

CyberCoreAPI

Overview

Version information

Version : v1

URI scheme

Host : localhost:44310

Schemes : HTTPS

Security

Bearer

API Key Authentication

Type : apiKey

Name : Authorizacion

In : HEADER

Paths

Crea una nueva unidad de almacenamiento en la base de datos.

POST /api/Almacenamientoes

Parameters

Type	Name	Description	Schema
Body	almacenamiento <i>required</i>	Los datos de la unidad de almacenamiento que se desea crear.	Almacenamiento

Responses

HTTP Code	Description	Schema
201	Creado. La unidad de almacenamiento se creó correctamente.	Almacenamiento
400	Solicitud incorrecta. El modelo no es válido.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Almacenamientos

Obtiene una lista de todas las unidades de almacenamiento disponibles en la base de datos.

```
GET /api/Almacenamientos
```

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de unidades de almacenamiento.	object
404	No encontrado. No se encontraron unidades de almacenamiento en la base de datos.	No Content

Produces

- `application/json`
- `text/json`

- `application/xml`
- `text/xml`

Tags

- Almacenamientos

Obtiene una unidad de almacenamiento específica por su ID.

```
GET /api/Almacenamientos/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la unidad de almacenamiento que se desea obtener.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la unidad de almacenamiento solicitada.	Almacenamiento
404	No encontrado. No se encontró una unidad de almacenamiento con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Almacenamientos

Actualiza una unidad de almacenamiento existente en la base de datos.

PUT /api/Almacenamientos/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la unidad de almacenamiento que se desea actualizar.	integer (int32)
Body	alm <i>required</i>	Los nuevos datos de la unidad de almacenamiento.	Almacenamiento

Responses

HTTP Code	Description	Schema
200	OK. La unidad de almacenamiento se actualizó correctamente.	No Content
400	Solicitud incorrecta. El modelo no es válido.	No Content
404	No encontrado. No se encontró una unidad de almacenamiento con el ID especificado.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- Almacenamientos

Elimina una unidad de almacenamiento específica por su ID.

DELETE /api/Almacenamientos/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la unidad de almacenamiento que se desea eliminar.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. La unidad de almacenamiento fue eliminada correctamente.	Almacenamiento
404	No encontrado. No se encontró una unidad de almacenamiento con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Almacenamientoes

Crea un nuevo carrito.

POST /api/Carritoes

Parameters

Type	Name	Description	Schema
Body	carrito <i>required</i>	Datos del nuevo carrito	Carrito

Responses

HTTP Code	Description	Schema
201	Created - Devuelve el carrito creado	Carrito
400	Bad Request - Datos del carrito inválidos	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Carrito

Obtiene todos los carritos existentes.

GET /api/Carrito

Responses

HTTP Code	Description	Schema
200	OK - Devuelve la lista de carritos	object
404	Not Found - No se encontraron carritos	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Carrito

Obtiene un carrito específico por su ID.

```
GET /api/Carritoes/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	ID del carrito (entero positivo)	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK - Devuelve el carrito solicitado	Carrito
404	Not Found - No se encontró el carrito con el ID especificado	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Carritoes

Actualiza un carrito existente.

```
PUT /api/Carritoes/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	ID del carrito a actualizar	integer (int32)
Body	carrito <i>required</i>	Datos actualizados del carrito	Carrito

Responses

HTTP Code	Description	Schema
204	No Content - Carrito actualizado correctamente	No Content
400	Bad Request - Datos del carrito inválidos	No Content
404	Not Found - Carrito no encontrado	No Content
409	Conflict - Error de concurrencia (datos modificados por otro usuario)	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- Carrito

Elimina un carrito existente.

```
DELETE /api/Carrito/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	ID del carrito a eliminar	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK - Devuelve el carrito eliminado	Carrito
404	Not Found - Carrito no encontrado	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Carrito

Crea una nueva fuente de poder en la base de datos.

