

# Fantasy Realms APIs

Esta API ofrece las herramientas necesarias para soportar las funcionalidades necesarias para jugar a "Fantasy Realms" en un entorno web. Automatiza las reglas del juego, gestiona partidas y las interacciones entre los jugadores, y proporciona una experiencia accesible y fiel al diseno original del juego fisico.

More information: <https://github.com/gii-is-DP1/DP1-2024-2025--l2-05>

Contact Info: [dp1-2425-group1-l2-05@gmail.com](mailto:dp1-2425-group1-l2-05@gmail.com)

Version: v4.0

Apache 2.0

<https://www.apache.org/licenses/LICENSE-2.0>

## Access

1.

## Methods

[ [Jump to Models](#) ]

## Table of Contents

### [Achievements](#)

- [POST /api/v1/achievements](#)
- [DELETE /api/v1/achievements/{achievementId}](#)
- [GET /api/v1/achievements](#)
- [GET /api/v1/achievements/{id}](#)
- [PUT /api/v1/achievements/{achievementId}](#)

### [Authentication](#)

- [POST /api/v1/auth/signin](#)
- [POST /api/v1/auth/signup](#)
- [GET /api/v1/auth/validate](#)

### [CurrentUser](#)

- [GET /api/v1/currentuser](#)

### [Decks](#)

- [GET /api/v1/deck/{id}](#)

### [Developers](#)

- [GET /api/v1/developers](#)

### [Friendship](#)

- [POST /api/v1/friendships/accept](#)
- [DELETE /api/v1/friendships/delete](#)
- [GET /api/v1/friendships/all](#)
- [GET /api/v1/friendships/friends](#)
- [GET /api/v1/friendships/pending](#)
- [POST /api/v1/friendships/request](#)

### [GameInvitation](#)

- [POST /api/v1/game-invitation/accept/{match}/{sender}](#)
- [GET /api/v1/game-invitation](#)
- [POST /api/v1/game-invitation/reject/{match}](#)
- [POST /api/v1/game-invitation/send/{receiver}/{match}](#)

### [Matches](#)

- [POST /api/v1/matches/create](#)
- [DELETE /api/v1/matches/{id}/delete](#)
- [GET /api/v1/matches/{id}/creator](#)
- [GET /api/v1/matches/{id}](#)
- [GET /api/v1/matches](#)
- [GET /api/v1/matches/myMatches](#)
- [POST /api/v1/matches/{id}/join](#)
- [DELETE /api/v1/matches/{matchId}/remove-player/{userIdToRemove}](#)

- [POST /api/v1/matches/{id}/start](#)
- [POST /api/v1/matches/{matchId}/toggle-role](#)

## Players

- [POST /api/v1/players/discard/{matchId}/{cardId}](#)
- [POST /api/v1/players/drawCard/deck/{matchId}](#)
- [POST /api/v1/players/drawCard/discardPile/{matchId}/{cardId}](#)

## Statistics

- [GET /api/v1/statistics/average-match-duration](#)
- [GET /api/v1/statistics/average-players-per-match](#)
- [GET /api/v1/statistics/average-points](#)
- [GET /api/v1/statistics/average-ranking-position](#)
- [GET /api/v1/statistics/average-turns-per-match](#)
- [GET /api/v1/statistics/global-match-duration-average](#)
- [GET /api/v1/statistics/global-match-players-average](#)
- [GET /api/v1/statistics/global-match-max-duration](#)
- [GET /api/v1/statistics/global-match-min-duration](#)
- [GET /api/v1/statistics/global-match-players-max](#)
- [GET /api/v1/statistics/max-match-duration](#)
- [GET /api/v1/statistics/match-players-max](#)
- [GET /api/v1/statistics/max-points](#)
- [GET /api/v1/statistics/global-match-players-min](#)
- [GET /api/v1/statistics/min-match-duration](#)
- [GET /api/v1/statistics/match-players-min](#)
- [GET /api/v1/statistics/min-points](#)
- [GET /api/v1/statistics/most-frequent-cards](#)
- [GET /api/v1/statistics/global-ranking](#)
- [GET /api/v1/statistics/total-match-duration](#)
- [GET /api/v1/statistics/total-matches](#)
- [GET /api/v1/statistics/win-percentage](#)

## Turns

- [POST /api/v1/turns/{matchId}/cancelTurn](#)
- [POST /api/v1/turns/{matchId}/specialTurn](#)

## UserConnection

- [POST /api/v1/heartbeat](#)

## Users

- [POST /api/v1/users](#)
- [DELETE /api/v1/users/{userId}](#)
- [GET /api/v1/users](#)
- [GET /api/v1/users/authorities](#)
- [GET /api/v1/users/{id}](#)
- [PUT /api/v1/users/{userId}](#)

# Achievements

## POST /api/v1/achievements

Crear un nuevo logro. Accesible solo por administradores. ([createAchievement](#))

Permite a los administradores crear un nuevo logro.

### Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

### Request body

body [AchievementDTO](#) (required)

*Body Parameter* —

### Return type

[AchievementDTO](#)

### Example data

Content-Type: application/json

```
{
  "condition" : "WIN_N_GAMES",
  "tier" : "FACIL",
  "extraData" : "extraData",
  "description" : "description",
  "id" : 0,
  "type" : "PROGRESO",
  "requiredValue" : 0
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

- 201  
Logro creado con Ã©xito [AchievementDTO](#)
- 400  
Existe un problema con los datos proporcionados con el cliente
- 403  
Acceso denegado: solo los administradores pueden crear logros
- 409  
Problema de unicidad en la base de datos
- 500  
Error interno inesperado en el servidor

DELETE /api/v1/achievements/{achievementId}

Up

Eliminar un logro existente. Accesible solo por administradores. (deleteAchievement)

Permite a los administradores eliminar un logro existente.

Path parameters

achievementId (required)  
Path Parameter — format: int32

Responses

- 204  
Logro eliminado con Ã©xito
- 403  
Acceso denegado: solo los administradores pueden eliminar logros
- 404  
Logro no encontrado

GET /api/v1/achievements

Up

Listar todos los logros. Accesible por cualquier usuario. (findAll1)

Devuelve una lista de todos los logros.

Return type

array[AchievementDTO]

Example data

Content-Type: application/json

```
[ {
  "condition" : "WIN_N_GAMES",
  "tier" : "FACIL",
  "extraData" : "extraData",
  "description" : "description",
  "id" : 0,
  "type" : "PROGRESO",
  "requiredValue" : 0
}, {
  "condition" : "WIN_N_GAMES",
  "tier" : "FACIL",
  "extraData" : "extraData",
  "description" : "description",
```

```
"id" : 0,
"type" : "PROGRESO",
"requiredValue" : 0
} ]
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

Lista de logros recuperada con Ã©xito

## GET /api/v1/achievements/{id}

Up

Obtener un logro por ID. Accesible por cualquier usuario. (**findById1**)

Recupera un logro por su ID.

