**Ficha Técnica del Servicio web**

**REVISIONES**

| **VERSIÓN** | **FECHA** | **NOMBRE** | **DESCRIPCIÓN** |
| --- | --- | --- | --- |
| 1.0 | 19/11/2022 | Zulma Patricia Salas Bautista | Versión inicial |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# INTRODUCCIÓN

El objetivo del presente documento es especificar, documentar y gestionar el perfil de un servicio web para Tienda Compulago.

Este perfil incluye todos los datos de identificación que habiliten su descubrimiento, así como la definición detallada de la estructura del contrato del servicio.

## Ficha Técnica de Servicio web

|  |  |
| --- | --- |
| **Servicio** | |
| Nombre | Customer-service |
| Descripción | Servicio que permite ingresar, buscar, actualizar y borrar u cliente |
| Versión | 1.0 |
| Estado | Desarrollo |
| URL del Servicio | https://servicio.terpel.com/path/recurso |
| Procesos y Actividades de Negocio relacionadas | createCustomer  getCustomer  updateCustomer  deleteCustomer |
| Fuente de datos del servicio | Nombre |
| Base de datos, aplicación que dispone el servicio. |
| Consumidores del servicio identificados | Nombre |
| Portal Web |
| Seguridad | Auntenticación |
| Nivel de Servicio (Disponibilidad) | Acuerdo de disponibilidad del servicio. Porcentaje de tiempo en línea. Ventanas de tiempo de disponibilidad. |
| Restricciones | Indica condiciones de operación dentro de las cuales el servicio puede ser consumido. Ej:  El servicio soporta máximo 50 usuarios concurrentes.  El servicio soporta máximo 1000 consultas por segundo.  El tiempo de respuesta del servicio en el límite superior de usuarios no puede superar 3 segundos.  La información de respuesta es actualizada cada 12 horas. |
| **Datos Técnicos** | |
| Tipo de Servicio | REST |
| Lenguaje | Java-Intellij, SpringBoot |
| **Operaciones** | |
| Creación cliente  Actualización datos cliente  Eliminar cliente | |

## Ficha de la Operación

Crear cliente

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operación** | | | | | | | | |
| Nombre | Crear cliente | | | | | | | |
| Precondiciones | El cliente no debe estar registrado | | | | | | | |
| Postcondiciones | Todos los datos se beben ingresar | | | | | | | |
| **Entradas** | | | | | | | | |
| **Nombre** | | **Tipo (numérico, Texto, etc)** | | | **Longitud** | | **Obligatorio (S/N)** | **Observación** |
| id | | long | | | - | | S | Generado automáticamente |
| name | | String | | | 25 | | S |  |
| surname | | String | | | 25 | | S |  |
| phone | | String | | | 10 | | S |  |
| direction | | String | | | 50 | | S |  |
| **Salidas** | | | | | | | | |
| **Nombre** | | **Tipo** | | | **Longitud** | | **Obligatorio** | **Observación** |
| id | | long | | | - | | S | Generado automáticamente |
| name | | String | | | 25 | | S |  |
| surname | | String | | | 25 | | S |  |
| phone | | String | | | 10 | | S |  |
| direction | | String | | | 50 | | S |  |
| **Transformaciones Asociadas** | | | | | | | | |
| **Campo Origen** | **Regla de la transformación** | | | | | | **Campo destino** | |
| id | Generado automáticamente | | | | | | id | |
| **Secuencia de Ejecución** | | | | | | | | |
|  | | | **Paso** | | | **Acción** | | |
| **Secuencia Normal** | | | 1 | | | Ingreso de datos | | |
| 2 | | | Verifica que todos los datos estén ingresados | | |
| 3 | | | Entrega del resultado | | |
| **Secuencia Alterna 1** | | | N/A | | | N/A | | |
| N/A | | | N/A | | |
| **Excepciones** | | | | | | | | |
| Datos incompletos | | | | No almacena cliente en base de datos | | | | |
|  | | | |  | | | | |