POST /api/FuentePoder

Parameters

Type	Name	Description	Schema
Body	fuentePoder <i>required</i>	Los datos de la fuente de poder que se desea crear.	FuentePoder

Responses

HTTP Code	Description	Schema
201	Creado. La fuente de poder se creó correctamente.	FuentePoder
400	Solicitud incorrecta. El modelo no es válido.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produces

- `application/json`
- `text/json`
- `application/xml`

- `text/xml`

Tags

- FuentePoders

Obtiene una lista de todas las fuentes de poder disponibles en la base de datos.

```
GET /api/FuentePoders
```

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de fuentes de poder.	object
404	No encontrado. No se encontraron fuentes de poder en la base de datos.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- FuentePoders

Obtiene una fuente de poder específica por su ID.

```
GET /api/FuentePoders/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la fuente de poder que se desea obtener.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la fuente de poder solicitada.	FuentePoder
404	No encontrado. No se encontró una fuente de poder con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- FuentePoders

Actualiza una fuente de poder existente en la base de datos.

```
PUT /api/FuentePoders/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la fuente de poder que se desea actualizar.	integer (int32)
Body	fp <i>required</i>	Los nuevos datos de la fuente de poder.	FuentePoder

Responses

HTTP Code	Description	Schema
200	OK. La fuente de poder se actualizó correctamente.	No Content
400	Solicitud incorrecta. El modelo no es válido.	No Content
404	No encontrado. No se encontró una fuente de poder con el ID especificado.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- FuentePoders

Elimina una fuente de poder específica por su ID.

```
DELETE /api/FuentePoders/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la fuente de poder que se desea eliminar.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. La fuente de poder fue eliminada correctamente.	FuentePoder
404	No encontrado. No se encontró una fuente de poder con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- FuentePoders

Crea una nueva GPU en la base de datos.

POST /api/GPUs

Parameters

Type	Name	Description	Schema
Body	gPU <i>required</i>	Los datos de la GPU que se desea crear.	GPU

Responses

HTTP Code	Description	Schema
201	Creado. La GPU se creó correctamente.	GPU
400	Solicitud incorrecta. El modelo no es válido.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- GPUs

Obtiene una lista de todas las GPU disponibles en la base de datos.

GET /api/GPUs

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de GPU.	object
404	No encontrado. No se encontraron GPU en la base de datos.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- GPUs

Obtiene una GPU específica por su ID.

```
GET /api/GPUs/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la GPU que se desea obtener.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la GPU solicitada.	GPU
404	No encontrado. No se encontró una GPU con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- GPUs

Actualiza una GPU existente en la base de datos.

```
PUT /api/GPUs/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la GPU que se desea actualizar.	integer (int32)
Body	gpu <i>required</i>	Los nuevos datos de la GPU.	GPU

Responses

HTTP Code	Description	Schema
200	OK. La GPU se actualizó correctamente.	No Content
400	Solicitud incorrecta. El modelo no es válido.	No Content
404	No encontrado. No se encontró una GPU con el ID especificado.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- GPUs

Elimina una GPU específica por su ID.

```
DELETE /api/GPUs/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la GPU que se desea eliminar.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. La GPU fue eliminada correctamente.	GPU
404	No encontrado. No se encontró una GPU con el ID especificado.	No Content

Produces

- [application/json](#)
- [text/json](#)
- [application/xml](#)
- [text/xml](#)

Tags

- GPUs

Crea un nuevo gabinete en la base de datos.

POST /api/Gabinetes

Parameters

Type	Name	Description	Schema
Body	gabinete <i>required</i>	Los datos del gabinete que se desea crear.	Gabinete

Responses

HTTP Code	Description	Schema
201	Creado. El gabinete se creó correctamente.	Gabinete
400	Solicitud incorrecta. El modelo no es válido.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Gabinetes

Obtiene una lista de todos los gabinetes disponibles en la base de datos.

```
GET /api/Gabinetes
```

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de gabinetes.	object
404	No encontrado. No se encontraron gabinetes en la base de datos.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Gabinetes

Obtiene un gabinete específico por su ID.

```
GET /api/Gabinetes/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID del gabinete que se desea obtener.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. Devuelve el gabinete solicitado.	Gabinete
404	No encontrado. No se encontró un gabinete con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Gabinetes

Actualiza un gabinete existente en la base de datos.