### Path parameters

**id (required)**

*Path Parameter* — format: int32

### Return type

[AchievementDTO](#)

### Example data

Content-Type: application/json

```
{
  "condition" : "WIN_N_GAMES",
  "tier" : "FACIL",
  "extraData" : "extraData",
  "description" : "description",
  "id" : 0,
  "type" : "PROGRESO",
  "requiredValue" : 0
}
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

Logro recuperado con Ã©xito [AchievementDTO](#)

404

Logro no encontrado

## PUT /api/v1/achievements/{achievementId}

Up

Actualizar un logro existente. Accesible solo por administradores. (**updateAchievement**)

Permite a los administradores actualizar un logro existente.

### Path parameters

**achievementId (required)**

*Path Parameter* — format: int32

### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

### Request body

body [AchievementDTO](#) (required)

*Body Parameter* —

Return type  
[AchievementDTO](#)

Example data  
Content-Type: application/json

```
{
  "condition" : "WIN_N_GAMES",
  "tier" : "FACIL",
  "extraData" : "extraData",
  "description" : "description",
  "id" : 0,
  "type" : "PROGRESO",
  "requiredValue" : 0
}
```

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**  
**200**  
Logro actualizado con Ã©xito [AchievementDTO](#)  
**403**  
Acceso denegado: solo los administradores pueden actualizar logros  
**404**  
Logro no encontrado  
**500**  
Error interno inesperado en el servidor

## Authentication

POST /api/v1/auth/signin

Up

Iniciar sesiÃ³n ([authenticateUser](#))  
Autentica un usuario basado en sus credenciales y devuelve un token JWT.

**Consumes**  
This API call consumes the following media types via the Content-Type request header:

- application/json

**Request body**  
body [LoginRequest](#) (required)  
*Body Parameter* —

**Return type**  
String

**Example data**  
Content-Type: application/json

```
" "
```

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**  
**200**  
Usuario autenticado exitosamente [String](#)  
**400**  
Credenciales invÃ¡lidas

POST /api/v1/auth/signup

Up

Registrar usuario ([registerUser](#))  
Registra un nuevo usuario en el sistema.

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [SignupRequest](#) (required)

*Body Parameter* —

Return type

[MessageResponse](#)

Example data

Content-Type: application/json

```
{
  "message" : "message"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

- 200  
Usuario registrado exitosamente [MessageResponse](#)
- 400  
Error al registrar el usuario
- 500  
Error interno del servidor al registrar el usuario

GET /api/v1/auth/validate

Up

Validar token (**validateToken**)

Valida la validez de un token JWT y devuelve true si es v  lido o false si no lo es.

Query parameters

token (required)  
*Query Parameter* —

Return type

Boolean

Example data

Content-Type: application/json

```
true
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

- 200  
Token validado exitosamente [Boolean](#)
- 400  
Token inv  lido o ausente
- 401  
El token no es v  lido o ha expirado
- 403  
El token es v  lido pero no tiene permisos para acceder al recurso
- 500  
Error inesperado al validar el token

Obtener usuario actual (`getCurrentUser`)

Devuelve la información del usuario actualmente autenticado.

### Return type

[UserDTO](#)

### Example data

Content-Type: application/json

```
{
  "password" : "password",
  "achievements" : [ {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  }, {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  } ],
  "authority" : "authority",
  "id" : 0,
  "avatar" : "avatar",
  "matches" : [ {
    "currentTurn" : {
      "turnCount" : 1,
      "drawSource" : "NONE",
      "username" : "username"
    },
    "discard" : {
      "cards" : [ {
        "image" : "image",
        "baseValue" : 9,
        "name" : "name",
        "cardType" : "MAGO",
        "modsDTO" : [ {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        }, {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        } ],
        "id" : 7,
        "finalValue" : 3
      }, {
        "image" : "image",
        "baseValue" : 9,
        "name" : "name",
        "cardType" : "MAGO",
        "modsDTO" : [ {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        }, {
          "primaryValue" : 4,
```

```

        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
} ],
"id" : 2
},
"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
} ],
"name" : "name",
"id" : 5,
"startDate" : "2000-01-23T04:56:07.000+00:00"
}, {
    "currentTurn" : {
        "turnCount" : 1,
        "drawSource" : "NONE",
        "username" : "username"
    },
    "discard" : {
        "cards" : [ {
            "image" : "image",
            "baseValue" : 9,
            "name" : "name",
            "cardType" : "MAGO",
            "modsDT0" : [ {
                "primaryValue" : 4,
                "modType" : "PENALTY",
                "secondaryValue" : 7,
                "description" : "description",
                "id" : 2
            }, {
                "primaryValue" : 4,
                "modType" : "PENALTY",
                "secondaryValue" : 7,
                "description" : "description",
                "id" : 2
            }
        ]
    }
}

```



```

    } ],
    "id" : 7,
    "finalValue" : 3
  }, {
    "image" : "image",
    "baseValue" : 9,
    "name" : "name",
    "cardType" : "MAGO",
    "modsDT0" : [ {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    }, {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
  } ],
  "id" : 2
},
"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
} ],
"name" : "name",
"id" : 5,
"startDate" : "2000-01-23T04:56:07.000+00:00"
} ],
"email" : "email",
"username" : "username"
}

```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

Usuario obtenido con Ã©xito [UserDTO](#)

401

No autorizado: El usuario no estÃ¡ autenticado

404

Usuario no encontrado

500

Error al intentar obtener la informaciÃ³n del usuario registrado

# Decks

## GET /api/v1/deck/{id}

Up

Buscar mazo por ID ([findDeckById](#))

Devuelve la informaciÃ³n del mazo correspondiente al ID especificado.

## Path parameters

**id (required)**

*Path Parameter* — format: int32

## Return type

[DeckDTO](#)

## Example data

Content-Type: application/json

```
{
  "cards" : [ {
    "image" : "image",
    "baseValue" : 9,
    "name" : "name",
    "cardType" : "MAGO",
    "modsDTO" : [ {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    }, {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
  }, {
    "image" : "image",
    "baseValue" : 9,
    "name" : "name",
    "cardType" : "MAGO",
    "modsDTO" : [ {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    }, {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
  } ],
  "initialCards" : [ null, null ]
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

200

Mazo encontrado con Ã©xito [DeckDTO](#)

401

No autorizado: se requiere autenticaci3n Bearer

404

Mazo no encontrado

Developers

GET /api/v1/developers [Up](#)

Obtener lista de desarrolladores ([getDevelopers](#))

Devuelve la lista de desarrolladores registrada en el archivo pom.xml. Si la lista no estÃ¡ cargada, se procesa dinÃ¡micamente.

Return type

[Developer](#)

Example data

Content-Type: application/json

```
{
  "timezone" : "timezone",
  "organization" : "organization",
  "roles" : [ "roles", "roles" ],
  "name" : "name",
  "id" : "id",
  "organizationUrl" : "organizationUrl",
  "email" : "email",
  "url" : "url",
  "properties" : {
    "key" : "properties"
  }
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Lista de desarrolladores obtenida con Ã©xito [Developer](#)

500

Error al intentar leer el archivo <code>pom.xml</code>.

Friendship

POST /api/v1/friendships/accept [Up](#)

Aceptar solicitud de amistad ([acceptFriendRequest](#))

Acepta la solicitud de amistad enviada por 'senderId'.

Query parameters

**senderId (required)**  
*Query Parameter* — format: int32

Return type

[FriendshipRequestDTO](#)

Example data

Content-Type: application/json

```
{
```

```
"senderIsOnline" : true,
"senderName" : "senderName",
"senderId" : 0,
"receiverId" : 6,
"receiverIsOnline" : true,
"receiverName" : "receiverName",
"status" : "status"
}
```

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**

- 200  
Solicitud de amistad aceptada [FriendshipRequestDTO](#)
- 400  
Bad Request [ErrorMessage](#)
- 401  
No autorizado [FriendshipRequestDTO](#)
- 403  
Forbidden [ErrorMessage](#)
- 404  
Solicitud o usuario no encontrada [ErrorMessage](#)
- 409  
Conflict [ErrorMessage](#)
- 500  
Internal Server Error [ErrorMessage](#)

**DELETE /api/v1/friendships/delete**

[Up](#)

Eliminar amistad (**deleteFriendship**)

Elimina la amistad con el usuario proporcionado en el par metro 'friendId'.

**Query parameters**

**friendId (required)**  
*Query Parameter* — format: int32

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**

- 204  
Amistad eliminada con  xito
- 400  
Bad Request [ErrorMessage](#)
- 401  
No autorizado
- 403  
Forbidden [ErrorMessage](#)
- 404  
Usuario o amistad no encontrada [ErrorMessage](#)
- 409  
Conflict [ErrorMessage](#)
- 500  
Internal Server Error [ErrorMessage](#)

**GET /api/v1/friendships/all**

[Up](#)

Obtener todas las amistades (**getAllFriendships**)

Retorna todas las amistades (en cualquier estado) para el usuario actual.

**Return type**

array[[FriendshipRequestDTO](#)]

**Example data**

Content-Type: application/json

```
[ {
  "senderIsOnline" : true,
  "senderName" : "senderName",
  "senderId" : 0,
  "receiverId" : 6,
  "receiverIsOnline" : true,
  "receiverName" : "receiverName",
  "status" : "status"
}, {
  "senderIsOnline" : true,
  "senderName" : "senderName",
  "senderId" : 0,
  "receiverId" : 6,
  "receiverIsOnline" : true,
  "receiverName" : "receiverName",
  "status" : "status"
} ]
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

200

Lista de amistades obtenida con Ã©xito

400

Bad Request [ErrorMessage](#)

401

No autorizado

403

Forbidden [ErrorMessage](#)

404

Not Found [ErrorMessage](#)

409

Conflict [ErrorMessage](#)

500

Internal Server Error [ErrorMessage](#)

## GET /api/v1/friendships/friends

[Up](#)

Obtener lista de amigos ([getFriends](#))

Retorna la lista de amigos para el usuario actual.

