

Interfaz hardware control de accesos

Esta aplicación levanta un servidor en Node.js con conexión cifrada que recibe peticiones para realizar las diferentes acciones que puede ejecutar el hardware de control de puertas, como por ejemplo abrir una puerta o grabar huellas en los sensores de las puertas.

Accesos

Accesos - Obtener la lista de accesos

GET

/accesos

Success 200

Campo	Tipo	Descripción
Lista	Object[]	Lista de accesos

- Success [#success-examples-Accesos-GetAccesos-2_0_0-0]

```
HTTP/2 200 OK
[
  {
    "id": 1,
    "fechaHora": "2017-01-01T08:00:00.000Z",
    "Puerta": {
      "id": 1,
      "nombre": "P1",
      "estadoInicial": true,
      "estadoActual": true,
      "detalle": "Puerta 1"
    },
    "Persona": {
      "id": 1,
      "nombre": "usuario1"
    }
  }, {
    "id": 2,
    "fechaHora": "2017-01-02T09:00:00.000Z",
    "Puerta": {
      "id": 2,
      "nombre": "P2",
      "estadoInicial": true,
      "estadoActual": true,
      "detalle": "Puerta 2"
    },
    "Persona": {
      "id": 2,
      "nombre": "usuario2"
    }
  }
]
```

Error 4xx

Nombre	Tipo	Descripción
--------	------	-------------

401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }
-----	------	--

- 401 No Autorizado [#error-examples-Accesos-GetAccesos-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Accesos - Obtener los datos de un registro específico de acceso

GET

/accesos/:id

Success 200

Campo	Tipo	Descripción
id	Number	Id de la tabla de historial de accesos
fechaHora	Date	Fecha y hora del acceso
persona	Number	Id de la persona que ingresó
puerta	Number	Id de la puerta que abrió

- Success [#success-examples-Accesos-GetAccesosId-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 1,
  "fechaHora": "2017-01-01T08:00:00.000Z",
  "Puerta": {
    "id": 1,
    "nombre": "P1",
    "estadoInicial": true,
    "estadoActual": true,
    "detalle": "Puerta 1"
  },
  "Persona": {
    "id": 1,
    "nombre": "usuario1"
  }
}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Accesos-GetAccesosId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Arduinos

Arduinos - Actualizar un arduino

PUT

/arduinios/:id

Parámetro

Campo	Tipo	Descripción
mac	String	Dirección de MAC de red del arduino
ip	String	Dirección de IP de red del arduino
control	Boolean	1 si el arduino es de control, 0 si el arduino es una interfaz sensorial
detalle	String	Nombre detallado o ubicación del arduino
pinesSalida	Number[]	Si el arduino es de control se deben especificar todos los pines de salida
clienteMqtt	Number	Id del cliente mqtt relacionado al arduino
puerta	Number	Si el arduino es una interfaz sensorial se debe especificar la puerta relacionada

- Arduino [#parameter-examples-Arduinos-PutArduinosId-2_0_0-0]

```
{
  "mac": "AA:BB:CC:DD:EE:01",
  "ip": "192.168.1.101",
  "control": false,
  "detalle": "Arduino Interfaz Sensorial",
  "pinesSalida": null,
  "clienteMqtt": 3,
  "puerta": 1
}
```

Success 200

Campo	Tipo	Descripción
id	Number	Id del arduino

Campo	Tipo	Descripción
mac	String	Dirección de MAC de red del arduino
ip	String	Dirección de IP de red del arduino
control	Boolean	1 si el arduino es de control, 0 si el arduino es una interfaz sensorial
detalle	String	Nombre detallado o ubicación del arduino
pinesSalida	Number[]	Si el arduino es de control se deben especificar todos los pines de salida
clienteMqtt	Number	Id del cliente mqtt relacionado al arduino
puerta	Number	Si el arduino es una interfaz sensorial se debe especificar la puerta relacionada
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-Arduinos-PutArduinosId-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 2,
  "mac": "AA:BB:CC:DD:EE:01",
  "ip": "192.168.1.101",
  "control": false,
  "detalle": "Arduino Interfaz Sensorial",
  "pinesSalida": null,
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-02T00:00:00.000Z",
  "clienteMqtt": 3,
  "puerta": 1
}
```

Error 4xx

Nombre	Tipo	Descripción
400	json	Consulta errónea HTTP/2 400 { "message": "Validation error", "errors": [{ "field": "PARÁMETRO", "message": "PARÁMETRO must be unique" }] }
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Arduinos-PutArduinosId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Arduinos - Crear un nuevo arduino

POST

/arduinosaurs

Parámetro

Campo	Tipo	Descripción
mac	String	Dirección de MAC de red del arduino
ip	String	Dirección de IP de red del arduino
control	Boolean	1 si el arduino es de control, 0 si el arduino es una interfaz sensorial
detalle	String	Nombre detallado o ubicación del arduino
pinesSalida	Number[]	Si el arduino es de control se deben especificar todos los pines de salida
clienteMqtt	Number	Id del cliente mqtt relacionado al arduino
puerta	Number	Si el arduino es una interfaz sensorial se debe especificar la puerta relacionada

- Arduino [#parameter-examples-Arduinos-PostArduinos-2_0_0-0]

