

# Universidad ORT Uruguay

Facultad de Ingeniería

## Diseño de aplicaciones II

### Obligatorio I

## Evidencia del diseño y especificación de la API.

Juan Manuel Gallicchio 233335

Federico Carbonell 224359

<https://github.com/ORT-DA2/CarbonellGallicchio>

# Índice

<b>Índice</b>	<b>2</b>
<b>Evidencia del diseño y especificación de la API</b>	<b>3</b>
<b>Estilo de arquitectura</b>	<b>3</b>
Client–server	3
Stateless	3
Cacheable	3
Uniform interface	3
Layered system	4
<b>Información sobre la implementación</b>	<b>4</b>
Verbos HTTP	4
Códigos de respuesta	5
Mecanismo de autenticación	5
<b>Endpoints</b>	<b>6</b>
Administrators	7
Audio Content	9
Categories	11
Consultation	12
Problematics	12
Psychologists	13
Sessions	15

## Evidencia del diseño y especificación de la API

El propósito de este documento es informar como esta compuesta la Api, para facilitar el uso de la misma. Se detallara el estilo de arquitectura utilizado y cómo está compuesta, también los códigos de error utilizados y el mecanismo de autenticación.

## Estilo de arquitectura

El estilo de arquitectura utilizado es REST este indica seis principios que deben cumplirse para obtener dicha arquitectura. A continuación una breve explicación de cómo se implementaron en nuestra Api.

### Client–server

El servidor el cual brinda los servicios al cliente se encuentra separado de la implementación que usará, expone una serie de endpoints los cuales se podrán utilizar para realizar operaciones. Esto nos genera portabilidad y extensibilidad ya que no depende de la implementación del cliente siendo la api independiente.

### Stateless

Cada request realizada el servidor crea los servicios nuevamente, por dicha razón es necesario enviarle todos los datos necesarios a cada llamada realizada para que el servidor sea capaz de procesarla.

### Cacheable

Se le indica al cliente que respuesta de servicios puede ser cacheada para no tener que realizarla nuevamente. En esta entrega no se implementó pero es posible hacerlo si se desea.

### Uniform interface

Rest nuevamente define cuatro principios para cumplir este principio, estos son: identification of resources, manipulation of resources through representations; self-descriptive messages, y hypermedia.

Se puede apreciar cómo se cumplen estos principios ya que se identifican claramente los recursos en cada endpoint, se serializan los recursos al mapearlos en modelos y tambien se crearon distintos proyectos de excepciones para poder customizar los mensajes dependiendo de la situación.

## Layered system

El sistema se encuentra dividido en varias capas, principalmente estas son WebApi, Adapter, Business Logic y Data Access. Para tener mayor información recomendamos consultar el documento “Descripción del diseño” donde profundizamos sobre cómo está compuesto en detalle todo el sistema.

## Información sobre la implementación

Profundizaremos sobre los verbos utilizados, los códigos de error manejados y los endpoints de la Api.

### Verbos HTTP

Los verbos utilizados para la implementación son cuatro y estos son:

- GET sirve para obtener información sobre un elemento.
- POST sirve para agregar un elemento.
- PUT sirve para modificar la información de un elemento.
- DELETE sirve para eliminar un elemento.

La combinación de todos los verbos genera que REST sea CRUD, todos los anteriores son impotentes a excepción del POST ya que modifica el estado del servidor.

## Códigos de respuesta

Los códigos de error utilizados se pueden separar en familias, son los siguientes:

Success:

- 200 Indica éxito en el request realizado y se retorna el elemento.
- 201 Indica que la creación del elemento fue correcta, se utiliza para el verbo POST.
- 204 Indica éxito en el request pero el retorno es nulo, se utiliza para los verbos PUT o DELETE.

Error:

- 400 Indica que existe un error sobre el elemento enviado, algún atributo obligatorio vacío por ejemplo.
- 404 Indica que el elemento solicitado no fue encontrado o no existe.

Authorization:

- 401 Indica la falta del token de administrador.
- 403 Indica la falta de permisos para realizar la operación.

Internal error:

- 500 Indica un error en el servidor que no fue esperado.

## Mecanismo de autenticación

Para la autenticación de ciertos servicios es necesario un token administrador. Para la implementación de esta fue utilizado un AuthorizationFilter, este filter es ejecutado antes de realizar la request al controller por lo que si no se presenta un token o este no presenta los permisos necesarios la request se rechaza y no se ejecuta el controller.

## Endpoints

Para cada recurso se implementó un controller para realizar una correcta separación de responsabilidades.

Para documentar estos se utilizó swagger, luego de compilar el proyecto WebApi se puede visualizar en <https://localhost:5001/swagger/index.html>.

Para esta entrega no logramos agregar el token por lo que los endpoints que necesitan este no se encuentran funcionales desde la UI.