#### Return type

array[[FriendshipRequestDTO](#)]

#### Example data

Content-Type: application/json

```
[ {
  "senderIsOnline" : true,
  "senderName" : "senderName",
  "senderId" : 0,
  "receiverId" : 6,
  "receiverIsOnline" : true,
  "receiverName" : "receiverName",
  "status" : "status"
}, {
  "senderIsOnline" : true,
  "senderName" : "senderName",
  "senderId" : 0,
  "receiverId" : 6,
  "receiverIsOnline" : true,
  "receiverName" : "receiverName",
  "status" : "status"
} ]
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

Lista de amigos obtenida con Ã©xito

400

Bad Request [ErrorMessage](#)

401

No autorizado (sin token o invÃ¡lido)

403

Forbidden [ErrorMessage](#)

404

Not Found [ErrorMessage](#)

409

Conflict [ErrorMessage](#)

500

Internal Server Error [ErrorMessage](#)

## GET /api/v1/friendships/pending

Up

Obtener solicitudes pendientes ([getPendingFriendships](#))

Retorna todas las solicitudes de amistad con estado 'PENDING' para el usuario actual.

### Return type

array[[FriendshipRequestDTO](#)]

### Example data

Content-Type: application/json

```
[ {
  "senderIsOnline" : true,
  "senderName" : "senderName",
  "senderId" : 0,
  "receiverId" : 6,
  "receiverIsOnline" : true,
  "receiverName" : "receiverName",
  "status" : "status"
}, {
  "senderIsOnline" : true,
  "senderName" : "senderName",
  "senderId" : 0,
  "receiverId" : 6,
  "receiverIsOnline" : true,
  "receiverName" : "receiverName",
  "status" : "status"
} ]
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

Lista de solicitudes pendientes obtenida con Ã©xito

400

Bad Request [ErrorMessage](#)

401

No autorizado

403

Forbidden [ErrorMessage](#)

404

Not Found [ErrorMessage](#)

409

Conflict [ErrorMessage](#)

500

Internal Server Error [ErrorMessage](#)

## POST /api/v1/friendships/request

Up

Enviar solicitud de amistad ([sendFriendRequest](#))

EnvÃ­a una solicitud de amistad al usuario especificado por 'receiverUsername'.

## Query parameters

**receiverUsername (required)**

*Query Parameter* —

## Return type

[FriendshipRequestDTO](#)

## Example data

Content-Type: application/json

```
{
  "senderIsOnline" : true,
  "senderName" : "senderName",
  "senderId" : 0,
  "receiverId" : 6,
  "receiverIsOnline" : true,
  "receiverName" : "receiverName",
  "status" : "status"
}
```

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

**200**

Solicitud de amistad enviada con Ã©xito [FriendshipRequestDTO](#)

**400**

Bad Request [ErrorMessage](#)

**401**

No autorizado [FriendshipRequestDTO](#)

**403**

Forbidden [ErrorMessage](#)

**404**

Usuario no encontrado [ErrorMessage](#)

**409**

Conflict [ErrorMessage](#)

**500**

Internal Server Error [ErrorMessage](#)

# GamelInvitation

**POST** /api/v1/game-invitation/accept/{match}/{sender}

Up

Aceptar invitaci3n de juego (acceptGameInvitation)

Acepta la invitaci3n de juego enviada por 'senderUsername' para la partida 'matchId'.

## Path parameters

**match (required)**

*Path Parameter* — ID de la partida format: int32

**sender (required)**

*Path Parameter* — Nombre de usuario que envi3 la invitaci3n

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

**200**

Invitaci3n aceptada con Ã©xito

**400**

Bad Request [ErrorMessage](#)

**401**

No autorizado

**403**

Forbidden [ErrorMessage](#)

**404**

Usuario, partida o invitaci3n no encontrada [ErrorMessage](#)

409

Conflict [ErrorMessage](#)

500

Internal Server Error [ErrorMessage](#)

## GET /api/v1/game-invitation

Listar invitaciones de juego (**getGameInvitations**)

Retorna todas las invitaciones de partidas dirigidas al usuario actual.

### Return type

array[[GameInvitationDTO](#)]

### Example data

Content-Type: application/json

```
[ {
  "senderUsername" : "senderUsername",
  "matchId" : 0,
  "receiverUsername" : "receiverUsername",
  "status" : true
}, {
  "senderUsername" : "senderUsername",
  "matchId" : 0,
  "receiverUsername" : "receiverUsername",
  "status" : true
} ]
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

Lista de invitaciones obtenida con  xito

400

Bad Request [ErrorMessage](#)

401

No autorizado, token inv lido o ausente

403

Forbidden [ErrorMessage](#)

404

Not Found [ErrorMessage](#)

409

Conflict [ErrorMessage](#)

500

Internal Server Error [ErrorMessage](#)

## POST /api/v1/game-invitation/reject/{match}

Rechazar invitaci3n de juego (**rejectGameInvitation**)

Rechaza la invitaci3n de juego para la partida con ID 'matchId'.

### Path parameters

**match (required)**

*Path Parameter* — ID de la partida format: int32

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

Invitaci3n rechazada con  xito

400

Bad Request [ErrorMessage](#)



401  
No autorizado  
403  
Forbidden [ErrorMessage](#)  
404  
Partida o invitaci3n no encontrada [ErrorMessage](#)  
409  
Conflict [ErrorMessage](#)  
500  
Internal Server Error [ErrorMessage](#)

---

## POST /api/v1/game-invitation/send/{receiver}/{match}

Up

Enviar invitaci3n de juego ([sendGameInvitation](#))

Env3a una invitaci3n de juego al usuario indicado para la partida especificada.

### Path parameters

#### receiver (required)

*Path Parameter* — Nombre de usuario que recibir3 la invitaci3n

#### match (required)

*Path Parameter* — ID de la partida a la que se invita format: int32

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200  
Invitaci3n enviada con 3xito

400  
Bad Request [ErrorMessage](#)

401  
No autorizado

403  
Forbidden [ErrorMessage](#)

404  
Usuario o partida no encontrada [ErrorMessage](#)

409  
Conflict [ErrorMessage](#)

500  
Internal Server Error [ErrorMessage](#)

---

## Matches

## POST /api/v1/matches/create

Up

Crear nueva partida ([createMatch](#))

Crea una partida con el usuario actual como CREADOR.

### Query parameters

#### matchName (required)

*Query Parameter* —

### Return type

[MatchDTO](#)

### Example data

Content-Type: application/json

