

Universidad ORT Uruguay

Obligatorio 1

Diseño de Aplicaciones 2

Diseño Web Api

Martin Edelman - 263630

Tatiana Poznanski - 221056

Tomas Bañales - 239825

Profesores: Nicolás Fierro, Alexander Wieler, Marco Fiorito

2023

1. Link al repositorio.....	3
2. Descripción general de la Web API.....	3
3. Filtros.....	4
4. Resources.....	4
4.1 Controlador Login.....	4
4.2 Controlador Productos.....	5
4.3 Controlador Usuario.....	7
4.4 Controlador Compra.....	11

1. Link al repositorio

https://github.com/IngSoft-DA2-2023-2/263630_221056_239825.git

2. Descripción general de la Web API

En nuestro proyecto, hemos implementado una API REST para facilitar la interacción con el sistema, y nuestros endpoints siguen los principios estándar de REST. En particular, hemos adoptado prácticas recomendadas, como el uso de sustantivos sobre verbos, lo que hace que los endpoints sean intuitivos y fáciles de entender. Además, hemos optado por nombres en plural en lugar de singular, lo que es coherente con las convenciones de REST.

Una característica clave de REST que hemos mantenido es la statelessness (ausencia de estado). Cada solicitud enviada al servidor contiene toda la información necesaria para su procesamiento, lo que significa que no se mantiene un estado asociado entre las solicitudes. Esto simplifica la gestión del sistema y asegura una comunicación eficiente entre el cliente y el servidor.

Asimismo, hemos definido los siguientes códigos de respuesta estándar de HTTP para reflejar el estado de las solicitudes y respuestas:

200 OK: para indicar que la solicitud se procesó de manera exitosa y se devolvió una respuesta válida.

201 Created: Se utiliza para informar que la solicitud se completó con éxito y resultó en la creación de un nuevo recurso.

204 No Content: para indicar que una solicitud se completó con éxito; no devuelve datos en la respuesta.

400 Bad Request: para indicar que algún parámetro ingresado es incorrecto.

401 Unauthorized: para indicar que la solicitud no se procesó porque el cliente no tiene permiso para acceder al recurso solicitado.

404 Not Found: para indicar que el servidor no puede encontrar el recurso solicitado.

500 Internal Server Error: para indicar que ocurrió un error interno en el servidor mientras procesaba la solicitud, lo que impide que se complete con éxito.

3. Filtros

Para devolver los resultados de la lista completa de productos, se creó un sistema de filtrado de los mismos. El mismo consiste en cinco opciones de filtros distintos con la conectiva lógica OR, lo que significa que cualquiera de dichos criterios, así como cualquier combinación de los mismos, puede ser seleccionada por el usuario hipotético.

Las opciones disponibles incluyen filtrado por Precio, Nombre, Marca y Categoría; con la quinta opción siendo sin ningún criterio en particular, es decir, retornando la lista completa sin procesar.

Asimismo, se creó una clase QueryProducto, la cual actúa como base para la elaboración de la lógica posteriormente. Dicha lógica se encuentra en RepositorioProducto, dentro del método RetornarLista. Se considera que la lógica necesaria es lo suficientemente simple y concreta para ser incluida dentro del alcance de las funciones de la capa de DataAccess.

4. Resources

URL no específica de la api: <https://localhost:7061/api/v1/>

A esto se agregan diferentes terminaciones, como pueden ser "compras", "usuarios", "productos", etc

Se expandirá sobre estas variaciones en las secciones contiguas.

4.1 Controlador Login

Este recurso permite que se cree una nueva sesión de un usuario. Además, retorna un token.

ControladorLogin

POST /api/v1/authentication

- **POST /api/v1/authentication**

Este endpoint crea una nueva sesión del usuario y devuelve un token de autorización.

➤ **Body:**

```
{
  "correoElectronico": "string",
  "contrasena": "string"
}
```

➤ **Responses:**

- 201: Created - Se creó una nueva sesión y devuelve un string que es el Authentication Token para ese usuario.
- 400: Bad Request - El correo electrónico o la contraseña son incorrectos.

4.2 Controlador Productos

Este recurso cumple con la funcionalidad de permitir que se agregue, modifique y elimine un producto. Además, retorna un único producto o una lista de todos los productos con sus atributos.