```
PUT /api/Gabinetes/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID del gabinete que se desea actualizar.	integer (int32)
Body	gabinete <i>required</i>	Los nuevos datos del gabinete.	Gabinete

Responses

HTTP Code	Description	Schema
200	OK. El gabinete se actualizó correctamente.	No Content
400	Solicitud incorrecta. El modelo no es válido.	No Content
404	No encontrado. No se encontró un gabinete con el ID especificado.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- Gabinetes

Elimina un gabinete específico por su ID.

```
DELETE /api/Gabinetes/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID del gabinete que se desea eliminar.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. El gabinete fue eliminado correctamente.	Gabinete
404	No encontrado. No se encontró un gabinete con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`

- `application/xml`
- `text/xml`

Tags

- Gabinetes

Crea una nueva orden.

POST /api/Ordens

Parameters

Type	Name	Description	Schema
Body	orden <i>required</i>	Datos de la nueva orden	Orden

Responses

HTTP Code	Description	Schema
201	Created - Devuelve la orden creada	Orden
400	Bad Request - Datos inválidos	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Ordens

Obtiene todas las órdenes existentes.

```
GET /api/Ordens
```

Responses

HTTP Code	Description	Schema
200	OK - Devuelve la lista de órdenes	object
404	Not Found - No se encontraron órdenes	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Ordens

Obtiene una orden específica por su ID.

```
GET /api/Ordens/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	ID de la orden (entero positivo)	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK - Devuelve la orden solicitada	Orden
404	Not Found - Orden no encontrada	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Ordens

Actualiza una orden existente.

```
PUT /api/Ordens/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	ID de la orden a actualizar	integer (int32)
Body	orden <i>required</i>	Datos actualizados de la orden	Orden

Responses

HTTP Code	Description	Schema
204	No Content - Orden actualizada correctamente	No Content
400	Bad Request - Datos inválidos	No Content
404	Not Found - Orden no encontrada	No Content
409	Conflict - Error de concurrencia	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- Ordens

Elimina una orden existente.

```
DELETE /api/Ordens/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	ID de la orden a eliminar	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK - Devuelve la orden eliminada	Orden
404	Not Found - Orden no encontrada	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Ordens

Crea una nueva placa base en la base de datos.

```
POST /api/PlacaBases
```

Parameters

Type	Name	Description	Schema
Body	placaBase <i>required</i>	Los datos de la placa base que se desea crear.	PlacaBase

Responses

HTTP Code	Description	Schema
201	Creado. La placa base se creó correctamente.	PlacaBase
400	Solicitud incorrecta. El modelo no es válido.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- PlacaBases

Obtiene una lista de todas las placas base disponibles en la base de datos.

```
GET /api/PlacaBases
```

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de placas base.	object
404	No encontrado. No se encontraron placas base en la base de datos.	No Content

Produces

- `application/json`

- `text/json`
- `application/xml`
- `text/xml`

Tags

- PlacaBases

Obtiene una placa base específica por su ID.

```
GET /api/PlacaBases/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la placa base que se desea obtener.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la placa base solicitada.	PlacaBase
404	No encontrado. No se encontró una placa base con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- PlacaBases

Actualiza una placa base existente en la base de datos.