```
{
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
```

```

"cards" : [ {
  "image" : "image",
  "baseValue" : 9,
  "name" : "name",
  "cardType" : "MAGO",
  "modsDT0" : [ {
    "primaryValue" : 4,
    "modType" : "PENALTY",
    "secondaryValue" : 7,
    "description" : "description",
    "id" : 2
  }, {
    "primaryValue" : 4,
    "modType" : "PENALTY",
    "secondaryValue" : 7,
    "description" : "description",
    "id" : 2
  } ],
  "id" : 7,
  "finalValue" : 3
}, {
  "image" : "image",
  "baseValue" : 9,
  "name" : "name",
  "cardType" : "MAGO",
  "modsDT0" : [ {
    "primaryValue" : 4,
    "modType" : "PENALTY",
    "secondaryValue" : 7,
    "description" : "description",
    "id" : 2
  }, {
    "primaryValue" : 4,
    "modType" : "PENALTY",
    "secondaryValue" : 7,
    "description" : "description",
    "id" : 2
  } ],
  "id" : 7,
  "finalValue" : 3
} ],
"id" : 2
},
"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
} ], {

```

```
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "name" : "name",
  "id" : 5,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
}
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

- 201**  
Partida creada con Ã©xito [MatchDTO](#)
- 400**  
Datos invÃ¡lidos en la peticiÃ³n
- 401**  
No autorizado
- 403**  
El usuario tiene un jugador asociado que estÃ¡ en partida
- 404**  
Mazo universal no encontrado
- 500**  
Error al crear la partida

## DELETE /api/v1/matches/{id}/delete

[Up](#)

Borrar partida (**deleteMatch**)

Solo se puede borrar si la partida no ha comenzado y el usuario es el creador.

### Path parameters

**id (required)**  
*Path Parameter* — format: int32

### Responses

- 204**  
Partida eliminada con Ã©xito
- 401**  
No autorizado
- 403**  
El usuario no es el creador de la partida
- 404**  
Partida no encontrada
- 409**  
La partida estÃ¡ en progreso o ha finalizado
- 500**  
Error al intentar eliminar la partida

## GET /api/v1/matches/{id}/creator

[Up](#)

Obtener creador de la partida (**findCreatorOfMatch**)

Devuelve el Player que creÃ³ la partida con ID especificado.

### Path parameters

**id (required)**  
*Path Parameter* — format: int32

### Return type

[PlayerDTO](#)

### Example data

Content-Type: application/json

```
{
```

```
"score" : 1,
"playerHand" : [ null, null ],
"id" : 1,
"rol" : "CREADOR",
"username" : "username"
}
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

Creador devuelto con Ã©xito [PlayerDTO](#)

401

No autorizado

404

Partida no encontrada

## GET /api/v1/matches/{id}

[Up](#)

Buscar partida por ID ([findMatchById](#))

Devuelve la informaci3n de la partida con el ID especificado.

### Path parameters

**id (required)**

*Path Parameter* — format: int32

### Return type

[MatchDTO](#)

### Example data

Content-Type: application/json

```
{
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ], {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ], {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ], {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    } ]
}
```

```

        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
} ],
"id" : 2
},
"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
} ],
"name" : "name",
"id" : 5,
"startDate" : "2000-01-23T04:56:07.000+00:00"
}

```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

**200**

Partida encontrada con Ã©xito [MatchDTO](#)

**401**

No autorizado

**404**

Partida no encontrada

## GET /api/v1/matches

Listar partidas filtradas por estado (**findMatchesByStatus**)

Devuelve partidas filtradas por estado (lobby, inProgress, finished) de forma paginada. Los usuarios no administradores solo pueden listar partidas en estado lobby. Los administradores pueden filtrar por cualquier estado.

## Query parameters

### page (optional)

*Query Parameter* — default: 0 format: int32

### size (optional)

*Query Parameter* — default: 10 format: int32

### status (optional)

*Query Parameter* — default: lobby

## Return type

[PageMatchDTO](#)

## Example data

Content-Type: application/json

```
{
  "number" : 5,
  "size" : 1,
  "numberOfElements" : 5,
  "last" : true,
  "totalPages" : 6,
  "pageable" : {
    "paged" : true,
    "pageNumber" : 7,
    "offset" : 2,
    "pageSize" : 9,
    "unpaged" : true
  },
  "sort" : {
    "unsorted" : true,
    "sorted" : true,
    "empty" : true
  },
  "content" : [ {
    "currentTurn" : {
      "turnCount" : 1,
      "drawSource" : "NONE",
      "username" : "username"
    },
    "discard" : {
      "cards" : [ {
        "image" : "image",
        "baseValue" : 9,
        "name" : "name",
        "cardType" : "MAGO",
        "modsDTO" : [ {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        }, {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        } ],
        "id" : 7,
        "finalValue" : 3
      }, {
        "image" : "image",
        "baseValue" : 9,
        "name" : "name",
        "cardType" : "MAGO",
        "modsDTO" : [ {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        }, {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        } ],
        "id" : 7,
        "finalValue" : 3
      } ]
    }
  ]
}
```

```

        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    } ],
    "id" : 2
  },
  "creator" : "creator",
  "inScoringPhase" : true,
  "inProgress" : true,
  "endDate" : "2000-01-23T04:56:07.000+00:00",
  "deckId" : 5,
  "players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "name" : "name",
  "id" : 5,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
}, {
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ], {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }
  }

```

```

    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDT0" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ], {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    } ],
    "id" : 2
  },
  "creator" : "creator",
  "inScoringPhase" : true,
  "inProgress" : true,
  "endDate" : "2000-01-23T04:56:07.000+00:00",
  "deckId" : 5,
  "players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "name" : "name",
  "id" : 5,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
} ],
"first" : true,
"totalElements" : 0,
"empty" : true
}

```

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*



## Responses

200

Partidas devueltas con éxito [PageMatchDTO](#)

400

Parámetro 'status' inválido. Solo se permite `lobby`, `inProgress` o `finished`

401

No autorizado (token inválido o ausente)

403

Prohibido: el usuario no tiene permisos para listar partidas en el estado solicitado

## GET /api/v1/matches/myMatches

Up

Obtener partidas del usuario actual (`findMatchesForCurrentUser`)

Devuelve una lista de partidas en las que el usuario actual ha participado, con opción de filtrar las creadas por él.

### Query parameters

#### page (optional)

*Query Parameter* — default: 0 format: int32

#### size (optional)

*Query Parameter* — default: 10 format: int32

#### onlyCreator (optional)

*Query Parameter* — default: false

### Return type

[PageMatchDTO](#)

### Example data

Content-Type: application/json

```
{
  "number" : 5,
  "size" : 1,
  "numberOfElements" : 5,
  "last" : true,
  "totalPages" : 6,
  "pageable" : {
    "paged" : true,
    "pageNumber" : 7,
    "offset" : 2,
    "pageSize" : 9,
    "unpaged" : true
  },
  "sort" : {
    "unsorted" : true,
    "sorted" : true,
    "empty" : true
  },
  "content" : [ {
    "currentTurn" : {
      "turnCount" : 1,
      "drawSource" : "NONE",
      "username" : "username"
    },
    "discard" : {
      "cards" : [ {
        "image" : "image",
        "baseValue" : 9,
        "name" : "name",
        "cardType" : "MAGO",
        "modsDTO" : [ {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        }, {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        } ],
        "id" : 7,

```

```

    "finalValue" : 3
  }, {
    "image" : "image",
    "baseValue" : 9,
    "name" : "name",
    "cardType" : "MAGO",
    "modsDT0" : [ {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    }, {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
  } ],
  "id" : 2
},
"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
} ],
"name" : "name",
"id" : 5,
"startDate" : "2000-01-23T04:56:07.000+00:00"
}, {
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,

```

```

    "name" : "name",
    "cardType" : "MAGO",
    "modsDT0" : [ {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    }, {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
  }, {
    "image" : "image",
    "baseValue" : 9,
    "name" : "name",
    "cardType" : "MAGO",
    "modsDT0" : [ {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    }, {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
  } ],
  "id" : 2
},
"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
} ],

```

```
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "name" : "name",
  "id" : 5,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
} ],
"first" : true,
"totalElements" : 0,
"empty" : true
}
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

Partidas obtenidas con Ã©xito [PageMatchDTO](#)

401

No autorizado: se necesita autenticaciÃ³n bearer para acceder a este endpoint

## POST /api/v1/matches/{id}/join

[Up](#)

Unirse a una partida (**joinMatch**)

El usuario actual se une como participante a la partida dada.