```
{
  "mac": "AA:BB:CC:DD:EE:01",
  "ip": "192.168.1.101",
  "control": false,
  "detalle": "Arduino Interfaz Sensorial",
  "pinesSalida": null,
  "clienteMqtt": 3,
  "puerta": 1
}
```

Success 200

Campo	Tipo	Descripción
id	Number	Id del arduino
mac	String	Dirección de MAC de red del arduino
ip	String	Dirección de IP de red del arduino
control	Boolean	1 si el arduino es de control, 0 si el arduino es una interfaz sensorial
detalle	String	Nombre detallado o ubicación del arduino

Campo	Tipo	Descripción
pinesSalida	Number[]	Si el arduino es de control se deben especificar todos los pines de salida
clienteMqtt	Number	Id del cliente mqtt relacionado al arduino
puerta	Number	Si el arduino es una interfaz sensorial se debe especificar la puerta relacionada
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-Arduinos-PostArduinos-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 2,
  "mac": "AA:BB:CC:DD:EE:01",
  "ip": "192.168.1.101",
  "control": false,
  "detalle": "Arduino Interfaz Sensorial",
  "pinesSalida": null,
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-02T00:00:00.000Z",
  "clienteMqtt": 3,
  "puerta": 1
}
```

Error 4xx

Nombre	Tipo	Descripción
400	json	Consulta errónea HTTP/2 400 { "message": "Validation error", "errors": [{ "field": "PARÁMETRO", "message": "PARÁMETRO must be unique" }] }
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Arduinos-PostArduinos-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Arduinos - Eliminar un arduino

DELETE

/arduinios/:id

Success 200

Campo	Tipo	Descripción
-------	------	-------------

Success	Object	Objeto vacío, elemento eliminado
---------	--------	----------------------------------

- Success [#success-examples-Arduinos-DeleteArduinosId-2_0_0-0]

```
HTTP/2 200 OK
{}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Arduinos-DeleteArduinosId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Arduinos - Obtener la lista de arduinos

GET

/arduinios

Success 200

Campo	Tipo	Descripción
Lista	Object[]	Lista de arduinos

- Success [#success-examples-Arduinos-GetArduinos-2_0_0-0]

```
HTTP/2 200 OK
[
  {
    "id": 1,
    "mac": "AA:BB:CC:DD:EE:00",
    "ip": "192.168.1.100",
    "control": true,
    "detalle": "Arduino Control",
    "pinesSalida": [
      16,
      17
    ],
    "createdAt": "2017-01-01T00:00:00.000Z",
    "updatedAt": "2017-01-02T00:00:00.000Z",
    "clienteMqtt": 2,
    "puerta": null
  },
  {
    "id": 2,
    "mac": "AA:BB:CC:DD:EE:01",
    "ip": "192.168.1.101",
    "control": false,
    "detalle": "Arduino Interfaz Sensorial",
    "pinesSalida": null,
  }
]
```

```
"createdAt": "2017-01-01T00:00:00.000Z",
"updatedAt": "2017-01-02T00:00:00.000Z",
"clienteMqtt": 3,
"puerta": 1
}
]
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Arduinos-GetArduinos-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Arduinos - Obtener los detalles de un arduino

GET

/arduinios/:id

Success 200

Campo	Tipo	Descripción
id	Number	Id del arduino
mac	String	Dirección de MAC de red del arduino
ip	String	Dirección de IP de red del arduino
control	Boolean	1 si el arduino es de control, 0 si el arduino es una interfaz sensorial
detalle	String	Nombre detallado o ubicación del arduino
pinsSalida	Number[]	Si el arduino es de control se deben especificar todos los pines de salida
clienteMqtt	Number	Id del cliente mqtt relacionado al arduino
puerta	Number	Si el arduino es una interfaz sensorial se debe especificar la puerta relacionada
createdAt	Date	Fecha de creación del registro

Campo	Tipo	Descripción
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-Arduinos-GetArduinosId-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 2,
  "mac": "AA:BB:CC:DD:EE:01",
  "ip": "192.168.1.101",
  "control": false,
  "detalle": "Arduino Interfaz Sensorial",
  "pinesSalida": null,
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-02T00:00:00.000Z",
  "clienteMqtt": 3,
  "puerta": 1
}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Arduinos-GetArduinosId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Autenticacion

Autenticacion - Obtener un token para las consultas

POST

/autenticar

Parámetro

Campo	Tipo	Descripción
usuario	String	Nombre de usuario de LDAP
clave	String	Contraseña para el usuario de LDAP

