Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Software Avanzado - Sección P

Ing. Marco Aldana

Aux. Fernando Paz

Práctica 1 - SOAP Y REST

Nombre	Registro Académico
Fernando Josué Tello Valiente	201800714

Explicación de la solución

Repositorio del proyecto: https://github.com/FerJoTello/sa_practica1

La aplicación fue elaborada en Next.js versión 13.4.4 que consiste en un sistema web donde se hace consumo de API's REST y SOAP. Es posible clonar el proyecto y ejecutarlo con NodeJS versión 18.14.0.

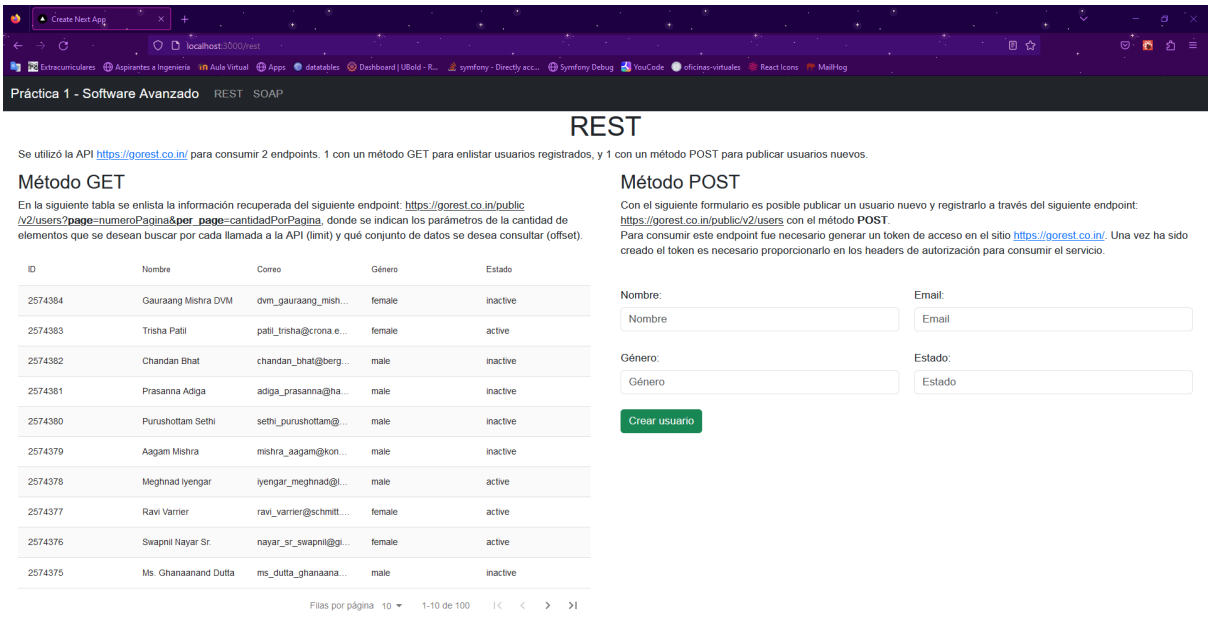
Las API's utilizadas fueron las siguientes:

- https://gorest.co.in/: API Rest que permite realizar peticiones basadas en recuperación de usuarios. Esta API es pública para consumir métodos GET, sin embargo es necesario registrarse para generar un Token de Acceso para consumir métodos POST, PUT y DELETE, por lo que fue necesario registrarse.
- http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso; API que se basa en protocolo SOAP que permite recuperar la información de países como su nombre y su código ISO.
- https://www.dataaccess.com/webservicesserver/NumberConversion.wso; API SOAP que permite convertir un número a letras devolviendo el nombre del número que se provea.