```
PUT /api/PlacaBases/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la placa base que se desea actualizar.	integer (int32)
Body	pb <i>required</i>	Los nuevos datos de la placa base.	PlacaBase

Responses

HTTP Code	Description	Schema
200	OK. La placa base se actualizó correctamente.	No Content
400	Solicitud incorrecta. El modelo no es válido.	No Content
404	No encontrado. No se encontró una placa base con el ID especificado.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- PlacaBases

Elimina una placa base específica por su ID.

```
DELETE /api/PlacaBases/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la placa base que se desea eliminar.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. La placa base fue eliminada correctamente.	PlacaBase
404	No encontrado. No se encontró una placa base con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- PlacaBases

Crea un nuevo procesador en la base de datos.

POST /api/Procesadors

Parameters

Type	Name	Description	Schema
Body	procesador <i>required</i>	Los datos del procesador que se desea crear.	Procesador

Responses

HTTP Code	Description	Schema
201	Creado. El procesador se creó correctamente.	Procesador
400	Solicitud incorrecta. El modelo no es válido.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produce

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Procesadores

Obtiene una lista de todos los procesadores disponibles en la base de datos.

```
GET /api/Procesadores
```

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de procesadores.	object
404	No encontrado. No se encontraron procesadores en la base de datos.	No Content

Produce

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Procesadores

Obtiene un procesador específico por su ID.

```
GET /api/Procesadores/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID del procesador que se desea obtener.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. Devuelve el procesador solicitado.	Procesador
404	No encontrado. No se encontró un procesador con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Procesadores

Actualiza un procesador existente en la base de datos.

```
PUT /api/Procesadores/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID del procesador que se desea actualizar.	integer (int32)
Body	procesador <i>required</i>	Los nuevos datos del procesador.	Procesador

Responses

HTTP Code	Description	Schema
200	OK. El procesador se actualizó correctamente.	No Content
400	Solicitud incorrecta. El modelo no es válido.	No Content
404	No encontrado. No se encontró un procesador con el ID especificado.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- Procesadors

Elimina un procesador específico por su ID.

```
DELETE /api/Procesadors/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID del procesador que se desea eliminar.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. El procesador fue eliminado correctamente.	Procesador
404	No encontrado. No se encontró un procesador con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Procesadors

Crea una nueva RAM en la base de datos.

POST /api/RAMs

Parameters

Type	Name	Description	Schema
Body	ram <i>required</i>	Los datos de la RAM que se desea crear.	RAM

Responses

HTTP Code	Description	Schema
201	Creado. La RAM se creó correctamente.	RAM
400	Solicitud incorrecta. El modelo no es válido.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- RAMs

Obtiene una lista de todas las RAM disponibles en la base de datos.

GET /api/RAMs

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de RAM.	object
404	No encontrado. No se encontraron RAM en la base de datos.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- RAMs

Obtiene una RAM específica por su ID.

```
GET /api/RAMs/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la RAM que se desea obtener.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la RAM solicitada.	RAM
404	No encontrado. No se encontró una RAM con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- RAMs

Actualiza una RAM existente en la base de datos.

```
PUT /api/RAMs/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la RAM que se desea actualizar.	integer (int32)
Body	ram <i>required</i>	Los nuevos datos de la RAM.	RAM

Responses

HTTP Code	Description	Schema
200	OK. La RAM se actualizó correctamente.	No Content
400	Solicitud incorrecta. El modelo no es válido.	No Content
404	No encontrado. No se encontró una RAM con el ID especificado.	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- RAMs

Elimina una RAM específica por su ID.

```
DELETE /api/RAMs/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	El ID de la RAM que se desea eliminar.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. La RAM fue eliminada correctamente.	RAM
404	No encontrado. No se encontró una RAM con el ID especificado.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- RAMs

Genera un reporte completo del inventario actual.

