }
};
```

Para el consumo de servicios rest, se utilizaron las apis de:
[PokéAPI (pokeapi.co)] y [JSONPlaceholder - Free Fake REST API (typicode.com)]

Petición GET

Donde a la pokeApi se hace consumo de su url get:
https://pokeapi.co/api/v2/pokemon?limit=25&offset=0

```
const getPokemons = async (url) => {
    const response = await getRestResquest(url);
    setData(response);
```

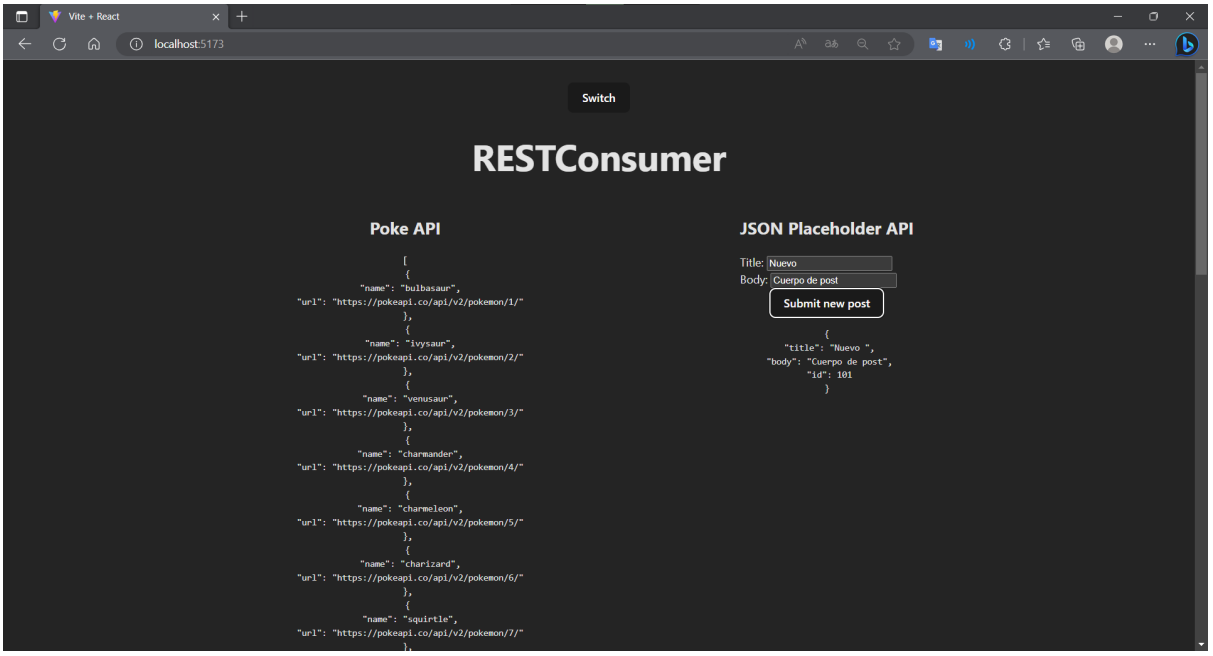
```
  };
```

Petición POST

Y JSONPlaceHolde postr:
https://jsonplaceholder.typicode.com/posts

```
const handleSubmit = async (e) => {
    e.preventDefault();
    console.log({ post });
    const response = await postRestRequest(
      "https://jsonplaceholder.typicode.com/posts",
      post
    );
    setResp(response);
    console.log({ response });
  };
```

Resultados del cliente React



SOAP

Para el desarrollo del backend se utilizó el framework Flask de Python, y la librería Zeep para soporte de peticiones SOAP.
Inicialización de cliente con el WSDL de countryinfo