Actualizar cliente

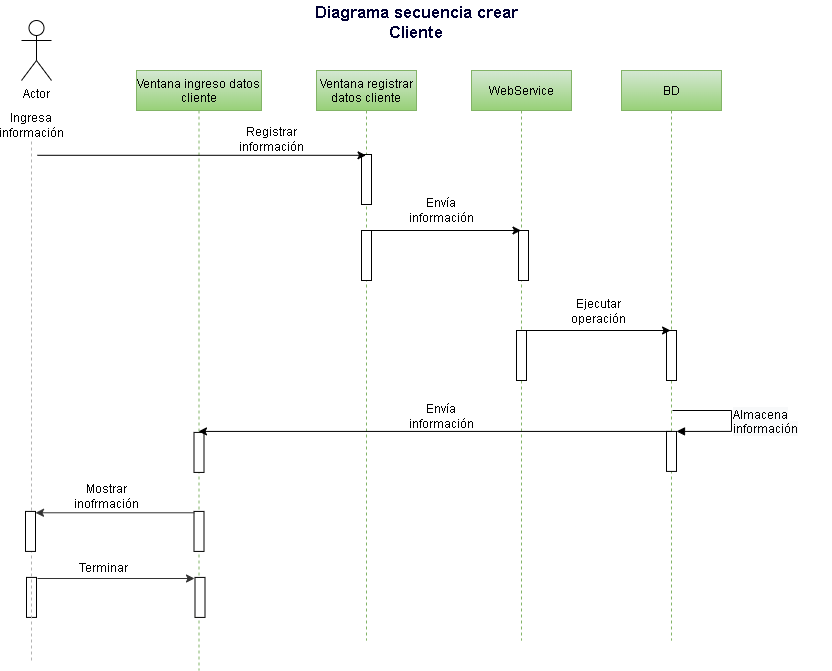
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operación** | | | | | | | | |
| Nombre | Actualiza cliente | | | | | | | |
| Precondiciones | El cliente debe estar registrado | | | | | | | |
| Postcondiciones | Todos los datos se beben ingresar | | | | | | | |
| **Entradas** | | | | | | | | |
| **Nombre** | | **Tipo (numérico, Texto, etc)** | | | **Longitud** | | **Obligatorio (S/N)** | **Observación** |
| id | | long | | | - | | S | Generado automáticamente |
| name | | String | | | 25 | | S |  |
| surname | | String | | | 25 | | S |  |
| phone | | String | | | 10 | | S |  |
| direction | | String | | | 50 | | S |  |
| **Salidas** | | | | | | | | |
| **Nombre** | | **Tipo** | | | **Longitud** | | **Obligatorio** | **Observación** |
| id | | long | | | - | | S | Generado automáticamente |
| name | | String | | | 25 | | S |  |
| surname | | String | | | 25 | | S |  |
| phone | | String | | | 10 | | S |  |
| direction | | String | | | 50 | | S |  |
| **Transformaciones Asociadas** | | | | | | | | |
| **Campo Origen** | **Regla de la transformación** | | | | | | **Campo destino** | |
| id | Generado automáticamente | | | | | | id | |
| **Secuencia de Ejecución** | | | | | | | | |
|  | | | **Paso** | | | **Acción** | | |
| **Secuencia Normal** | | | 1 | | | Ingresa número de documento | | |
| 2 | | | Se busca número de documento | | |
| 3 | | | Se cargan datos del cliente | | |
| 4 | | | Se actualizan datos del cliente | | |
| 5 | | | Mensaje datos cliente actualizados | | |
| **Secuencia Alterna 1** | | | N/A | | | N/A | | |
| N/A | | | N/A | | |
| **Excepciones** | | | | | | | | |
| Datos incompletos | | | | No actualiza datos del cliente en base de datos | | | | |