```
GET /api/ReporteInventarios
```

Description

Este endpoint: 1. Recupera todos los componentes del inventario 2. Calcula el stock total automáticamente 3. Incluye la fecha de generación del reporte

Responses

HTTP Code	Description	Schema
200	OK - Devuelve el reporte generado	ReporteInventario
500	Internal Server Error - Error al procesar el reporte	No Content

Produces

- `application/json`

- `text/json`
- `application/xml`
- `text/xml`

Tags

- ReporteInventarios

POST /api/Values

Parameters

Type	Name	Schema
Body	value <i>required</i>	string

Responses

HTTP Code	Description	Schema
204	No Content	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- Values

GET /api/Values

Responses

HTTP Code	Description	Schema
200	OK	< string > array

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Values

GET /api/Values/{id}

Parameters

Type	Name	Schema
Path	id <i>required</i>	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK	string

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Values

PUT /api/Values/{id}

Parameters

Type	Name	Schema
Path	id <i>required</i>	integer (int32)

Type	Name	Schema
Body	value <i>required</i>	string

Responses

HTTP Code	Description	Schema
204	No Content	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Tags

- Values

DELETE /api/Values/{id}

Parameters

Type	Name	Schema
Path	id <i>required</i>	integer (int32)

Responses

HTTP Code	Description	Schema
204	No Content	No Content

Tags

- Values

Autentica un usuario y genera un token JWT

POST /api/auth/login

Description

Ejemplo de solicitud: POST /api/auth/login { "correo": "usuario@example.com", "contraseña": "MiClaveSegura123" }

El token JWT devuelto debe incluirse en el header 'Authorization' de solicitudes subsiguientes:
Authorization: Bearer [token]

Parameters

Type	Name	Description	Schema
Body	loginDTO <i>required</i>	Credenciales de acceso	LoginDTO

Responses

HTTP Code	Description	Schema
200	OK - Devuelve el token JWT	TokenResponse
400	Bad Request - Datos inválidos	No Content
401	Unauthorized - Credenciales incorrectas	No Content

Consumes

- application/json
- text/json
- application/xml
- text/xml
- application/x-www-form-urlencoded

Produces

- application/json
- text/json
- application/xml
- text/xml

Tags

- Auth

Elimina un componente específico de un carrito.

```
DELETE /api/carritos/{carritoId}/componentes/{componenteId}
```

Parameters

Type	Name	Description	Schema
Path	carritoId <i>required</i>	ID del carrito	integer (int32)
Path	componenteId <i>required</i>	ID del componente a eliminar	integer (int32)

Responses

HTTP Code	Description	Schema
204	No Content - Componente eliminado correctamente	object
404	Not Found - Carrito o componente no encontrado	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Carrito

Busca componentes de computadora cuyo nombre contenga una cadena específica.

```
GET /api/componentes/buscar-por-nombre
```

Description

Ejemplo de solicitud: GET: `api/Componentes/buscar-por-nombre?nombre=Procesador`

Parameters

Type	Name	Description	Schema
Query	nombre <i>required</i>	La cadena de texto que se buscará dentro de los nombres de los componentes. No distingue entre mayúsculas y minúsculas.	string

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de componentes que coinciden con la cadena especificada.	object
404	No encontrado. No se encontraron componentes que coincidan con la cadena especificada.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Componentes

Filtra los componentes de computadora por una marca específica.

```
GET /api/componentes/filtrar-por-marca
```

Description

Ejemplo de solicitud: GET: `api/Componentes/filtrar-por-marca?marca=Intel`

Parameters

Type	Name	Description	Schema
Query	marca <i>required</i>	La marca de los componentes que se desean filtrar. Solo se incluirán componentes que coincidan con esta marca.	string

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de componentes filtrados por la marca especificada.	object
404	No encontrado. No se encontraron componentes de la marca especificada.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Componentes

Filtra los componentes de computadora por un rango de precios especificado.

```
GET /api/componentes/filtrar-por-precio
```

Description

Ejemplo de solicitud: GET: `api/Componentes/filtrar-por-precio?precioMin=100&precioMax=500`

Parameters

Type	Name	Description	Schema
Query	precioMax <i>required</i>	El precio máximo del rango. Solo se incluirán componentes con un precio igual o inferior a este valor.	number (double)
Query	precioMin <i>required</i>	El precio mínimo del rango. Solo se incluirán componentes con un precio igual o superior a este valor.	number (double)

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de componentes filtrados.	object
404	No encontrado. No se encontraron componentes en el rango de precios especificado.	No Content

Produce

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Componentes

Filtra los componentes de computadora por un stock mínimo especificado.