- Ejemplo [#parameter-examples-Autenticacion-PostAutenticar-2_0_0-0]

```
{
  "usuario": "pepito",
  "clave": "grillo"
}
```

Success 200

Campo	Tipo	Descripción
mensaje	String	Mensaje de grabación correcta

- Success [#success-examples-Autenticacion-PostAutenticar-2_0_0-0]

```
HTTP/2 200 OK
{
  "token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vhcm91IjoicGVwaXRvIiwicm9sZSI6InVzZXIiLCJpYXQiOiE0OTMwNjE4MTEsI
}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "error": "Usuario o contraseña incorrecta" }

- 401 No Autorizado [#error-examples-Autenticacion-PostAutenticar-2_0_0-0]

```
HTTP/2 401
{
  "error": "Usuario o contraseña incorrecta"
}
```

CientesMqtt

CientesMqtt - Actualizar un Cientes Mqtt

PUT

/clientesMqtt/:id

Parámetro

Campo	Tipo	Descripción
usuario	String	Nombre de usuario del cliente mqtt
clave	String	Clave para el nuevo usuario del cliente mqtt
admin	Boolean	1 si el cliente es administrador, 0 si el cliente no es administrador

- Cliente [#parameter-examples-CientesMqtt-PutClientesmqttId-2_0_0-0]

```
{
  "usuario": "actualizacionCliente",
  "clave": "actualizacionClave",
  "admin": true
}
```

Success 200

Campo	Tipo	Descripción
id	Number	Id del cliente mqtt
usuario	String	Nombre de usuario del cliente mqtt
conectado	Boolean	1 si el cliente está conectado, 0 si el cliente está desconectado
admin	Boolean	1 si el cliente es administrador, 0 si el cliente no es administrador
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-CientesMqtt-PutClientesmqttId-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 1,
  "usuario": "admin",
  "conectado": false,
  "admin": true,
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-02T00:00:00.000Z"
}
```

Error 4xx

Nombre	Tipo	Descripción
400	json	Consulta errónea HTTP/2 400 { "message": "Validation error", "errors": [{ "field": "PARÁMETRO", "message": "PARÁMETRO must be unique" }] }
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-CientesMqtt-PutClientesmqttId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

ClientesMqtt - Crear un nuevo Clientes Mqtt

POST

/clientesMqtt

Parámetro

Campo	Tipo	Descripción
usuario	String	Nombre de usuario del cliente mqtt
clave	String	Clave para el nuevo usuario del cliente mqtt
admin	Boolean	1 si el cliente es administrador, 0 si el cliente no es administrador

- Cliente [#parameter-examples-CientesMqtt-PostClientesmqtt-2_0_0-0]

```
{
  "usuario": "nuevoCliente",
  "clave": "nuevaClave",
  "admin": false
}
```

Success 200

Campo	Tipo	Descripción
id	Number	Id del cliente mqtt
usuario	String	Nombre de usuario del cliente mqtt
conectado	Boolean	1 si el cliente está conectado, 0 si el cliente está desconectado
admin	Boolean	1 si el cliente es administrador, 0 si el cliente no es administrador
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-CientesMqtt-PostClientesmqtt-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 1,
  "usuario": "admin",
  "conectado": false,
  "admin": true,
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-01T00:00:00.000Z"
}
```

Error 4xx

Nombre	Tipo	Descripción
--------	------	-------------

400	json	Consulta errónea HTTP/2 400 { "message": "Validation error", "errors": [{ "field": "PARÁMETRO", "message": "PARÁMETRO must be unique" }] }
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-ClientesMqtt-PostClientesmqtt-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

ClientesMqtt - Eliminar un Clientes Mqtt

DELETE

```
/clientesMqtt/:id
```

Success 200

Campo	Tipo	Descripción
Success	Object	Objeto vacío, elemento eliminado

- Success [#success-examples-ClientesMqtt-DeleteClientesmqttId-2_0_0-0]

```
HTTP/2 200 OK
{}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-ClientesMqtt-DeleteClientesmqttId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

ClientesMqtt - Obtener la lista de Clientes Mqtt

GET

```
/clientesMqtt
```

Success 200

Campo	Tipo	Descripción
Lista	Object[]	Lista de Clientes Mqtt

- Success [#success-examples-ClientesMqtt-GetClientesmqtt-2_0_0-0]

```
HTTP/2 200 OK
[
  {
    "id": 1,
    "usuario": "admin",
    "conectado": false,
    "admin": true,
    "createdAt": "2017-01-01T00:00:00.000Z",
    "updatedAt": "2017-01-01T00:00:00.000Z"
  }
]
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-ClientesMqtt-GetClientesmqtt-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

ClientesMqtt - Obtener los detalles de un Clientes Mqtt

GET