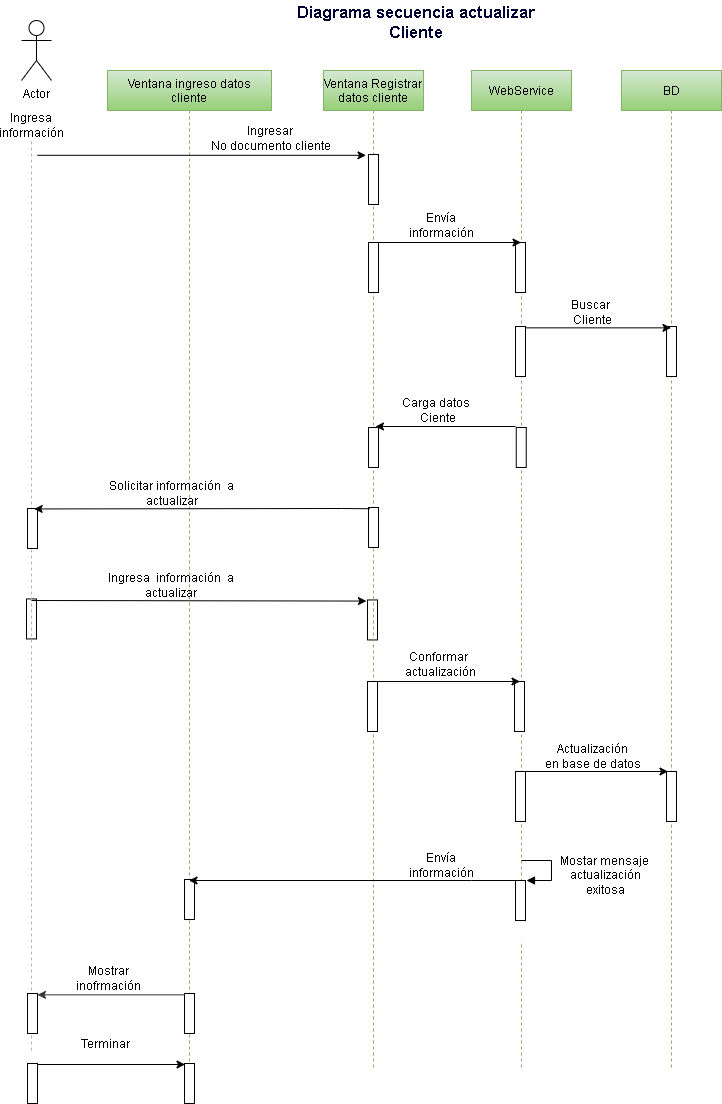
Eliminar cliente

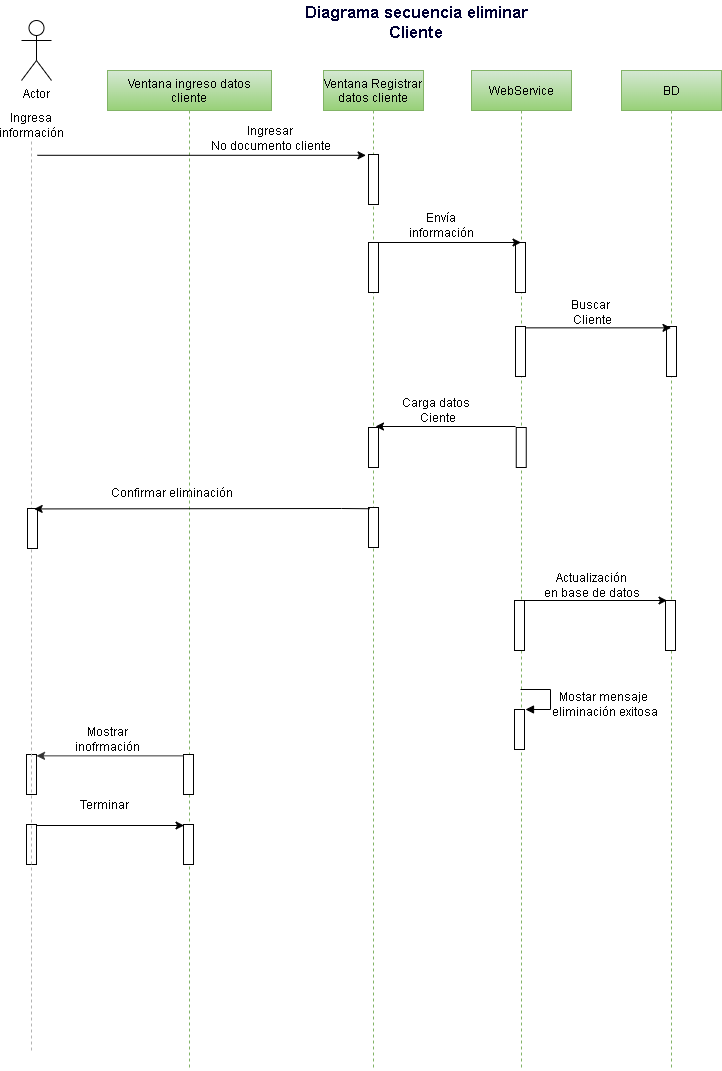
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Operación** | | | | | | | | |
| Nombre | Eliminar cliente | | | | | | | |
| Precondiciones | El cliente debe estar registrado | | | | | | | |
| Postcondiciones | N/A | | | | | | | |
| **Entradas** | | | | | | | | |
| **Nombre** | | **Tipo (numérico, Texto, etc)** | | | **Longitud** | | **Obligatorio (S/N)** | **Observación** |
| id | | long | | | - | | S | Generado automáticamente |
| name | | String | | | 25 | | S |  |
| surname | | String | | | 25 | | S |  |
| phone | | String | | | 10 | | S |  |
| direction | | String | | | 50 | | S |  |
| **Salidas** | | | | | | | | |
| **Nombre** | | **Tipo** | | | **Longitud** | | **Obligatorio** | **Observación** |
| id | | long | | | - | | S | Generado automáticamente |
| name | | String | | | 25 | | S |  |
| surname | | String | | | 25 | | S |  |
| phone | | String | | | 10 | | S |  |
| direction | | String | | | 50 | | S |  |
| **Transformaciones Asociadas** | | | | | | | | |
| **Campo Origen** | **Regla de la transformación** | | | | | | **Campo destino** | |
| id | Generado automáticamente | | | | | | id | |
| **Secuencia de Ejecución** | | | | | | | | |
|  | | | **Paso** | | | **Acción** | | |
| **Secuencia Normal** | | | 1 | | | Ingresa número de documento | | |
| 2 | | | Se busca número de documento | | |
| 3 | | | Se cargan datos del cliente | | |
| 4 | | | Se solicita confirmación eliminación cliente | | |
| 5 | | | Mensaje datos cliente eliminado | | |
| **Secuencia Alterna 1** | | | N/A | | | N/A | | |
| N/A | | | N/A | | |
| **Excepciones** | | | | | | | | |
| Datos incompletos | | | | No actualiza datos del cliente en base de datos | | | | |