```
GET /api/componentes/filtrar-por-stock
```

Description

Ejemplo de solicitud: GET: `api/Componentes/filtrar-por-stock?stockMin=10`

Parameters

Type	Name	Description	Schema
Query	stockMin <i>required</i>	El stock mínimo requerido. Solo se incluirán componentes con un stock igual o superior a este valor.	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de componentes filtrados por el stock mínimo especificado.	object
404	No encontrado. No se encontraron componentes con el stock mínimo especificado.	No Content

Produce

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Componentes

Filtra los componentes de computadora por un tipo específico, utilizando nombres comunes para la búsqueda.

```
GET /api/componentes/filtrar-por-tipo
```

Parameters

Type	Name	Description	Schema
Query	tipo <i>required</i>	El tipo de componente que se desea filtrar. Puede ser uno de los siguientes valores: - "Procesador" - "Tarjeta gráfica", "Tarjeta grafica", "GPU" - "Memoria RAM", "RAM" - "Disco duro", "Disco duro HDD", "SSD" - "Placa base", "Placa madre" - "Fuente de poder", "Fuente" - "Caja", "Gabinete"	string

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de componentes que coinciden con el tipo especificado.	object
400	Solicitud incorrecta. El tipo proporcionado no es válido. Se incluye un mensaje con los tipos válidos.	No Content
404	No encontrado. No se encontraron componentes del tipo especificado.	No Content

Produce

- `application/json`

- `text/json`
- `application/xml`
- `text/xml`

Tags

- Componentes

Ordena los componentes de computadora por precio, en orden ascendente o descendente.

GET `/api/componentes/ordenar-por-precio`

Description

Ejemplos de solicitud: - Orden ascendente: GET: `api/Componentes/ordenar-por-precio?orden=asc` - Orden descendente: GET: `api/Componentes/ordenar-por-precio?orden=desc` - Por defecto (ascendente): GET: `api/Componentes/ordenar-por-precio`

Parameters

Type	Name	Description	Schema
Query	orden <i>optional</i>	El criterio de ordenación. Puede ser "asc" para orden ascendente o "desc" para orden descendente. Si no se proporciona un valor, por defecto se ordenará en orden ascendente.	string

Responses

HTTP Code	Description	Schema
200	OK. Devuelve la lista de componentes ordenados por precio.	object
400	Solicitud incorrecta. El parámetro 'orden' no es válido.	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Componentes

Actualiza el estado de una orden existente.

PUT /api/ordenes/{id}/estado

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	ID de la orden	integer (int32)
Query	nuevoEstado <i>required</i>	Nuevo estado de la orden	enum (0, 1, 2, 3, 4)

Responses

HTTP Code	Description	Schema
204	No Content - Estado actualizado correctamente	object
404	Not Found - Orden no encontrada	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Ordens

Genera un reporte de ventas dentro de un rango de fechas específico.

POST /api/reportes/ventas

Description

Criterios importantes: - Las fechas deben estar en formato (YYYY-MM-DD) - El rango máximo permitido es de 1 año - Fechas futuras no son permitidas

Parameters

Type	Name	Description	Schema
Query	fechaFinal <i>required</i>	Fecha de fin del reporte	string (date-time)
Query	fechaInicio <i>required</i>	Fecha de inicio del reporte	string (date-time)

Responses

HTTP Code	Description	Schema
200	OK - Devuelve el reporte generado	object
400	Bad Request - Parámetros inválidos	No Content
404	Not Found - No se encontraron ventas en el rango	No Content
500	Internal Server Error - Error al procesar el reporte	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- ReporteVentas

GET /api/test/ping

Responses

HTTP Code	Description	Schema
200	OK	object

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Default

Crea un nuevo usuario (No requiere autenticación)

POST /api/usuario

Parameters

Type	Name	Description	Schema
Body	usuario <i>required</i>	Datos del nuevo usuario	Usuario

Responses

HTTP Code	Description	Schema
201	Created - Usuario registrado	Usuario
400	Bad Request - Datos inválidos	No Content
409	Conflict - El correo ya está registrado	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`
- `application/x-www-form-urlencoded`

Produces

- `application/json`
- `text/json`

- `application/xml`
- `text/xml`

Tags

- Usuarios

Obtiene todos los usuarios (Solo Admin)

GET /api/usuario

Responses

HTTP Code	Description	Schema
200	OK - Lista de usuarios	object
401	Unauthorized - No autenticado	No Content
403	Forbidden - No tiene rol Admin	No Content
404	Not Found - No hay usuarios registrados	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Usuarios

Obtiene un usuario específico por ID

GET /api/usuario/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	ID del usuario (requiere ser el propio usuario o Admin)	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK - Usuario encontrado	Usuario
401	Unauthorized - No autenticado	No Content
403	Forbidden - No es el usuario dueño ni Admin	No Content
404	Not Found - Usuario no existe	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Usuarios

PUT /api/usuario/{id}

Parameters

Type	Name	Schema
Path	id <i>required</i>	string
Body	usuario <i>required</i>	Usuario

Responses

HTTP Code	Description	Schema
204	No Content	No Content

Consumes

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

- `application/x-www-form-urlencoded`

Tags

- Usuarios

Elimina un usuario (Solo Admin)

