

Diseño de Aplicaciones 2

Obligatorio 1

Evidencia del diseño y especificación de la API

Entregado como requisito de la materia Diseño de Aplicaciones 2

2021

Diego Scaffo - 196673
José Pedro Amado - 188885

Índice

1. Criterios seguidos para asegurar que la API cumple con los criterios REST	2
2. Descripción del mecanismo de autenticación de requests	2
3. Descripción general de códigos de error	3
4. Descripción de los resources de la API	3

1. Criterios seguidos para asegurar que la API cumple con los criterios REST

Para asegurar que la API cumple con los criterios REST nos aseguramos de que:

- Los nombres de los recursos estén en plural
- Se utilicen correctamente los verbos (Get, Post, Patch, Delete)
- No utilizar acciones o verbos como nombres de los recursos
- No tener más de 3 niveles en los endpoints
- Los requests son independientes, ninguno depende del estado/resultado de otro.

2. Descripción del mecanismo de autenticación de requests

El mecanismo de autenticación de requests consiste en recibir un Authorization Header con un token que debe ser validado por parte de la Web API.

Ese token es provisto por el login, al que se puede llegar haciendo un POST en /api/sessions.

Para la validación del token, se utiliza un Authorization Filter, que se ejecuta antes del método al que estaba destinado el request en el controlador. Ese authorization filter controla que se tenga un Authorization Header en el request. En caso de no tenerlo, devuelve un 401 - Not Authorized, si lo tiene, consulta a la BL si es válido, si no es válido también devuelve un 401 - Not Authorized. En caso de que sea válido, también se consulta si el token pertenece a un usuario que a su vez tenga el rol adecuado (generalmente "Administrator"), en caso de que esto sea negativo, se devuelve un 403 - You don't have permission to access this resource. Caso contrario, el request es permitido y se continúa con la ejecución.

El Authorization Filter implementado para esta solución puede ser utilizado tanto a nivel de métodos como a nivel de clases en los controllers. Además, se puede parametrizar el rol a controlar:

```
[Route("api/[controller]")]
[ApiController]
[AuthorizationFilter("Administrator")]
9 references
public class AdministratorsController : ControllerBase
{
    private readonly IUserLogic userLogic;
```

3. Descripción general de códigos de error

En general, se utilizaron los siguientes códigos de error:

- 200: Ejecución correcta, acompañado de la respuesta correspondiente (objeto o string).
- 400: Bad Request, acompañado de mensaje de error.
- 401: Not Authorized, cuando el usuario no envía un token en el Authorization Header o el mismo no es válido.
- 403: You don't have permission to access this resource, cuando se envía un token pero no se corresponde con el rol esperado (generalmente Administrator)
- 500: Something went wrong, cuando sucede alguna excepción no controlada

4. Descripción de los resources de la API

Url base: `http://localhost:{puerto}/api`

- Para categorías:

Get /categories

StatusCode:

200:

```
[
  {
    "categoryId": int,
    "categoryName": string,
    "playLists": [
      {
        "id": int,
        "name": string,
        "description": string,
        "imageUrl": string
      }
    ]
  }
]
```

500: string

Headers: -

Body: -

Descripción: Devuelve todas las categorías existentes en el sistema.

Get /categories/{categoryId}

StatusCode:

200:

```
{
  "id": int,
  "name": string,
  "playlists" : [
    {
      "id": int,
      "name": string,
      "description": string,
      "imageUrl": string,
    }
  ]
}
```

```

    ],
    "contents" : [
        {
            "id": int,
            "name": string,
            "artistName": string,
            "contentLength": string,
            "imageUrl": string,
            "audioUrl": string
        }
    ]
}
400: string
500: string
Headers: -
Body: -
Descripción: Devuelve el contenido de una categoría.

```

Get /categories/{categoryId}/contents

Status Code:

```

200:
[
    {
        "id": string
        "name": string,
        "contentLength" : string,
        "artistName": string,
        "imageUrl": string,
        "audioUrl": string
    }
]
400: string
500: string

```

Headers: -

Body: -

Descripción: Devuelve los contenidos reproducibles de una categoría.

Get /categories/{categoryId}/playlists

Status Code:

```

200:
[
    {
        "id": int,
        "name" : string
        "description": string,
        "imageUrl": string,
    }
]
400: string
500: string

```

Headers: -

Body: -

Descripción: Devuelve las playlists de una categoría.

- Para playlists:

Get /playlists

StatusCode:

200:

```
[
  {
    "id": int,
    "name" : string,
    "description": string,
    "imageUrl": string
  }
]
```

500: string

Headers: -

Body: -

Descripción: Devuelve las playlists existentes en el sistema.

Get /playlists/{playlistId}

StatusCode:

200:

```
{
  "id": int,
  "name": string,
  "contents" : [
    {
      "id": int,
      "name": string,
      "artistName": string,
      "contentLength": string,
      "imageUrl": string,
      "audioUrl": string
    }
  ],
  "description": string,
  "imageUrl": string,
}
```

400: string

500: string

Headers: -

Body: -

Descripción: Devuelve el contenido de una playlist.

Get /playlists/{playlistId}/contents

StatusCode:

200:

```
[
```

```

    {
        "id": string
        "name": string,
        "contentLength" : string,
        "artistName": string,
        "imageUrl": string,
        "audioUrl": string
    }
]
400: string
500: string
Headers: -
Body: -
Descripción: Devuelve los contenidos reproducibles de una playlist.

```

- Para contenidos reproducibles:

Get /contents

StatusCode:

```

200:
[
    {
        "id": string
        "name": string,
        "contentLength" : string,
        "artistName": string,
        "imageUrl": string,
        "audioUrl": string
    }
]
500: string

```

Headers: -

Body: -

Descripción: Devuelve los contenidos reproducibles en el sistema.

