



***Universidad Nacional de Lanús***

**DESARROLLO DE SOFTWARE EN SISTEMAS  
DISTRIBUIDOS**

**Web Services: SOAP/REST**

**Departamento: Desarrollo Productivo y Tecnológico**

**Carrera: Licenciatura en Sistemas**

**Año: 2024**

**Cuatrimestre: 2° Cuatrimestre**

**Docentes:**

**Ing. Diego Andrés Azcurra  
Lic. Marcos Amaro**

**Alumnos:**

**Fernando Jose Luis Camacho (DNI: 35.459.731)  
Franco Alberto Astorga (DNI: 42.359.479)  
Franco Ariel Figueroa (DNI: 42.684.291)  
Juan Leandro Gauna (DNI: 31.654.945)**

# Índice

<u>1. Código fuente</u> .....	3
<u>2. Roles y tareas desempeñadas</u> .....	3
<u>3. Estrategia de resolución</u> .....	3
<u>3.1. Base de datos</u> .....	3
<u>3.2. Cliente SOAP</u> .....	4
<u>3.3. Servidor SOAP</u> .....	5
<u>3.4. Interfaz gráfica</u> .....	6
<u>3.5. Ejecutables</u> .....	6
<u>4. Pruebas realizadas</u> .....	7
<u>5. Pendiente del segundo trabajo práctico</u> .....	7

# 1. Código fuente

Enlace al repositorio público de Github del tercer trabajo práctico:

<https://github.com/Franco2900/soapStockearteGrupo5.git>

## 2. Roles y tareas desempeñadas

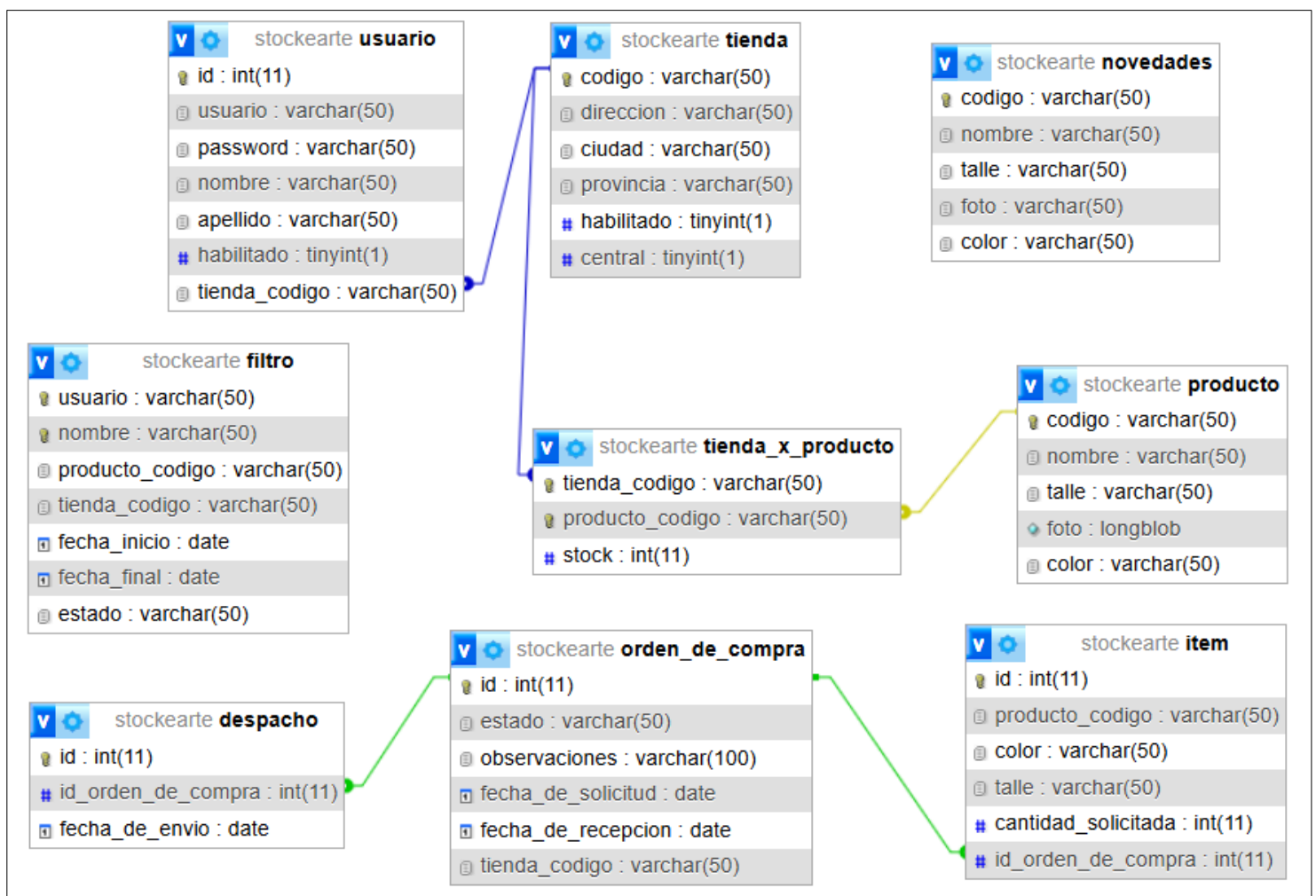
- Desarrollo del cliente SOAP y servidor SOAP: Franco Astorga y Franco Figueroa
- Desarrollo de lo pendiente en el segundo trabajo práctico: Fernando Camacho y Leandro Gauna

