

API PAPELERÍA

Carmen Trillo Navarro

Contenido

1. NEGOCIO ELEGIDO Y FUNDAMENTACIÓN	3
2. DISEÑO DE TABLAS	3
3. TECNOLOGÍAS, LENGUAJES Y VERSIONES.....	4
4. FUNCIONALIDAD DE LA WEB API.....	4
OBJETIVOS	4
ARQUITECTURA	5
CONTROLLERS Y MÉTODOS	5
Controller Rol	6
Controller Tipo Cliente	7
Controller Persona	7
Controller Usuario	9
Controller Cliente	10
Controller Producto.....	11
Controller Pedido	12
5. EXTENSIONES POSIBLES	14

1. NEGOCIO ELEGIDO Y FUNDAMENTACIÓN

Papelería Martín es una pequeña papelería de barrio que a lo largo de los años ha ido evolucionando y creciendo tanto en su cifra de ventas, como en el número de clientes a los que debe atender. La innovación continua es la base de su crecimiento, aportando continuamente nuevas soluciones a sus clientes y productos de calidad que se adaptan a las últimas tendencias tecnológicas.

Este gran crecimiento ha traído, a su vez, una serie de dificultades en la gestión del negocio. Algunas de éstas son:

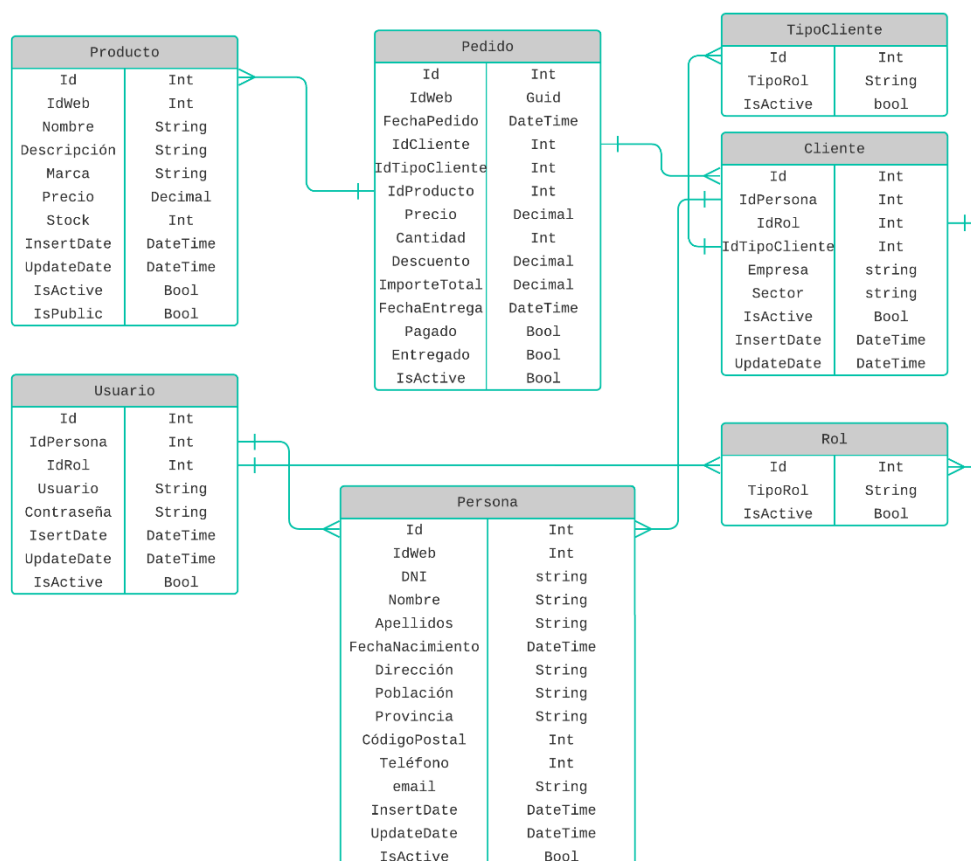
- Grandes colas por parte de los clientes cuando acuden a realizar compras al establecimiento
- Tiempos de espera amplios en la atención a los clientes
- Establecimiento pequeño que dificulta la gran asistencia de clientes al negocio
- Poco personal para atender a los clientes

Para tratar de solventar estas dificultades, el propietario del negocio ha decidido crear una web, para gestionar los encargos y los pedidos online, para lo que será necesario el desarrollo de una API que lo integre.

2. DISEÑO DE TABLAS

Para desarrollar la API y la Base de Datos, partimos del siguiente diseño de tablas, que presenta una serie de tablas y relaciones entre ellas.

API-Papelería
Carmen Trillo Navarro



La Base de Datos presenta un diseño de siete tablas:

1. Rol: roles de acceso o uso de la API o base de datos.
2. Persona: datos personales de usuarios y clientes.
3. Usuario: personas con acceso mediante contraseña a la aplicabilidad de la API o base de datos.
4. Cliente: clientes que realizan compras en el negocio.
5. Tipo de Cliente: según el tipo de cliente, se aplicarán distintos precios en los pedidos.
6. Producto: inventario de los productos de la papelería y stock existente.
7. Pedido: relación de pedidos realizados y del estado de éstos.