<b>Administrator</b>		▼
GET	/api/administrators/{id}	Obtains the information of an administrator by the id given.
DELETE	/api/administrators/{id}	Deletes an administrator.
POST	/api/administrators	Creates an administrator.
PUT	/api/administrators	Updates an administrator.
<b>AudioContent</b>		▼
GET	/api/audioContents/{id}	Obtains the information of an audio content by the id given.
DELETE	/api/audioContents/{id}	Deletes an audio content.
POST	/api/audioContents	Creates an audio content.
PUT	/api/audioContents	Updates an audio content.
<b>Category</b>		▼
GET	/api/categories	Obtains all the categories.
GET	/api/categories/{id}/playlists	Obtains all the playlists related to a category.
<b>Consultation</b>		▼
POST	/api/consultations	Creates a consultation with a psychologist.
<b>Problematic</b>		▼
GET	/api/problematics	Obtains all the problematics.
<b>Psychologist</b>		▼
GET	/api/psychologists/{id}	Obtains the information of a psychologist by the id given.
DELETE	/api/psychologists/{id}	Deletes a psychologist.
POST	/api/psychologists	Creates a psychologist.
PUT	/api/psychologists	Updates a psychologist.
<b>Session</b>		▼
POST	/api/sessions	Creates an administrator apikey token.

# Administrators

**Administrator**

**GET** `/api/administrators/{id}` Obtains the information of an administrator by the id given.

Obtains an administrator by the id given. An administrator token is required.

**Parameters**

Try it out

Name	Description
<b>id</b> <small>required</small> <code>integer(\$int32)</code> <i>(path)</i>	<input type="text" value="id"/>

**Responses**

Code	Description	Links
200	Success. Returns the requested object.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
404	NotFound. There is no administrator registered for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

**DELETE** `/api/administrators/{id}` Deletes an administrator.

Deletes the administrator with the id given. An administrator token is required.

**Parameters**

Try it out

Name	Description
<b>id</b> <small>required</small> <code>integer(\$int32)</code> <i>(path)</i>	<input type="text" value="id"/>

**Responses**

Code	Description	Links
200	Success	No links
204	No content.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
404	NotFound. There is no administrator registered for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

## Errores faltantes para los verbos POST y PUT de Administrators:

- 400: Error. The administrator's email can't be empty.
- 400: Error. The administrator's password can't be empty.

POST

/api/administrators

Creates an administrator.

Creates the administrator with the information send in the body. An administrator token is required.

Parameters

Try it out

No parameters

Request body

application/json

Example Value

Schema

```
{  "id": 0,  "name": "string",  "email": "string",  "password": "string"}
```

Responses

Code	Description	Links
200	Success	No links
201	Created.	No links
400	Error. The administrator's password can't be empty.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
500	InternalServerError. Server problems, unexpected error.	No links

PUT

/api/administrators

Updates an administrator.

Updates an administrator with the information send in the body. An administrator token is required.

Parameters

Try it out

No parameters

Request body

application/json

Example Value

Schema

```
{  "id": 0,  "name": "string",  "email": "string",  "password": "string"}
```

Responses

Code	Description	Links
200	Success	No links
204	No content.	No links
400	Error. The administrator's password can't be empty.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
404	NotFound. There is no administrator registered for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links



# Audio Content

**AudioContent**

**GET** `/api/audioContents/{id}` Obtains the information of an audio content by the id given.

Obtains an audio content related by the id given.

Parameters

Try it out

Name	Description
<b>id</b> <small>required</small> <code>integer(\$int32)</code> <i>(path)</i>	<input type="text" value="id"/>

Responses

Code	Description	Links
200	Success. Returns the requested object.	No links
404	NotFound. Audio content not exist for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

**DELETE** `/api/audioContents/{id}` Deletes an audio content.

Deletes the audio content with the id given. An administrator token is required.

Parameters

Try it out

Name	Description
<b>id</b> <small>required</small> <code>integer(\$int32)</code> <i>(path)</i>	<input type="text" value="id"/>

Responses

Code	Description	Links
200	Success	No links
204	No content.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
404	NotFound. Audio content not exist for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

## Errores faltantes para los verbos POST y PUT de Audio Content:

- 400: Error. The audio content must contain a playlist.
- 400: Error. The audio content name can't be empty.

POST

/api/audioContents

Creates an audio content.

Creates the audio content send in the body. An administrator token is required.

Parameters

Try it out

No parameters

Request body

application/json

Example Value

Schema

```

{
  "id": 0,
  "name": "string",
  "duration": {},
  "creatorName": "string",
  "imageUrl": "string",
  "audioUrl": "string",
  "categories": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "playlists": [
    {
      "id": 0,
      "name": "string",
      "description": "string"
    }
  ]
}

```

Responses

Code	Description	Links
200	Success	No links
201	Created.	No links
400	Error. The audio content must contain a category.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
500	InternalServerError. Server problems, unexpected error.	No links

PUT

/api/audioContents

Updates an audio content.

Updates the audio content with the content send in the body. An administrator token is required.

