

UNIVERSIDAD ORT URUGUAY

Facultad de Ingeniería

Obligatorio 1 Diseño de Aplicaciones 2

Entregado como requisito para la obtención del  
título de Ingeniero en Sistemas

Martín Slepian - 266959

Leopoldo Perez - 257341

Eric Poplawski - 258327

Profesores: Nicolás Fierro, Alexander Wieler, Sofia  
Duclos

2024

Link al repositorio: <https://github.com/IngSoft-DA2/257341-266959-258327>

## Declaración de autoría

Nosotros, Martín Slepian, Leopoldo Perez y Eric Poplawski declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

1. La obra fue producida en su totalidad mientras cursamos Diseño de Aplicaciones.
2. Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad.
3. Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra.
4. En la obra, hemos acusado recibo de las ayudas recibidas.
5. Cuando la obra se basa en el trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y que fue construido por nosotros.
6. Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Martín Slepian

7/10/2024

Leopoldo Perez

7/10/2024

Eric Poplawski

7/10/2024

## **Abstract**

Este documento detalla la especificación y diseño de una API REST, conforme a los principios arquitecturales de REST. La API es escalable, segura y robusta, utilizando de manera coherente los métodos HTTP (GET, POST, PUT, DELETE) para operaciones CRUD y asegurando rutas semánticas claras.

Para garantizar la seguridad, se ha implementado un mecanismo de autenticación basado en tokens JWT, permitiendo que solo usuarios autenticados accedan a los recursos protegidos. Además, se ha diseñado un manejo de errores exhaustivo, abarcando validaciones de entrada y fallos del servidor, proporcionando retroalimentación precisa y clara a los desarrolladores.

La especificación incluye una descripción detallada de los endpoints, los parámetros necesarios, las respuestas esperadas y los códigos de estado HTTP posibles, asegurando una interacción segura y predecible. El diseño sigue los principios de stateless, donde cada solicitud es autocontenida, lo que asegura la escalabilidad y eficiencia de la API.

# Índice

<b>Carátula.....</b>	<b>1</b>
<b>Declaración de autoría.....</b>	<b>2</b>
<b>Abstract.....</b>	<b>3</b>
<b>Índice.....</b>	<b>4</b>
Buenas prácticas en las API REST.....	5
Restricciones de REST que se cumplen en nuestra API.....	6
Mecanismo de Autenticación.....	7
Códigos de Error.....	8
Especificación de la API.....	8
<b>Referencias Bibliográficas.....</b>	<b>23</b>

## Buenas prácticas en las API REST

La API está diseñada siguiendo buenas prácticas RESTful, comenzando por el uso apropiado de URIs para representar recursos de manera clara y coherente. Las rutas como `/api/homes` y `/api/homes/{homelId}` utilizan sustantivos en plural para describir los recursos, lo cual es una recomendación habitual en el diseño de APIs RESTful, ya que refleja la naturaleza de los datos que se manejan. Esta estructura no solo proporciona claridad, sino que también permite la extensibilidad al facilitar la adición de nuevos puntos de acceso de manera predecible.

Un aspecto destacado de nuestra implementación son las rutas jerárquicas, como `/api/homes/{homelId}/devices` y `/api/homes/{homelId}/residents`. Estas rutas anidadas muestran la relación lógica entre los recursos, permitiendo una navegación intuitiva y comprensible para los usuarios de la API.

Además, incluimos identificadores únicos como `{homelId}` y `{hardwareDeviceId}` en las rutas para referirse a instancias específicas de recursos. Esto permite un acceso flexible a recursos individuales sin la necesidad de crear rutas complejas o añadir parámetros adicionales innecesarios.

La API también cumple con la buena práctica del uso coherente de métodos HTTP, asegurando que cada operación esté asociada al método correcto. Por ejemplo, se utiliza POST para crear recursos como `CreateHome`, y GET se utiliza para obtener datos sin modificar el estado del recurso lo cual es fundamental. Esta correspondencia entre operaciones y métodos HTTP mejora la semántica de la API, haciéndola más intuitiva y fácil de usar.