```
def __init__(self):
```

```
        #URL proporcionado con todos los servicios
        self.url =
'http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoSer
vice.wso?WSDL'
        #objeto tipo cliente referente a la libreria zeep
        self.client = Client(self.url)
```

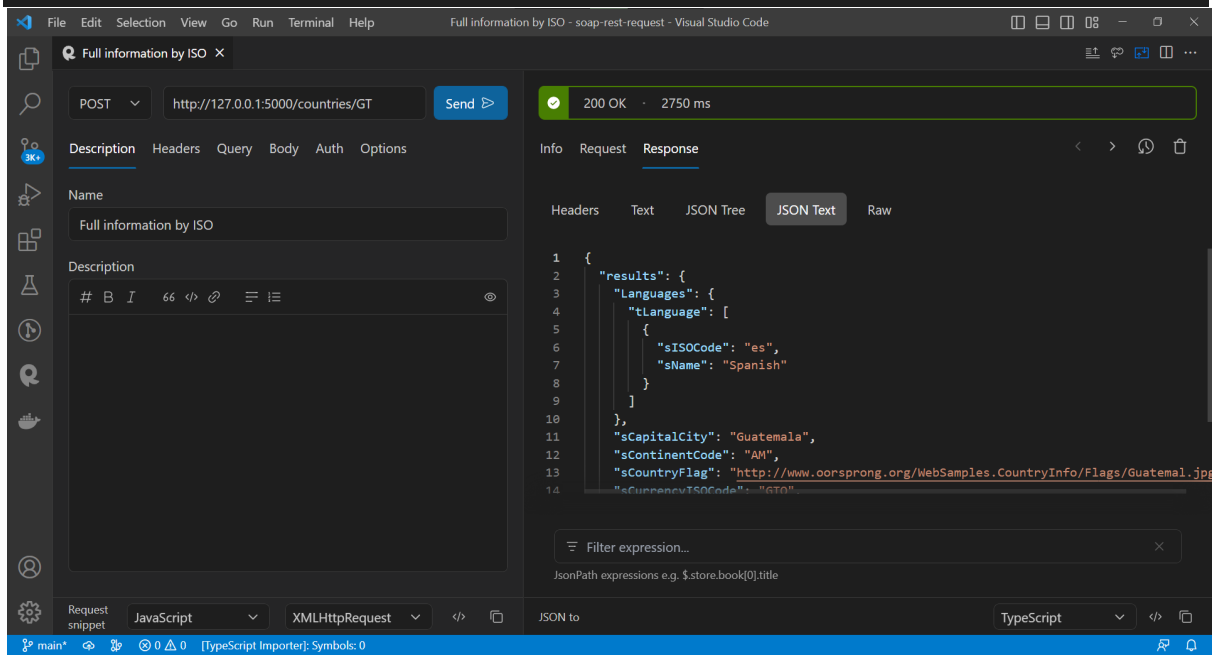
Peticiones POST

Se utilizó el método FullCountryInfo para obtener la información del País solicitado mediante el código ISO

```
@app.route("/countries/<isocode>", methods =['POST'])
def post_country_by_iso(isocode):
    # ejecutamos fullInformationCountry en base a el codigo iso
    response = countryInfoService.fullInformationCountry(isocode)

    return jsonify({'results': response}), 200
```

```
# Metodo para obtener la capital en base a su codigo iso (POST)
    def fullInformationCountry(self, isoCode):
        fullInformation = self.client.service.FullCountryInfo(isoCode)
        time.sleep(2.4)
        fullInformationFormat = {
            'sISOCode': str(fullInformation['sISOCode']),
            'sName': str(fullInformation['sName']),
            'sCapitalCity': str(fullInformation['sCapitalCity']),
            'sPhoneCode': str(fullInformation['sPhoneCode']),
            'sContinentCode': str(fullInformation['sContinentCode']),
            'sCurrencyISOCode':
str(fullInformation['sCurrencyISOCode']),
            'sCountryFlag': str(fullInformation['sCountryFlag']),
            'Languages': {
                'tLanguage': [
                    {
                        'sISOCode':
str(fullInformation['Languages']['tLanguage'][0]['sISOCode']),
                        'sName':
str(fullInformation['Languages']['tLanguage'][0]['sName'])
                    }
                ]
            }
        }
```