ControladorProductos		^
GET	/api/v1/productos/{id}	▼
DELETE	/api/v1/productos/{id}	▼
PUT	/api/v1/productos/{id}	▼
GET	/api/v1/productos	▼
POST	/api/v1/productos	▼

- **GET /api/v1/productos/{id}**

Este endpoint devuelve un único producto con sus atributos.

➤ **Parámetros:**

- Key: "id"
- Value: "int"

➤ **Responses:**

- 200: Success - Detalle de un producto específico que se quiere leer
- 404: Not Found - El id del producto no se encuentra.

- **DELETE /api/v1/productos/{id}**

Este endpoint elimina un producto, es necesario enviar la id del producto a eliminar y un Authentication Token de un usuario administrador.

➤ **Parámetros:**

- Key: "id"
- Value: "int"

➤ **Header**

- Key: "Authentication"
- Value: "String"

➤ **Responses:**

- 204: Sin contenido
- 401: Unauthorized - "Token inválido. El usuario debe estar logueado o ingresar un token válido"
- 404: Not Found - El id del producto no se encuentra.

- **PUT /api/v1/productos/{id}**

Este endpoint actualiza un producto, es necesario enviar la id del producto a modificar y un Authentication Token de un usuario administrador..

- **Parámetros:**

- Key: "id"
 - Value: "int"

- **Header**

- Key: "Authentication"
 - Value: "String"

- **Body:**

```
{
  "nombre": "string",
  "precio": 0,
  "descripcion": "string",
  "marcald": 0,
  "categoriald": 0,
  "colores": [
    0
  ]
}
```

- **Responses:**

- 200: Success - "Producto actualizado con éxito"
 - 400: Bad Request - El nombre, precio, descripción, marca Id, categoría Id o colores es incorrecta.
 - 401: Unauthorized - "Token inválido. El usuario debe estar logueado o ingresar un token válido"
 - 404: Not Found - El id del producto no se encuentra.

- **GET /api/v1/productos**

Este endpoint devuelve la lista de productos con sus atributos, si se agrega parámetros, se puede filtrar por el nombre, precio, marca y categoría del producto..

- **Parámetros:**

Key	Value
"Nombre"	"String"
"Precio"	"Integer"
"Marcald"	"Integer"
"Categoriald"	"Integer"

➤ **Responses:**

- 200: Success - "Detalle completo de la lista de productos"
- 404: Not Found - El id del producto no se encuentra.

● **POST /api/v1/productos**

Este endpoint crea un nuevo producto, es necesario el Authentication Token de un usuario administrador.

➤ **Header**

- Key: "Authentication"
- Value: "String"

➤ **Body:**

```
{
  "nombre": "string",
  "precio": 0,
  "descripcion": "string",
  "marcaId": 0,
  "categoriaId": 0,
  "colores": [
    0
  ]
}
```

➤ **Responses:**

- 201: Created - "Producto creado con éxito"
- 401: Unauthorized - "Token inválido. El usuario debe estar logueado o ingresar un token válido"
- 400: Bad Request - El nombre, precio, descripción, marca Id, categoría Id o colores es incorrecta.

4.3 Controlador Usuario

Este recurso cumple con la funcionalidad de permitir que se registre, se modifique o se elimine un usuario, así como también crear o devolver una compra. Además, retorna un usuario o una lista de todos los usuarios con todas sus propiedades.

ControladorUsuario		^
POST	/api/v1/usuarios	▼
GET	/api/v1/usuarios	▼
GET	/api/v1/usuarios/{id}	▼
PUT	/api/v1/usuarios/{id}	▼
DELETE	/api/v1/usuarios/{id}	▼
GET	/api/v1/usuarios/{id}/compras	▼
POST	/api/v1/usuarios/{id}/compras	▼

- **POST /api/v1/usuarios**

Este endpoint crea un nuevo usuario.

- **Body:**

```
{
  "correoElectronico": "string",
  "direccionEntrega": "string",
  "rol": 0,
  "contrasena": "string"
}
```

- **Responses:**

- 201: Created - "Se creó un usuario con éxito y devuelve el usuario creado"
- 400: Bad Request - El correo electrónico, dirección de entrega, rol o contraseña es incorrecta.

- **GET /api/v1/usuarios**

Este endpoint devuelve la lista de usuarios, solo los administradores pueden ver todos los usuarios.