Por último, la API proporciona soporte para filtrado y paginación, mejorando la experiencia del cliente. Un ejemplo de esto se encuentra en el método `ListCompanies`, donde se utilizan parámetros de consulta (como `pageNumber`, `pageSize`, `companyName` y `ownerName`) para filtrar y paginar los resultados. Esto permite a los clientes acceder únicamente a la información necesaria y gestionar grandes conjuntos de datos de manera eficiente, garantizando respuestas rápidas y evitando el envío de datos innecesarios, lo cual está alineado con las recomendaciones de diseño de API.

## **Restricciones de REST que se cumplen en nuestra API**

### **Client-Server**

El diseño de la API sigue la separación de responsabilidades entre cliente y servidor. Los controladores (DeviceController, HomeController, NotificationController, etc.) actúan como intermediarios entre el cliente y la lógica de negocio, delegando la lógica a los servicios especializados (DeviceService, NotificationService). Esto permite que la interfaz de usuario evolucione de forma independiente del backend, mejorando la portabilidad y la flexibilidad del sistema. Por ejemplo, para obtener una lista de dispositivos, el cliente interactúa con el DeviceController, que consulta los datos a través de los servicios y repositorios. Esto facilita el mantenimiento y la actualización de la API sin impactar a los clientes conectados.

### **Stateless**

Cada solicitud HTTP contiene toda la información necesaria para ser procesada, eliminando la necesidad de mantener el estado en el servidor entre solicitudes. Esto se logra a través de tokens de autenticación enviados en los headers de cada solicitud. Aunque se podría discutir si existe cierto grado de stateful en el sentido que un cliente usando Postman puede guardar la información del token que utiliza para autenticarse entre cada Request.

### **Uniform Interface**

La API sigue una interfaz uniforme, garantizando que los recursos sean accesibles de manera consistente. Las rutas de la API están diseñadas de forma clara, lo que facilita entender a qué recurso se refieren, como por ejemplo, /api/devices para dispositivos. Se utilizan verbos HTTP adecuados para cada operación (GET para obtener datos, POST para crear, PUT para actualizar y DELETE para eliminar). Las respuestas incluyen códigos de estado significativos (200 OK, 400 Bad Request, 404 Not Found), facilitando la comprensión y manejo de errores por parte de los clientes.

### **Layered System**

La solución está organizada en capas: presentación (controladores), lógica de negocio (servicios) y acceso a datos (repositorios y ApplicationDbContext). Esto mejora la modularidad, la seguridad y la escalabilidad de la aplicación, permitiendo mantener y actualizar cada capa de forma independiente.

## **Uso de Códigos de Estado HTTP**

La API utiliza correctamente los códigos de estado HTTP para que los clientes puedan interpretar las respuestas. Se devuelve 200 OK para operaciones exitosas, 400 Bad Request cuando los datos son inválidos, 401 Unauthorized si el token de autenticación no es válido, y 403 Forbidden cuando el usuario no tiene los permisos necesarios. Esto, combinado con mensajes descriptivos, facilita la gestión de errores y asegura que los clientes puedan adaptarse a las distintas situaciones.

## **Orientado a Recursos**

La API está centrada en la manipulación de recursos como User, Device, Home y Notification, cada uno con una URI única. Esto permite realizar operaciones específicas sobre cada recurso de manera predecible, como acceder a un dispositivo con `/api/devices/{id}`. Las respuestas se estructuran principalmente en formato JSON, asegurando la compatibilidad y la fácil integración con diferentes clientes.

## **Mecanismo de Autenticación**

### **Mecanismo de Autenticación basado en Token**

La autenticación se maneja mediante un sistema de tokens que se genera al iniciar sesión. Este token se envía en cada solicitud a través del encabezado Authorization, usando el formato Bearer {token}. Los tokens permiten que el servidor identifique al usuario sin mantener una sesión en el servidor. Esto facilita la seguridad y la escalabilidad del sistema, permitiendo que la autenticación sea manejada de forma sencilla y eficiente.

### **AuthenticationFilterAttribute**

La clase AuthenticationFilterAttribute se encarga de validar la autenticación antes de que las solicitudes lleguen a los controladores. Este filtro revisa el encabezado Authorization para verificar que el token esté presente y tenga el formato correcto. Si el token es válido, la información del usuario se almacena en el contexto HTTP para su uso posterior. En caso contrario, se devuelve un código HTTP 401 Unauthorized, indicando que la solicitud no está autenticada. Este mecanismo asegura que solo los usuarios autenticados puedan acceder a los recursos protegidos, lo que refuerza la seguridad de la API.

