Evaluación Continua MF0492_3 - UF1846

Susanna Signoretti

Tipo test:

1 - A

2 - C

3 - A

4 - B

5 - B

6 - D

7- B

8- A

9 - D

10 - C

11 - B

12- C

Problema práctico:

Gestionar el Catálogo de Libros:

1.1. Obtener la lista completa de libros disponibles:

Método HTTP: GET /api/books

Descripción: Este endpoint devuelve la lista completa de libros disponibles en el catálogo.

Justificación: Utilizamos HTTPS para garantizar la seguridad en la

transmisión de datos, especialmente al obtener información sensible como la lista completa de libros disponibles. Además, implementamos medidas de seguridad adicionales, como la autenticación y autorización basadas en roles, para proteger los datos contra accesos no autorizados. Esto contribuye a la estructura y claridad del diseño del servicio web al integrar la seguridad como parte fundamental del proceso de lectura. Además, cumplimos con las especificaciones y estándares de los protocolos web al priorizar la confidencialidad e integridad de la información mediante el uso de HTTPS.

1.2. Añadir un nuevo libro al catálogo:

- Método HTTP: POST /api/books

- Descripción: Este endpoint permite añadir un nuevo libro al catálogo.

- Justificación: Al utilizar HTTPS, garantizamos la transmisión segura de los datos durante la operación de creación. Además, implementamos medidas de seguridad como la validación y sanitización de los datos de entrada para proteger contra ataques de inyección. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de agregar libros al catálogo, cumpliendo así con las especificaciones y estándares de los protocolos web.

1.3. Actualizar la información de un libro existente:

- Método HTTP: PUT /api/books/{id}

- Descripción: Este endpoint permite actualizar la información de un libro existente en el catálogo.

- Justificación: Al utilizar HTTPS, aseguramos la transmisión segura de los datos durante la operación de actualización. Además, implementamos controles de seguridad adicionales, como la autenticación basada en roles, para garantizar que solo los usuarios

autorizados puedan realizar actualizaciones en el catálogo. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de actualizar la información del libro, cumpliendo con las especificaciones y estándares de los protocolos web.

1.4. Eliminar un libro del catálogo:

- Método HTTP: DELETE /api/books/{id}

- Descripción: Este endpoint permite eliminar un libro del catálogo.

- Justificación: Al emplear HTTPS, aseguramos la transmisión segura de los datos durante la operación de eliminación. Además, implementamos medidas de seguridad como la autorización basada en roles para garantizar que solo los usuarios con los permisos adecuados puedan eliminar libros del catálogo. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de eliminar recursos del catálogo, cumpliendo con las especificaciones y estándares de los protocolos web.

2.1. Crear una nueva orden de compra:

1. Método HTTP: POST /api/orders

2. Descripción: Este endpoint permite crear una nueva orden de compra en el sistema.

3. Justificación: Al utilizar HTTPS, garantizamos la transmisión segura de los datos durante la operación de creación. Además, implementamos medidas de seguridad como la autenticación y autorización basadas en roles para proteger contra accesos no autorizados. Esto contribuye a la estructura y claridad del diseño del servicio web al integrar la seguridad como parte fundamental del proceso de creación de órdenes. Además, cumplimos con las especificaciones y estándares de los protocolos web al priorizar la

confidencialidad e integridad de la información mediante el uso de HTTPS.

2.2. Obtener los detalles de una orden específica:

- Método HTTP: GET /api/orders/{id}

- Descripción: Este endpoint permite obtener los detalles de una orden específica en el sistema.

- Justificación: Utilizamos HTTPS para garantizar la seguridad en la transmisión de datos, especialmente al obtener información sensible como los detalles de una orden específica. Además, implementamos medidas de seguridad adicionales, como la autenticación basada en roles, para garantizar que solo los usuarios autorizados puedan acceder a los detalles de las órdenes. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de acceder a los detalles de las órdenes, cumpliendo así con las especificaciones y estándares de los protocolos web.

2.3. Actualizar el estado de una orden:

- Método HTTP: PUT /api/orders/{id}

- Descripción: Este endpoint permite actualizar el estado de una orden existente en el sistema.

- Justificación: Al utilizar HTTPS, aseguramos la transmisión segura de los datos durante la operación de actualización. Además, implementamos controles de seguridad adicionales, como la autenticación basada en roles, para garantizar que solo los usuarios autorizados puedan actualizar el estado de las órdenes. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de actualizar el estado de las órdenes, cumpliendo con las especificaciones y estándares de los protocolos web.

2.4. Eliminar una orden si es necesario:

- Método HTTP: DELETE /api/orders/{id}

- Descripción: Este endpoint permite eliminar una orden del sistema si es necesario.

- Justificación: Al emplear HTTPS, aseguramos la transmisión segura de los datos durante la operación de eliminación. Además, implementamos medidas de seguridad como la autorización basada en roles para garantizar que solo los usuarios con los permisos adecuados puedan eliminar órdenes del sistema. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de eliminar órdenes, cumpliendo con las especificaciones y estándares de los protocolos web.

Gestionar los Detalles del Usuario:

3.1. Crear un nuevo perfil de usuario:

- Método HTTP: POST /api/user-profiles

- Descripción: Este endpoint permite crear un nuevo perfil de usuario en el sistema.

- Justificación: Al utilizar HTTPS, garantizamos la transmisión segura de los datos durante la operación de creación. Además, implementamos medidas de seguridad como la validación y sanitización de los datos de entrada para proteger contra ataques de inyección. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de crear perfiles de usuario, cumpliendo así con las especificaciones y estándares de los protocolos web.

3.2. Obtener la información de un usuario específico:

- Método HTTP: GET /api/user-profiles/{id}
- Descripción: Este endpoint permite obtener la información de un usuario específico en el sistema.

- Justificación: Utilizamos HTTPS para garantizar la seguridad en la transmisión de datos, especialmente al obtener información sensible como la información de un usuario específico. Además, implementamos medidas de seguridad adicionales, como la autenticación basada en roles, para garantizar que solo los usuarios autorizados puedan acceder a la información de otros usuarios. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de acceder a la información de los usuarios, cumpliendo con las especificaciones y estándares de los protocolos web.

3.3. Actualizar la información del usuario:

- Método HTTP: PUT /api/user-profiles/{id}

- Descripción: Este endpoint permite actualizar la información de un usuario existente en el sistema.

- Justificación: Al utilizar HTTPS, aseguramos la transmisión segura de los datos durante la operación de actualización. Además, implementamos controles de seguridad adicionales, como la autenticación basada en roles, para garantizar que solo los usuarios autorizados puedan actualizar su propia información de usuario. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de actualizar la información de usuario, cumpliendo con las especificaciones y estándares de los protocolos web.

3.4. Eliminar un perfil de usuario:

- Método HTTP: DELETE /api/user-profiles/{id}

- Descripción: Este endpoint permite eliminar un perfil de usuario del sistema si es necesario.

- Justificación: Al emplear HTTPS, aseguramos la transmisión segura de los datos durante la operación de eliminación. Además, implementamos medidas de seguridad como la autorización basada

en roles para garantizar que solo los usuarios con los permisos adecuados puedan eliminar perfiles de usuario. Esto contribuye a la estructura y claridad del diseño del servicio web al proporcionar una forma segura y coherente de eliminar perfiles de usuario, cumpliendo con las especificaciones y estándares de los protocolos web.