### Path parameters

**id (required)**

*Path Parameter* — format: int32

### Return type

[MatchDTO](#)

### Example data

Content-Type: application/json

```
{
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
```

```

        "description" : "description",
        "id" : 2
    }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
} ],
"id" : 2
},
"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
} ],
"name" : "name",
"id" : 5,
"startDate" : "2000-01-23T04:56:07.000+00:00"
}

```

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

**200**

Unión a la partida exitosa [MatchDTO](#)

**401**

No autorizado

**403**

El usuario tiene un jugador asociado que está en partida

**404**

Partida no encontrada

**409**

<p>La solicitud no puede ser procesada debido a un conflicto en el estado de la partida:</p> <p>Motivos:</p> <ol> <li> <p><strong>La

partida está en progreso o ha finalizado.

- Excepción generada: `MatchStateException`
- Descripción: No se pueden realizar acciones en partidas que ya están en progreso o han concluido.

La partida ha alcanzado el límite de jugadores.

- Excepción generada: `MatchStateException`
- Descripción: No se pueden añadir más jugadores a la partida porque se ha alcanzado el número máximo permitido.

500

Error al intentar unirse a la partida

DELETE /api/v1/matches/{matchId}/remove-player/{userIdToRemove}

(removePlayerFromMatch)

Path parameters

- matchId (required)

Path Parameter — format: int32
- userIdToRemove (required)

Path Parameter — format: int32

Return type

[MatchDTO](#)

Example data

Content-Type: application/json

```
{
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ], {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ], {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    } ],
    "id" : 2
  },
```

```

"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
} ],
"name" : "name",
"id" : 5,
"startDate" : "2000-01-23T04:56:07.000+00:00"
}

```

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK [MatchDTO](#)

400

Bad Request [ErrorMessage](#)

403

Forbidden [ErrorMessage](#)

404

Not Found [ErrorMessage](#)

409

Conflict [ErrorMessage](#)

500

Internal Server Error [ErrorMessage](#)

# POST /api/v1/matches/{id}/start

Up

Iniciar la partida (**startMatch**)

Solo el usuario creador de la partida puede iniciarla.

## Path parameters

**id (required)**

*Path Parameter* — format: int32

## Return type

[MatchDTO](#)

## Example data

Content-Type: application/json

```
{
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    } ],
    "id" : 2
  },
  "creator" : "creator",
  "inScoringPhase" : true,
  "inProgress" : true,
  "endDate" : "2000-01-23T04:56:07.000+00:00",
  "deckId" : 5,
  "players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
```



```
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "name" : "name",
  "id" : 5,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
}
```

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**

- 200  
Partida iniciada con Ã©xito [MatchDTO](#)
- 401  
No autorizado
- 403  
El usuario actual no es creador
- 404  
Partida no encontrada
- 409  
<p>La solicitud no puede ser procesada debido a un conflicto en el estado de la partida:</p> <p>Motivos:</p> <ol> <li> <p><strong>La partida estÃ¡ en progreso o ha finalizado.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>MatchStatesException</code></li> <li>DescripciÃ³n: No se pueden realizar acciones en partidas que ya estÃ¡n en progreso o han concluido.</li> </ul> </li> <li> <p><strong>La partida no tiene entre 3 y 6 jugadores.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>MatchStatesException</code></li> <li>DescripciÃ³n: La partida no cumple con el requisito de tener entre 3 y 6 jugadores para continuar con esta acciÃ³n.</li> </ul> </li> </ol>
- 500  
Error al comenzar la partida

**POST /api/v1/matches/{matchId}/toggle-role** [Up](#)

(toggleRole)

**Path parameters**

**matchId (required)**  
*Path Parameter* — format: int32

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**

- 200  
OK
- 400  
Bad Request [ErrorMessage](#)
- 403  
Forbidden [ErrorMessage](#)
- 404  
Not Found [ErrorMessage](#)
- 409

# Players

POST /api/v1/players/discard/{matchId}/{cardId}

Up

Descartar una carta (**discardCard**)

El jugador actual descarta la carta con ID 'cardId' en la partida especificada.

### Path parameters

- matchId (required)**  
*Path Parameter* — format: int32
- cardId (required)**  
*Path Parameter* — format: int32

### Return type

[PlayerDTO](#)

### Example data

Content-Type: application/json

```
{
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

**200**  
Carta descartada con Éxito, se devuelve el jugador actualizado [PlayerDTO](#)

**401**  
No autorizado

**403**  
Forbidden [ErrorMessage](#)

**404**  
Partida o carta no encontrada

**409**  
<p>La solicitud no puede ser procesada debido a uno o más de los siguientes motivos:</p> <p>Motivos:</p> <ol> <li> <p><strong>No hay jugador en la partida asociado al usuario registrado.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>PlayerStatesException</code></li> <li>DescripciÃ³n: El usuario registrado no estÃ¡ vinculado como jugador a esta partida.</li> </ul> </li> <li> <p><strong>La partida no estÃ¡ en progreso.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>MatchStatesException</code></li> <li>DescripciÃ³n: La partida aÃºn no ha iniciado o ya ha finalizado.</li> </ul> </li> <li> <p><strong>La partida estÃ¡ en fase de turnos especiales.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>MatchStatesException</code></li> <li>DescripciÃ³n: No es posible realizar la acciÃ³n porque la partida estÃ¡ en una fase de turnos especiales.</li> </ul> </li> <li> <p><strong>El usuario registrado no tiene el turno actual de la partida.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>TurnStatesException</code></li> <li>DescripciÃ³n: El usuario intentÃ³ realizar una acciÃ³n fuera de su turno.</li> </ul> </li> <li> <p><strong>El usuario no ha realizado acciones previas en su turno.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>TurnStatesException</code></li> <li>DescripciÃ³n: El usuario no ha completado las acciones previas de su turno (robo).</li> </ul> </li> <li> <p><strong>Ya se ha descartado una carta en el turno actual.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>TurnStatesException</code></li> <li>DescripciÃ³n: Solo se permite descartar una vez por turno.</li> </ul> </li> <li> <p><strong>La carta que se quiere descartar no pertenece a la mano del jugador</strong></p> <ul> <li>ExcepciÃ³n generada: <code>CardStatesException</code></li> <li>DescripciÃ³n: El jugador solo puede descartar una carta que estÃ© disponible en su mano.</li> </ul> </li> </ol>

**500**  
Error al intentar descartar una carta

POST /api/v1/players/drawCard/deck/{matchId}

Up

Robar carta del mazo principal (**drawCardFromDeck**)

El jugador actual roba la carta superior del mazo en la partida dada.

Path parameters

matchId (required)  
Path Parameter — format: int32

Return type  
[PlayerDTO](#)

Example data  
Content-Type: application/json

```
{
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}
```

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**  
**200**

Carta robada con Ãxito, se devuelve el jugador actualizado [PlayerDTO](#)

**401**  
No autorizado (token invÃlido o ausente)

**403**  
Forbidden [ErrorMessage](#)

**404**  
Partida no encontrada o jugador inexistente

**409**  
<p>La solicitud no puede ser procesada debido a uno o mÃ¡s de los siguientes motivos:</p> <p>Motivos:</p> <ol> <li> <p><strong>No hay jugador en la partida asociado al usuario registrado.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>PlayerStatesException</code></li> <li>DescripciÃ³n: El usuario registrado no estÃ¡ vinculado como jugador a esta partida.</li> </ul> </li> <li> <p><strong>La partida no estÃ¡ en progreso.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>MatchStatesException</code></li> <li>DescripciÃ³n: La partida aÃºn no ha iniciado o ya ha finalizado.</li> </ul> </li> <li> <p><strong>La partida estÃ¡ en fase de turnos especiales.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>MatchStatesException</code></li> <li>DescripciÃ³n: No es posible realizar la acciÃ³n porque la partida estÃ¡ en una fase de turnos especiales.</li> </ul> </li> <li> <p><strong>El usuario registrado no tiene el turno actual de la partida.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>TurnStatesException</code></li> <li>DescripciÃ³n: El usuario intentÃ³ realizar una acciÃ³n fuera de su turno.</li> </ul> </li> <li> <p><strong>El usuario ha realizado acciones previas en su turno.</strong></p> <ul> <li>ExcepciÃ³n generada: <code>TurnStatesException</code></li> <li>DescripciÃ³n: El usuario ya completÃ³ las acciones permitidas para su turno actual.</li> </ul> </li> </ol>

**500**  
Error al tratar de robar del mazo

POST /api/v1/players/drawCard/discardPile/{matchId}/{cardId} [Up](#)

Robar carta de la pila de descartes (drawCardFromDiscard)  
El jugador actual roba la carta con ID 'cardId' de la zona de descarte de la partida.

Path parameters

matchId (required)  
Path Parameter — format: int32

cardId (required)  
Path Parameter — format: int32

Return type  
[PlayerDTO](#)

Example data  
Content-Type: application/json