## **Autorización de Acceso a través de Permisos**

El proyecto utiliza un sistema de autorización que se basa en permisos específicos del usuario. Estos permisos definen las acciones permitidas para cada usuario, como crear o listar recursos. `AuthorizationFilterAttribute` se encarga de verificar si el usuario tiene los permisos necesarios antes de permitir el acceso a una ruta. Si no tiene los permisos requeridos, se devuelve un código HTTP 403 Forbidden. Esto asegura que, aunque un usuario esté autenticado, solo pueda acceder a los recursos que le están autorizados, mejorando la seguridad y control del sistema.

## **Seguridad en las Rutas de API**

Las rutas de la API están protegidas con filtros que exigen autenticación y autorización. Esto se implementa a través de los filtros `AuthenticationFilterAttribute` y `AuthorizationFilterAttribute`, aplicados en controladores como `DeviceController` y `CompanyController`. Así, cada ruta define quién puede acceder y bajo qué condiciones. Esto permite que solo usuarios autenticados y autorizados interactúen con recursos sensibles, como dispositivos y empresas. Esta estrategia asegura que los datos y las funcionalidades críticas del sistema solo sean accesibles para aquellos usuarios que tienen el permiso adecuado, protegiendo la integridad de la aplicación.

## **Códigos de Error**

- **1xx**: Información (no aplicable en esta API).
- **2xx**: Respuestas exitosas (200 OK, 201 Created).
- **3xx**: Redirecciones (no aplicable en esta API).
- **4xx**: Errores del cliente (400 Bad Request, 401 Unauthorized, 403 Forbidden, 404).
- **5xx**: Errores del servidor (500 Internal Server Error).

## **Especificación de la API**

**URL base:** /api



## **/administrators**

**Recurso:** Usuario

**Verbo:** POST

Headers: Authorization: {token}

**Entradas:** {"FirstName": string (FromBody), "LastName": string (FromBody), "Email": string (FromBody), "Password": string (FromBody)}

Salidas: {"Id": string}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad Request: "Argument can not be null or empty. Value of email cannot be null or empty"

400 Bad Request: "Argument can not be null or empty. Value of password cannot be null or empty"

400 Bad Request: "Argument can not be null or empty. Value of firstName cannot be null or empty"

400 Bad Request: "Argument can not be null or empty. Value of lastName cannot be null or empty"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission CreateAdministrator"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Creación de usuario con rol administrador

## **/companyOwners**

**Recurso:** Usuario

**Verbo:** POST

Headers: Authorization: {token}

**Entradas:** {"FirstName": string (FromBody), "LastName": string (FromBody), "Email": string (FromBody), "Password": string (FromBody)}

Salidas: {"Id": string}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of firstName cannot be

null or empty"

400 Bad request: "Argument can not be null or empty. Value of lastName cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of email cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of password cannot be null or empty"

400 Bad Request: "Argument is not valid. Incorrect format"

400 Bad Request: "Argument is not valid. Must have at least one number and special character"

400 Bad Request: "Exception thrown in Service. There is already a user with this email"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission CreateCompanyOwner"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Creación de usuario con rol de dueño de empresa

## **/homeOwners**

**Recurso:** Usuario

**Verbo:** POST

Headers: Authorization: {token}

**Entradas:** {"FirstName": string (FromBody), "LastName": string (FromBody), "Email": string (FromBody), "Password": string (FromBody), "ProfilePicture": string (FromBody)}

Salidas: {"Id": string}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of firstName cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of lastName cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of email cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of password cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of profilePicture cannot be null or empty"

400 Bad Request: "Argument is not valid. Incorrect format"

400 Bad Request: "Argument is not valid. Must have at least one number and special character"

400 Bad Request: "Exception thrown in Service. There is already a user with this email"

400 Bad Request: "Argument is not valid. Profile picture must be a valid URL"

401 Unauthorized: "You are not authenticated"

404 Not Found: "Entity not found. Entity not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Creación de usuario con rol de dueño de hogar

