



Instituto Politécnico Nacional
Escuela Superior de Cómputo
Licenciatura en Ciencia de Datos



Unidad de Aprendizaje: Desarrollo de Aplicaciones Web

Profesor: Efren Cruz Carlin

Proyecto: Entry security system
Documentación de APIs

Equipo 2

Aguilar Hernández Rodrigo Javier

González Chacón Mónica

Osorio García Gabriel

Luna Ruíz Luis Ernesto

4AM1

Fecha: 11 de enero de 2024

EndPoints Administrativo/datos:

- **HttpGet:**

Permite obtener una lista de todos los usuarios de tipo administrativo registrados en la base de datos, por lo tanto no requiere enviarse nada dentro de la petición.

Devuelve un objeto de tipo JsonResult que contiene la lista de los administrativos con su información correspondiente. Ejemplo:

```
[
  {
    "idAdmin": "A2022631269",
    "password": "contra3698",
    "name": "Alberto",
    "apePat": "Mendoza",
    "apeMat": "Hernández",
    "photo": "https://historiadeltiempopresente.com/uploads/20/01/userdefaultimage.jpg",
    "departamentoAdmin": "Gestión escolar",
    "correoAdmin": "albert23@gmail.com"
  }
]
```

- **HttpPost:**

Permite insertar en la base de datos un nuevo usuario de tipo administrativo, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idAdmin": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string",
  "departamentoAdmin": "string",
  "correoAdmin": "string"
}
```

Devuelve una respuesta de tipo bool, en donde es True si se guardaron los datos correctamente en la base de datos y False si no se guardaron.

- **HttpPatch:**

Permite actualizar el registro de un usuario de tipo administrativo que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idAdmin": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",

```

```

    "apeMat": "string",
    "photo": "string",
    "departamentoAdmin": "string",
    "correoAdmin": "string"
}

```

Es importante que el "idAdmin" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y se pueda modificar con la nueva información.

Devuelve una respuesta de tipo bool, la cual es True si se actualizaron correctamente los datos y False si no se encontró el registro con el "idAdmin" de la solicitud.

- **HttpDelete:**

Permite eliminar el registro de un usuario de tipo administrativo que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```

{
    "idAdmin": "string",
    "password": "string",
    "name": "string",
    "apePat": "string",
    "apeMat": "string",
    "photo": "string",
    "departamentoAdmin": "string",
    "correoAdmin": "string"
}

```

Es importante que el "idAdmin" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y eliminarlo.

Devuelve una respuesta de tipo bool, la cual es True si se eliminó el registro del usuario de tipo administrativo correctamente y False si no se encontró el registro con el "idAdmin" de la solicitud.

- **HttpGet("{id}"):**

Permite obtener cierta información de un usuario de tipo administrativo en específico que se encuentre registrado en la base de datos, por lo tanto la ruta de la solicitud debe de contener un parámetro llamado "id".

En caso de que el identificador enviado en la url de la solicitud se encuentre dentro de la base de datos, se devuelve un objeto de tipo JsonResult que contiene la información del administrativo, a excepción de su identificador. Ejemplo:

```

{
    "password": "contra3698",
    "name": "Alberto",
    "apePat": "Mendoza",
    "apeMat": "Hernández",
    "photo": "https://historiadeltiempopresente.com/uploads/20/01/userdefaultimage.jpg",
    "departamentoAdmin": "Gestión escolar",
}

```

```
        "correoAdmin": "albert23@gmail.com"
    }
}
```

En caso contrario se devuelve un objeto JsonResult con el mensaje: "No se ha encontrado un registro de administrativo con el ID {id}".

Endpoints Alumno/datos:

- **HttpGet:**

Permite obtener una lista de todos los usuarios de tipo alumno registrados en la base de datos, por lo tanto no requiere enviarse nada dentro de la petición.

Devuelve un objeto de tipo JsonResult que contiene la lista de los alumnos con su información correspondiente. Ejemplo:

```
[
    {
        "boleta": "2022630751",
        "password": "contra_01",
        "name": "Daniel",
        "apePat": "Camacho",
        "apeMat": "García",
        "photo": "https://historiadeltiempopresente.com/uploads/20/01/userdefaultimage.jpg",
        "estado": "Inscrito"
    }
]
```

- **HttpPost:**

Permite insertar en la base de datos un nuevo usuario de tipo alumno, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
    "boleta": "string",
    "password": "string",
    "name": "string",
    "apePat": "string",
    "apeMat": "string",
    "photo": "string",
    "estado": "string"
}
```

Devuelve una respuesta de tipo bool, en donde es True si se guardaron los datos correctamente en la base de datos y False si no se guardaron.