```
        return fullInformationFormat
```



Peticiones GET

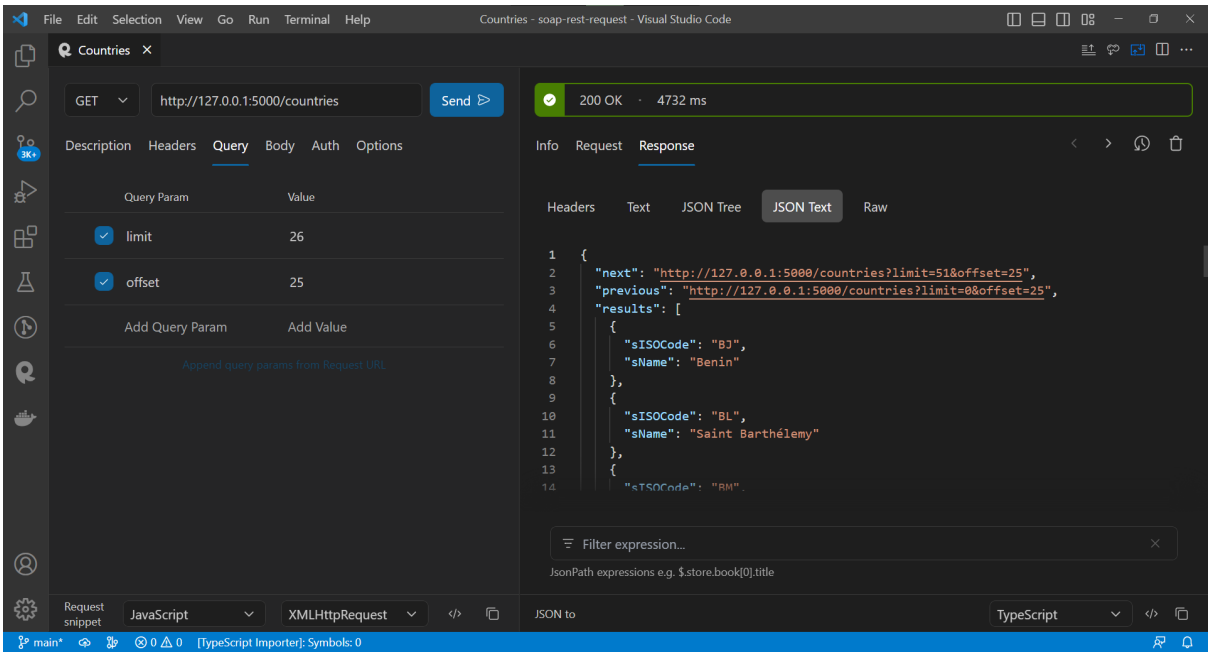
Se utilizó el método ListOfCountryNamesByCode para obtener todos los países con su respectivo ISO

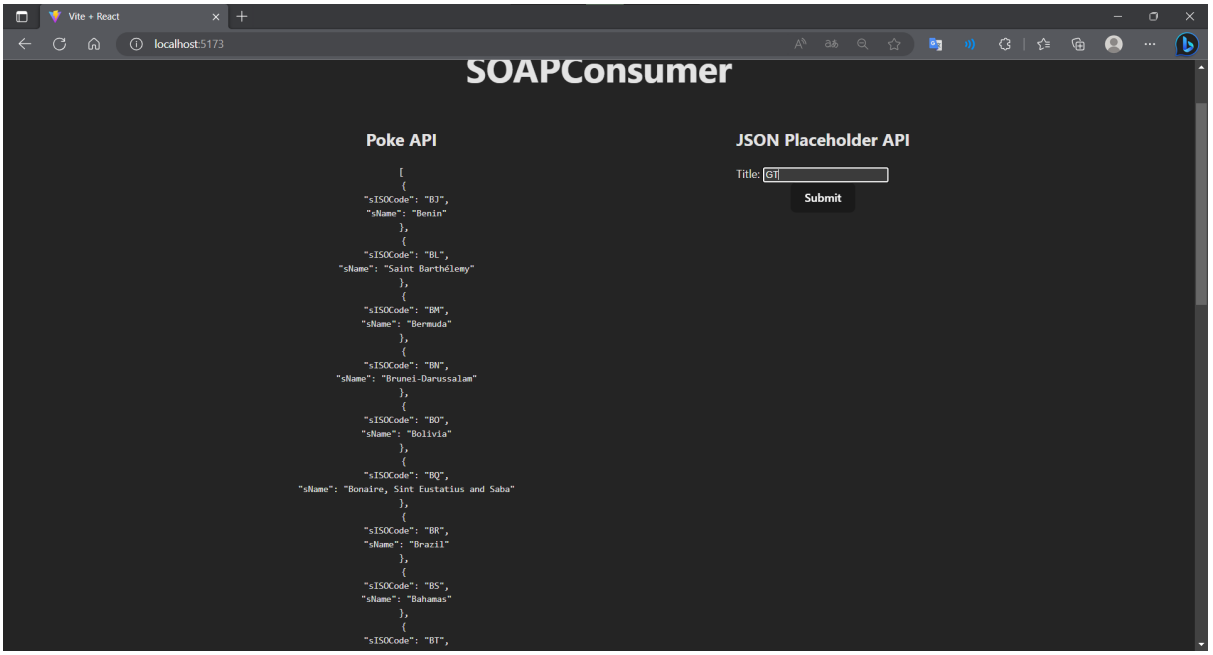
```
# Metodo para listar paises con su nombre y codigo iso (GET)
    def listOfCountryNamesByCode(self):