```
{
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}
```

```
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

200

Carta robada con Éxito, se devuelve el jugador actualizado [PlayerDTO](#)

401

No autorizado

403

Forbidden [ErrorMessage](#)

404

Partida o carta no encontrada

409

<p>La solicitud no puede ser procesada debido a uno o más de los siguientes motivos:</p> <p>Motivos:</p> <ol> <li> <p><strong>No hay jugador en la partida asociado al usuario registrado.</strong></p> <ul> <li>Excepción generada: <code>PlayerStatesException</code></li> <li>Descripción: El usuario registrado no está vinculado a esta partida.</li> </ul> </li> <li> <p><strong>La partida no está en progreso.</strong></p> <ul> <li>Excepción generada: <code>MatchStatesException</code></li> <li>Descripción: La partida aún no ha iniciado o ya ha finalizado.</li> </ul> </li> <li> <p><strong>La partida está en fase de turnos especiales.</strong></p> <ul> <li>Excepción generada: <code>MatchStatesException</code></li> <li>Descripción: No es posible realizar la acción porque la partida está en una fase de turnos especiales.</li> </ul> </li> <li> <p><strong>El usuario registrado no tiene el turno actual de la partida.</strong></p> <ul> <li>Excepción generada: <code>TurnStatesException</code></li> <li>Descripción: El usuario intentó realizar una acción fuera de su turno.</li> </ul> </li> <li> <p><strong>El usuario ha realizado acciones previas en su turno.</strong></p> <ul> <li>Excepción generada: <code>TurnStatesException</code></li> <li>Descripción: El usuario ya completó las acciones permitidas para su turno actual.</li> </ul> </li> <li> <p><strong>Se está intentando robar de la zona de descarte en el primer turno.</strong></p> <ul> <li>Excepción generada: <code>TurnStatesException</code></li> <li>Descripción: No se permite robar del descarte en el primer turno.</li> </ul> </li> </ol>

500

Error al tratar de robar de la zona de descarte

Statistics

GET /api/v1/statistics/average-match-duration

Up

Obtener duración promedio de las partidas del usuario ([getAverageMatchDuration](#))

Devuelve la duración promedio de las partidas jugadas por el usuario actual.

Return type

Double

Example data

Content-Type: application/json

```
0.8008281904610115
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

200

Duración promedio obtenida con Éxito [Double](#)

401

No autorizado: se necesita autenticación bearer para acceder a este endpoint

GET /api/v1/statistics/average-players-per-match

Up

Obtener número promedio de jugadores por partida ([getAveragePlayersPerMatch](#))

Devuelve el número promedio de jugadores por partida en las partidas jugadas por el usuario actual.

Return type

Double

Example data

Content-Type: application/json

0.8008281904610115

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

200

Promedio de jugadores por partida obtenido con Ã©xito [Double](#)

401

No autorizado: se necesita autenticaciÃ³n bearer para acceder a este endpoint

## GET /api/v1/statistics/average-points

[Up](#)

Obtener puntaje promedio del usuario ([getAveragePoints](#))

Devuelve el puntaje promedio obtenido por el usuario en las partidas en las que ha participado.

#### Return type

Double

#### Example data

Content-Type: application/json

0.8008281904610115

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

200

Puntaje promedio obtenido con Ã©xito [Double](#)

401

No autorizado: se necesita autenticaciÃ³n bearer para acceder a este endpoint

## GET /api/v1/statistics/average-ranking-position

[Up](#)

Obtener posiciÃ³n promedio en el ranking del usuario ([getAverageRankingPosition](#))

Devuelve la posiciÃ³n promedio en el ranking del usuario en las partidas que ha jugado.

#### Return type

Double

#### Example data

Content-Type: application/json

0.8008281904610115

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

200

PosiciÃ³n promedio en el ranking obtenida con Ã©xito [Double](#)

401

No autorizado: se necesita autenticaciÃ³n bearer para acceder a este endpoint

## GET /api/v1/statistics/average-turns-per-match

[Up](#)

Obtener promedio de turnos por partida del usuario ([getAverageTurnsPerMatch](#))

Devuelve el promedio de turnos jugados por el usuario en cada partida.

#### Return type

Double

#### Example data

Content-Type: application/json

```
0.8008281904610115
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

200

Promedio de turnos por partida obtenido con Ã©xito [Double](#)

401

No autorizado: se necesita autenticaciÃ³n bearer para acceder a este endpoint

## GET /api/v1/statistics/global-match-duration-average

[Up](#)

Obtener duraciÃ³n promedio global de las partidas (**getGlobalAverageMatchDuration**)

Devuelve la duraciÃ³n promedio de todas las partidas registradas en el sistema.

#### Return type

Double

#### Example data

Content-Type: application/json

```
0.8008281904610115
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

200

DuraciÃ³n promedio global obtenida con Ã©xito [Double](#)

401

No autorizado: se necesita autenticaciÃ³n bearer para acceder a este endpoint

## GET /api/v1/statistics/global-match-players-average

[Up](#)

Obtener nÃºmero promedio de jugadores por partida a nivel global (**getGlobalAveragePlayersPerMatch**)

Devuelve el promedio de jugadores registrados en todas las partidas del sistema.

#### Return type

Double

#### Example data

Content-Type: application/json

```
0.8008281904610115
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

200

NÃºmero promedio de jugadores global obtenido con Ã©xito [Double](#)

401

No autorizado: se necesita autenticaciÃ³n bearer para acceder a este endpoint

## GET /api/v1/statistics/global-match-max-duration

[Up](#)

Obtener duraciÃ³n mÃ¡xima global de las partidas (**getGlobalMaxMatchDuration**)

Devuelve la duraci3n m3xima registrada de todas las partidas en el sistema.

**Return type**  
Integer

**Example data**  
Content-Type: application/json

0

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**  
**200**  
Duraci3n m3xima global obtenida con 3xito [Integer](#)  
**401**  
No autorizado: se necesita autenticaci3n bearer para acceder a este endpoint

**GET /api/v1/statistics/global-match-min-duration** [Up](#)

Obtener duraci3n m3nima global de las partidas (**getGlobalMinMatchDuration**)  
Devuelve la duraci3n m3nima registrada de todas las partidas en el sistema.

**Return type**  
Integer

**Example data**  
Content-Type: application/json

0

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**  
**200**  
Duraci3n m3nima global obtenida con 3xito [Integer](#)  
**401**  
No autorizado: se necesita autenticaci3n bearer para acceder a este endpoint

**GET /api/v1/statistics/global-match-players-max** [Up](#)

Obtener n3mero m3ximo de jugadores por partida a nivel global (**getMaxGlobalPlayersPerMatch**)  
Devuelve el n3mero m3ximo de jugadores registrados en todas las partidas del sistema.

