

1.-NEGOCIO ELEGIDO Y FUNDAMENTACIÓN DEL PRODUCTO.

El negocio elegido ha sido una tienda de velas, inciensos, aceite esenciales, jabones naturales y varios productos más relacionados con la artesanía. La dueña del negocio ha decidido dar el paso a la digitalización y es por eso que necesita de una API para manejar los flujos de información de su actividad comercial.

2.-DISEÑO TABLAS.

Se han establecido tres tablas. La tabla productos, la tabla pedidos y la tabla clientes. De esta manera se va a organizar óptimamente la información. Se podrán ver las relaciones que tienen los pedidos con sus respectivos productos y el cliente que lo ha realizado. La persona que consuma esta API, puede consultar, agregar, modificar y eliminar productos de su catálogo, puede ver los pedidos realizados

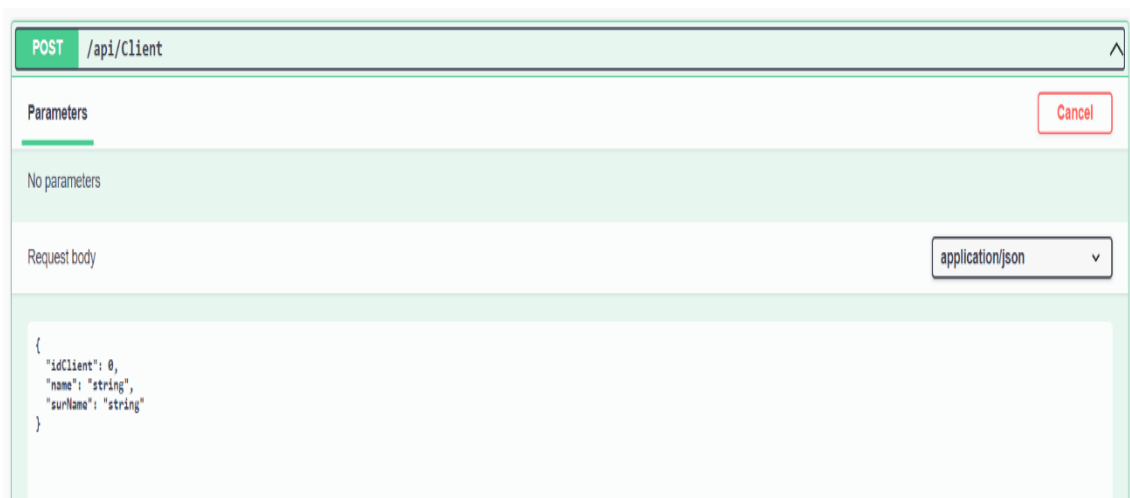
3.-TECNOLOGÍAS, LENGUAJES Y VERSIONES.

Para la creación de la API se ha utilizado como lenguaje C# con el Framework .NET CORE 6.0 y, en concreto, la librería Entity Framework.

4.-DOCUMENTACIÓN FUNCIONAL DE LA WEB API:

Reúne las solicitudes referidas a alta, baja, modificación y consulta de usuarios.

POST: inserta **cliente** nuevo.



The screenshot shows an API client interface with the following components:

- Method and URL:** POST /api/Client
- Parameters:** No parameters
- Request body:** application/json
- Request body content:**

```
{  "idClient": 0,  "name": "string",  "surName": "string"}
```

GET: obtiene los clientes introducidos.

Response body

```
[
  {
    "idClient": 2,
    "name": "Alda",
    "surName": "Mar"
  },
  {
    "idClient": 3,
    "name": "Alicia",
    "surName": "Mar"
  },
  {
    "idClient": 1003,
    "name": "Salem",
    "surName": "Mar"
  },
  {
    "idClient": 1004,
    "name": "string",
    "surName": "string"
  },
  {
    "idClient": 1005,
    "name": "Alicia",
    "surName": "Mar"
  },
]
```

Download

DELETE: Borra clientes por id:

DELETE /api/Client/{id}

Parameters

Name	Description
id * required	
integer(\$int32)	1004
(path)	