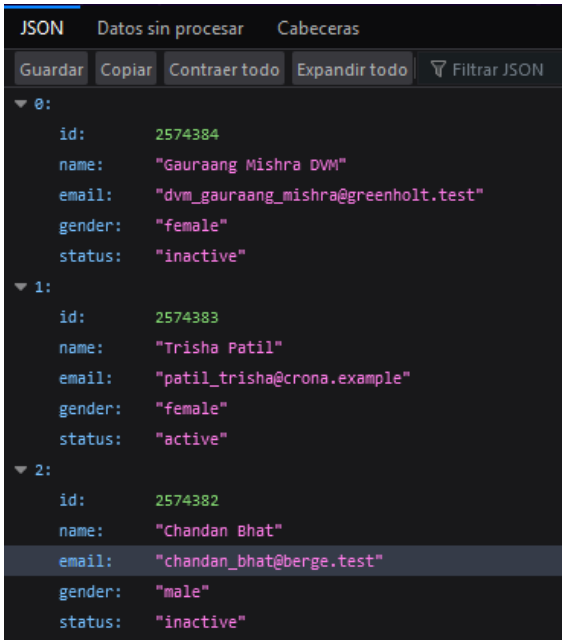
Las API's basadas en SOAP fueron halladas en el siguiente enlace: https://documenter.getpostman.com/view/8854915/Szf26WHn.

Consumos REST

Para visualizar el funcionamiento de los servicios REST se elaboró una vista donde se indica las funcionalidades disponibles y qué método está utilizando.



Para poner en práctica el método GET del servicio REST se consume un endpoint que consulta una cantidad de usuarios indicando un paginado a través de la siguiente ruta: https://gorest.co.in/public/v2/users?page=1per_page=10 donde los parámetros page y per_page pueden ser modificados para consultar e implementar un paginado. La respuesta JSON de este servicio sigue el siguiente formato de ejemplo:



Estos datos son mostrados en una tabla mostrando de forma ordenada la información que devuelve el servicio.

Método GET

En la siguiente tabla se enlista la información recuperada del siguiente endpoint: https://gorest.co.in/public/v2/users?**page**=numeroPagina&**per_page**=cantidadPorPagina, donde se indican los parámetros de la cantidad de elementos que se desean buscar por cada llamada a la API (limit) y qué conjunto de datos se desea consultar (offset).

ID	Nombre	Correo	Género	Estado
2574384	Gauraang Mishra DVM	dvm_gauraang_mish...	female	inactive
2574383	Trisha Patil	patil_trisha@crona.e...	female	active
2574382	Chandan Bhat	chandan_bhat@berg...	male	inactive
2574381	Prasanna Adiga	adiga_prasanna@ha...	male	inactive
2574380	Purushottam Sethi	sethi_purushottam@...	male	inactive
2574379	Aagam Mishra	mishra_aagam@kon...	male	inactive
2574378	Meghnad Iyengar	iyengar_meghnad@l...	male	active
2574377	Ravi Varrier	ravi_varrier@schmitt....	female	active
2574376	Swapnil Nayar Sr.	nayar_sr_swapnil@gi...	female	active
2574375	Ms. Ghanaanand Dutta	ms_dutta_ghanaana...	male	inactive

Filas por página 10 1-10 de 100

En el archivo **UserTable.jsx** del código fuente se puede encontrar cuál es la forma de consumo, implementando el método *fetch* nativo de Javascript dentro de un componente React que hace una petición HTTP-GET a la ruta indicada con los parámetros detallados de “página” y “porPagina” que definen la cantidad de usuarios que se desea solicitar.

A su vez, para aplicar el método POST se consume un endpoint que permite crear un usuario por medio de un formulario.

Método POST

Con el siguiente formulario es posible publicar un usuario nuevo y registrarlo a través del siguiente endpoint: https://gorest.co.in/public/v2/users con el método **POST**.
Para consumir este endpoint fue necesario generar un token de acceso en el sitio https://gorest.co.in/. Una vez ha sido creado el token es necesario proporcionarlo en los headers de autorización para consumir el servicio.

Nombre: Fernando

Email: fernando@fakemail.com

Género: male

Estado: active

Crear usuario

Al enviar la información, el servicio nos devuelve una respuesta con el detalle del usuario que se está creando, incluyendo un ID, el cual fue el que la API le asignó cuando se registró. A continuación se muestra un ejemplo al realizar una petición con *curl*.

En la vista de la página web esto es desplegado de la siguiente manera después de haber llenado los campos y presionar el botón **Crear usuario**.