**Return type**  
Integer

**Example data**  
Content-Type: application/json

0

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**  
**200**  
N3mero m3ximo de jugadores global obtenido con 3xito [Integer](#)  
**401**  
No autorizado: se necesita autenticaci3n bearer para acceder a este endpoint

## GET /api/v1/statistics/max-match-duration

[Up](#)

Obtener duraci3n m3xima de las partidas jugadas (**getMaxMatchDuration**)

Devuelve la duraci3n m3xima registrada de las partidas jugadas por el usuario actual.

### Return type

Integer

### Example data

Content-Type: application/json

```
0
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

Duraci3n m3xima obtenida con 3xito [Integer](#)

401

No autorizado: se necesita autenticaci3n bearer para acceder a este endpoint

## GET /api/v1/statistics/match-players-max

[Up](#)

Obtener n3mero m3ximo de jugadores por partida del usuario (**getMaxPlayersPerMatch**)

Devuelve el n3mero m3ximo de jugadores en las partidas jugadas por el usuario actual.

### Return type

Integer

### Example data

Content-Type: application/json

```
0
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

N3mero m3ximo de jugadores obtenido con 3xito [Integer](#)

401

No autorizado: se necesita autenticaci3n bearer para acceder a este endpoint

## GET /api/v1/statistics/max-points

[Up](#)

Obtener puntaje m3ximo del usuario (**getMaxPoints**)

Devuelve el puntaje m3ximo obtenido por el usuario en las partidas en las que ha participado.

### Return type

Integer

### Example data

Content-Type: application/json

```
0
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

Puntaje m3ximo obtenido con 3xito [Integer](#)



GET /api/v1/statistics/global-match-players-min

Up

Obtener n3mero m3ximo de jugadores por partida a nivel global (**getMinGlobalPlayersPerMatch**)

Devuelve el n3mero m3ximo de jugadores registrados en todas las partidas del sistema.

Return type  
Integer

Example data  
Content-Type: application/json

0

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**  
200  
N3mero m3ximo de jugadores global obtenido con 3xito [Integer](#)  
401  
No autorizado: se necesita autenticaci3n bearer para acceder a este endpoint

GET /api/v1/statistics/min-match-duration

Up

Obtener duraci3n m3xima de las partidas jugadas (**getMinMatchDuration**)

Devuelve la duraci3n m3xima registrada de las partidas jugadas por el usuario actual.

Return type  
Integer

Example data  
Content-Type: application/json

0

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**  
200  
Duraci3n m3xima obtenida con 3xito [Integer](#)  
401  
No autorizado: se necesita autenticaci3n bearer para acceder a este endpoint

GET /api/v1/statistics/match-players-min

Up

Obtener n3mero m3ximo de jugadores por partida del usuario (**getMinPlayersPerMatch**)

Devuelve el n3mero m3ximo de jugadores en las partidas jugadas por el usuario actual.

Return type  
Integer

Example data  
Content-Type: application/json

0

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

200

Número mínimo de jugadores obtenido con éxito [Integer](#)

401

No autorizado: se necesita autenticación bearer para acceder a este endpoint

GET /api/v1/statistics/min-points

Up

Obtener puntaje mínimo del usuario (**getMinPoints**)

Devuelve el puntaje mínimo obtenido por el usuario en las partidas en las que ha participado.

Return type

Integer

Example data

Content-Type: application/json

```
0
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

200

Puntaje mínimo obtenido con éxito [Integer](#)

401

No autorizado: se necesita autenticación bearer para acceder a este endpoint

GET /api/v1/statistics/most-frequent-cards

Up

Obtener cartas más frecuentes en las manos finales del usuario (**getMostFrequentCards**)

Devuelve una lista con los nombres de las cartas más frecuentes en las manos finales del usuario.

Return type

array[String]

Example data

Content-Type: application/json

```
[ "", "" ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

200

Cartas más frecuentes obtenidas con éxito

401

No autorizado: se necesita autenticación bearer para acceder a este endpoint

GET /api/v1/statistics/global-ranking

Up

Obtener cartas más frecuentes en las manos finales del usuario (**getRanking**)

Devuelve una lista con los nombres de las cartas más frecuentes en las manos finales del usuario.

Query parameters

sortBy (optional)

Query Parameter — default: WINS

Return type

map[String, array[Integer]]

Example data

Content-Type: application/json

```
{
  "key" : [ 0, 0 ]
}
```

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**

**200**  
Ranking obtenido con Ã©xito

**400**  
ParÃ¡metro 'sortBy' invalido. Solo se permite 'WINS' y 'POINTS' [ErrorMessage](#)

**401**  
No autorizado: se necesita autenticaciÃ³n bearer para acceder a este endpoint

**403**  
Forbidden [ErrorMessage](#)

**404**  
Not Found [ErrorMessage](#)

**409**  
Conflict [ErrorMessage](#)

**500**  
Internal Server Error [ErrorMessage](#)

GET /api/v1/statistics/total-match-duration

[Up](#)

Obtener duraciÃ³n total de las partidas jugadas (**getTotalMatchDuration**)

Devuelve la suma de las duraciones de las partidas jugadas por el usuario actual.

**Return type**

Integer

**Example data**

Content-Type: application/json

0

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**

**200**  
DuraciÃ³n total obtenida con Ã©xito [Integer](#)

**401**  
No autorizado: se necesita autenticaciÃ³n bearer para acceder a este endpoint

GET /api/v1/statistics/total-matches

[Up](#)

Obtener total de partidas jugadas por el usuario (**getTotalMatches**)

Devuelve el nÃºmero total de partidas jugadas por el usuario actual.

**Return type**

Integer

**Example data**

Content-Type: application/json

0

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**

**200**  
Total de partidas obtenidas con Ã©xito [Integer](#)

401  
No autorizado: se necesita autenticaci3n bearer para acceder a este endpoint

GET /api/v1/statistics/win-percentageUp

Obtener porcentaje de victorias del usuario (**getWinPercentage**)  
Devuelve el porcentaje de victorias del usuario en todas las partidas que ha jugado.

Return type  
Double

Example data  
Content-Type: application/json

0.8008281904610115

Produces  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses  
200  
Porcentaje de victorias obtenido con 3xito Double  
401  
No autorizado: se necesita autenticaci3n bearer para acceder a este endpoint

Turns

POST /api/v1/turns/{matchId}/cancelTurnUp

Cancelar turno actual (**cancelTurn**)  
Anula las acciones realizadas en el turno en curso de la partida, si el usuario es quien est3 jugando el turno.

Path parameters  
matchId (required)  
Path Parameter — format: int32

Responses  
200  
Turno cancelado con 3xito  
401  
No autorizado  
404  
Partida no encontrada  
409  
<p>La solicitud no puede ser procesada debido a uno o m3s de los siguientes motivos:</p> <p>Motivos:</p> <ol> <li> <p><strong>El usuario no posee el turno actual de la partida</strong></p> <ul> <li>Excepci3n generada: <code>TurnStatesException</code></li> <li>Descripci3n: El usuario registrado no posee el turno actual de la partida.</li> </ul> </li> <li> <p><strong>La partida no est3 en progreso.</strong></p> <ul> <li>Excepci3n generada: <code>MatchStatesException</code></li> <li>Descripci3n: La partida a3n no ha iniciado o ya ha finalizado.</li> </ul> </li> </ol>  
500  
Error al intentar cancelar el turno

POST /api/v1/turns/{matchId}/specialTurnUp

Turno especial (fase de scoring) (**specialTurn**)  
Aplica las decisiones din3micas (modificadores) del jugador en el 3ltimo tramo de la partida (si tiene).

Path parameters  
matchId (required)  
Path Parameter — format: int32

Consumes  
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [DecisionRequest](#) (required)  
*Body Parameter* —

Responses

200  
Turno especial procesado con Éxito

401  
No autorizado

404  
Partida no encontrada

409  
<p>La solicitud no puede ser procesada debido a uno o más de los siguientes motivos:</p> <p>Motivos:</p> <ol> <li> <p><strong>El usuario no posee el turno actual de la partida</strong></p> <ul> <li>Excepci3n generada: <code>TurnStatesException</code></li> <li>Descripci3n: El usuario registrado no posee el turno actual de la partida.</li> </ul> </li> <li> <p><strong>La partida no estÁ en fase de turnos especiales (scoring).</strong></p> <ul> <li>Excepci3n generada: <code>MatchStatesException</code></li> <li>Descripci3n: La partida aÃn no estÁ en fase de turnos especiales.</li> </ul> </li> </ol>

500  
Error al intentar aplicar el turno especial

UserConnection

POST /api/v1/heartbeat [Up](#)

Actualizar heartbeat (**updateHeartbeat**)  
Marca que el usuario autenticado sigue conectado, actualizando su Éltima conexi3n.

**Produces**  
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**

200  
Heartbeat actualizado correctamente.

400  
Bad Request [ErrorMessage](#)

401  
Usuario no autenticado o token invÁlido.

403  
Forbidden [ErrorMessage](#)

404  
Not Found [ErrorMessage](#)

409  
Conflict [ErrorMessage](#)

500  
Internal Server Error [ErrorMessage](#)

Users

POST /api/v1/users [Up](#)

Crear usuario (**create**)  
Crea un nuevo usuario en el sistema. Debes ser administrador para acceder a este endpoint.

**Consumes**  
This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [UserUpdateProfileDTO](#) (required)  
*Body Parameter* —

**Return type**  
[User](#)

## Example data

Content-Type: application/json

