

LanaApi

1)Negocio elegido y fundamentación del producto.

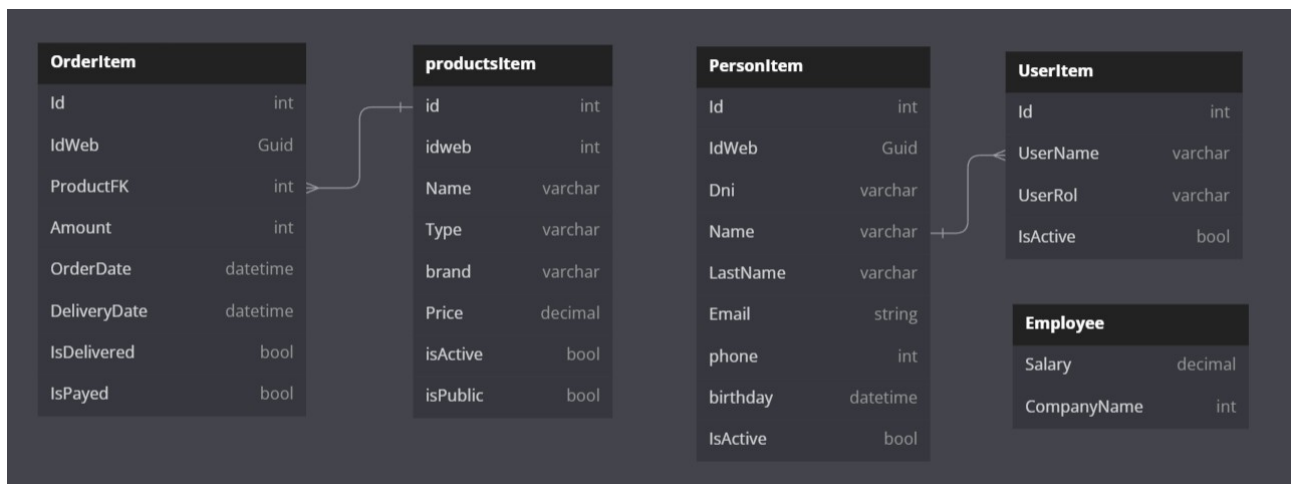
La api desarrollada en este proyecto esta pensada para ser utilizada para negocios especializados en la distribución de lanas y productos relacionados con la artesanía textil (punto, crochet, labores textiles, etc).

La necesidad de crear una api para gestionar este tipo de productos nace de la creciente cantidad de marcas que están llenando el mercado, así como la diversificación en el tipo de hilado y los materiales especializados que se desarrollan para estás labores.

Este producto, por tanto, se desarrolla para facilitar la labor de pequeños negocios dedicados al sector (principalmente mercerías).

En esta primera versión la api esta diseñada para gestionar una base de datos que almacene la información relativa a: Productos, Pedidos, Usuarios, así como una función para clasificar los diferentes usuarios.

2) Diseño tablas.



Como puede ver en la siguiente imagen el diseño de las tablas esta diseñado para organizar las diferentes partes del negocio.

3) Tecnologías/lenguajes/versiones.

Tecnologías: .netcore (Framework), Microsoft Visual Studio

lenguajes: C#

versiones: .netcore versión 6

4) Documentación funcional de la Web API:

Objetivo: desarrollar una api capaz de gestionar las tablas básicas para el desarrollo de un e-commerce.

Arquitectura: Entity Framework

Funcionalidades: Métodos Get, Post, Patch, Delete y un filtro get por marcas.

Endpoint /Order

Reúne las solicitudes referidas al alta y consulta de pedidos

GET: Solicitud de consulta de ordenes

No parameters

```
{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "productForeignKey": 0,
  "amount": 4,
  "orderDate": "2023-02-02T14:20:25.104Z",
  "deliveryDate": "2023-02-02T14:20:25.104Z",
  "isDelivered": true,
  "isPayed": true
}
```

POST: solicitud de nuevo item en la tabla de ordenes

-No parameters

-Request Body

```
{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "productForeignKey": 0,
  "amount": 0,
  "orderDate": "2023-02-02T16:23:28.394Z",
  "deliveryDate": "2023-02-02T16:23:28.394Z",
  "isDelivered": true,
  "isPayed": true
}
```

Endpoint/Person

POST

Solicitud de nuevo usuario en la tabla person

-No parameters

-Request Body

```
{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "dni": "string",
  "name": "string",
  "lastName": "string",
  "email": "string",
  "phone": "string",
  "birthday": "2023-02-02T16:33:22.361Z",
  "isActive": true
}
```

GET

Solicitud de consulta de personas

-No parameters

```
{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "dni": "string",
  "name": "string",
  "lastName": "string",
  "email": "string",
  "phone": "string",
  "birthday": "2023-02-02T16:33:22.361Z",
  "isActive": true
}
```

DELETE

Solicitud para eliminar usuarios de la base de datos

-Parameters: Name: Id Description: “ “

Endpoint/Products

POST

Solicitud de nuevo productos

-No parameters

-Request from body

```
{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "name": "string",
  "type": "string",
  "brand": "string",
  "price": 0,
  "isActive": true,
  "isPublic": true
}
```

GET

Solicitud de consulta de productos

-No parameters

```
{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "name": "string",
  "type": "string",
  "brand": "string",
  "price": 0,
  "isActive": true,
  "isPublic": true
}
```

DELETE

Solicitud para eliminar un producto de la base de datos

-Parameter Name Id, Descripción: “ ”