```
/clientesMqtt/:id
```

Success 200

Campo	Tipo	Descripción
id	Number	Id del cliente mqtt
usuario	String	Nombre de usuario del cliente mqtt
conectado	Boolean	1 si el cliente está conectado, 0 si el cliente está desconectado
admin	Boolean	1 si el cliente es administrador, 0 si el cliente no es administrador
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-CientesMqtt-GetClientesmqttId-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 1,
  "usuario": "admin",
  "conectado": false,
  "admin": true,
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-01T00:00:00.000Z"
}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-CientesMqtt-GetClientesmqttId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Huellas

Huellas - Borrar huellas

PUT

/huellas/borrar

Parámetro

Campo	Tipo	Descripción
destino	Number	ID del sensor en el cual se grabará la o las nuevas huellas, "0" en caso de grabar a todos los sensores
huellas	Number[]	ID o IDs de las huellas que se desean borrar

- Ejemplo [#parameter-examples-Huellas-PutHuellasBorrar-2_0_0-0]

```
{
  "destino": 0,
  "huellas": [
    1,
    21,
    45
  ]
}
```

Success 200

Campo	Tipo	Descripción
mensaje	String	Mensaje de grabación correcta

- Success [#success-examples-Huellas-PutHuellasBorrar-2_0_0-0]

```
HTTP/1.1 200 OK
{
  "mensaje": "Eliminando huellas 1,2"
}
```

Error 4xx

Nombre	Tipo	Descripción
409	json	Servidor ocupado HTTP/1.1 409 Conflict { "error": "Servidor ocupado" }

- 409 Servidor ocupado [#error-examples-Huellas-PutHuellasBorrar-2_0_0-0]

```
HTTP/1.1 409 Conflict
{
  "error": "Servidor ocupado"
}
```

Huellas - Enviar huellas a los sensores

PUT

```
/huellas/enviar
```

Parámetro

Campo	Tipo	Descripción
destino	Number	ID del sensor en el cual se grabará la o las nuevas huellas, "0" en caso de grabar a todos los sensores
huellas	Number[]	ID o IDs de las huellas que se desean enviar

- Ejemplo [#parameter-examples-Huellas-PutHuellasEnviar-2_0_0-0]

```
{
  "destino": 0,
  "huellas": [
    1,
    21,
    45
  ]
}
```

Success 200

Campo	Tipo	Descripción
mensaje	String	Mensaje de grabación correcta

- Success [#success-examples-Huellas-PutHuellasEnviar-2_0_0-0]

```
HTTP/1.1 200 OK
{
  "mensaje": "Grabando huellas 1,2"
}
```

Error 4xx

Nombre	Tipo	Descripción
409	json	Servidor ocupado HTTP/1.1 409 Conflict { "error": "Servidor ocupado" }

- 409 Servidor ocupado [#error-examples-Huellas-PutHuellasEnviar-2_0_0-0]

```
HTTP/1.1 409 Conflict
{
  "error": "Servidor ocupado"
}
```

Huellas - Envía la imagen de la huella seleccionada

GET

/huellas/:id

Parámetro

Campo	Tipo	Descripción
id	Number	ID de la persona de la que se desea recuperar la imagen

Error 4xx

Nombre	Tipo	Descripción
500	json	Error interno HTTP/2 500 { "error": true, "mensaje": "Conexión perdida con el sensor" }

- 500 Error interno [#error-examples-Huellas-GetHuellasId-2_0_0-0]

```
HTTP/2 500
{
  "error": true,
  "mensaje": "Error interno del servidor"
}
```

Huellas - Verifica si existe la imagen de una huella

GET

/huellas/:id

Parámetro

Campo	Tipo	Descripción
-------	------	-------------

id	Number	ID de la persona de la que se desea recuperar la imagen
----	--------	---

Success 200

Campo	Tipo	Descripción
200	json	Exitoso

Error 4xx

Nombre	Tipo	Descripción
404	json	Huella no existente
500	json	Error interno

Huellas - Grabar una nueva huella o actualizar una existente

POST

/huellas/grabar

Parámetro

Campo	Tipo	Descripción
id	Number	ID de la persona que se desea grabar sus huellas

- Ejemplo [#parameter-examples-Huellas-PostHuellasGrabar-2_0_0-0]

```
{
  "id": 2
}
```

Success 200

Campo	Tipo	Descripción
error	Boolean	Estado de error
mensaje	Object	Mensaje de respuesta

- Success [#success-examples-Huellas-PostHuellasGrabar-2_0_0-0]

```
HTTP/2 200 OK
{
  "error": false,
  "mensaje": "Huella de usuario jdoe actualizada"
}
```

Error 4xx

Nombre	Tipo	Descripción
--------	------	-------------

500	json	Error interno HTTP/2 500 { "error": true, "mensaje": "Conexión perdida con el sensor" }
-----	------	---

- 401 No Autorizado [#error-examples-Huellas-PostHuellasGrabar-2_0_0-0]