Método POST

Con el siguiente formulario es posible publicar un usuario nuevo y registrarlo a través del siguiente endpoint: https://gorest.co.in/public/v2/users con el método **POST**.
Para consumir este endpoint fue necesario generar un token de acceso en el sitio https://gorest.co.in/. Una vez ha sido creado el token es necesario proporcionarlo en los headers de autorización para consumir el servicio.

Nombre: Nombre

Email: Email

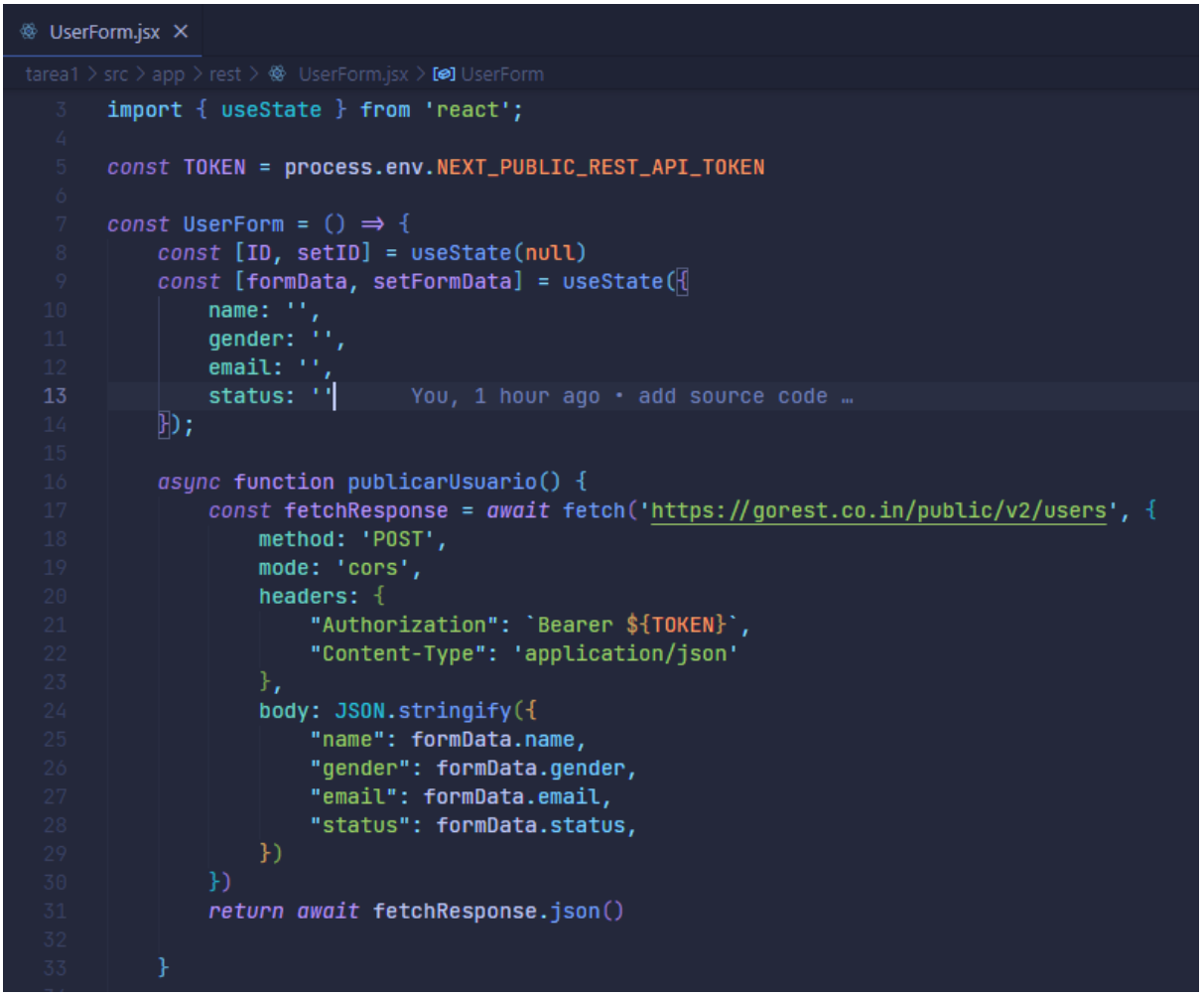
Género: Género

Estado: Estado

Crear usuario

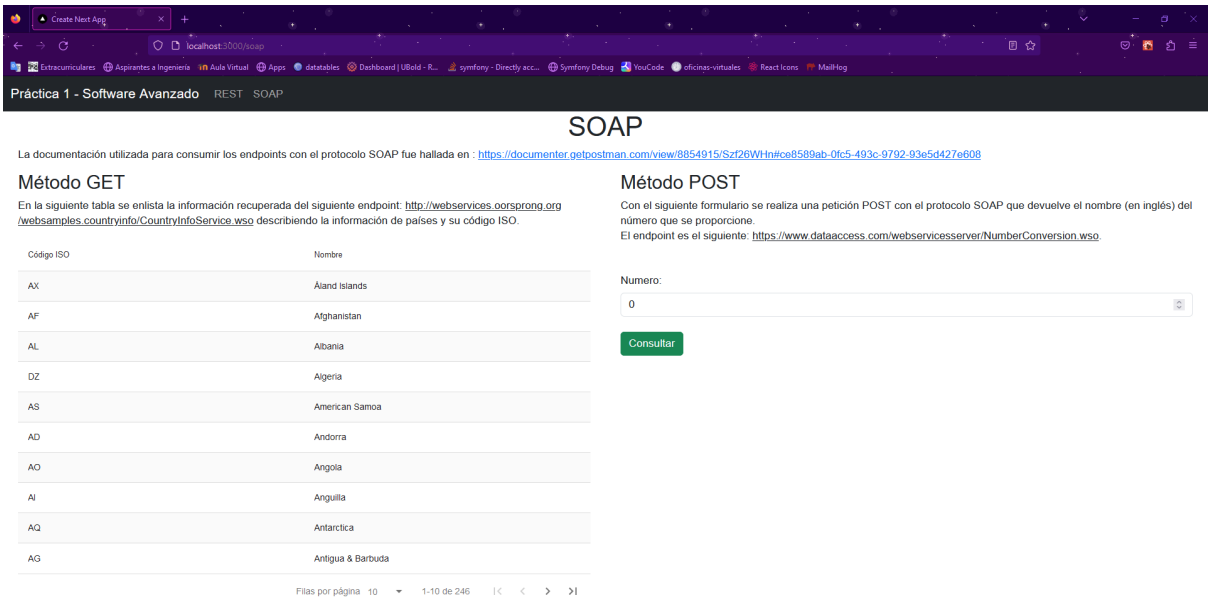
Usuario creado con ID: 2576198

De igual manera, la implementación del consumo del método HTTP-POST se puede encontrar en el archivo **UserForm.jsx**, donde se vuelve a usar el método fetch pero se especifica el método y los parámetros que el endpoint necesita para funcionar, como los headers de autorización y el body.