        response = self.client.service.ListOfCountryNamesByCode()
        time.sleep(2.4)
        return response
```

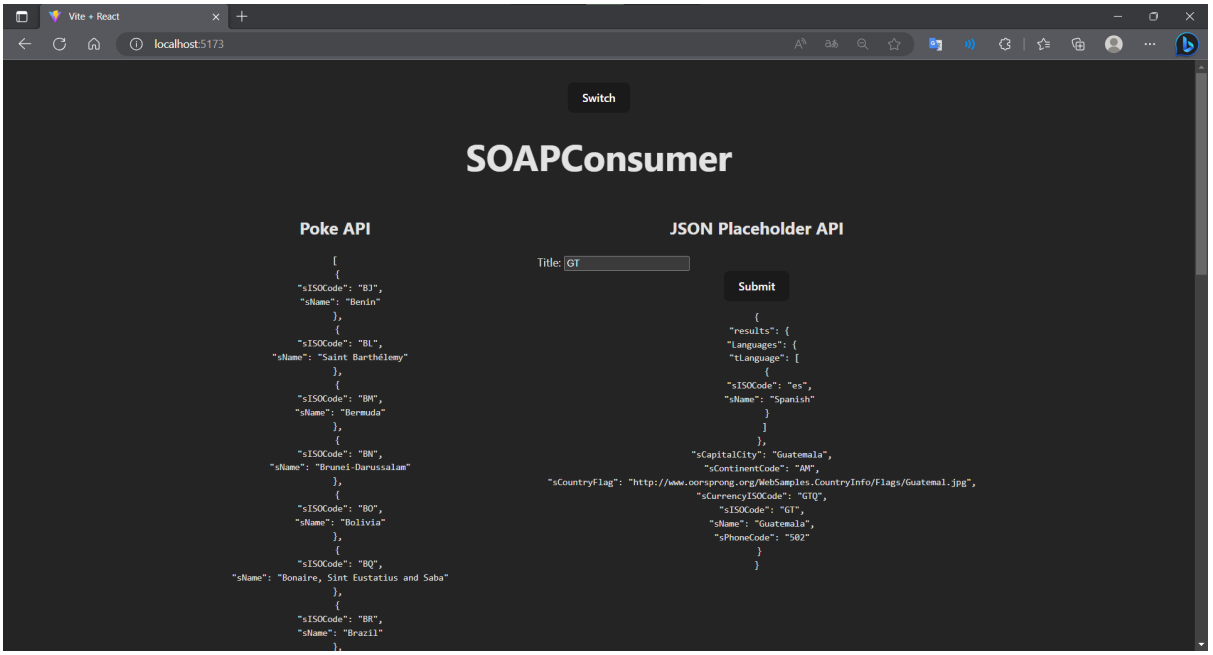
Donde mediante el manejo del arreglo de países se hizo una paginación en base a un limite y un desplazamiento.