#### **/users**

**Recurso:** Usuario

**Verbo:** GET

Headers: Authorization: {token}

**Entradas:** {"pageNumber": int (FromQuery), "pageSize": int (FromQuery), "roleName": string (FromQuery), "fullName": string (FromQuery)}

Salidas: [{"FirstName": string, "LastName": string, "FullName": string, "RoleName": string, "CreationDate": DateTime}]

**Respuestas:**

200 Ok

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission ListUsers"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Listar todos los usuarios

#### **/administrators/{id}**

**Recurso:** Usuario

**Verbo:** DELETE

Headers: Authorization: {token}

**Entradas:** {"id": string (FromRoute)}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Invalid ID format"

400 Bad Request : "Exception thrown in Service. Only users with role 'Administrator' can be deleted"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission DeleteAdministrator"

404 Not Found: "Entity not found. User not found"

404 Not Found: "Entity not found. Entity not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Borrar un usuario con rol administrador

**/companies**

**Recurso:** Empresa

**Verbo:** GET

Headers: Authorization: {token}

**Entradas:** {"pageNumber": int (FromQuery), "pageSize": int (FromQuery), "companyName": string (FromQuery), "OwnerName": string (FromQuery)}

Salidas: [{"CompanyName": string, "OwnerName": string, "OwnerEmail": string, "CompanyRUT": int}]

**Respuestas:**

200 Ok

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission ListCompanies"

404 Not Found: "Entity not found. Entity not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Listar todas las empresas

**/companies**

**Recurso:** Empresa

**Verbo:** POST

Headers: Authorization: {token}

**Entradas:** {"Name": string (FromBody), "Logo": string (FromBody), "RUT": int (FromBody)}

Salidas: {"Id": string}

**Respuestas:**

200 Ok

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission CreateCompany"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Creación de una empresa

**/devices/securityCamera**

**Recurso:** Dispositivo

**Verbo:** POST

Headers: Authorization: {token}

**Entradas:** {"Name": string (FromBody), "ModelNumber": int (FromBody),  
"Description": string (FromBody), "MainPicture": string (FromBody), "Photographies":  
string[] (FromBody), "MotionDetectionEnabled": string (FromBody),  
"PersonDetectionEnabled": string (FromBody), "UsageType": string (FromBody)}

Salidas: {"Id": string}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of name cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of modelNumber cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of description cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of mainPicture cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of usageType cannot be null or empty"

400 Bad Request: "Exception thrown in Service. There is already a device with this name"

400 Bad Request: "Exception thrown in Service. There is already a device with this modelNumber"

400 Bad Request: "Exception thrown in Service. Usage type must be interior or

exterior"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission RegisterSecurityCamera"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Registrar una cámara de seguridad

### **/devices/windowSensor**

**Recurso:** Dispositivo

**Verbo:** POST

Headers: Authorization: {token}

**Entradas:** {"Name": string (FromBody), "ModelNumber": string (FromBody),

"Description": string (FromBody), "Photographies": string[] (FromBody)}

Salidas: {"Id": string}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of name cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of modelNumber cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of description cannot be null or empty"

400 Bad Request: "Exception thrown in Service. There is already a device with this name"

400 Bad Request: "Exception thrown in Service. There is already a device with this modelNumber"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission RegisterWindowSensor"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Registrar un sensor de ventana

### **/devices**

**Recurso:** Dispositivo

**Verbo:** GET

Headers: Authorization: {token}

**Entradas:** {"pageNumber": int (FromQuery), "pageSize": int (FromQuery),  
"deviceName": string (FromQuery), "modelName": string (FromQuery),  
"companyName": string (FromQuery), "deviceType": string (FromQuery)}

Salidas: [{"Name": string, "ModelNumber": string, "MainPicture": string,  
"CompanyName": string}]

**Respuestas:**

200 Ok

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission ListDevices"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Listado de dispositivos

**/supportedDevices**

**Recurso:** Dispositivo

**Verbo:** GET

Headers: Authorization: {token}

Salidas: [{"Type": string}]

**Respuestas:**

200 Ok

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission ListSupportedDevices"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Nos dice los tipos de dispositivos soportados