Parameters

Try it out

No parameters

Request body

application/json

Example Value

Schema

```

{
  "id": 0,
  "name": "string",
  "duration": {},
  "creatorName": "string",
  "imageUrl": "string",
  "audioUrl": "string",
  "categories": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "playlists": [
    {
      "id": 0,
      "name": "string",
      "description": "string"
    }
  ]
}

```

Responses

Code	Description	Links
200	Success	No links
204	No content.	No links
400	Error. The audio content must contain a category.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
404	NotFound. There is no administrator registered for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

# Categories

GET

/api/categories

Obtains all the categories.

Obtains all the information of the categories on the system.

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Success. Returns the list of categories.	No links
500	InternalServerError. Server problems, unexpected error.	No links

GET

/api/categories/{id}/playlists

Obtains all the playlists related to a category.

Obtains all the playlist related to a category.

Parameters

Try it out

Name	Description
<b>id</b> * required integer(int32) (path)	<div>id</div>

Responses

Code	Description	Links
200	Success. Returns the list of playlists.	No links
404	NotFound. Playlist not exist for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

# Consultation

POST

/api/consultations

Creates a consultation with a psychologist.

Creates a consultation with a psychologist. It will return the information of the psychologist and the direction or the link if it's virtual.

Parameters

Try it out

No parameters

Request body

application/json

Example Value

Schema

```
{
  "problematicId": 0,
  "patient": {
    "name": "string",
    "surname": "string",
    "birthDate": "2021-05-05T23:24:22.179Z",
    "email": "string",
    "cellphone": "string"
  }
}
```

Responses

Code	Description	Links
200	Success	No links
201	Created. Returns the information of the psychologist and the direction or the link if it's virtual	No links
404	Not found. There is no psychologist registered for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

# Problematics

GET

/api/problematics

Obtains all the problematics.

Obtains all the information of the problematics on the system.

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Success. Returns the list of problematics.	No links
500	InternalServerError. Server problems, unexpected error.	No links

# Psychologists

**GET** `/api/psychologists/{id}` Obtains the information of a psychologist by the id given.

Obtains the information of a psychologist by the id give.

Parameters

Try it out

Name	Description
<b>id</b> * required <code>integer(int32)</code> <i>(path)</i>	<input type="text" value="id"/>

Responses

Code	Description	Links
200	Success. Returns the psychologist information.	No links
404	NotFound. There is no psychologist registered for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

**DELETE** `/api/psychologists/{id}` Deletes a psychologist.

Deletes the psychologist with the id given. An administrator token is required.

Parameters

Try it out

Name	Description
<b>id</b> * required <code>integer(int32)</code> <i>(path)</i>	<input type="text" value="id"/>

Responses

Code	Description	Links
200	Success	No links
204	No content.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
404	NotFound. There is no psychologist registered for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

## Errores faltantes para los verbos POST y PUT de Psychologist:

- 400: Error. The psychologist's consultation mode can't be empty.
- 400: Error. The psychologist's direction can't be empty.
- 400: Error. The psychologist's problematics can't be zero.
- 400: Error. The psychologist's consultation mode must be 'Virtual' or 'Presencial'.
- 400: Error. The psychologist's problematics must be exactly three

POST

/api/psychologists

Creates a psychologist.

Creates the psychologist with the information send in the body. An administrator token is required.

Parameters

No parameters

Try it out

Request body

application/json

Example Value

Schema

```
{
  "id": 0,
  "name": "string",
  "consultationMode": "string",
  "direction": "string",
  "problematics": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "creationDate": "2021-05-05T23:30:39.307Z"
}
```

Responses

Code	Description	Links
200	Success	No links
201	Created. Returns the psychologist created.	No links
400	Error. The psychologist's problematics can't be zero.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
500	InternalServerError. Server problems, unexpected error.	No links

PUT

/api/psychologists

Updates a psychologist.

Updates a psychologist with the information send in the body. An administrator token is required.

Parameters

No parameters

Try it out

Request body

application/json

Example Value

Schema

```
{
  "id": 0,
  "name": "string",
  "consultationMode": "string",
  "direction": "string",
  "problematics": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "creationDate": "2021-05-05T23:33:58.131Z"
}
```

Responses

Code	Description	Links
200	Success	No links
204	No content.	No links
400	Error. The psychologist's problematics can't be zero.	No links
401	Unauthorized. Must contain a token to access Api.	No links
403	Unauthorized. Forbidden, ask for permission.	No links
404	NotFound. There is no psychologist registered for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links

# Sessions

POST

/api/sessions

Creates an administrator apikey token.

Creates an administrator apikey token from administrator user credentials.

Parameters

Try it out

No parameters

Request body

application/json

Example Value

Schema

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

Responses

Code	Description	Links
200	Success	No links
201	Created.	No links
400	Error. The password can't be empty.	No links
404	Not Found. There is no administrator registered for the given data.	No links
500	InternalServerError. Server problems, unexpected error.	No links