- **HttpPatch:**

Permite actualizar el registro de un usuario de tipo alumno que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
```

```

    "boleta": "string",
    "password": "string",
    "name": "string",
    "apePat": "string",
    "apeMat": "string",
    "photo": "string",
    "estado": "string"
}

```

Es importante que la “boleta” que se envía en la solicitud sea correcta, ya que este valor es el que se utiliza para evaluar que el registro exista y se pueda modificar con la nueva información.

Devuelve una respuesta de tipo bool, la cual es True si se actualizaron correctamente los datos y False si no se encontró el registro con la “boleta” de la solicitud.

- **HttpDelete:**

Permite eliminar el registro de un usuario de tipo alumno que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```

{
    "boleta": "string",
    "password": "string",
    "name": "string",
    "apePat": "string",
    "apeMat": "string",
    "photo": "string",
    "estado": "string"
}

```

Es importante que la “boleta” que se envía en la solicitud sea correcta, ya que este valor es el que se utiliza para evaluar que el registro exista y eliminarlo.

Devuelve una respuesta de tipo bool, la cual es True si se eliminó el registro del usuario de tipo alumno correctamente y False si no se encontró el registro con la “boleta” de la solicitud.

- **HttpGet("{id}"):**

Permite obtener cierta información de un usuario de tipo alumno en específico que se encuentre registrado en la base de datos, por lo tanto la ruta de la solicitud debe de contener un parámetro llamado “id”.

En caso de que el identificador enviado en la url de la solicitud se encuentre dentro de la base de datos, se devuelve un objeto de tipo JsonResult que contiene la información del alumno, a excepción de su identificador y estado. Ejemplo:

```

{
    "password": "contra_01",
    "name": "Daniel",
    "apePat": "Camacho",
    "apeMat": "García",
    "photo": "https://historiadeltiempopresente.com/uploads/20/01/userdefaultimage.jpg"
}

```

```
}
```

En caso contrario se devuelve un objeto JsonResult con el mensaje: “No se ha encontrado un registro de alumno con el ID {id}”.

EndPoints Cafeteria/datos:

- **HttpGet:**

Permite obtener una lista de todos los usuarios de tipo cafetería registrados en la base de datos, por lo tanto no requiere enviarse nada dentro de la petición.

Devuelve un objeto de tipo JsonResult que contiene la lista de las cafeterías con su información correspondiente. Ejemplo:

```
[
  {
    "idCafe": "C202200001",
    "password": "contra123",
    "name": "Luis",
    "apePat": " Sanchez",
    "apeMat": "Cruz ",
    "photo": "https://historiadeltiempopresente.com/uploads/20/01/cafe_default_image.jpg",
    "correoCafe": "luis.sancruz@gmail.com"
  }
]
```

- **HttpPost:**

Permite insertar en la base de datos una nueva cafetería, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idCafe": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string",
  "correoCafe": "string"
}
```

Devuelve una respuesta de tipo bool, donde es True si se guardaron los datos correctamente en la base de datos y False si no se guardaron.

- **HttpPatch:**

Permite actualizar el registro de una cafetería que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idCafe": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
}
```

```

    "apeMat": "string",
    "photo": "string",
    "correoCafe": "string"
}

```

Es importante que el "idCafe" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y se pueda modificar con la nueva información.

Devuelve una respuesta de tipo bool, la cual es True si se actualizaron correctamente los datos y False si no se encontró el registro con el "idCafe" de la solicitud.

- **HttpDelete:**

Permite eliminar el registro de una cafetería que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```

{
  "idCafe": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string",
  "correoCafe": "string"
}

```

Es importante que el "idCafe" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y eliminarlo.

Devuelve una respuesta de tipo bool, la cual es True si se eliminó el registro de la cafetería correctamente y False si no se encontró el registro con el "idCafe" de la solicitud.

- **HttpGet("{id}"):**

Permite obtener cierta información de una cafetería en específico que se encuentre registrada en la base de datos, por lo tanto la ruta de la solicitud debe contener un parámetro llamado "id".

En caso de que el identificador enviado en la URL de la solicitud se encuentre dentro de la base de datos, se devuelve un objeto de tipo JsonResult que contiene la información de la cafetería, a excepción de su identificador. Ejemplo:

```

{
  "password": "contra123",
  "name": "Luis",
  "apePat": " Sanchez",
  "apeMat": "Cruz ",
  "photo": "https://historiadeltiempopresente.com/uploads/20/01/cafe_default_image.jpg",
  "correoCafe": "luis.sancruz@gmail.com"
}

```

En caso contrario se devuelve un objeto JsonResult con el mensaje: "No se ha encontrado un registro de cafetería con el ID {id}".

EndPoints Intendencia/datos:

- **HttpGet:**

Permite obtener una lista de todos los usuarios de tipo intendencia registrados en la base de datos, por lo tanto no requiere enviarse nada dentro de la petición.

Devuelve un objeto de tipo JsonResult que contiene la lista de las intendencias con su información correspondiente. Ejemplo:

```
[
  {
    "idInten": "I202200001",
    "password": "contra123",
    "name": "Alex",
    "apePat": "Cruz ",
    "apeMat": "Perez ",
    "photo": "https://historiadeltiempopresente.com/uploads/20/01/inte_default_image.jpg",
    "correoInten": "alex.cruzper@gmail.com"
  }
]
```

- **HttpPost:**

Permite insertar en la base de datos una nueva intendencia, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idInten": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string",
  "correoInten": "string"
}
```

Devuelve una respuesta de tipo bool, donde es True si se guardaron los datos correctamente en la base de datos y False si no se guardaron.

- **HttpPatch:**

Permite actualizar el registro de una intendencia que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idInten": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string",
  "correoInten": "string"
}
```


Es importante que el "idInten" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y se pueda modificar con la nueva información.

Devuelve una respuesta de tipo bool, la cual es True si se actualizaron correctamente los datos y False si no se encontró el registro con el "idInten" de la solicitud.

- **HttpDelete:**

Permite eliminar el registro de una intendencia que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idInten": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string",
  "correoInten": "string"
}
```

Es importante que el "idInten" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y eliminarlo.

Devuelve una respuesta de tipo bool, la cual es True si se eliminó el registro de la intendencia correctamente y False si no se encontró el registro con el "idInten" de la solicitud.

- **HttpGet("{id}"):**

Permite obtener cierta información de una intendencia en específico que se encuentre registrada en la base de datos, por lo tanto la ruta de la solicitud debe contener un parámetro llamado "id".

En caso de que el identificador enviado en la URL de la solicitud se encuentre dentro de la base de datos, se devuelve un objeto de tipo JsonResult que contiene la información de la intendencia, a excepción de su identificador. Ejemplo:

```
{
  "password": "contra123",
  "name": "Alex",
  "apePat": "Cruz ",
  "apeMat": "Perez ",
  "photo": "https://historiadeltiempopresente.com/uploads/20/01/inte_default_image.jpg",
  "correoInten": "alex.cruzper@gmail.com"
}
```

En caso contrario se devuelve un objeto JsonResult con el mensaje: "No se ha encontrado un registro de intendencia con el ID {id}".

EndPoint Post Login:

Permite iniciar sesión en la aplicación, por lo que requiere los datos del identificador del usuario y su contraseña para verificar que exista dentro de la base de datos y que pueda acceder.

```
{  
    "isUsuario": "string",  
    "password": "string"  
}
```

Devuelve una respuesta de tipo bool, donde es True si se pudo hacer el login (se encontró al usuario) y False si no se pudo hacer el login.

Endpoints Maestro/datos:

- **HttpGet:**

Permite obtener una lista de todos los usuarios de tipo maestro registrados en la base de datos, por lo tanto no requiere enviarse nada dentro de la petición.

Devuelve un objeto de tipo JsonResult que contiene la lista de los maestros con su información correspondiente. Ejemplo:

```
[  
    {  
        "idMaestro": "M20226326",  
        "password": "contra6789",  
        "name": "Ana Paola Guadalupe",  
        "apePat": "Vazquez",  
        "apeMat": "Quintero ",  
        "photo": "https://historiadeltiempopresente.com/uploads/20/01/inte_default_image.jpg"  
    }  
]
```

- **HttpPost:**

Permite insertar en la base de datos un nuevo maestro, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{  
    "idMaestro": "string",  
    "password": "string",  
    "name": "string",  
    "apePat": "string",  
    "apeMat": "string",  
    "photo": "string"  
}
```

Devuelve una respuesta de tipo bool, donde es True si se guardaron los datos correctamente en la base de datos y False si no se guardaron.

- **HttpPatch:**

Permite actualizar el registro de un maestro que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idMaestro": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string"
}
```

Es importante que el "idMaestro" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y se pueda modificar con la nueva información.

Devuelve una respuesta de tipo bool, la cual es True si se actualizaron correctamente los datos y False si no se encontró el registro con el "idMaestro" de la solicitud.