**/homes**

**Recurso:** Hogar

**Verbo:** POST

Headers: Authorization: {token}

**Entradas:** {"Street": string (FromBody), "HouseNumber": int (FromBody), "Latitude":  
double (FromBody), "Longitude": double (FromBody), "QuantityOfResidents": int  
(FromBody), "QuantityOfResidentsAllowed": int (FromBody)}

Salidas: {"Id": string}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of street cannot be null or empty"

400 Bad request: "Exception thrown in Service. Quantity of residents cannot be greater than quantity of residents allowed"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission CreateHome"

404 Not Found: "Entity not found. Entity not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Creación de un hogar

**/homes/{homeld}**

**Recurso:** Hogar

**Verbo:** POST

Headers: Authorization: {token}

**Entradas:** {"homeld": string (FromRoute), "userId": string (FromBody)}

Salidas: {"id": string}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of userId cannot be null or empty"

400 Bad request: "Exception thrown in Service. Home is full"

400 Bad request: "Exception thrown in Service. User is not a home owner"

400 Bad request: "Exception thrown in Service. User is already the owner of this home"

400 Bad request: "Exception thrown in Service. User is already a resident of this home"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission AddResidentToHome"

404 Not Found: "Entity not found. Entity not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Agregar un usuario con rol dueño de hogar a un hogar

**/homes/{id}/devices**

**Recurso:** Hogar



**Verbo:** POST

Headers: Authorization: {token}

**Entradas:** {"homeld": string (FromRoute), "DeviceId": string (FromBody)}

Salidas: {"HardwareDeviceId": string}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Exception thrown in Service. User is not a resident of this home"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission AddDeviceToHome"

404 Not Found: "Entity not found. Entity not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Asociar un dispositivo al hogar

**/homes/{id}/residents**

**Recurso:** Hogar

**Verbo:** GET

Headers: Authorization: {token}

**Entradas:** {"homeld": string (FromRoute)}

Salidas: [{"FullName": string, "Email": string, "ProfilePicture": string, "DoesUserMustBeNotified": bool, "Permissions": string[]}]

**Respuestas:**

200 Ok

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission ListHomeResidents"

404 Not Found: "Entity not found. Entity not found" 404 Not Found: "Entity not found. Home not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Listado de miembros de un hogar

**/homes/{id}/hardwareDevices**

**Recurso:** Hogar

**Verbo:** GET

Headers: Authorization: {token}

**Entradas:** {"homeld": string (FromRoute)}

Salidas: [{"DeviceName": string, "DeviceModelNumber": int, "MainPicture": string, "ConnectionState": string}]

**Respuestas:**

200 Ok

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission ListHomeDevices"

404 Not Found: "Entity not found. Entity not found" 404 Not Found: "Entity not found. Home not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Listado de dispositivos en un hogar

**/homes/{id}/residentPermissions**

**Recurso:** Hogar

**Verbo:** PUT

Headers: Authorization: {token}

**Entradas:** {"homelId": string (FromRoute), "UserId": string (FromBody), "HomePermissionName": string (FromBody)}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of userId cannot be null or empty"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission User is not the owner of this home"

404 Not Found: "Entity not found. Entity not found"

404 Not Found: "Entity not found. Home not found"

404 Not Found: "Entity not found. User not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Configurar los permisos de un usuario que es residente de un hogar

**/hardwareDevices/{hardwareDeviceId}**

**Recurso:** Hogar

**Verbo:** PUT

**Entradas:** {"hardwareDeviceId": string (FromRoute), "NewConnectionState": string (FromBody)}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of new connection state cannot be null or empty"

400 Bad request: "Exception thrown in Service. Connection state must only be connected or disconnected"

401 Unauthorized: "You are not authenticated"

404 Not Found: "Entity not found. Entity not found"

404 Not Found: "Entity not found. HardwareDevice not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Cambio del estado de conexión de un dispositivo físico

**/devices/{id}/motionDetectionNotifications**

**Recurso:** Notificación

**Verbo:** POST

**Entradas:** {"hardwareDeviceId": string (FromRoute)}

**Respuestas:**

200 Ok

400 Bad Request: "Exception thrown in Service. Hardware device is not a security camera"

400 Bad Request: "Exception thrown in Service. Hardware device is not connected"

401 Unauthorized: "You are not authenticated"

404 Not Found: "Entity not found. Entity not found"

404 Not Found: "Entity not found. Hardware device not found"

404 Not Found: "Entity not found. Home not found"

404 Not Found: "Entity not found. User not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Ejecución de una notificación por detección de movimiento