Get /contents/{contentId}

StatusCode:

```

200:
{
    "id": string
    "name": string,
    "contentLength" : string,
    "artistName": string,
    "imageUrl": string,
    "audioUrl": string
}
400: string
500: string

```

Headers: -

Body: -

Descripción: Devuelve toda la información de un contenido reproducible.

Post /contents

Body:

```
{
  "name": string,
  "contentLength": string,
  "artistName": string,
  "imageUrl": string,
  "audioUrl": string,
  "categories": int[],
  "playlists": [{
    "id": int,
    "name": string,
    "description": string
  }]
}
```

Headers:

```
{
  "Authorization": string
}
```

StatusCode:

```
201: string
400: string
401: string
403: string
500: string
```

Descripción: Crea un contenido reproducible.

Delete /contents/{id}

Body: -

Headers:

```
{
  "Authorization": string
}
```

StatusCode:

```
200: string
400: string
401: string //Para el caso en que no está autenticado
403: string //Para el supuesto caso en que está autenticado, pero no tiene
permisos
500: string
```

Descripción: Crea un contenido reproducible.

Patch /contents

Body:

```
{
  "id": int,
  "name": string,
  "contentLength": string,
```



```

    "artistName": string,
    "imageUrl": string,
    "audioUrl": string,
    "categories": int[],
    "playlists": [{
        "id": int,
        "name": string,
        "description": string
    }]
}
Headers:
{
    "Authorization": string
}
StatusCode:
    200 - string
    400 - string
    401 - string
    403 - string
    500 - string
Descripción: Actualiza un contenido reproducible.

```

- Para agendar citas

Post /appointments

Body:

```

{
    "illnessId": int,
    "name": string,
    "lastName": string,
    "birthDate": string,
    "eMail": string,
    "phone": string
}
Headers: -
StatusCode:
    201:
    {
        "psychologistName": string,
        "format": string,
        "address": string
    }
    400 - string
    404 - string
    500 - string
Descripción: Crea una nueva cita en el sistema.

```

- Para las dolencias

Get /illnesses

Body: -

Headers: -

```

StatusCode:
  200:
    [
      {
        "id": string,
        "name": string
      }
    ]
  500 - string

```

Descripción: Devuelve la lista de posibles dolencias.

- Para iniciar sesión

Post /sessions

Body:

```

{
  "eMail": string,
  "password": string
}

```

Headers: -

StatusCode:

```

  200:
    {
      "token": string
    }
  400 - string
  500 - string

```

Descripción: Permite iniciar sesión.

- Para cerrar sesión

Delete /sessions

Body:

```

{
  "token": string
}

```

Headers: -

StatusCode:

```

  200 - string
  400 - string
  500 - string

```

Descripción: Permite cerrar sesión

- Para los psicólogos

GET /psychologists

Body: -

Headers:

```

{
  "Authorization": string
}

```

StatusCode:

```

  200:

```

```
[
  {
    "id": int,
    "firstName": string,
    "lastName": string,
    "address": string,
    "format": string
  }
],
401 - string
403 - string
500 - string
```

Descripción: Devuelve la lista de psicólogos existentes en el sistema.

Get /psychologists/{psychologistId}

Body: -

Headers:

```
{
  "Authorization": string
}
```

StatusCode:

```
200:
[
  {
    "id": int,
    "firstName": string,
    "lastName": string,
    "address": string,
    "illnessModels": [
      {
        "id": 3,
        "name": "depresión"
      }
    ]
  }
]
400 - string
401 - string
403 - string
500 - string
```

Descripción: Devuelve toda la información de un psicólogo.

Post /psychologists

Body:

```
{
  "firstName": string,
  "lastName": string,
  "address": string,
  "format": string,
  "illnessModels": [{
    "id": 3
```

```

    ]]
  }
  Headers:
  {
    "Authorization": string
  }
  StatusCode:
    200 - string
    400 - string
    401 - string
    403 - string
    500 - string

```

Descripción: Crea un nuevo psicólogo en el sistema.

Patch /psychologists/{psychologistId}

```

Body:
{
  "id": int,
  "firstName": string,
  "lastName": string,
  "address": string,
  "format": string,
  "illnessModels": [{
    "id": 3
  }]
}

```

```

}
Headers:
{
  "Authorization": string
}
StatusCode:
  204
  400 - string
  401 - string
  403 - string
  500 - string

```

Descripción: Actualiza los datos de un psicólogo en el sistema.

Delete /psychologists

Body: int

```

Headers:
{
  "Authorization": string
}
StatusCode:
  204
  400 - string
  401 - string
  403 - string
  500 - string

```

Descripción: Elimina un psicólogo del sistema.

- Para los administradores

Get /administrators

Body: -

Headers:

```
{
    "Authorization": string
}
StatusCode:
200:
[
    {
        "id": int,
        "eMail": string
    }
]
400 - string
401 - string
403 - string
500 - string
```

Descripción: Devuelve los administradores registrados en el sistema.

Post /administrators

Body:

```
{
    "name": string,
    "eMail": string,
    "password": string
}
Headers:
{
    "Authorization": string
}
StatusCode:
200 - string
400 - string
401 - string
403 - string
500 - string
```

Descripción: Crea un nuevo administrador en el sistema.

Patch /administrators

Body:

```
{
    "id": int,
    "name": string,
    "eMail": string,
    "password": string
}
Headers:
```

```
{
  "Authorization": string
}
```

StatusCode:

200 - string

400 - string

401 - string

403 - string

500 - string

Descripción: Actualiza los datos de un administrador en el sistema.

Delete /administrators/{administratorId}

Body: -

Headers:

```
{
  "Authorization": string
}
```

StatusCode:

200 - string

400 - string

401 - string

403 - string

500 - string