- **HttpDelete:**

Permite eliminar el registro de un maestro que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idMaestro": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string"
}
```

Es importante que el "idMaestro" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y eliminarlo.

Devuelve una respuesta de tipo bool, la cual es True si se eliminó el registro de la intendencia correctamente y False si no se encontró el registro con el "idMaestro" de la solicitud.

- **HttpGet("{id}"):**

Permite obtener cierta información de un maestro en específico que se encuentre registrado en la base de datos, por lo tanto la ruta de la solicitud debe contener un parámetro llamado "id".

En caso de que el identificador enviado en la URL de la solicitud se encuentre dentro de la base de datos, se devuelve un objeto de tipo JsonResult que contiene la información de la intendencia, a excepción de su identificador. Ejemplo:

```
{
  "password": "contra6789",
  "name": "Ana Paola Guadalupe",
  "apePat": "Vazquez",
  "apeMat": "Quintero ",
}
```

```
}    "photo": "https://historiadeltiempopresente.com/uploads/20/01/inte_default_image.jpg"
}
```

En caso contrario se devuelve un objeto JsonResult con el mensaje: "No se ha encontrado un registro de maestro con el ID {id}".

EndPoints Policia/datos:

- **HttpGet:**

Permite obtener una lista de todos los usuarios de tipo policía registrados en la base de datos, por lo tanto no requiere enviarse nada dentro de la petición.

Devuelve un objeto de tipo JsonResult que contiene la lista de los policías con su información correspondiente. Ejemplo:

```
[
  {
    "idPoli": "P2022631594",
    "password": "adal6454",
    "name": "Adalberto",
    "apePat": "Carlos",
    "apeMat": "Anaya",
    "photo": "https://historiadeltiempopresente.com/uploads/20/01/userdefaultimage.jpg",
    "correoPoli": "adalbert@gmail.com"
  }
]
```

- **HttpPost:**

Permite insertar en la base de datos un nuevo usuario de tipo policía, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idPoli": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string",
  "correoPoli": "string"
}
```

Devuelve una respuesta de tipo bool, en donde es True si se guardaron los datos correctamente en la base de datos y False si no se guardaron.

- **HttpPatch:**

Permite actualizar el registro de un usuario de tipo policía que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idPoli": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string",
  "correoPoli": "string"
}
```

Es importante que el "idPoli" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y se pueda modificar con la nueva información.

Devuelve una respuesta de tipo bool, la cual es True si se actualizaron correctamente los datos y False si no se encontró el registro con el "idPoli" de la solicitud.

- **HttpDelete:**

Permite eliminar el registro de un usuario de tipo policía que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "idPoli": "string",
  "password": "string",
  "name": "string",
  "apePat": "string",
  "apeMat": "string",
  "photo": "string",
  "correoPoli": "string"
}
```

Es importante que el "idPoli" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y eliminarlo.

Devuelve una respuesta de tipo bool, la cual es True si se eliminó el registro del usuario de tipo policía correctamente y False si no se encontró el registro con el "idPoli" de la solicitud.

- **HttpGet("{id}"):**

Permite obtener cierta información de un usuario de tipo policía en específico que se encuentre registrado en la base de datos, por lo tanto la ruta de la solicitud debe de contener un parámetro llamado "id".

En caso de que el identificador enviado en la url de la solicitud se encuentre dentro de la base de datos, se devuelve un objeto de tipo JsonResult que contiene la información del policía, a excepción de su identificador. Ejemplo:

```
{
  "password": "adal5464",
  "name": "Adalberto",

```

```

    "apePat": "Carlos",
    "apeMat": "Anaya",
    "photo": "https://historiadeltiempopresente.com/uploads/20/01/userdefaultimage.jpg",
    "correoPoli": "adalbert@gmail.com"
}

```

En caso contrario se devuelve un objeto JsonResult con el mensaje: “No se ha encontrado un registro de administrativo con el ID {id}”.

EndPoints RegistroEntradaSalida:

- **HttpGet:**

Permite obtener una lista de todos los registros de entrada y salida registrados en la base de datos, por lo tanto no requiere enviarse nada dentro de la petición.

Devuelve un objeto de tipo JsonResult que contiene la lista de los registros con su información correspondiente. Ejemplo:

```

[
  {
    "ID": 1,
    "Fecha": "2024-01-10T05:53:12.086Z",
    "Hora": {
      "ticks": 0
    },
    "IdUsuario": "1234567890",
    "TipoUsuario": "Administrador",
    "Instalacion": "Edificio A",
    "Registro": "Entrada"
  }
]

```

- **HttpPost:**

Permite insertar en la base de datos un nuevo registro de entrada o salida, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```

{
  "ID": number,
  "Fecha": "string",
  "Hora": {
    "ticks": number
  },
  "idUsuario": "string",
  "TipoUsuario": "string",
  "Instalacion": "string",
  "Registro": "string"
}

```

Devuelve una respuesta de tipo bool, en donde es True si se guardaron los datos correctamente en la base de datos y False si no se guardaron.

- **HttpPatch:**

Permite actualizar el registro de una entrada o salida que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "ID": number,
  "Fecha": "string",
  "Hora": {
    "ticks": number
  },
  "idUsuario": "string",
  "TipoUsuario": "string",
  "Instalacion": "string",
  "Registro": "string"
}
```

Es importante que el "ID" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y se pueda modificar con la nueva información.

