**PRUEBA PRÁCTICA DE IMPLEMENTACION**

**DE CAPA DE DATOS Y SERVICIOS CON C# / SQL Server**

RESULTADO:

|  |  |
| --- | --- |
| NOMBRE: |  |
| FECHA: |  |
| TECNOLOGIAS: |  |

**Por favor lea detenidamente el contexto de la prueba, es importante que se familiarice con el objetivo que se quiere alcanzar para que pueda salir adelante en la misma.**

i-Route le ha pedido a sus colaboradores que realicen un software interno para el control de asistencia a los cursos de capacitación que se dan en la empresa. Para esto se le ha asignado a usted el módulo de Clientes.

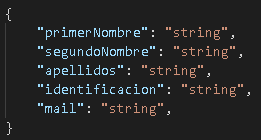
Para esto se le solicita a usted implementar los servicios **RESTful** que permitan crear, consultar y anular clientes. Así también, realizar los scripts de base de datos para poder crear la base, la estructura de clientes y los SPs requeridos de acuerdo a las operaciones descritas a continuación:

**Estructura Requerida**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Clientes** | | | | | | |
| **ID** | **Primer Nombre** | **Segundo Nombre** | **Apellidos** | **Identificacion** | **Correo** | **Estado** |
| 1 | Juan | Esteban | Perez | 1234567890 | juan.perez@iroute.com.ec | 1 |
| 2 | Pedro | Alberto | Perez | 2234567890 | pedro.perez@iroute.com.ec | 0 |
| 3 | Perico |  | De los Palotes | 3334567890 | perico.delospalotes@iroute.com.ec | 1 |

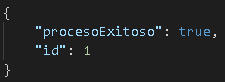
**Operaciones Requeridas**

**Operación de Crear Cliente:** Una de las operaciones que se solicita implementar es la que permita crear clientes para esto el equipo de arquitectura ha definido que los campos de entrada que se deben solicitar son los siguientes:

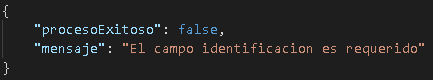


De los campos indicados todos son requeridos a excepción del “SegundoNombre” que es opcional. Adicional se debe considerar que al momento de crear un cliente este se crea con Estado activo(1).

En caso de ser un proceso exitoso la operación devolverá la siguiente estructura de respuesta:



En caso de ser un proceso fallido la operación devolverá la siguiente estructura de respuesta:



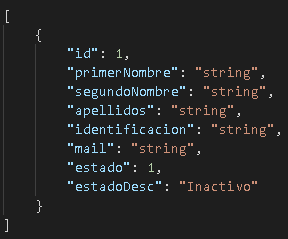
|  |  |
| --- | --- |
| **CASOS DE ERROR** | |
| **Caso** | **Mensaje Devuelto** |
| Parámetros Requeridos | El campo [campo] es requerido |
| Cliente ya existe | Ya existe un cliente con dicha Identificación |

**Operación de Consultar Clientes:** Una de las operaciones que se solicita implementar es la que permita consultar clientes para esto el equipo de arquitectura ha definido que los campos de entrada que se deben solicitar son los siguientes:

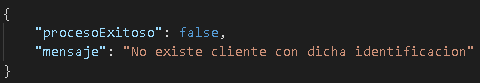
* Identificación (opcional)

En caso de no enviar el campo identificación deberá devolver todos los clientes que existen en la estructura de la base de datos.

En caso de ser un proceso exitoso la operación devolverá la siguiente estructura de respuesta:



En caso de ser un proceso fallido la operación devolverá la siguiente estructura de respuesta:



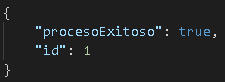
|  |  |
| --- | --- |
| **CASOS DE ERROR** | |
| **Caso** | **Mensaje Devuelto** |
| Cliente no existe | La identificación del cliente no es valida |

**Operación de Anular Cliente:** Una de las operaciones que se solicita implementar es la que permita anular clientes para esto el equipo de arquitectura ha definido que los campos de entrada que se deben solicitar son los siguientes:

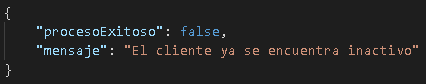


Se debe considerar que el proceso de anular hará una eliminación lógica cambiando el Estado del cliente a Inactivo(0).