```
HTTP/2 500
{
  "error": true,
  "mensaje": "No se detectó ninguna huella"
}
```

MonitorMicroservicio

MonitorMicroservicio - Sincronizar manualmente ID's y usuarios desde el microservicio

GET

```
/monitorMicroservicio/sincronizar
```

Success 200

Campo	Tipo	Descripción
mensaje	String	Mensaje de disponibilidad

- Success [#success-examples-MonitorMicroservicio-GetMonitormicroservicioSincronizar-2_0_0-0]

```
HTTP/1.1 200 OK
{
  "mensaje": "Sincronizando"
}
```

Error 4xx

Nombre	Tipo	Descripción
409	json	Servidor ocupado HTTP/1.1 409 Conflict { "error": "Servidor ocupado" }

- 409 Servidor ocupado [#error-examples-MonitorMicroservicio-GetMonitormicroservicioSincronizar-2_0_0-0]

```
HTTP/1.1 409 Conflict
{
  "error": "Servidor ocupado"
}
```

PermisosAcceso

PermisosAcceso - Actualizar un permiso de acceso

PUT

/permisosAcceso/:id

- PermisosAcceso [#parameter-examples-PermisosAcceso-PutPermisosaccesold-2_0_0-0]

```
{
  "fechaInicio": "2017-01-01T04:00:00.000Z",
  "fechaFin": "2017-01-02T04:00:00.000Z",
  "persona": 2,
  "puerta": 2
}
```

Success 200

Campo	Tipo	Descripción
fechaInicio	Date	Fecha de inicio del permiso de acceso
fechaFin	Date	Fecha final del permiso de acceso, NULL si el permiso es indefinido
persona	Number	Id de la persona a la que se brinda el acceso
puerta	Number	Id de la puerta a la que se brinda el acceso
id	Number	Id del permiso de acceso

- Success [#success-examples-PermisosAcceso-PutPermisosaccesold-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 2,
  "fechaInicio": "2017-01-01T04:00:00.000Z",
  "fechaFin": "2017-01-02T04:00:00.000Z",
  "persona": 2,
  "puerta": 2
}
```

Error 4xx

Nombre	Tipo	Descripción
400	json	Consulta errónea HTTP/2 400 { "message": "Validation error", "errors": [{ "field": "PARÁMETRO", "message": "PARÁMETRO must be unique" }] }
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-PermisosAcceso-PutPermisosaccesold-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

PermisosAcceso - Crear un nuevo permiso de acceso

POST

/permisosAcceso

- PermisosAcceso [#parameter-examples-PermisosAcceso-PostPermisosacceso-2_0_0-0]

```
{  "fechaInicio": "2017-01-01T04:00:00.000Z",  "fechaFin": "2017-01-02T04:00:00.000Z",  "persona": 2,  "puerta": 2}
```

Success 200

Campo	Tipo	Descripción
fechaInicio	Date	Fecha de inicio del permiso de acceso
fechaFin	Date	Fecha final del permiso de acceso, NULL si el permiso es indefinido
persona	Number	Id de la persona a la que se brinda el acceso
puerta	Number	Id de la puerta a la que se brinda el acceso
id	Number	Id del permiso de acceso

- Success [#success-examples-PermisosAcceso-PostPermisosacceso-2_0_0-0]

```
HTTP/2 200 OK
{  "id": 2,  "fechaInicio": "2017-01-01T04:00:00.000Z",  "fechaFin": "2017-01-02T04:00:00.000Z",  "persona": 2,  "puerta": 2}
```

Error 4xx

Nombre	Tipo	Descripción
400	json	Consulta errónea HTTP/2 400 { "message": "notNull Violation: PARÁMETRO cannot be null", "errors": [{ "field": "PARÁMETRO", "message": "PARÁMETRO cannot be null" }] }
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-PermisosAcceso-PostPermisosacceso-2_0_0-0]

```
HTTP/2 401
{
```

```
"code": "InvalidCredentials",
"message": "No authorization token was found"
}
```

PermisosAcceso - Eliminar un permiso de acceso

DELETE

/permisosAcceso/:id

Success 200

Campo	Tipo	Descripción
Success	Object	Objeto vacío, elemento eliminado

- Success [#success-examples-PermisosAcceso-DeletePermisosaccesold-2_0_0-0]

```
HTTP/2 200 OK
{}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-PermisosAcceso-DeletePermisosaccesold-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

PermisosAcceso - Obtener la lista de permisos de acceso indefinido

GET

/permisosAccesoIndefinido

Success 200

Campo	Tipo	Descripción
Lista	Object[]	Lista de accesos indefinidos

- Success [#success-examples-PermisosAcceso-GetPermisosaccesoindefinido-2_0_0-0]

```
HTTP/2 200 OK
[
  {
    "persona": "1",
    "nombre": "usuario1",
    "puertas": [
```

