



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>	5
<u>3.3. Servidor SOAP</u>	6
<u>3.4. Interfaz gráfica</u>	7
<u>3.5. Ejecutables</u>	7
<u>4. Pruebas realizadas</u>	8
<u>5. Pendiente del segundo trabajo práctico</u>	8
<u>6. Diagrama de comunicación</u>	8

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 las vistas: 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, esta tabla está relacionada con la tabla usuario. También se añadieron dos tablas para manejar los catálogos, la tabla catalogo para saber a qué tienda pertenece el catálogo y almacenar el archivo PDF del catálogo y la tabla intermedia catalogo_x_producto para saber los productos que contiene los catálogos y los detalles de los productos.

También se arregló la falta de relación entre la tabla tienda y la tabla orden_de_compra, pendiente del trabajo práctico anterior.

LO QUE ESTA EN AMARILLO ES LO QUE FALTA HACER

Diagrama Entidad Relación (DER) de la base de datos stockearte antes de los cambios

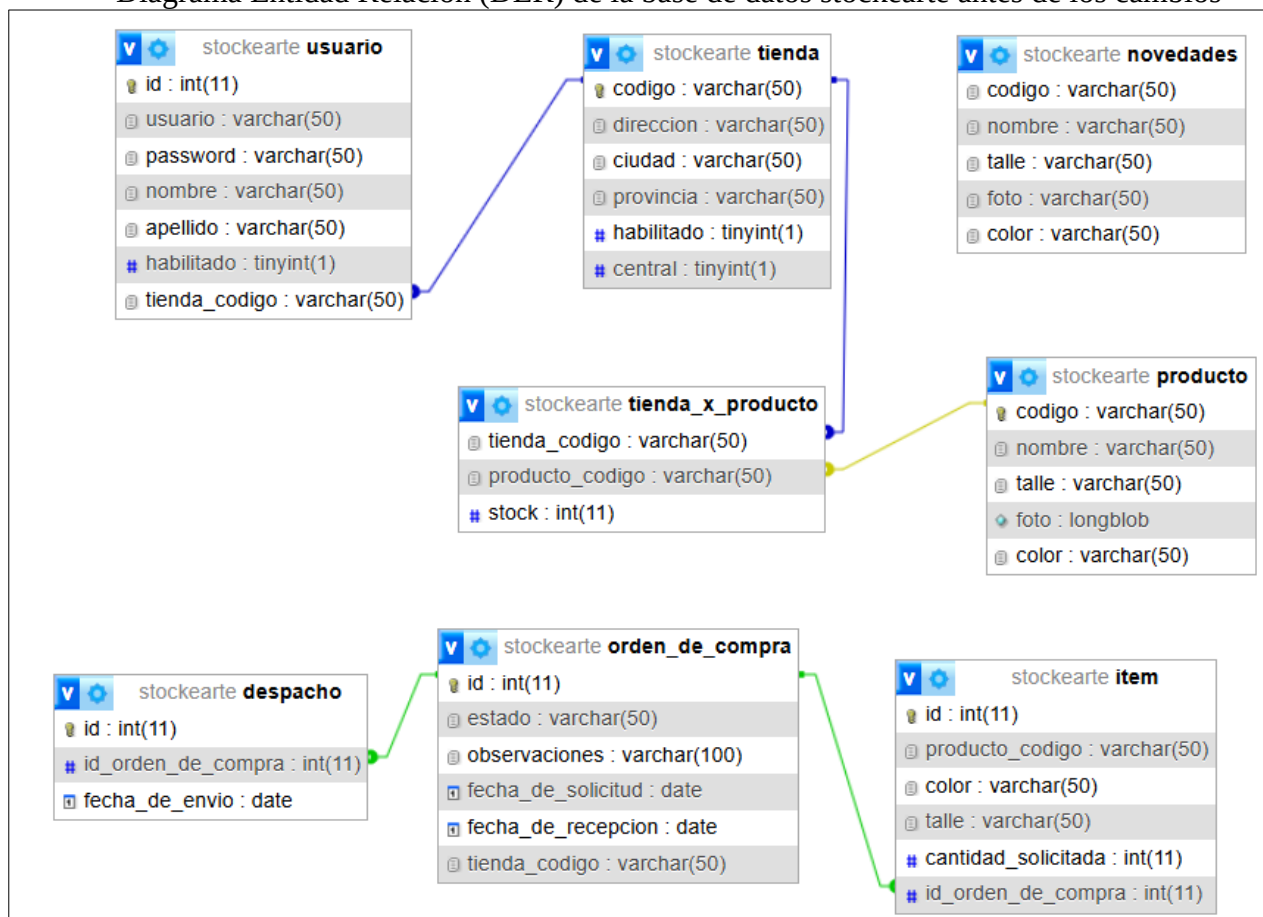
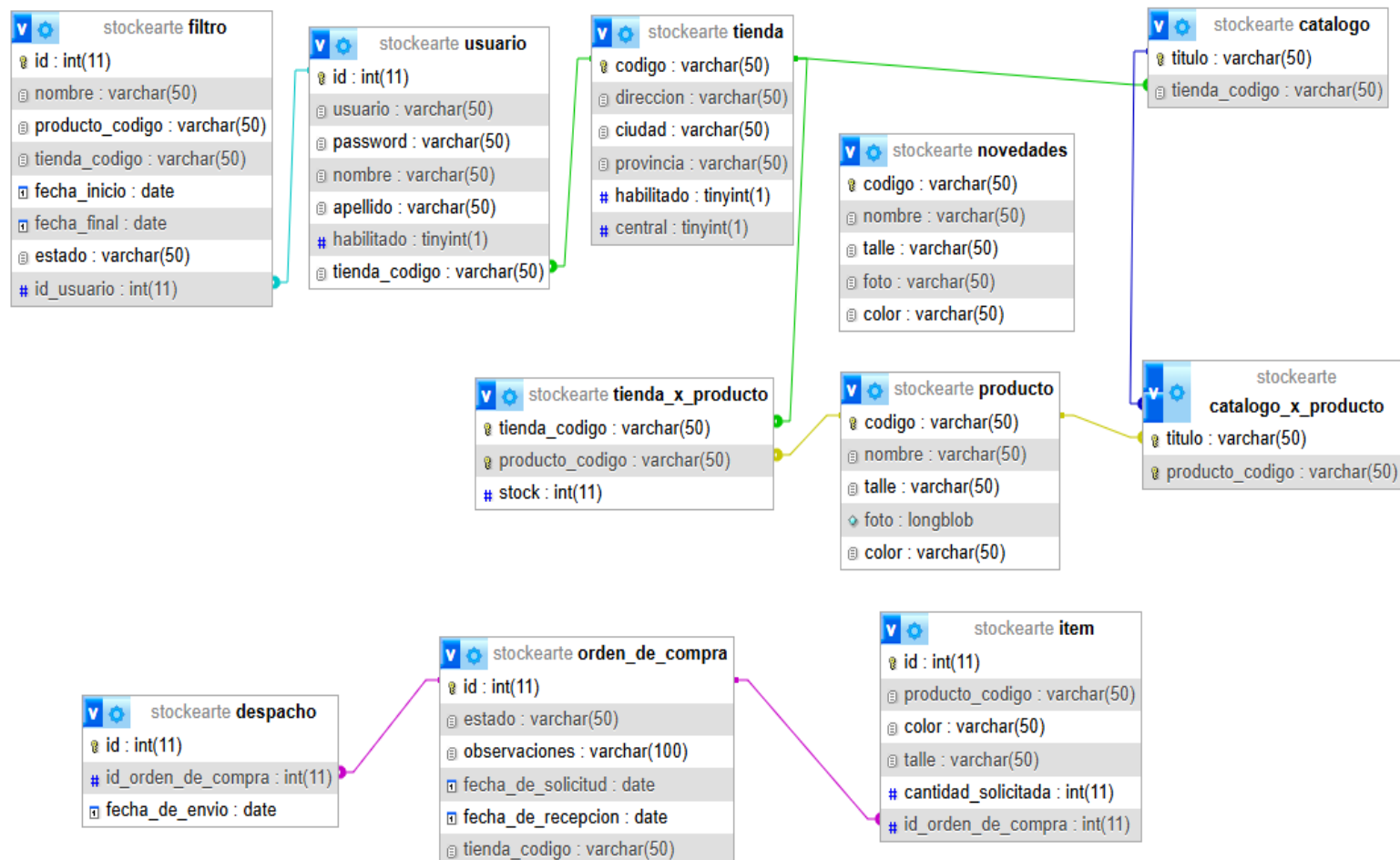


Diagrama Entidad Relación (DER) de la base de datos stockearte después de los cambios