1. **Diagramas del Diseño**

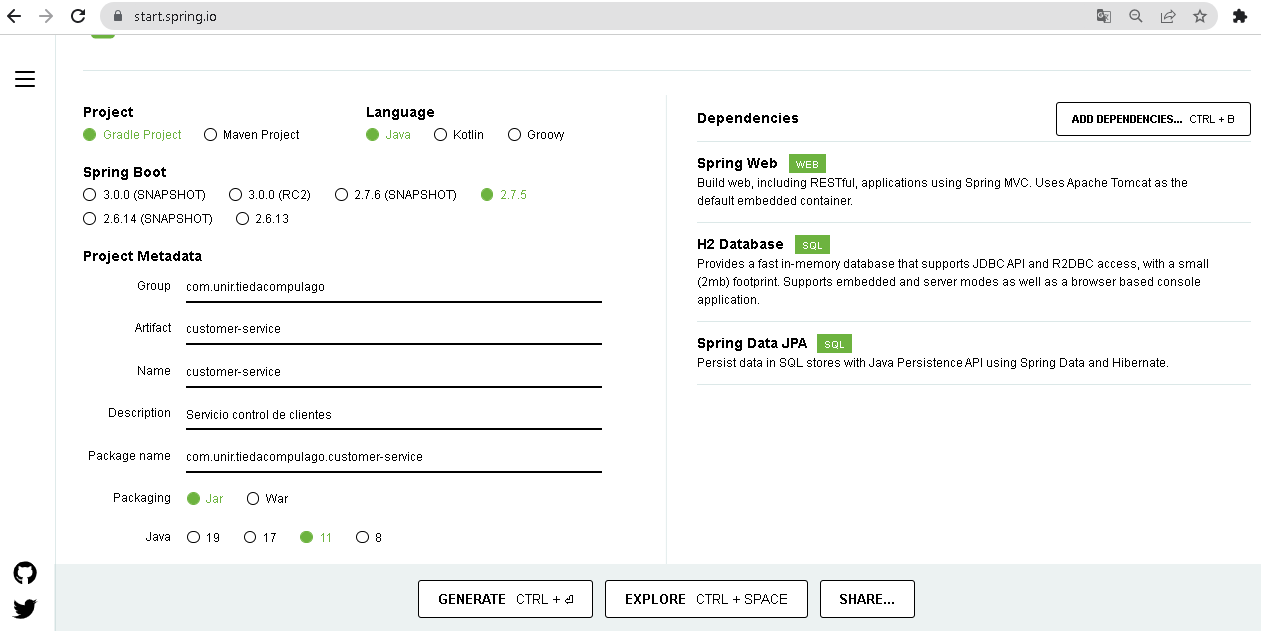
**Diagrama de Secuencia.**

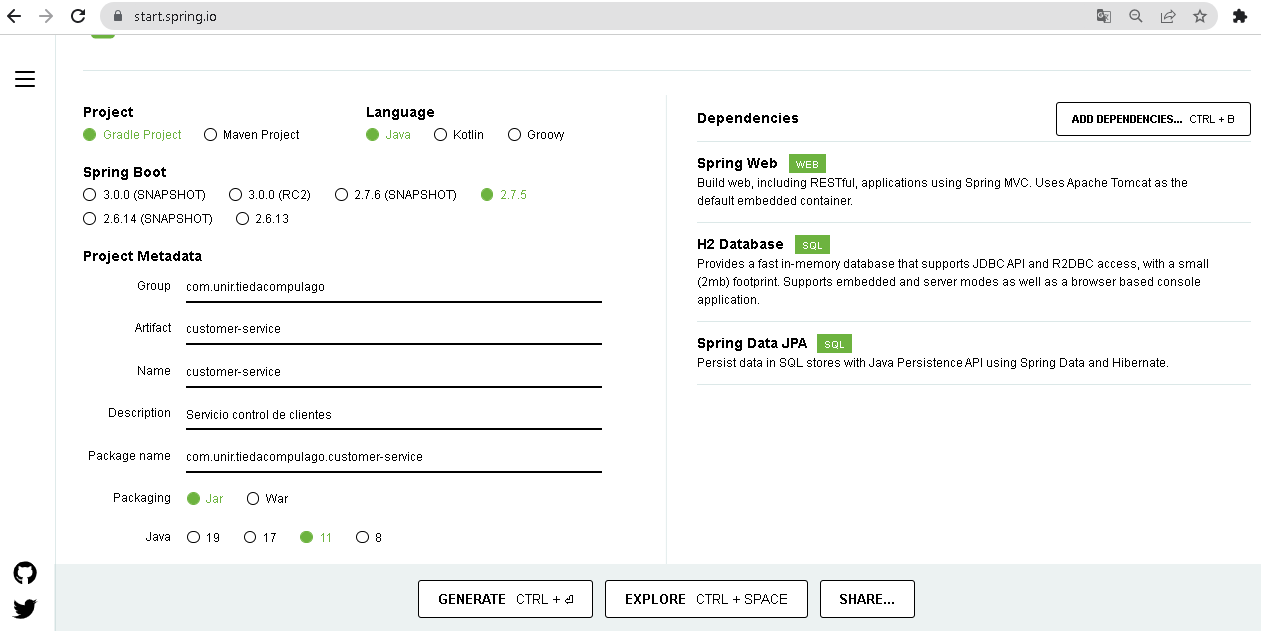


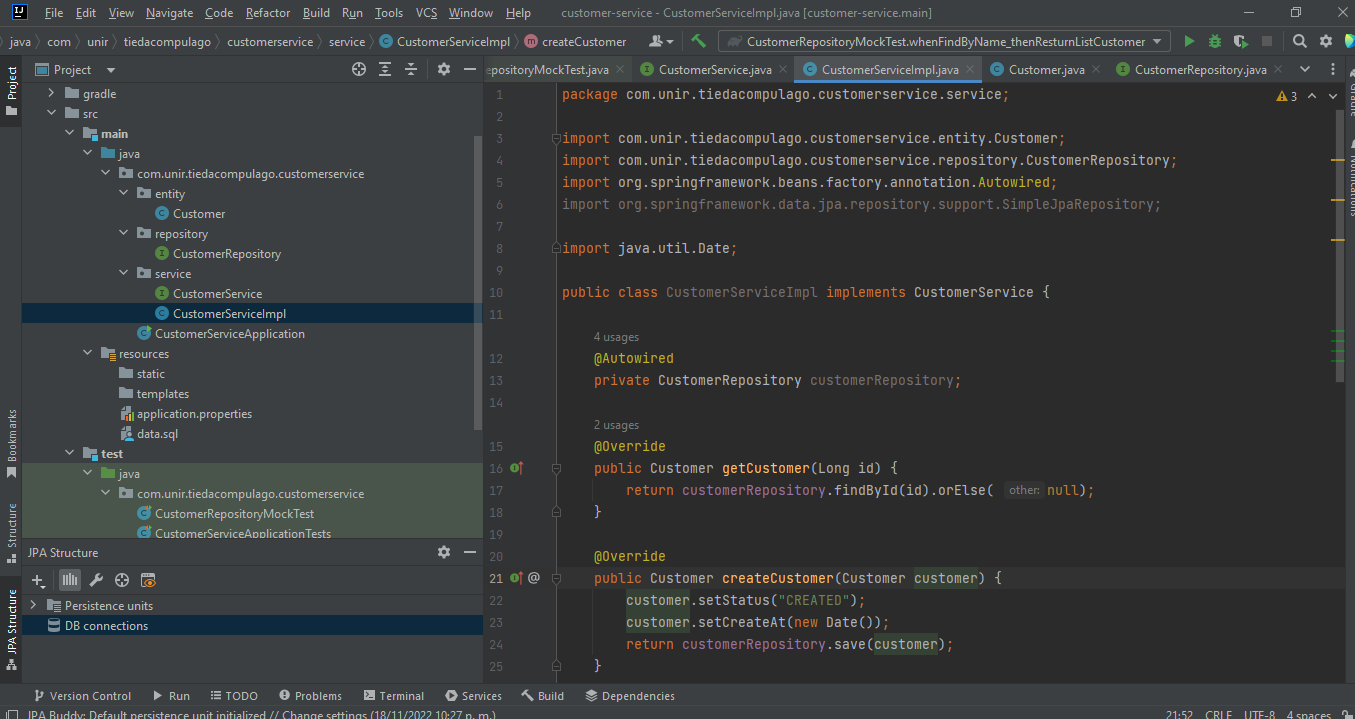




1. **Ejemplo Consumo del Servicio**

****

****

****

|  |
| --- |
| Creación entidad Customer |
| package com.unir.tiedacompulago.customerservice.entity;  import lombok.\*;  import org.springframework.boot.context.properties.bind.Name;  import javax.persistence.\*;  import java.util.Date;  @Entity  @Table(name="tbl\_customer")  @Data  @AllArgsConstructor @NoArgsConstructor @Builder  public class Customer {  @Id  @GeneratedValue (strategy = GenerationType.IDENTITY)  private long id;  private String name;  private String surname;  private String phone;  private String direction;  public void setStatus(String created) {  }  public void setCreateAt(Date date) {  }  //public Name getCustomer() { }  } |

|  |
| --- |
| Servicio: getCustomer, createCustomer, updateCustomer, deleteCustomer |
| package com.unir.tiedacompulago.customerservice.service;  import com.unir.tiedacompulago.customerservice.entity.Customer;  import com.unir.tiedacompulago.customerservice.repository.CustomerRepository;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.data.jpa.repository.support.SimpleJpaRepository;  import java.util.Date;  public class CustomerServiceImpl implements CustomerService {  @Autowired  private CustomerRepository customerRepository;  @Override  public Customer getCustomer(Long id) {  return customerRepository.findById(id).orElse( null);  }  @Override  public Customer createCustomer(Customer customer) {  customer.setStatus("CREATED");  customer.setCreateAt(new Date());  return customerRepository.save(customer);  }  @Override  public Customer updateCustomer(Customer customer) {  Customer customerDB = getCustomer(customer.getId());  if (null == customerDB ){  return null;  }  customerDB.setName(customer.getName());  customerDB.setSurname(customer.getSurname());  customerDB.setPhone(customer.getPhone());  customerDB.setDirection(customer.getDirection());  return customerRepository.save(customerDB);  }  @Override  public Customer deleteCustomer(Long id) {  Customer customerDB = getCustomer(id);  if (null == customerDB ){  return null;  }  customerDB.setStatus("DELETE");  return customerRepository.save(customerDB);  }  } |