## 3. Estrategia de resolución

### 3.1. Base de datos

Para manejar los ABM de los filtros se añadió una nueva tabla a la base de datos stockearte. Ya que distintos usuarios pueden crear filtros con el mismo nombre, se decidió ponerle doble llave primaria, el nombre del filtro y el usuario que lo creó.

Diagrama Entidad Relación (DER) de la base de datos stockearte

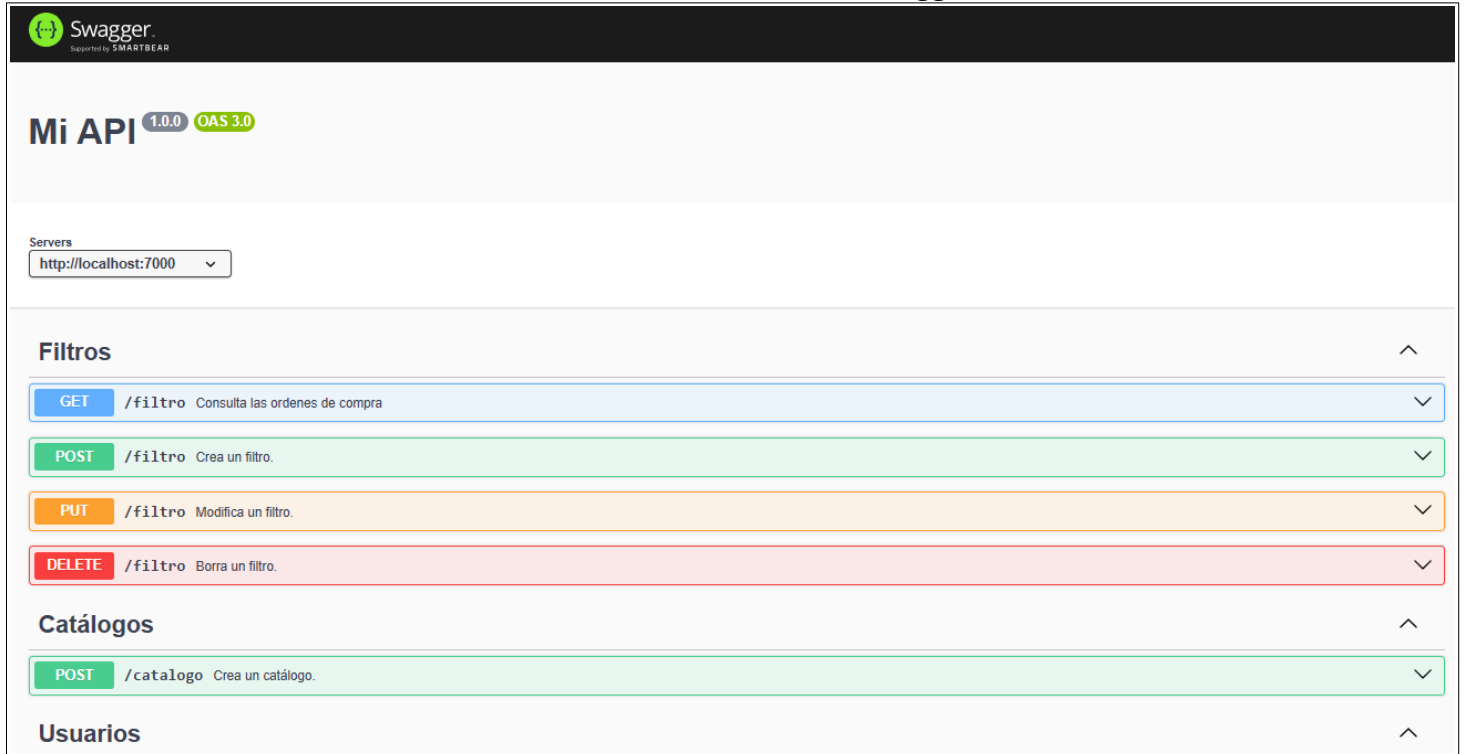


### 3.2. Cliente SOAP

Se desarrolló un nuevo cliente SOAP con Node.js para que actué como intermediario entre la interfaz front-end y el servidor SOAP.