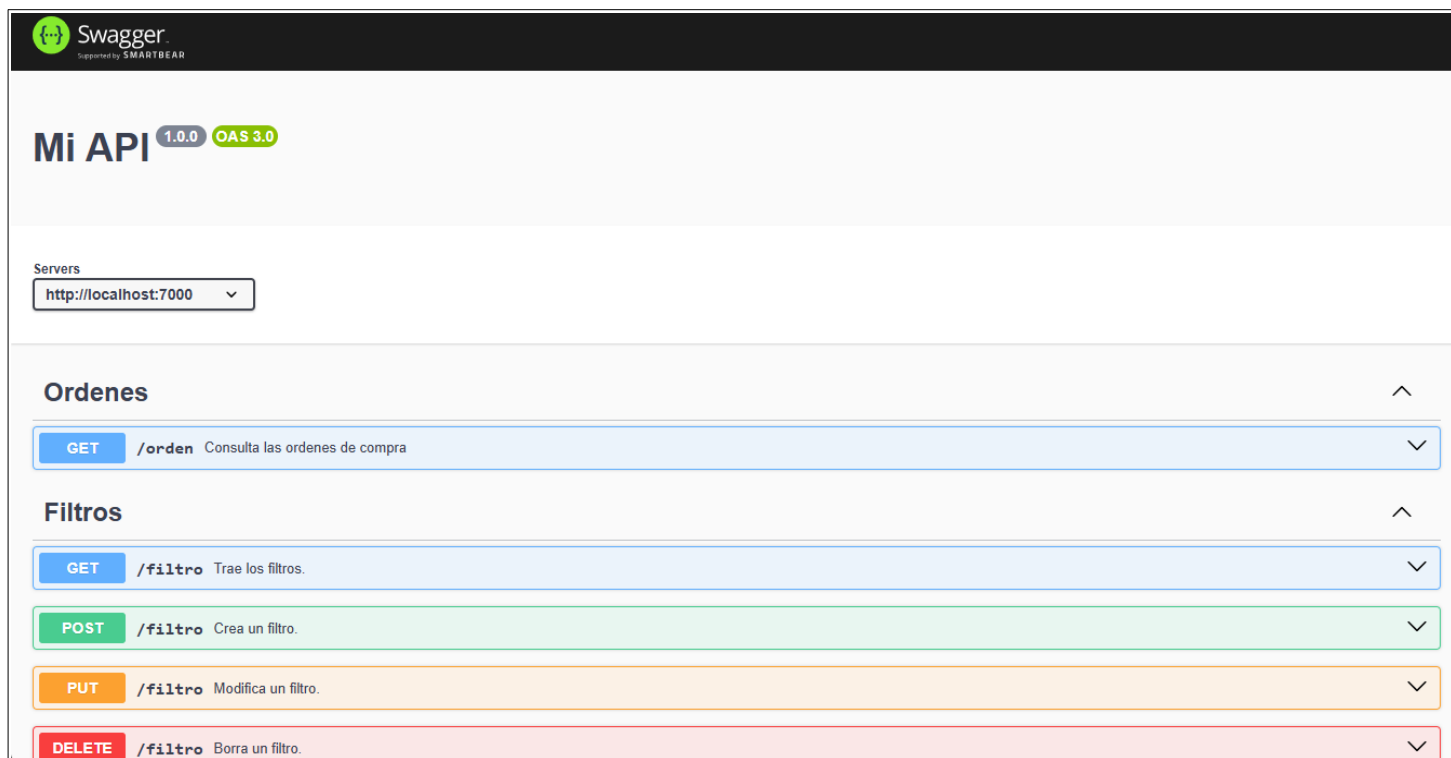


3.2. Cliente SOAP

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



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..

Hay cuatro rutas, una para cada elemento que solicita la consigna:

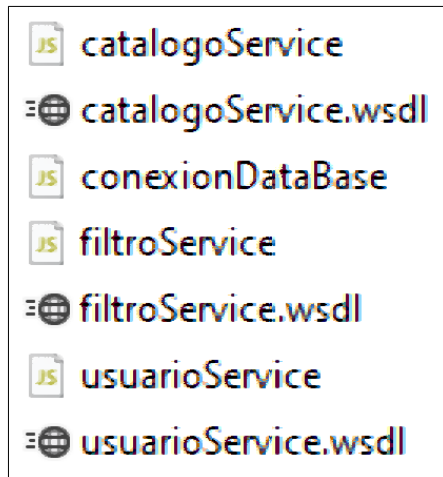
- /orden: Para manejar las consultas de las ordenes de compras.
- /filtro: Para manejar el CRUD de los filtros.
- /catalogo: Para manejar el CRUD de los catalogos
- /usuario: Para hacer el alta de usuarios con un archivo .csv.