```
@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

@app.route("/countries", methods =['GET'])
def get_countries():
    # obtenemos el limite y el desplazamiento en los parametros
    args = request.args
    limit = int(args.get('limit')) # inicio
    limit2 =  limit
    limit = limit if limit == 0 else limit -1

    offset = int(args.get('offset')) # desplazamiento
    for_next = limit + offset # link de siguientes
```

```
    # listamos los paises con su nombre y codigo iso
    response = countryInfoService.listOfCountryNamesByCode()

    countries_list = []
    for country in response:
        countries_list.append({
            'sISOCode': str(country['sISOCode']),
            'sName': str(country['sName']),
        })

    countries_list = countries_list[limit:for_next]

    calc_previous = str(limit2-offset) if limit2-offset >= 0 and
limit2-offset != 1 else str(0)
    next  =
'http://127.0.0.1:5000/countries?limit={limit}&offset={offset}'.format(
limit = str(for_next+1), offset = offset)
    previous =
'http://127.0.0.1:5000/countries?limit={limit}&offset={offset}'.format(
limit = calc_previous, offset = offset)

    time.sleep(2)
    # calc_previous =
    return jsonify({'results': countries_list,
                    'next': next,
                    'previous':previous}), 200
```




Vite + React
localhost:5173
SOAPConsumer
Poke API
[
{
"sISOCode": "BJ",
"sName": "Benin"
},
{
"sISOCode": "BL",
"sName": "Saint Barthélemy"
},
{
"sISOCode": "BM",
"sName": "Bermuda"
},
{
"sISOCode": "BN",
"sName": "Brunei-Darussalam"
},
{
"sISOCode": "BO",
"sName": "Bolivia"
},
{
"sISOCode": "BQ",
"sName": "Bonaire, Sint Eustatius and Saba"
},
{
"sISOCode": "BR",
"sName": "Brazil"
},
{
"sISOCode": "BS",
"sName": "Bahamas"
},
{
"sISOCode": "BT",
JSON Placeholder API
Title: GT
Submit


Vite + React
localhost:5173
Switch
SOAPConsumer
Poke API
[
{
"sISOCode": "BJ",
"sName": "Benin"
},
{
"sISOCode": "BL",
"sName": "Saint Barthélemy"
},
{
"sISOCode": "BM",
"sName": "Bermuda"
},
{
"sISOCode": "BN",
"sName": "Brunei-Darussalam"
},
{
"sISOCode": "BO",
"sName": "Bolivia"
},
{
"sISOCode": "BQ",
"sName": "Bonaire, Sint Eustatius and Saba"
},
{
"sISOCode": "BR",
"sName": "Brazil"
},
JSON Placeholder API
Title: GT
Submit
{
"results": {
"Languages": {
"tLanguage": [
{
"sISOCode": "es",
"sName": "Spanish"
}
]
},
"sCapitalCity": "Guatemala",
"sContinentCode": "AM",
"sCountryFlag": "http://www.oorsprong.org/WebSamples.CountryInfo/Flags/Guatemal.jpg",
"sCurrencyISOCode": "GTQ",
"sISOCode": "GT",
"sName": "Guatemala",
"sPhoneCode": "502"
}
}