El cliente recibe las peticiones de la interfaz gráfica mediante una API REST que tiene incorporado. Dicha API esta documentada con Swagger y se puede ver y usar en la siguiente dirección: <http://localhost:7000/api-docs/>

#### API REST documentada con Swagger



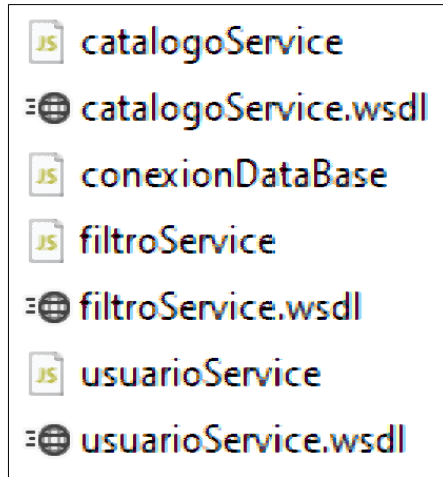
The screenshot shows the Swagger UI for an API. At the top, it says "Swagger" with a logo and "powered by SMARTBEAR". Below that, it says "Mi API" with version "1.0.0" and "OAS 3.0". There is a "Servers" dropdown menu showing "http://localhost:7000". The main content is divided into sections: "Filtros" (Filters) with four items (GET, POST, PUT, DELETE) for the "/filtro" endpoint, "Catálogos" (Catalogs) with one item (POST) for the "/catalogo" endpoint, and "Usuarios" (Users) at the bottom. Each item has a description of the action and a dropdown arrow.

Los rutas GET reciben los datos mediante la URL mientras que las rutas POST, PUT y DELETE reciben los datos en el cuerpo de la solicitud. Con los datos que le llegan de la interfaz gráfica, el cliente hace solicitudes SOAP al servidor y lo que este devuelva es enviado a la interfaz gráfica.

### 3.3. Servidor SOAP

El servidor recibe peticiones en formato SOAP y devuelve las respuestas en el mismo formato. Cada elemento del que se encarga el servidor tiene su propio archivo WSDL, los filtros tienen su propio archivo .wsdl, los catalogos tienen el suyo y así. Los archivos WSDL junto con los archivos que definen la lógica de los métodos de dichos archivos se encuentran dentro de la carpeta Lógica.

Archivos dentro de la carpeta Lógica



El servidor trabaja con la base de datos stockearte, misma base de datos usada en anteriores trabajos prácticos, para hacer el CRUD de los datos.

El servidor escucha por peticiones en la siguiente dirección: <http://localhost:9000/>

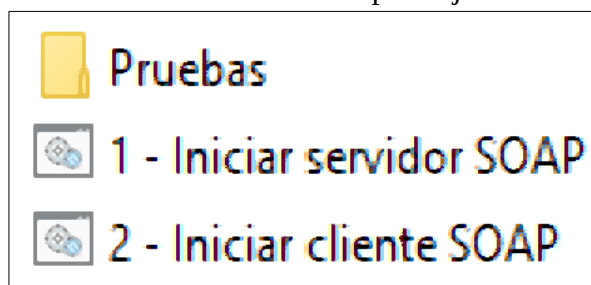
### 3.4. Interfaz gráfica

PONER ALGO ACÁ

### 3.5. Ejecutables

Esta carpeta tiene archivos .bat que sirven para ejecutar comandos automáticamente en la consola de comandos de Windows (cmd). Los ejecutables que estan en dicha carpeta permiten iniciar el cliente SOAP y el servidor SOAP. También hay un ejecutable en la carpeta Pruebas para cargar datos de prueba directamente a la base de datos stockearte.

Archivos .bat en la carpeta Ejecutables



## 4. Pruebas realizadas

Todas las pruebas fueron realizadas en Windows 10 Home. Para poder realizar las pruebas es necesario tener Xampp y Node.js instalados.

Enlace al video de las pruebas realizadas:

ENLACE ACÁ

## 5. Pendiente del segundo trabajo práctico

Enlace al repositorio público de Github del primer trabajo práctico:

<https://github.com/Franco2900/stockearteGrupo5.git>

Enlace al repositorio público de Github del segundo trabajo práctico:

<https://github.com/Franco2900/proveedorKafkaGrupo5.git>

Enlace para mostrar que se completo la interfaz gráfica faltante en el segundo trabajo práctico:

ENLACE ACÁ