

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Curso: Laboratorio Software Avanzado

1er. Semestre Vacaciones 2023






# **Tarea 1: Practica 1 (REST-SOAP)**

Anggelo Santiago Son Mux

201709502

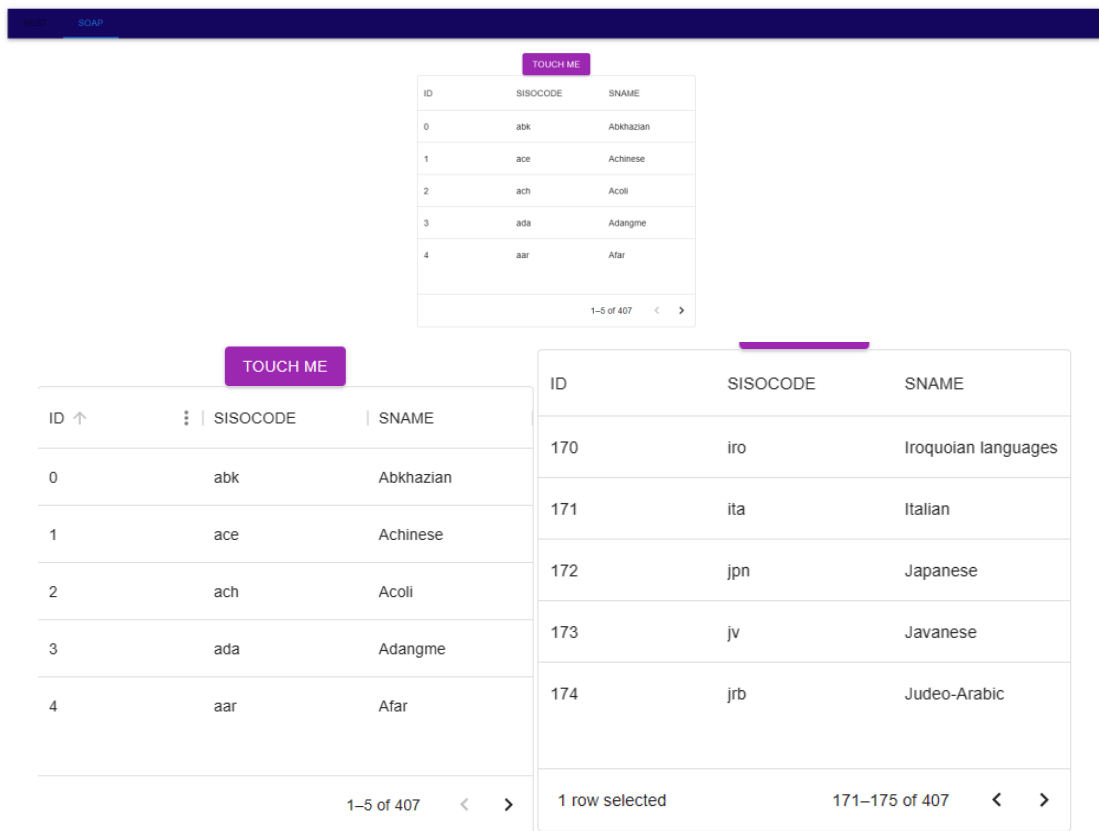
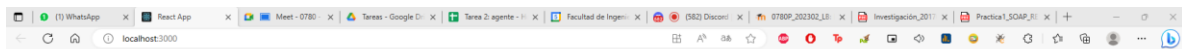
# REST



ID ↑	Picture_url	Title	Description
0		Maria†Holic	Both tries to turn t...
1		Birdie Wing: Golf ...	Both shows are s...
2		Ouran Koukou Ho...	Funny love storie...
3		Isekai Ojisan	Both series are co...
4		Ookami Kodomo ...	The series are ab...
1-5 of 100 < >			

Se realizo una petición post la cual consulta las recomendaciones que han realizado los usuarios de una página de anime mostrando en una tabla los resultados obtenidos, mostrando la imagen, titulo y descripción de una recomendación.

## SOAP



Para la petición SOAP se utilizó un botón el cual realiza una petición SOAP la cual devuelve las siglas y el nombre de un lenguaje, mostrando los resultados en una tabla.