Tal como podemos observar en el diseño anterior, Usuario y Cliente están relacionadas con la tabla Personas a través de Id Personas. Además, ambas tablas presentan una relación con Rol, a través de Id Rol, que nos indicará qué rol tiene ese usuario o cliente en el acceso al programa.

Clientes está relacionado con Tipo de Clientes por el Id Tipo de Clientes, lo que nos indicará de qué tipo de cliente se trata para poder calcular el precio en función del cliente.

La tabla de Pedidos está relacionada con la tabla de Productos a través de Id Productos, así como con la tabla de Clientes por el Id Clientes.

3. TECNOLOGÍAS, LENGUAJES Y VERSIONES

En el desarrollo de la WEB API se ha hecho uso de las siguientes tecnologías:

- Microsoft Visual Studio 2022
- Microsoft SQL Server 18
- Git
- Tablero Trello
- Lucidchart Diagramas

Los lenguajes utilizados durante el proyecto son:

- .Net Core v6.0
- Entity Framework v7.0
- C#
- SQL

4. FUNCIONALIDAD DE LA WEB API

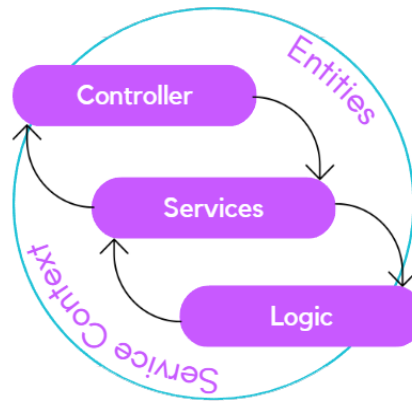
OBJETIVOS

El desarrollo de la WEB API busca alcanzar tres grandes objetivos principalmente:

1. Objetivo 1: Gestión del **inventario de productos** de la papelería.
2. Objetivo 2: Gestión **base de datos de Personas**: Clientes y Usuarios.
3. Objetivo 3: Gestionar la **venta de productos** con mayor agilidad.

ARQUITECTURA

Para conseguir los tres objetivos anteriores, se ha establecido una arquitectura acorde con las necesidades del proyecto, basada en unos Controller que llaman a la acción a través del Service, para que finalmente sea la Logic quien lleve a cabo las funciones que se solicita. Todo esto enmarcado en un Service Context y en base a las Entities que se han desarrollado para crear la estructura de la base de datos que hemos visto anteriormente.



CONTROLLERS Y MÉTODOS

La API presenta siete Controllers con distintos métodos cada uno de ellos.

1. Rol
 - Post: insertar nuevo rol
 - Get: ver roles actuales
 - Patch: modificar un rol
 - Delete: eliminar un rol
2. Tipo Cliente
 - Post: insertar nuevo tipo de cliente
 - Get: ver clasificación de clientes actual
 - Patch: modificar tipo de cliente
 - Delete: eliminar tipo de cliente
3. Persona
 - Post: insertar una persona nueva
 - Get: ver personas actuales en la base de datos
 - Patch: modificar una persona
 - Delete: eliminar una persona
4. Usuario
 - Post: insertar un usuario
 - Get: ver usuarios actuales
 - Get con Filtro: ver usuarios por rango de fechas
 - Patch: modificar un usuario
 - Delete: eliminar un usuario
5. Cliente
 - Post: insertar un cliente
 - Get: ver clientes actuales
 - Get con Filtro: ver usuarios por tipo de cliente y/o rango de fecha

- Patch: modificar un cliente
 - Delete: eliminar un cliente
6. Producto
- Post: insertar un producto
 - Get: ver productos actuales
 - Get con Filtro: ver productos por si están activos, rango de fechas, rango de precio o marca
 - Patch: modificar un producto
 - Delete: eliminar un producto
7. Pedido
- Post: insertar un pedido
 - Get: ver pedidos realizados
 - Get con Filtro: ver pedidos por si están activos, rango de fechas, rango de importes de venta, rango de fecha de entrega, por cliente, por producto, si están pagados y si están entregados.
 - Patch: modificar un pedido
 - Delete: eliminar un pedido

Controller Rol

MÉTODO POST:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "id": 0,
  "tipoRol": "string",
  "isActive": true
}
```

MÉTODO GET:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

MÉTODO PATCH:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "id": 0,
  "tipoRol": "string",
  "isActive": true
}
```

MÉTODO DELETE:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Id

Controller Tipo Cliente

MÉTODO POST:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "id": 0,
  "tipoRol": "string",
  "isActive": true
}
```

MÉTODO GET:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

MÉTODO PATCH:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "id": 0,
  "tipoRol": "string",
  "isActive": true
}
```