Devuelve una respuesta de tipo bool, la cual es True si se actualizaron correctamente los datos y False si no se encontró el registro con el "ID" de la solicitud.

- **HttpDelete:**

Permite eliminar el registro de una entrada o salida que se encuentre en la base de datos, por lo tanto requiere que dentro de la petición se le envíe la siguiente información:

```
{
  "ID": number,
  "Fecha": "string",
  "Hora": {
    "ticks": number
  },
  "idUsuario": "string",
  "TipoUsuario": "string",
  "Instalacion": "string",
  "Registro": "string"
}
```

Es importante que el "ID" que se envía en la solicitud sea correcto, ya que este valor es el que se utiliza para evaluar que el registro exista y eliminarlo.

Devuelve una respuesta de tipo bool, la cual es True si se eliminó el registro de la entrada o salida correctamente y False si no se encontró el registro con el "ID" de la solicitud.

EndPoint Usuario/datos:

- **HttpGet("{id}"):**

Permite obtener cierta información de cualquier tipo de usuario en específico que se encuentre registrado en la base de datos, por lo tanto la ruta de la solicitud debe de contener un parámetro llamado "id".

En caso de que el identificador enviado en la url de la solicitud se encuentre dentro de la base de datos (en cualquier tabla de los distintos tipos de usuarios), se devuelve un objeto de tipo JsonResult que contiene la información esencial del usuario y su tipo de usuario (el cual cambia dependiendo de la tabla de usuario en donde se encontró el registro). Ejemplo:

```
{
  "name": "Daniel",
  "apePat": "Camacho",
  "apeMat": "García",
  "photo": "https://historiadeltiempopresente.com/uploads/20/01/userdefaultimage.jpg",
  "tipoUsuario": "Alumno"
}
```

En caso contrario se devuelve un objeto JsonResult con el mensaje: "No se ha encontrado un registro del usuario con el identificador {id}"

Endpoints QR/datos:

- **HttpGet("{id}"):**

Permite obtener el código QR de un usuario específico identificado por su ID. La ruta de la solicitud debe contener un parámetro llamado "id". Se generará un código QR que contiene información sobre el usuario, y se devolverá una imagen en formato PNG.

En caso de que el identificador enviado en la URL de la solicitud coincida con algún usuario registrado en la base de datos, se generará un código QR con la información correspondiente y se devolverá una imagen. Ejemplo:

Para un usuario de tipo Alumno:

```
{
  "boleta": "2022630751",
  "name": "Daniel",
  "apePat": "Camacho",
  "apeMat": "García",
  "tipoUsuario": "Alumno"
}
```

Para un usuario de tipo Administrativo:

```
{
  "idAdmin": "A2022631269",
  "name": "Alberto",
  "apePat": "Mendoza",
  "apeMat": "Hernández",
  "tipoUsuario": "Administrativo"
}
```


Para un usuario de tipo Cafetería:

```
{
  "idCafe": "C202200002",
  "name": "Cafetería ABC",
  "apePat": " ",
  "apeMat": " ",
  "tipoUsuario": "Cafeteria"
}
```

Para un usuario de tipo Intendencia:

```
{
  "idInten": "I202200001",
  "name": "Intendencia XYZ",
  "apePat": " ",
  "apeMat": " ",
  "tipoUsuario": "Intendencia"
}
```

Para un usuario de tipo Maestro:

```
{
  "idMaestro": "M202200003",
  "name": "Profesor Ejemplo",
  "apePat": "Pérez",
  "apeMat": "Martínez",
  "tipoUsuario": "Maestro"
}
```

Para un usuario de tipo Policía:

```
{
  "idPoli": "P202200004",
  "name": "Oficial García",
  "apePat": " ",
  "apeMat": " ",
  "tipoUsuario": "Policia"
}
```

En caso contrario, si el identificador no se encuentra en la base de datos, se devolverá un mensaje indicando que no se ha encontrado un registro del usuario con el identificador específico.

Nota: El código QR contiene información específica según el tipo de usuario, y se genera utilizando la biblioteca QRCoder. La imagen del código QR se devuelve en formato PNG.