El cliente SOAP escucha por las peticiones de la interfaz gráfica en la siguiente dirección: <http://localhost:7000/>

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 lo mismo con los demás elementos. 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.

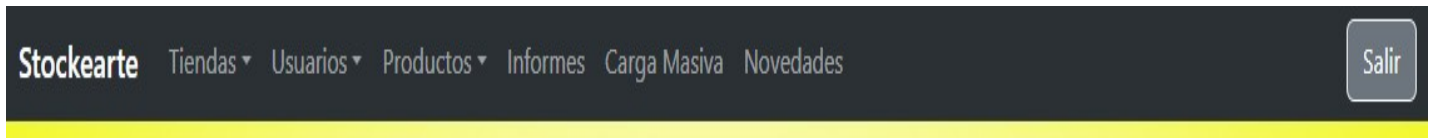
El servidor SOAP escucha por las peticiones del cliente SOAP en la siguiente dirección:
<http://localhost:9000/>

3.4. Interfaz gráfica

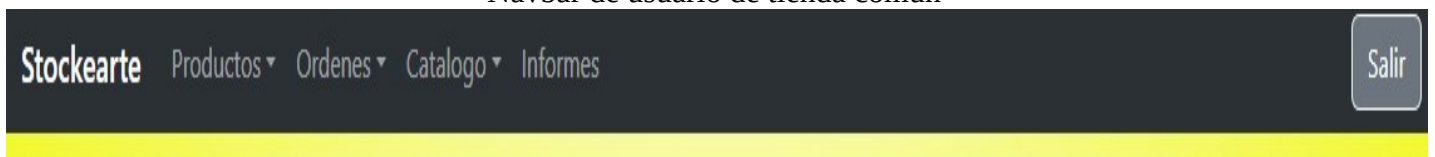
Se añadieron 3 vistas nuevas para tratar con los filtros personalizados de informe, los catálogos y la carga masiva de usuarios mediante archivos .csv.

Un usuario de tienda central tiene acceso a los informes de todas las tiendas y a la carga masiva de usuarios mientras que un usuario de tienda común solo tiene acceso a los informes de su respectiva tienda y a los catálogos.

Navbar de usuario de tienda central



Navbar de usuario de tienda común



Vista Informes


Vista Informes. Interfaz de usuario para generar informes. Incluye campos para Nombre, Código, Estado (dropdown), Desde (dd/mm/aaaa), y Hasta (dd/mm/aaaa). Hay botones para 'Informes 2...', 'RECIBIDA', y un botón de acción con un icono de documento y un icono de basura. El fondo es amarillo y hay un recuadro azul grande con el texto 'Informe SIN INFORME'.

Vista Catálogos

Prueba


ModificarEliminar^

Nombre: Chaqueta de CueroCodigo: P004



Nombre: Chaqueta de Cuero	Talle: S
Codigo: P004	Color: Marrón

Nombre: Vestido de VeranoCodigo: P005



Nombre: Vestido de Verano	Talle: M
Codigo: P005	Color: Rojo

Prueba_2

ModificarEliminar^

Vista Carga Masiva de Usuarios

Carga Masiva de Usuarios

Elegir archivo

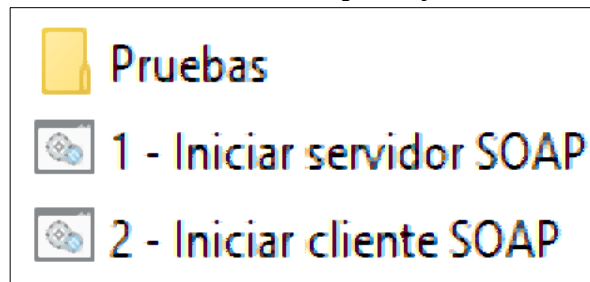
No se ha seleccionado ningún archivo

Cargar Archivo

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:

HACER NUEVO VIDEO

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:

https://drive.google.com/file/d/19NtRev9TkGkA0-pxg_cSAnOiwZJ5Ql-x/view?usp=sharing

6. Diagrama de comunicación

A continuación se muestra un diagrama que resume como se realiza la comunicación entre todas las partes del sistema. Este diagrama muestra los componentes de todos los trabajos prácticos, no solo las partes de este último trabajo práctico.