```
DELETE /api/usuario/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	ID del usuario a eliminar	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK - Usuario eliminado	Usuario
401	Unauthorized - No autenticado	No Content
403	Forbidden - No tiene rol Admin	No Content
404	Not Found - Usuario no existe	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Usuarios

Obtiene el historial de órdenes de un usuario específico.

```
GET /api/usuarios/{usuarioId}/ordenes
```

Parameters

Type	Name	Description	Schema
Path	usuarioId <i>required</i>	ID del usuario (entero positivo)	integer (int32)

Responses

HTTP Code	Description	Schema
200	OK - Devuelve el historial de órdenes	object
404	Not Found - No se encontraron órdenes para el usuario	No Content

Produces

- `application/json`
- `text/json`
- `application/xml`
- `text/xml`

Tags

- Ordenes

Definitions

Almacenamiento

Name	Schema
Capacidad <i>optional</i>	integer (int32)
Categoria <i>optional</i>	string
Descripcion <i>optional</i>	string
ID <i>optional</i>	integer (int32)
Marca <i>optional</i>	string
Nombre <i>optional</i>	string

Name	Schema
Precio <i>optional</i>	number (double)
Stock <i>optional</i>	integer (int32)
TipoAlmacenamiento <i>optional</i>	string
VelocidadEscritura <i>optional</i>	integer (int32)
VelocidadLectura <i>optional</i>	number (double)

Carrito

Name	Schema
Componentes <i>optional</i>	< Componentes > array
ID <i>optional</i>	integer (int32)
Total <i>optional</i>	number (double)
Usuario <i>optional</i>	Usuario
UsuarioID <i>optional</i>	integer (int32)

Componentes

Name	Schema
Categoria <i>optional</i>	string
Descripcion <i>optional</i>	string
ID <i>optional</i>	integer (int32)
Marca <i>optional</i>	string
Nombre <i>optional</i>	string

Name	Schema
Precio <i>optional</i>	number (double)
Stock <i>optional</i>	integer (int32)

FuentePoder

Name	Schema
Categoria <i>optional</i>	string
Certificacion <i>optional</i>	string
Descripcion <i>optional</i>	string
ID <i>optional</i>	integer (int32)
Marca <i>optional</i>	string
Nombre <i>optional</i>	string
Potencia <i>optional</i>	integer (int32)
Precio <i>optional</i>	number (double)
Stock <i>optional</i>	integer (int32)