```
[
  {
    "puerta": 1,
    "permiso": true
  },
  {
    "puerta": 2,
    "permiso": false
  },
  {
    "puerta": 3,
    "permiso": false
  },
  {
    "puerta": 4,
    "permiso": false
  }
],
{
  "persona": "2",
  "nombre": "usuario2",
  "puertas": [
    {
      "puerta": 1,
      "permiso": true
    },
    {
      "puerta": 2,
      "permiso": true
    },
    {
      "puerta": 3,
      "permiso": false
    },
    {
      "puerta": 4,
      "permiso": true
    }
  ]
}
]
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-PermisosAcceso-GetPermisosaccesoindefinido-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

PermisosAcceso - Obtener la lista de permisos de acceso

GET

/permisosAcceso

Success 200

Campo	Tipo	Descripción
Lista	Object[]	Lista de permisos de acceso

- Success [#success-examples-PermisosAcceso-GetPermisosacceso-2_0_0-0]

```
HTTP/2 200 OK
[
  {
    "id": 1,
    "fechaInicio": "2017-01-01T04:00:00.000Z",
    "fechaFin": null
    "Puerta": {
      "id": 1,
      "nombre": "P1",
      "pin": 0,
      "estadoInicial": true,
      "estadoActual": true,
      "detalle": "Puerta 1",
      "arduinoControl": 1
    },
    "Persona": {
      "id": 1,
      "nombre": "usuario1",
      "grabado": true
    }
  }, {
    "id": 2,
    "fechaInicio": "2017-01-01T04:00:00.000Z",
    "fechaFin": "2017-01-02T04:00:00.000Z",
    "Puerta": {
      "id": 2,
      "nombre": "P2",
      "pin": 1,
      "estadoInicial": true,
      "estadoActual": true,
      "detalle": "Puerta 2",
      "arduinoControl": 2
    },
    "Persona": {
      "id": 2,
      "nombre": "usuario2",
      "grabado": true
    }
  }
]
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-PermisosAcceso-GetPermisosacceso-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

PermisosAcceso - Obtener los datos de un permiso de acceso

GET

/permisosAcceso/:id

Success 200

Campo	Tipo	Descripción
id	Number	Id del permiso de acceso
fechaInicio	Date	Fecha de inicio del permiso de acceso
fechaFin	Date	Fecha final del permiso de acceso, NULL si el permiso es indefinido
persona	Number	Id de la persona a la que se brinda el acceso
puerta	Number	Id de la puerta a la que se brinda el acceso

- Success [#success-examples-PermisosAcceso-GetPermisosaccesold-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 1,
  "fechaInicio": "2017-01-01T04:00:00.000Z",
  "fechaFin": null
  "Puerta": {
    "id": 1,
    "nombre": "P1",
    "pin": 0,
    "estadoInicial": true,
    "estadoActual": true,
    "detalle": "Puerta 1",
    "arduinoControl": 1
  },
  "Persona": {
    "id": 1,
    "nombre": "usuario1",
    "grabado": true
  }
}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-PermisosAcceso-GetPermisosaccesold-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Personas

Personas - Obtener la lista de personas

GET

/personas

Success 200

Campo	Tipo	Descripción
Lista	Object[]	Lista de personas

- Success [#success-examples-Personas-GetPersonas-2_0_0-0]

```
HTTP/2 200 OK
[
  {
    "id": 1,
    "nombre": "usuario1",
    "grabado": false,
    "createdAt": "2017-01-01T00:00:00.000Z",
    "updatedAt": "2017-01-01T00:00:00.000Z"
  },
  {
    "id": 2,
    "nombre": "usuario2",
    "grabado": true,
    "createdAt": "2017-01-01T00:00:00.000Z",
    "updatedAt": "2017-01-01T00:00:00.000Z"
  }
]
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Personas-GetPersonas-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Personas - Obtener los datos de una persona en específico

GET

/personas/:id

Success 200

Campo	Tipo	Descripción
-------	------	-------------

id	Number	Id de la persona
nombre	Number	Nombre de usuario de LDAP
grabado	Number	1 si la huella se há grabado en los sensores, 0 si la huella aún no se há grabado en los sensores
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-Personas-GetPersonasId-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 1,
  "nombre": "usuario1",
  "grabado": false,
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-01T00:00:00.000Z"
}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Personas-GetPersonasId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Programas

Programas - Arduino GPIO

GET

```
/programas/:arduinoId
```

Parámetro

Campo	Tipo	Descripción
arduinoId	Number	ID del arduino de control de puertas

- Success [#success-examples-Programas-GetProgramasArduinoid-2_0_0-0]

```
HTTP/1.1 200 OK
#include <SPI.h>
#include <UIPEthernet.h>
...
void setup() {
...
void loop() {
...
}
```

Error 4xx

Nombre	Tipo	Descripción
403	json	Datos erróneos HTTP/1.1 409 Forbidden { "error": "Datos erróneos" }
500	json	Error interno HTTP/1.1 500 Internal Error { "error": "Error interno del servidor" }