MÉTODO DELETE:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Id

Controller Persona

MÉTODO POST:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "id": 0,
```

```

    "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "dni": "string",
    "nombre": "string",
    "apellidos": "string",
    "fechaNacimiento": "2023-02-02T23:44:14.182Z",
    "dirección": "string",
    "población": "string",
    "provincia": "string",
    "códigoPostal": 0,
    "teléfono": 0,
    "email": "string",
    "insertDate": "2023-02-02T23:44:14.182Z",
    "updateDate": "2023-02-02T23:44:14.182Z",
    "isActive": true
  }

```

MÉTODO GET:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

MÉTODO PATCH:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```

{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "dni": "string",
  "nombre": "string",
  "apellidos": "string",
  "fechaNacimiento": "2023-02-02T23:44:14.182Z",
  "dirección": "string",
  "población": "string",
  "provincia": "string",
  "códigoPostal": 0,
  "teléfono": 0,
  "email": "string",
  "insertDate": "2023-02-02T23:44:14.182Z",
  "updateDate": "2023-02-02T23:44:14.182Z",
  "isActive": true
}

```

MÉTODO DELETE:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Id

Controller Usuario

MÉTODO POST:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "idPersona": 0,
  "idRol": 0,
  "usuario": "string",
  "password": "string"
}
```

MÉTODO GET:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

MÉTODO GETBYCRITERIA:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- InsertDateFrom
- InsertDateTo

MÉTODO PATCH:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "idPersona": 0,
  "idRol": 0,
  "usuario": "string",
  "password": "string"
}
```

MÉTODO DELETE:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Id

Controller Cliente

MÉTODO POST:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "id": 0,
  "idPersona": 0,
  "idRol": 0,
  "idTipoCliente": 0,
  "empresa": "string",
  "sector": "string",
  "isActive": true,
  "insertDate": "2023-02-03T13:06:18.500Z",
  "updateDate": "2023-02-03T13:06:18.500Z"
}
```

MÉTODO GET:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

MÉTODO GETBYCRITERIA:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- isActive
- insertDateFrom
- insertDateTo

MÉTODO PATCH:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "id": 0,
  "idPersona": 0,
  "idRol": 0,
  "idTipoCliente": 0,
  "empresa": "string",
  "sector": "string",
  "isActive": true,
  "insertDate": "2023-02-03T13:06:18.500Z",
  "updateDate": "2023-02-03T13:06:18.500Z"
}
```

```
}
```

MÉTODO DELETE:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Id

Controller Producto

MÉTODO POST:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "nombre": "string",
  "descripción": "string",
  "marca": "string",
  "precio": 0,
  "stock": 0,
  "insertDate": "2023-02-02T23:51:35.748Z",
  "updateDate": "2023-02-02T23:51:35.748Z",
  "isActive": true
}
```

MÉTODO GET:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

MÉTODO GETBYCRITERIA:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- IsActive
- InsertDateFrom
- InsertDateTo
- PrecioDesde
- PrecioHasta

MÉTODO PATCH:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "id": 0,
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "nombre": "string",
  "descripción": "string",
  "marca": "string",
  "precio": 0,
  "stock": 0,
  "insertDate": "2023-02-02T23:51:35.748Z",
  "updateDate": "2023-02-02T23:51:35.748Z",
  "isActive": true
}
```

MÉTODO GETBYMARCA:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Marca

MÉTODO DELETE:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Id

Controller Pedido

MÉTODO POST:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "fechaPedido": "2023-02-02T23:55:44.708Z",
  "idCliente": 0,
  "idTipoCliente": 0,
  "idProducto": 0,
  "precio": 0,
  "cantidad": 0,
  "fechaEntrega": "2023-02-02T23:55:44.708Z",
  "pagado": true,
  "entregado": true,
  "isActive": true
}
```

MÉTODO GET:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

MÉTODO GETBYCRITERIA:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- IsActive
- InsertDateFrom
- InsertDateTo
- ImporteTotalDesde
- ImporteTotalHasta
- FechaEntregaDesde
- FechaEntregaHasta

MÉTODO GETBYCLIENTE:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- IdCliente

MÉTODO GETBYPRODUCTO:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- IdProducto

MÉTODO GETBYPAGADO:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Pagado

MÉTODO GETBYENTREGADO:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Entregado

MÉTODO PATCH:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Body:

```
{
  "idWeb": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "fechaPedido": "2023-02-02T23:55:44.708Z",
  "idCliente": 0,
  "idTipoCliente": 0,
  "idProducto": 0,
  "precio": 0,
  "cantidad": 0,
  "fechaEntrega": "2023-02-02T23:55:44.708Z",
  "pagado": true,
  "entregado": true,
  "isActive": true
}
```

MÉTODO DELETE:

Request parámetros from Header:

- "userName": "string"
- "userPassword": "string"

Request parámetros from Query:

- Id

5. EXTENSIONES POSIBLES

En próximas extensiones de la WEB API se perfeccionará el sistema de filtrado de los distintos métodos, además de añadir extras al producto como crear gestión de stock según órdenes de pedido o seguimiento del pedido por parte del cliente final, entre otros.