PATCH

Solicitud para editar/modificar una entrada de la base de datos

-No parameters

-Request from body

```
{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "name": "string",
  "type": "string",
  "brand": "string",
  "price": 0,
  "isActive": true,
  "isPublic": true
}
```

GET BY CRITERIA

Solicitud de consulta con filtro definido por la marca de los productos

-Parameters: Name: "Brand", Description: "Introducir marca"

-Request from query

Endpoint/User

POST

Solicitud para introducir un nuevo usuario

-Parameters

-Request from body

```
{
  "id": 0,
  "userName": "string",
  "userRol": "string"
}
```

GET

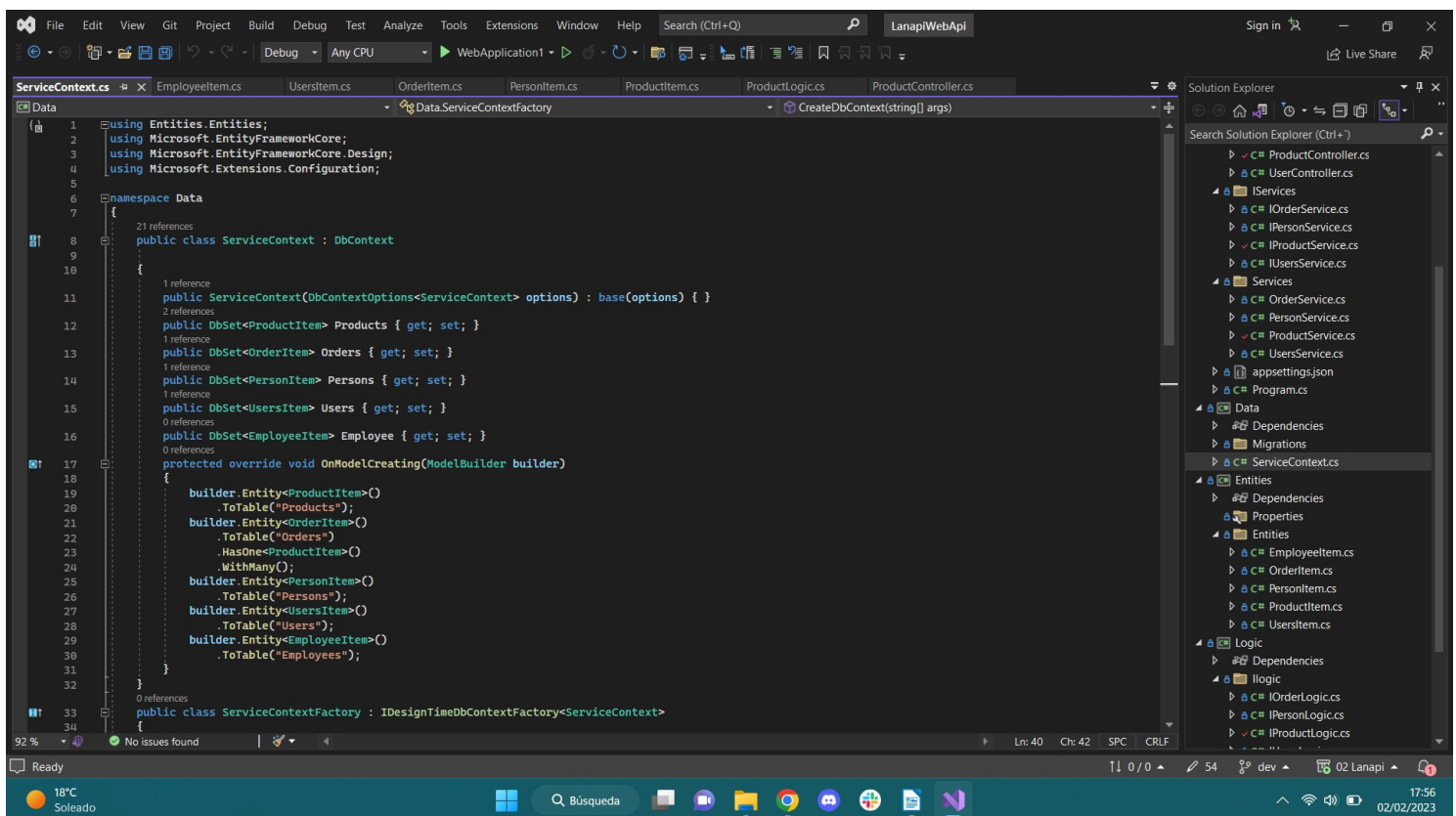
Solicitud de consulta de los usuarios

-No parameters

5) Extensiones posibles/pendientes

1. Primal & Foreign Key: Desarrollar las conexiones entre las tablas.
2. Validaciones de usuarios: Establecer diferentes usuarios con mayor y menor grado de poder sobre la api y sus debidas validaciones.
3. Introducir todos los métodos en las rutas: Actualmente el proyecto solo tiene todas los métodos en una de las rutas por lo que en un futuro me gustaría establecer los métodos en todas las rutas.
4. Desarrollo de tablas complementarias: Desarrollar por completo tablas complementarias como la de "empleados" y desarrollar y conectar otras tablas complementarias para perfeccionar el funcionamiento de la api.

6) Ejemplo del código trabajado



Una de las partes más satisfactorias para mi fue crear la arquitectura siguiendo los modelos de ejemplo desarrollados por Marcos para trabajar en mi proyecto ya que sentí que, a pesar de no tener claras todas las

partes del código, fui capaz de desarrollar toda esa parte del proyecto con autonomía y sin grandes complicaciones.