- 409 Servidor ocupado [#error-examples-Programas-GetProgramasArduinoid-2_0_0-0]

```
HTTP/1.1 409 Conflict
{
  "error": "Datos erróneos"
}
```

Puertas

Puertas - Abrir una puerta mediante el usuario conectado

POST

/puerta/abrirPuerta

Parámetro

Campo	Tipo	Descripción
puerta	Number	ID de la puerta que se desea abrir

- Ejemplo [#parameter-examples-Puertas-PostPuertaAbrirpuerta-2_0_0-0]

```
{
  "puerta": 1
}
```

Success 200

Campo	Tipo	Descripción
mensaje	String	Mensaje de grabación correcta

- Success [#success-examples-Puertas-PostPuertaAbrirpuerta-2_0_0-0]

```
HTTP/2 200 OK
{
  "mensaje": "Abriendo puerta 4"
}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "error": "Usuario 1 no autorizado" }

- 401 No Autorizado [#error-examples-Puertas-PostPuertaAbrirpuerta-2_0_0-0]

```
HTTP/2 401
{
  "error": "Usuario 1 no autorizado"
}
```

Puertas - Actualizar una puerta

PUT

/puertas/:id

Parámetro

Campo	Tipo	Descripción
nombre	String	Nombre código de la puerta
pin	Number	Pin del arduino de control al cual se conecta la puerta
estadoInicial	Boolean	1 si el estado inicial debe ser alto, 0 si el estado inicial debe ser bajo
estadoActual	Boolean	1 si se encuentra en estado alto, 0 si se encuentra en estado bajo
detalle	String	Detalle o ubicación de la puerta
arduinoControl	Number	Id del arduino de control que activa o desactiva esta puerta

- Puerta [#parameter-examples-Puertas-PutPuertasId-2_0_0-0]

```
{
  "nombre": "PNU",
  "pin": 10,
  "estadoInicial": true,
  "estadoActual": true,
  "detalle": "Puerta nueva",
  "arduinoControl": 1
}
```

Success 200

Campo	Tipo	Descripción
-------	------	-------------

id	Number	Id de la puerta
nombre	String	Nombre código de la puerta
pin	Number	Pin del arduino de control al cual se conecta la puerta
estadoInicial	Boolean	1 si el estado inicial debe ser alto, 0 si el estado inicial debe ser bajo
estadoActual	Boolean	1 si se encuentra en estado alto, 0 si se encuentra en estado bajo
detalle	String	Detalle o ubicación de la puerta
arduinoControl	Number	Id del arduino de control que activa o desactiva esta puerta
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-Puertas-PutPuertasId-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 1,
  "nombre": "PNU",
  "pin": 10,
  "estadoInicial": true,
  "estadoActual": true,
  "detalle": "Puerta nueva",
  "arduinoControl": 1
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-02T00:00:00.000Z"
}
```

Error 4xx

Nombre	Tipo	Descripción
400	json	Consulta errónea HTTP/2 400 { "message": "Validation error", "errors": [{ "field": "PARÁMETRO", "message": "PARÁMETRO must be unique" }] }
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Puertas-PutPuertasId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Puertas - Crear una nueva puerta

POST

/puertas

Parámetro

Campo	Tipo	Descripción
nombre	String	Nombre código de la puerta
pin	Number	Pin del arduino de control al cual se conecta la puerta
estadoInicial	Boolean	1 si el estado inicial debe ser alto, 0 si el estado inicial debe ser bajo
estadoActual	Boolean	1 si se encuentra en estado alto, 0 si se encuentra en estado bajo
detalle	String	Detalle o ubicación de la puerta
arduinoControl	Number	Id del arduino de control que activa o desactiva esta puerta

- Puerta [#parameter-examples-Puertas-PostPuertas-2_0_0-0]

```
{
  "nombre": "PNU",
  "pin": 10,
  "estadoInicial": true,
  "estadoActual": true,
  "detalle": "Puerta nueva",
  "arduinoControl": 1
}
```

Success 200

Campo	Tipo	Descripción
id	Number	Id de la puerta
nombre	String	Nombre código de la puerta
pin	Number	Pin del arduino de control al cual se conecta la puerta
estadoInicial	Boolean	1 si el estado inicial debe ser alto, 0 si el estado inicial debe ser bajo
estadoActual	Boolean	1 si se encuentra en estado alto, 0 si se encuentra en estado bajo
detalle	String	Detalle o ubicación de la puerta
arduinoControl	Number	Id del arduino de control que activa o desactiva esta puerta

Campo	Tipo	Descripción
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-Puertas-PostPuertas-2_0_0-0]

```
HTTP/2 200 OK
{
  "nombre": "PNU",
  "pin": 10,
  "estadoInicial": true,
  "estadoActual": true,
  "detalle": "Puerta nueva",
  "arduinoControl": 1
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-01T00:00:00.000Z"
}
```