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:5015/api/Client/1004' \
-H 'accept: */*'
```

Request URL

PUT: Modifica clientes por id:

PUT /api/Client/{id}

Parameters

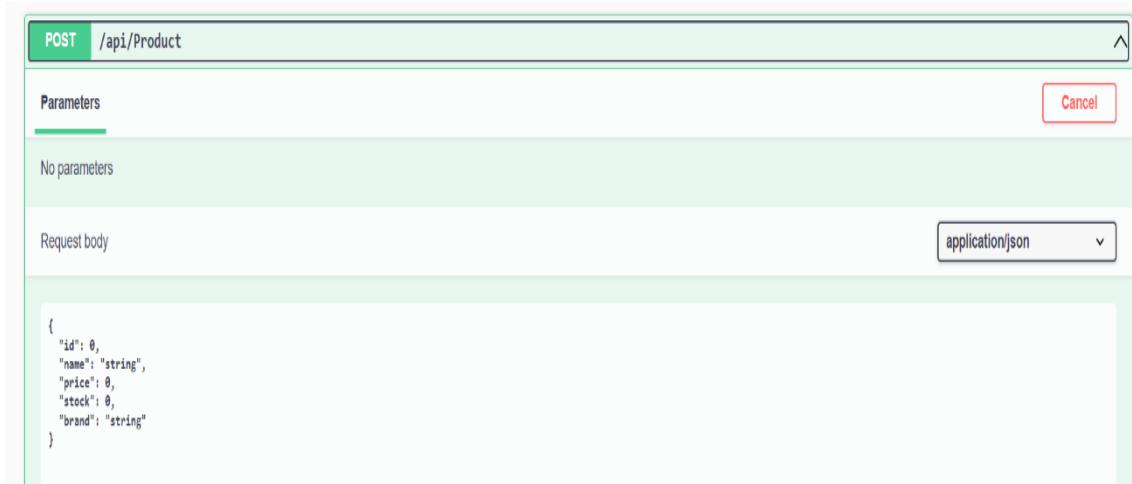
Name	Description
id * required	
integer(\$int32)	id
(path)	

Request body

application/json

```
{
  "idClient": 0,
  "name": "string",
  "surName": "string"
}
```

POST: inserta **productos** nuevos.

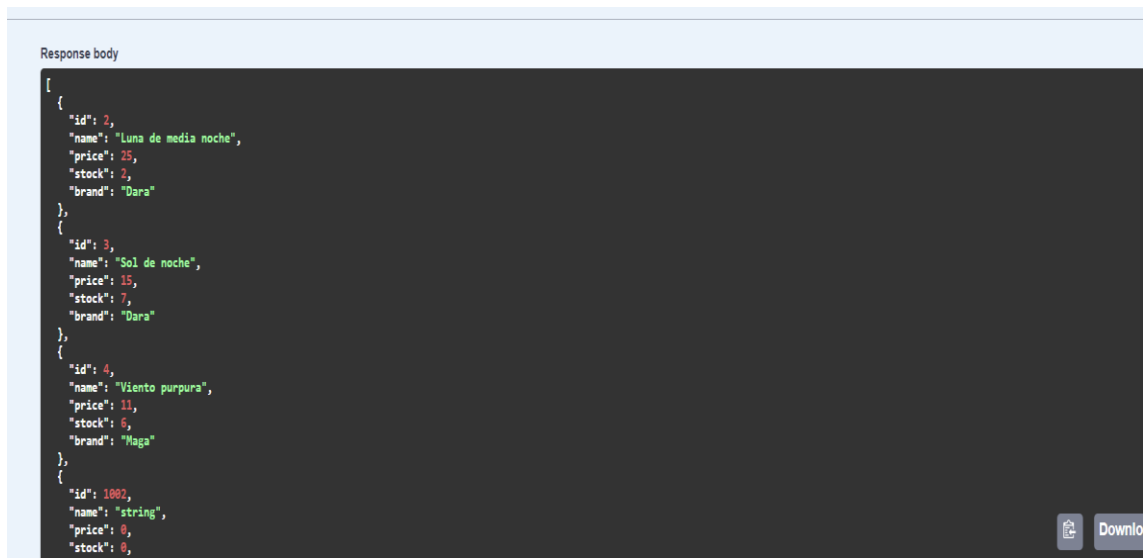


The screenshot shows a REST client interface for a POST request to the endpoint `/api/Product`. The request body is a JSON object with the following structure:

```
{  "id": 0,  "name": "string",  "price": 0,  "stock": 0,  "brand": "string"}
```

The interface includes a 'Parameters' section with 'No parameters' and a 'Request body' section with a dropdown menu set to 'application/json'. A 'Cancel' button is visible in the top right corner.

GET: Recupera los productos introducidos:



The screenshot shows the response body of a GET request, displaying a JSON array of product objects. The response is as follows:

```
[  {    "id": 2,    "name": "Luna de media noche",    "price": 25,    "stock": 2,    "brand": "Dara"  },  {    "id": 3,    "name": "Sol de noche",    "price": 15,    "stock": 7,    "brand": "Dara"  },  {    "id": 4,    "name": "Viento purpura",    "price": 11,    "stock": 6,    "brand": "Maga"  },  {    "id": 1002,    "name": "string",    "price": 0,    "stock": 0  }]
```

The interface includes a 'Response body' section with a dark background. A 'Download' button is visible in the bottom right corner.

DELETE: Borra productos por id:

DELETE
/api/Product/{id}

Parameters
Cancel

Name	Description
id required	1002
integer(\$int32)	
(path)	

Execute

Responses

Code	Description	Links
200	Success	No links

PUT: Modifica productos por id:

PUT
/api/Product/{id}

Parameters
Cancel

Name	Description
id required	id
integer(\$int32)	
(path)	

Request body

application/json

```

{
  "id": 0,
  "name": "string",
  "price": 0,
  "stock": 0,
  "brand": "string"
}

```

POST: Crea pedidos nuevos:

POST
/api/Order

Parameters
Cancel
Reset

No parameters

Request body

application/json

```

{
  "id": 0,
  "idClient": 0,
  "idProduct": 0,
  "amount": 0,
  "orderDate": "2023-02-02T18:53:25.793Z",
  "products": [
    {
      "id": 0,
      "name": "string",
      "price": 0,
      "stock": 0,
      "brand": "string"
    }
  ],
  "cliente": f
}

```

GET: recupera pedidos:

```
Response body

[
  {
    "id": 1005,
    "idClient": 2,
    "idProduct": 2,
    "amount": 2,
    "orderDate": "2023-02-01T20:20:40.248",
    "products": null,
    "clients": null
  },
  {
    "id": 1006,
    "idClient": 3,
    "idProduct": 3,
    "amount": 1,
    "orderDate": "2023-02-01T20:20:40.248",
    "products": null,
    "clients": null
  },
  {
    "id": 1007,
    "idClient": 2,
    "idProduct": 2,
    "amount": 1,
    "orderDate": "2023-02-01T20:20:40.248",
    "products": null,
    "clients": null
  }
]
```

DELETE: borra pedidos por id:

DELETE /api/Order/{id}

Parameters

Cancel

Name	Description
id * required	id
integer(\$int32)	
(path)	

Execute

Responses

Code	Description	Links
200	Success	No links

PUT: modifica pedidos por id.

PUT /api/Order/{id}

Parameters

Cancel

Reset

Name	Description
id * required	id
integer(\$int32)	
(path)	

Request body

application/json

```
{
  "id": 0,
  "idClient": 0,
  "idProduct": 0,
  "amount": 0,
  "orderDate": "2023-02-02T19:03:26.990Z",
  "products": [
    {
      "id": 0,
      "name": "string",
      "price": 0,
      "quantity": 0
    }
  ]
}
```

5) Extensiones posibles/pendientes:

*Los productos, a su vez, se agruparán por marcas. El cliente deberá poder consultar, agregar, modificar y eliminar marcas de productos.

*La persona regente del negocio puede administrar los clientes

*Incorporar medidas de seguridad adecuadas..