En caso de ser un proceso exitoso la operación devolverá la siguiente estructura de respuesta:



En caso de ser un proceso fallido la operación devolverá la siguiente estructura de respuesta:



|  |  |
| --- | --- |
| **CASOS DE ERROR** | |
| **Caso** | **Mensaje Devuelto** |
| Parámetros Requeridos | El campo ID es requerido |
| Cliente no existe | El ID del cliente no es valido |

**Problemas a Resolver:**

El colaborador de i-Route que está desarrollando el programa se ha encontrado con algunos inconvenientes para terminar la aplicación y su misión es lograr a resolver estos inconvenientes:

1. Se implementa y ejecuta correctamente el proyecto **(0.5 Puntos).**
2. Se implementa correctamente la operación de consultar cumpliendo con el formato de entrada y salida establecido **(1 Puntos).**
3. Se implementa correctamente la operación de guardar cumpliendo con el formato de entrada y salida establecido **(1.5 Puntos).**
4. Se implementa correctamente la operación de anular cumpliendo con el formato de entrada y salida establecido **(1 Puntos).**
5. El programa debe cumplir con las validaciones solicitadas **(1.5 Puntos).**
6. Por estándares de desarrollo los atributos de las entidades deben iniciar con la primera letra en mayúscula mientras que los atributos de salida de los servicios RESTful deben empezar con letra minúscula. **(0.5 Puntos).**
7. Se crea correctamente la estructura de la base de datos y SPs requeridos **(1 Puntos).**
8. Se crea correctamente la tabla clientes de manera que el campo ID sea un secuencial que se incremente en 1 **(0.5 Puntos).**
9. Los datos en la base se deben registrar según lo enviado en los servicios web.

**(1 Puntos).**

1. La consulta de clientes debe devolver la información tal cual esta almacenada en la base de datos

**(0.5 Puntos).**

1. Se deben describir los pasos y presentar evidencias (Print de Pantallas) de cómo ejecutar los servicios programados para poder certificar los numerales requeridos **(1 Puntos).**

**Endpoint “Crear” / Post:**

Ejecucion:

Se ejecuta el servicio enviando por Request un objeto json de la siguiente forma:

{

"primerNombre": "Wellington",

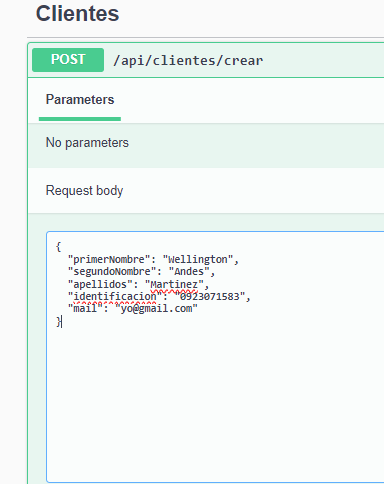
"segundoNombre": "Andes",

"apellidos": "Martinez",

"identificacion": "0923071583",

"mail": "yo@gmail.com"

}



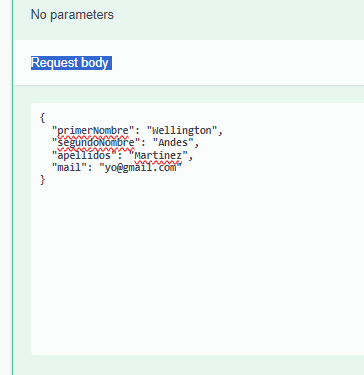
Respuesta correcta:

Una vez enviada la solicitud, se recibe la siguiente respuesta:



Respuesta errónea:

Para la siguiente solicitud errónea:



Una vez enviada la solicitud, de tener un error, este es el siguiente problema:



**Enpoint “Consultar” / Get:**

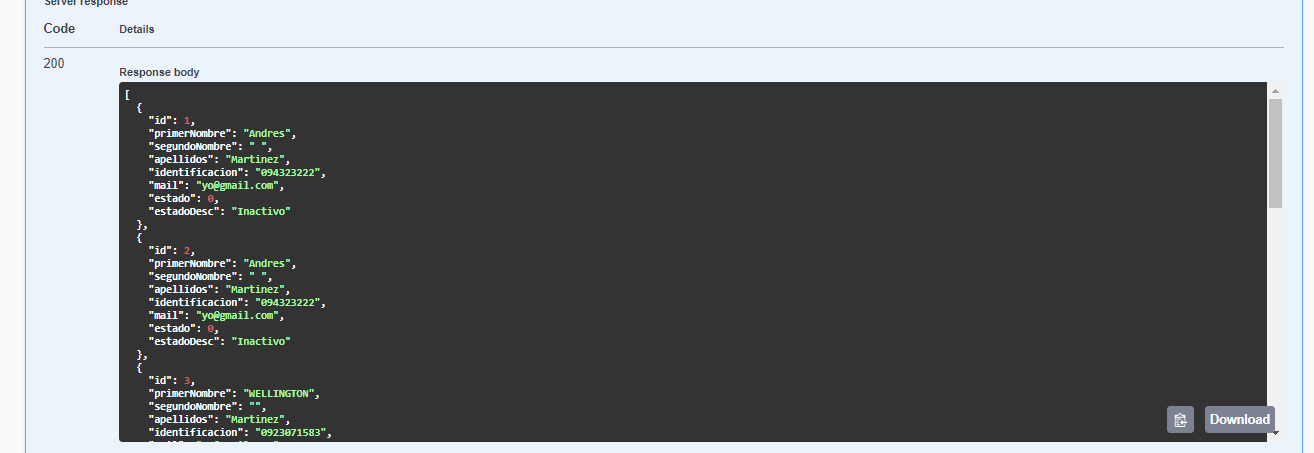
Consulta de todos los datos:

Se ejecuta el servicio sin parámetro identificación



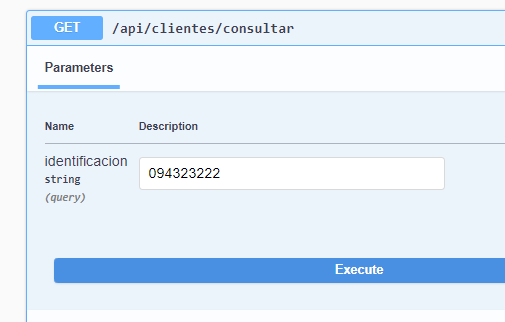
Respuesta de la consulta de todos los datos:

Se recibe un json con todos los clientes existentes:

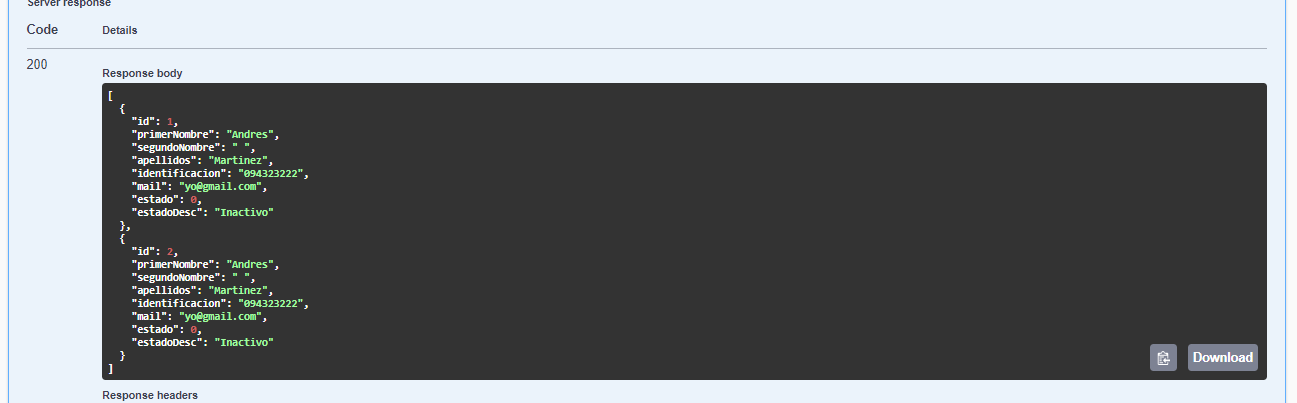


Consulta exitosa de cliente con filtro por identificacion:

Se ejecuta el servicio con una identificación existente:

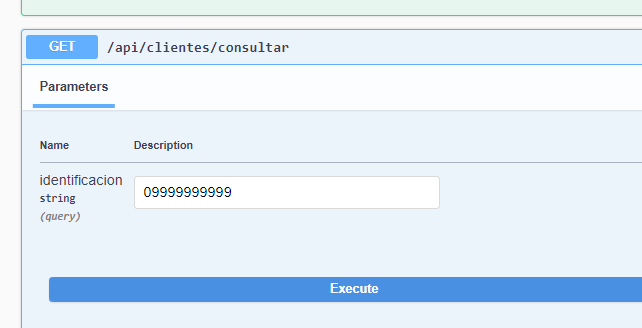


El servicio responde con un json con el cliente que tiene dicha identificación

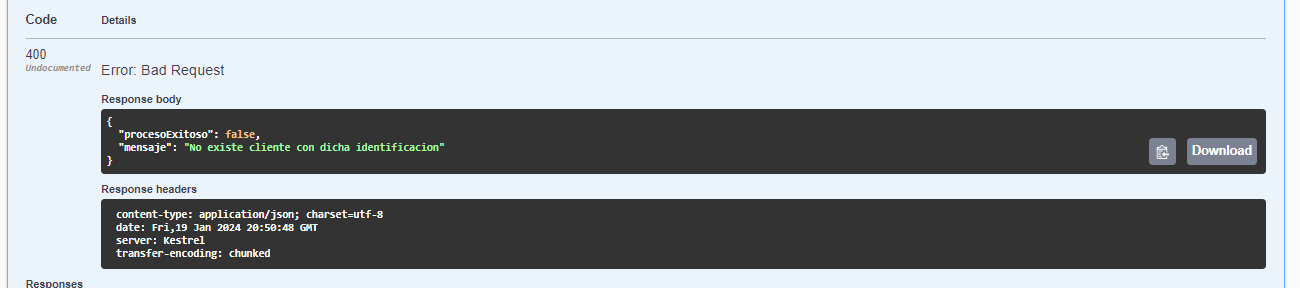


Consulta errónea cliente de con filtro por identificacion:

Se ejecuta el servicio con una identificación inexistente:



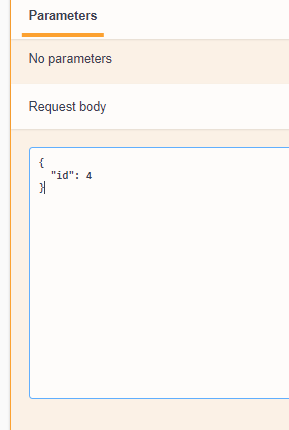
El servicio responde con un json con un mensaje de error:



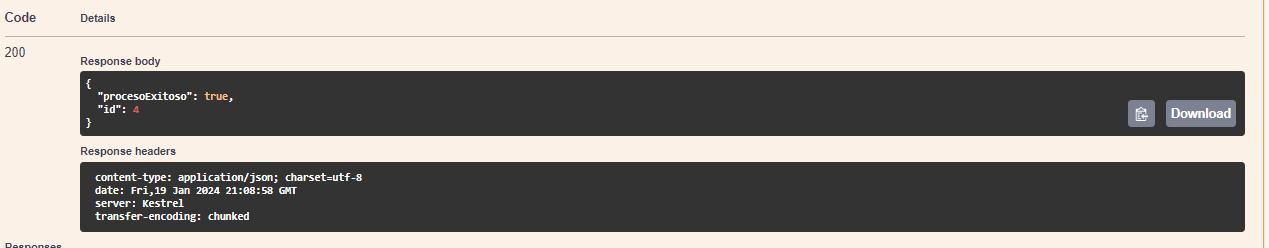
**Endpoint “Anular” / Put:**

Ejecucion de una anulación exitosa:

Se ejecuta el servicio enviando por Request un objeto json de la siguiente forma:

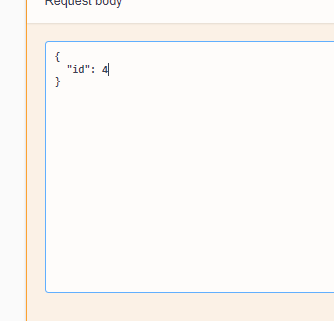


El servicio retorna una respuesta con la siguiente estructura:

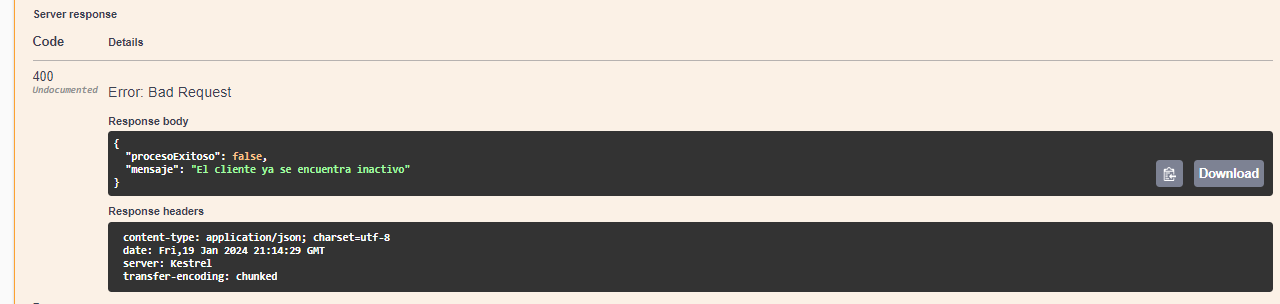


Ejecucion de una anulación errada por cliente ya anulado:

Se ejecuta el servicio enviando por Request un objeto json que posea en su estructura el id del cliente ya eliminado:



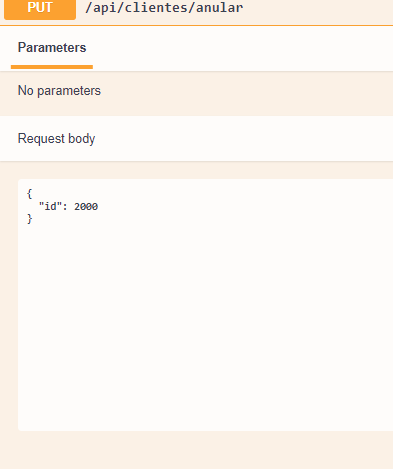
El servicio responde con el siguiente mensaje:



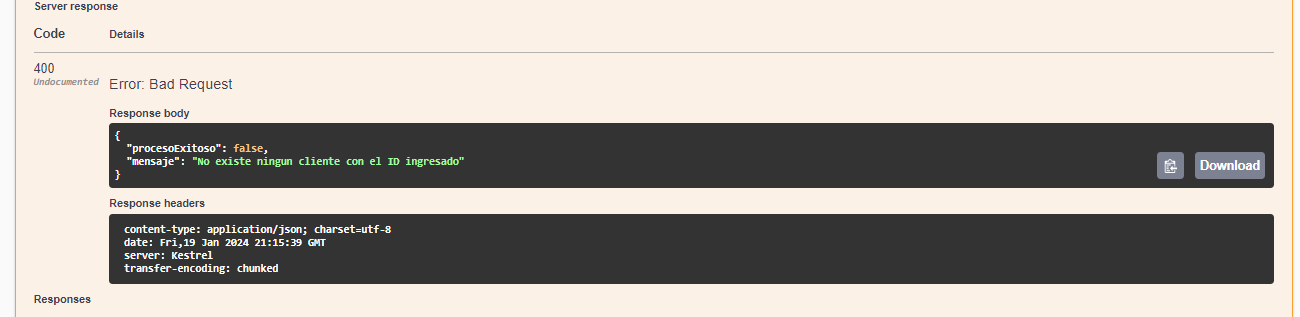
El cual indica que el cliente con dicho id ya se encuentra inactivo

Ejecucion de una anulación errada por cliente inexistente:

Se ejecuta el servicio enviando por Request un objeto json que posea en su estructura el id del cliente que no existe



El servicio responde con el siguiente mensaje:



El cual indica que el id no existe

Direccion del backup y script de la base de datos:

Se encuentra ubicados en PruebC#/Base de datos