Error 4xx

Nombre	Tipo	Descripción
400	json	Consulta errónea HTTP/2 400 { "message": "Validation error", "errors": [{ "field": "PARÁMETRO", "message": "PARÁMETRO must be unique" }] }
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Puertas-PostPuertas-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Puertas - Eliminar una puerta

DELETE

```
/puertas/:id
```

Success 200

Campo	Tipo	Descripción
Success	Object	Objeto vacío, elemento eliminado

- Success [#success-examples-Puertas-DeletePuertasId-2_0_0-0]

```
HTTP/2 200 OK
{ }
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Puertas-DeletePuertasId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Puertas - Obtener la lista de puertas

GET

/puertas

Success 200

Campo	Tipo	Descripción
Lista	Object[]	Lista de puertas

- Success [#success-examples-Puertas-GetPuertas-2_0_0-0]

```
HTTP/2 200 OK
[
  {
    "id": 1,
    "nombre": "P1",
    "pin": 0,
    "estadoInicial": true,
    "estadoActual": true,
    "detalle": "Puerta 1",
    "createdAt": "2017-01-01T00:00:00.000Z",
    "updatedAt": "2017-01-02T00:00:00.000Z"
    "arduinoControl": 1
  },
  {
    "id": 2,
    "nombre": "P2",
    "pin": 1,
    "estadoInicial": true,
    "estadoActual": true,
    "detalle": "Puerta madera derecha",
    "createdAt": "2017-01-01T00:00:00.000Z",
    "updatedAt": "2017-01-02T00:00:00.000Z"
    "arduinoControl": 1
  }
]
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Puertas-GetPuertas-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Puertas - Obtener los detalles de una puerta

GET

/puertas/:id

Success 200

Campo	Tipo	Descripción
id	Number	Id de la puerta
nombre	String	Nombre código de la puerta
pin	Number	Pin del arduino de control al cual se conecta la puerta
estadoInicial	Boolean	1 si el estado inicial debe ser alto, 0 si el estado inicial debe ser bajo
estadoActual	Boolean	1 si se encuentra en estado alto, 0 si se encuentra en estado bajo
detalle	String	Detalle o ubicación de la puerta
arduinoControl	Number	Id del arduino de control que activa o desactiva esta puerta
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-Puertas-GetPuertasId-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 1,
  "nombre": "P1",
  "pin": 0,
  "estadoInicial": true,
  "estadoActual": true,
  "detalle": "Puerta 1",
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-02T00:00:00.000Z"
  "arduinoControl": 1
}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Puertas-GetPuertasId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Respuestas

Respuestas - Obtener la lista de acciones y respuestas de los sensores

GET

```
/respuestas
```

Success 200

Campo	Tipo	Descripción
Lista	Object[]	Lista de acciones y respuestas

- Success [#success-examples-Respuestas-GetRespuestas-2_0_0-0]

```
HTTP/2 200 OK
[
  {
    "id": 1,
    "comando": "Enviando huella 1",
    "respuesta": null,
    "createdAt": "2017-01-01T00:00:00.000Z",
    "updatedAt": "2017-01-01T00:00:00.000Z",
    "arduino": null
  },
  {
    "id": 2,
    "comando": null,
    "respuesta": "Proceso ejecutado con exito",
    "createdAt": "2017-01-01T00:00:00.000Z",
    "updatedAt": "2017-01-01T00:00:00.000Z",
    "arduino": 2
  }
]
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Respuestas-GetRespuestas-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
  "message": "No authorization token was found"
}
```

Respuestas - Obtener los datos de una respuesta o acción específica

GET

/respuestas/:id

Success 200

Campo	Tipo	Descripción
id	Number	Id de la acción o la respuesta
comando	String	Comando enviado a los sensores, Null si se está recibiendo una respuesta
respuesta	String	Respuesta recibida de los sensores, Null si se está enviando una acción
arduino	Number	Id del arduino que responde, Null si se está enviando una acción
createdAt	Date	Fecha de creación del registro
updatedAt	Date	Fecha de alteración del registro

- Success [#success-examples-Respuestas-GetRespuestasId-2_0_0-0]

```
HTTP/2 200 OK
{
  "id": 1,
  "comando": "Enviando huella 1",
  "respuesta": null,
  "createdAt": "2017-01-01T00:00:00.000Z",
  "updatedAt": "2017-01-01T00:00:00.000Z",
  "arduino": null
}
```

Error 4xx

Nombre	Tipo	Descripción
401	json	No Autorizado HTTP/2 401 { "code": "InvalidCredentials", "message": "No authorization token was found" }

- 401 No Autorizado [#error-examples-Respuestas-GetRespuestasId-2_0_0-0]

```
HTTP/2 401
{
  "code": "InvalidCredentials",
```

```
"message": "No authorization token was found"
}
```