- **Header**

- Key: "Authentication"
- Value: "String"

- **Responses**

- 200: Success - "Detalle completo de la lista de usuarios"

- 401: Unauthorized - "Token inválido. El usuario debe estar logueado o ingresar un token válido"

- **GET /api/v1/usuarios/{id}**

Este endpoint devuelve un único usuario dado como parámetro, el Authentication Token debe ser de un usuario administrador o el usuario del id pasado como parámetro.

- **Header**

- Key: "Authentication"
- Value: "String"

- **Parámetros:**

- Key: "id"
- Value: "int"

- **Responses**

- 200: Success - Detalle completo del usuario específico que se quiere leer
- 401: Unauthorized - "Token inválido. El usuario debe estar logueado o ingresar un token válido"
- 404: Not Found - El id del usuario no se encuentra.

- **PUT /api/v1/usuarios/{id}**

Este endpoint modifica un usuario dado como parámetro, el Authentication Token debe ser de un usuario administrador o el usuario del id pasado como parámetro.

- **Header**

- Key: "Authentication"
- Value: "String"

- **Parámetros:**

- Key: "id"
- Value: "int"

- **Body:**

```
{
  "correoElectronico": "string",
  "direccionEntrega": "string",
  "rol": 0,
  "contrasena": "string"
}
```

- **Responses:**

- 200: Success - "Usuario modificado con éxito"
- 400: Bad Request - El correo electrónico, dirección de entrega, rol o contraseña es incorrecta.

- 401: Unauthorized - "Token inválido. El usuario debe estar logueado o ingresar un token válido"
- 404: Not Found - El id del usuario no se encuentra.

- **DELETE /api/v1/usuarios/{id}**

Este endpoint elimina un usuario dado como parámetro, el Authentication Token debe ser de un usuario administrador.

- **Header**

- Key: "Authentication"
- Value: "String"

- **Parámetros:**

- Key: "id"
- Value: "int"

- **Responses**

- 200: Success - "Se borró correctamente el usuario indicado"
- 401: Unauthorized - "Token inválido. El usuario debe estar logueado o ingresar un token válido"
- 404: Not Found - El id del usuario no se encuentra.

- **GET /api/v1/usuarios/{id}/compras**

Este endpoint devuelve una compra pasando una id como parámetro, el Authentication Token debe ser del usuario del id pasado como parámetro.

- **Header**

- Key: "Authentication"
- Value: "String"

- **Parámetros:**

- Key: "id"
- Value: "int"

- **Responses**

- 200: Success - Detalle completo de la compra específica que se quiere leer
- 401: Unauthorized - "Token inválido. El usuario debe estar logueado o ingresar un token válido"
- 404: Not Found - El id del usuario no se encuentra.

- **POST /api/v1/usuarios/{id}/compras**

Este endpoint crea una compra pasando una id como parámetro, el Authentication Token debe ser del usuario del id pasado como parámetro.

- **Header**

- Key: "Authentication"
- Value: "String"

➤ **Parámetros:**

- Key: "id"
- Value: "int"

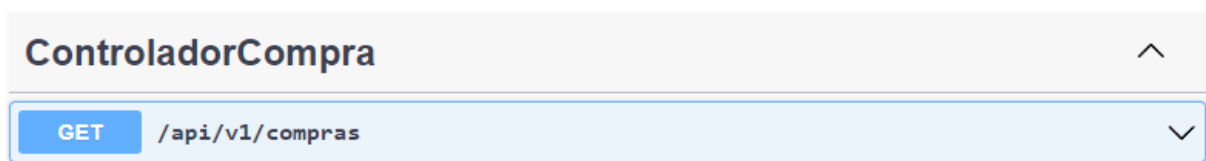
➤ **Body:**

```
{  
  "idProductos": [  
    0  
  ]  
}
```

➤ **Responses**

- 201: Created - "Compra creada con éxito"
- 400: Bad Request - La id de los productos es incorrecta.
- 401: Unauthorized - "Token inválido. El usuario debe estar logueado o ingresar un token válido"
- 404: Not Found - El id del usuario no se encuentra.

4.4 Controlador Compra



- **GET /api/v1/compras**

Este endpoint retorna todas las compras hechas por todos los usuarios en la base de datos, solo los administradores pueden acceder a ella, por ende, tienen que mandar un authorization token mediante el header.

➤ **Header**

- Key: "Authentication"
- Value: "String"

➤ 200: Success - Lista con un detalle completo de las compras.

➤ 401: Unauthorized - Token inválido. El usuario debe estar logueado o ingresar un token válido