GPU

Name	Schema
Categoria <i>optional</i>	string
Descripcion <i>optional</i>	string
ID <i>optional</i>	integer (int32)
Marca <i>optional</i>	string

Name	Schema
Nombre <i>optional</i>	string
Precio <i>optional</i>	number (double)
Stock <i>optional</i>	integer (int32)
TipoMemoria <i>optional</i>	string
VelocidadRelej <i>optional</i>	number (double)
Vram <i>optional</i>	integer (int32)

Gabinete

Name	Schema
Categoria <i>optional</i>	string
Descripcion <i>optional</i>	string
Formato <i>optional</i>	string
ID <i>optional</i>	integer (int32)
Marca <i>optional</i>	string
Nombre <i>optional</i>	string
Precio <i>optional</i>	number (double)
Stock <i>optional</i>	integer (int32)
VentanaAcrilico <i>optional</i>	boolean

LoginDTO

Name	Schema
Contraseña <i>required</i>	string
Correo <i>required</i>	string

MetodoPago

Name	Schema
Detalles <i>optional</i>	string
ID <i>optional</i>	integer (int32)
TipoPago <i>optional</i> <i>read-only</i>	string

Orden

Name	Schema
CarritoID <i>optional</i>	integer (int32)
Componentes <i>optional</i>	< Componentes > array
Estado <i>optional</i>	enum (0, 1, 2, 3, 4)
Fecha <i>optional</i>	string (date-time)
ID <i>optional</i>	integer (int32)
MetodoPago <i>optional</i>	MetodoPago
Total <i>optional</i>	number (double)
Usuario <i>optional</i>	Usuario
UsuarioID <i>optional</i>	integer (int32)

PlacaBase

Name	Schema
Categoria <i>optional</i>	string
Descripcion <i>optional</i>	string
Formato <i>optional</i>	string
ID <i>optional</i>	integer (int32)
Marca <i>optional</i>	string
Nombre <i>optional</i>	string
Precio <i>optional</i>	number (double)
SlotsPCIe <i>optional</i>	integer (int32)
SlotsRAM <i>optional</i>	integer (int32)
Socket <i>optional</i>	string
Stock <i>optional</i>	integer (int32)

Procesador

Name	Schema
Categoria <i>optional</i>	string
Descripcion <i>optional</i>	string
Frecuencia <i>optional</i>	number (double)
Hilos <i>optional</i>	integer (int32)
ID <i>optional</i>	integer (int32)

Name	Schema
Marca <i>optional</i>	string
Nombre <i>optional</i>	string
Nucleos <i>optional</i>	integer (int32)
Precio <i>optional</i>	number (double)
Socket <i>optional</i>	string
Stock <i>optional</i>	integer (int32)

RAM

Name	Schema
Capacidad <i>optional</i>	integer (int32)
Categoria <i>optional</i>	string
Descripcion <i>optional</i>	string
Frecuencia <i>optional</i>	integer (int32)
ID <i>optional</i>	integer (int32)
Marca <i>optional</i>	string
Nombre <i>optional</i>	string
Precio <i>optional</i>	number (double)
Stock <i>optional</i>	integer (int32)
TipoMemoria <i>optional</i>	string

ReporteInventario

Name	Schema
Componentes <i>optional</i>	< Componentes > array
Fecha <i>optional</i>	string (date-time)
ID <i>optional</i>	integer (int32)
TotalStock <i>optional</i>	integer (int32)

TokenResponse

Name	Schema
Token <i>optional</i>	string

Usuario

Name	Schema
Contraseña <i>optional</i>	string
Correo <i>optional</i>	string
ID <i>optional</i>	integer (int32)
Nombre <i>optional</i>	string
esAdmin <i>optional</i>	boolean