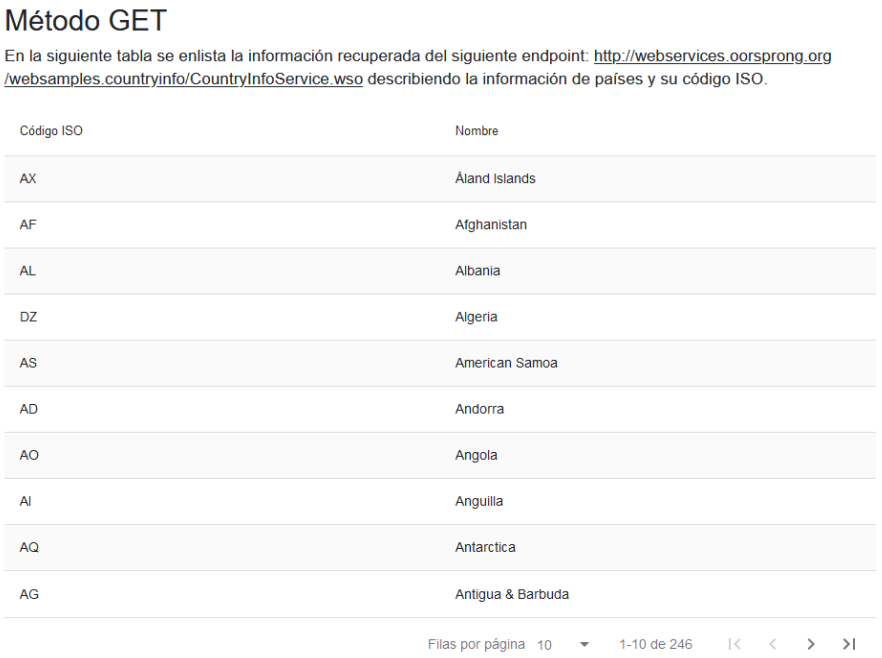
Consumos SOAP

Para visualizar el funcionamiento de los servicios SOAP se elaboró una vista donde se indica las funcionalidades disponibles y qué método está utilizando.



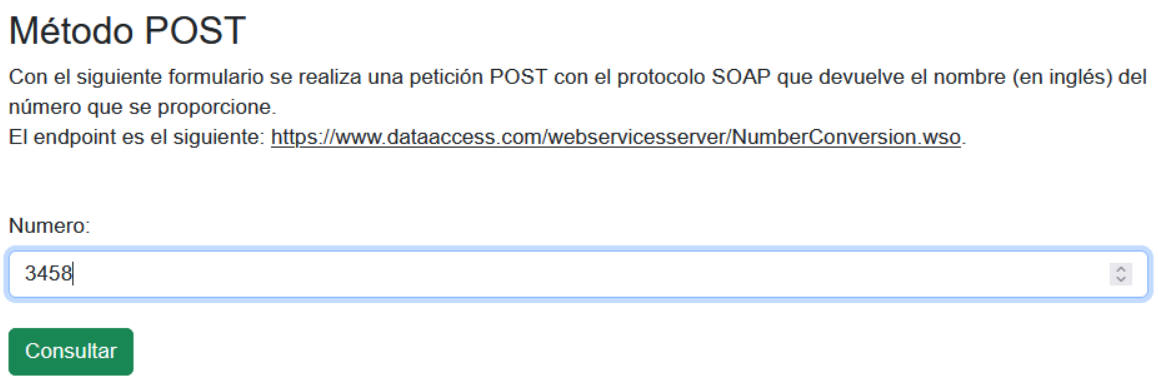
Para poner en práctica el método GET del servicio SOAP se consume un endpoint que consulta un listado de países indicando su nombre y su código ISO a través del siguiente enlace: http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso. La respuesta XML de este servicio sigue el siguiente formato de ejemplo:

Estos datos son mostrados en una tabla mostrando de forma ordenada la información que devuelve el servicio.



En el archivo **obtenerPaises.js** del código fuente se puede encontrar cuál es la forma de consumo, usando la biblioteca de Node llamada *axios* que permite hacer peticiones y procesar la respuesta de forma práctica.

A su vez, para aplicar el método POST se consume un endpoint que permite convertir un número a letras devolviendo el nombre del número en palabras (en inglés).



Después de proporcionar un número y presionar el botón **Consultar** se despliega la respuesta del servicio indicando el nombre del número en inglés.

Método POST

Con el siguiente formulario se realiza una petición POST con el protocolo SOAP que devuelve el nombre (en inglés) del número que se proporcione.
El endpoint es el siguiente: https://www.dataaccess.com/webservicesserver/NumberConversion.wso.

Numero:

0

Consultar

Respuesta en palabras: three thousand four hundred and fifty eight

De la misma manera, la implementación del consumo del método SOAP-POST se puede encontrar en el archivo **api/soap/route.js**, dónde Next funciona como un backend y realiza la petición SOAP resolviendo así los problemas de CORS que se presentan al consumir los endpoints desde un cliente web (navegador). Después de recuperar la información de la petición con axios nuevamente, se procesa para acceder al dato de la conversión del número y queda disponible para consultas desde el cliente.