**/devices/{id}/personDetectionNotifications**

**Recurso:** Notificación

**Verbo:** POST

**Entradas:** {"hardwareDeviceId": string (FromRoute)}

**Respuestas:**

200 Ok

400 Bad Request: "Exception thrown in Service. Hardware device is not a security camera"

400 Bad Request: "Exception thrown in Service. Hardware device is not connected"

401 Unauthorized: "You are not authenticated"

404 Not Found: "Entity not found. Entity not found"

404 Not Found: "Entity not found. Hardware device not found"

404 Not Found: "Entity not found. Home not found"

404 Not Found: "Entity not found. User not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Ejecución de una notificación por detección de persona

### **/devices/{id}/openingStateNotifications**

**Recurso:** Notificación

**Verbo:** POST

**Entradas:** {"hardwareDeviceId": string (FromRoute), "NewOpeningState": string (FromBody)}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of NewOpeningState cannot be null or empty"

400 Bad Request: "Exception thrown in Service. Hardware device is not connected"

400 Bad Request: "Exception thrown in Service. Device is already in the specified state"

401 Unauthorized: "You are not authenticated"

404 Not Found: "Entity not found. Entity not found"

404 Not Found: "Entity not found. Hardware device not found"

404 Not Found: "Entity not found. Home not found"

404 Not Found: "Entity not found. User not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Ejecución de una notificación por apertura o cierre del dispositivo

### **/users/{id}/notifications**

**Recurso:** Notificación

**Verbo:** GET

Headers: Authorization: {token}

**Entradas:** {"userId": string (FromRoute), "deviceType": string (FromQuery),  
"creationDatetime": DateTime (FromQuery), "hasBeenRead": bool (FromQuery)}

Salidas: [{"Event": string, "HardwareDeviceId": string, "CreationDatetime": DateTime,  
"DeviceName": string, "DeviceModelNumber": string, "HasBeenRead": bool}]

**Respuestas:**

200 Ok

400 Bad Request: "Exception thrown in Service. Argument has been read must be 'Yes', 'No' or 'Empty'"

401 Unauthorized: "You are not authenticated"

403 Forbidden: "Missing permission ListNotifications"

404 Not Found: "Entity not found. Entity not found"

404 Not Found: "Entity not found. User not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Listar notificaciones de un usuario

## **/sessions**

**Recurso:** Sesiones

**Verbo:** POST

**Entradas:** {"Email": string (FromBody), "Password": string (FromBody)}

Salidas: {"Token": string, "User": SessionUserResponse, "Permissions":  
SessionPermissionsResponse[]}

**Respuestas:**

200 Ok

400 Bad request: "Exception thrown in Controller. Request cannot be null"

400 Bad request: "Argument can not be null or empty. Value of email cannot be null or empty"

400 Bad request: "Argument can not be null or empty. Value of password cannot be null or empty"

400 Bad request: "Exception thrown in Service. Invalid password"

401 Unauthorized: "You are not authenticated"

404 Not Found: "Entity not found. Entity not found"

404 Not Found: "Entity not found. User not found"

404 Not Found: "Entity not found. Session not found"

500 Internal server error: "There was an error when processing the request"

**Descripción:** Iniciar sesión

## Referencias Bibliográficas

Notion, “Teórico APIs REST / WebApis,” Sep. 2024. [Online]. Available: <https://diseño-de-aplicaciones-2.notion.site/Te-rico-APIs-REST-WebApis-5dc5617999dc478d9059d938bc22920e>. Accessed on: Oct. 03, 2024.

Universidad ORT Uruguay, “Arquitecturas web y REST”, 2024. [Online]. Available: [https://aulas.ort.edu.uy/pluginfile.php/450652/mod\\_resource/content/3/Arquitecturas%20web%20y%20REST.pdf](https://aulas.ort.edu.uy/pluginfile.php/450652/mod_resource/content/3/Arquitecturas%20web%20y%20REST.pdf). Accessed on: Oct. 8, 2024.

Microsoft, “Best practices for designing REST APIs”, 2024. [Online]. Available: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design>. Accessed on: Oct. 8, 2024.