## FUNCIONES

### GET

Se utilizó fetch para poder acceder a un canal HTTP para poder realizar peticiones y respuestas, la api que se utilizó es de myanimelist que se encuentra en rapidAPI, esta API devuelve las recomendaciones que los usuarios realizaron en la página de myanimelist y luego se paginan los resultados con la ayuda de dataGrid el cual muestra de cinco en cinco los resultados de la consulta.

```
const fetchMe = async () =>{
  const url = 'https://myanimelist.p.rapidapi.com/anime/recommendations/1';
  const options = {
    method: 'GET',
    headers: {
      'X-RapidAPI-Key': '3768b2a0femshcccb442eb31be89p193c98jsn48a32ed93aeb',
      'X-RapidAPI-Host': 'myanimelist.p.rapidapi.com'
    }
  };

  try {
    const response = await fetch(url, options);
    const result = await response.json();
    const data = await result.recommendations
    console.log(result);
    setContainer(data)
  } catch (error) {
    console.error(error);
  }
}
```

Del resultado obtenido de la consulta se toman los campos de la url de la imagen, el titulo y la descripción de una recomendación para crear las columnas y filas del dataGrid

```
const columns = [
  {field: "id", headerName: "ID", width: 150},
  {field: "picture_url", headerName: "Picture_url", width: 150, editable: true, renderCell: (params)=><img src={params.value} alt=""/>,
  {field: "title", headerName: "Title", width: 150},
  {field: "description", headerName: "Description", width: 150}
];

const rows = container.map((item, index)=>({
  id: index,
  picture_url: item.recommendation.picture_url,
  title: item.recommendation.title,
  description: item.description
})))
```

## SOAP

Para soap se utilizó axios el cual realiza una petición a una API SOAP gratuita la cual devuelve el resultado de un listado de los lenguajes que existen y sus abreviaciones.

El resultado XML se convirtió a JSON para poder acceder a los elementos necesarios de la respuesta siendo el nombre del lenguaje y su abreviación.

```

const soapRequest = async () => {
  const options = {
    method: 'POST',
    url: 'http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso',
    headers: {
      'Content-Type': 'text/xml',
    },
    data: '<?xml version="1.0" encoding="utf-8"?>'+
    '<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">'+
    '  <soap:Body>'+
    '    <ListOfLanguagesByName xmlns="http://www.oorsprong.org/websamples.countryinfo">'+
    '      </ListOfLanguagesByName>'+
    '    </soap:Body>'+
    '  </soap:Envelope>'
  };
};

try {
  const response = await axios.request(options);
  const xml = response.data;
  const convertXML = require("simple-xml-to-json").convertXML;
  const json = convertXML(xml);
  //console.log(json)
  const dataJSON = json['soap:Envelope'].children[0]['soap:Body'].children[0]['m:ListOfLanguagesByNameResponse'].children[0]['m:ListOfLanguagesByNameResult'].children
  console.log(dataJSON);
  setContainer2(dataJSON)
} catch (error) {
  console.error(error);
}
}

```

Del resultado obtenido de la consulta se toman los campos del nombre y la abreviación de un lenguaje para crear las columnas y filas del dataGrid

```

const columns2 = [
  {field: "id", headerName: "ID", width: 150},
  {field: "sISOCode", headerName: "SISOCODE", width: 150},
  {field: "sName", headerName: "SNAME", width: 150}
];

const rows2 = container2.map((item, index)=>({
  id: index,
  sISOCode: item['m:tLanguage'].children[0]['m:sISOCode'].content,
  sName: item['m:tLanguage'].children[1]['m:sName'].content
})))

```

## RESPUESTA XML

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <m:ListOfLanguagesByNameResponse
xmlns:m="http://www.oorsprong.org/websamples.countryinfo">
      <m:ListOfLanguagesByNameResult>
        <m:tLanguage>
          <m:sISOCode>abk</m:sISOCode>
          <m:sName>Abkhazian</m:sName>
        </m:tLanguage>
        <m:tLanguage>
          <m:sISOCode>ace</m:sISOCode>
          <m:sName>Achinese</m:sName>
        </m:tLanguage>
        <m:tLanguage>
          <m:sISOCode>ach</m:sISOCode>
          <m:sName>Acoli</m:sName>
        </m:tLanguage>

```