```
{
  "password" : "password",
  "achievements" : [ {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  }, {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  } ],
  "players" : [ {
    "score" : 5,
    "role" : "CREADOR",
    "inLobbyOrPlaying" : true,
    "playerHand" : [ {
      "image" : "image",
      "mods" : [ {
        "primaryValue" : 3,
        "modType" : "PENALTY",
        "secondaryValue" : 2,
        "description" : "description",
        "id" : 9,
        "target" : [ null, null ]
      }, {
        "primaryValue" : 3,
        "modType" : "PENALTY",
        "secondaryValue" : 2,
        "description" : "description",
        "id" : 9,
        "target" : [ null, null ]
      } ],
      "baseValue" : 0,
      "name" : "name",
      "cardType" : "MAGO",
      "id" : 5,
      "finalValue" : 7
    }, {
      "image" : "image",
      "mods" : [ {
        "primaryValue" : 3,
        "modType" : "PENALTY",
        "secondaryValue" : 2,
        "description" : "description",
        "id" : 9,
        "target" : [ null, null ]
      }, {
        "primaryValue" : 3,
        "modType" : "PENALTY",
        "secondaryValue" : 2,
        "description" : "description",
        "id" : 9,
        "target" : [ null, null ]
      } ],
      "baseValue" : 0,
      "name" : "name",
      "cardType" : "MAGO",
      "id" : 5,
      "finalValue" : 7
    } ],
    "id" : 1,
    "matchPlayed" : {
      "currentTurn" : {
```

```

        "discarded" : true,
        "drawSource" : "NONE",
        "turnCount" : 1,
        "id" : 1
    },
    "discard" : {
        "cards" : [ null, null ],
        "id" : 1
    },
    "inProgress" : true,
    "endDate" : "2000-01-23T04:56:07.000+00:00",
    "players" : [ null, null ],
    "deck" : {
        "cards" : [ null, null ],
        "id" : 7,
        "initialCards" : [ null, null ],
        "empty" : true
    },
    "scoringStartTurn" : 6,
    "finished" : true,
    "maxSpecialTurn" : 7,
    "inScoringPhase" : true,
    "name" : "name",
    "id" : 4,
    "startDate" : "2000-01-23T04:56:07.000+00:00"
}
}, {
    "score" : 5,
    "role" : "CREADOR",
    "inLobbyOrPlaying" : true,
    "playerHand" : [ {
        "image" : "image",
        "mods" : [ {
            "primaryValue" : 3,
            "modType" : "PENALTY",
            "secondaryValue" : 2,
            "description" : "description",
            "id" : 9,
            "target" : [ null, null ]
        }, {
            "primaryValue" : 3,
            "modType" : "PENALTY",
            "secondaryValue" : 2,
            "description" : "description",
            "id" : 9,
            "target" : [ null, null ]
        } ],
        "baseValue" : 0,
        "name" : "name",
        "cardType" : "MAGO",
        "id" : 5,
        "finalValue" : 7
    }, {
        "image" : "image",
        "mods" : [ {
            "primaryValue" : 3,
            "modType" : "PENALTY",
            "secondaryValue" : 2,
            "description" : "description",
            "id" : 9,
            "target" : [ null, null ]
        }, {
            "primaryValue" : 3,
            "modType" : "PENALTY",
            "secondaryValue" : 2,
            "description" : "description",
            "id" : 9,
            "target" : [ null, null ]
        } ],
        "baseValue" : 0,
        "name" : "name",
        "cardType" : "MAGO",
        "id" : 5,
        "finalValue" : 7
    } ],
    "id" : 1,
    "matchPlayed" : {
        "currentTurn" : {
            "discarded" : true,

```

```
    "drawSource" : "NONE",
    "turnCount" : 1,
    "id" : 1
  },
  "discard" : {
    "cards" : [ null, null ],
    "id" : 1
  },
  "inProgress" : true,
  "endDate" : "2000-01-23T04:56:07.000+00:00",
  "players" : [ null, null ],
  "deck" : {
    "cards" : [ null, null ],
    "id" : 7,
    "initialCards" : [ null, null ],
    "empty" : true
  },
  "scoringStartTurn" : 6,
  "finished" : true,
  "maxSpecialTurn" : 7,
  "inScoringPhase" : true,
  "name" : "name",
  "id" : 4,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
}
} ],
"authority" : {
  "authority" : "authority",
  "id" : 6
},
"playing" : true,
"id" : 0,
"avatar" : "avatar",
"email" : "email",
"username" : "username"
}
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

201

Usuario creado con Ã©xito [User](#)

400

Error al validar los datos para la creaci3n del usuario

403

Acceso denegado: Se necesita tener permisos de administrador para acceder a este endpoint

409

El usuario no puede ser creado porque los datos proporcionados generan un conflicto

500

Error al intentar crear el usuario

## DELETE /api/v1/users/{userId}

Up

Eliminar usuario (**delete**)

Elimina un usuario del sistema por su ID. Debes ser administrador o el propio usuario a eliminar para acceder a este endpoint.

### Path parameters

**userId (required)**

*Path Parameter* — format: int32

### Return type

[MessageResponse](#)

### Example data

Content-Type: application/json

```
{
  "message" : "message"
}
```

### Produces



This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

Responses

- 200  
Usuario eliminado con Ã©xito [MessageResponse](#)
- 401  
No autorizado
- 403  
Acceso denegado: El usuario no tiene permisos para realizar esta acciÃ³n
- 404  
Usuario no encontrado
- 500  
Error al intentar eliminar el usuario

GET /api/v1/users

Up

Listar usuarios (findAll)

Devuelve una lista paginada de usuarios, permitiendo ordenar y filtrar por diferentes criterios. Debes ser administrador para acceder a este endpoint.

Query parameters

- page (optional)  
Query Parameter — default: 0 format: int32
- size (optional)  
Query Parameter — default: 5 format: int32
- sortBy (optional)  
Query Parameter — default: username
- sortDirection (optional)  
Query Parameter — default: asc

Return type

[PageUserDTO](#)

Example data

Content-Type: application/json

```
{
  "number" : 5,
  "size" : 1,
  "numberOfElements" : 5,
  "last" : true,
  "totalPages" : 6,
  "pageable" : {
    "paged" : true,
    "pageNumber" : 7,
    "offset" : 2,
    "pageSize" : 9,
    "unpaged" : true
  },
  "sort" : {
    "unsorted" : true,
    "sorted" : true,
    "empty" : true
  },
  "content" : [ {
    "password" : "password",
    "achievements" : [ {
      "condition" : "WIN_N_GAMES",
      "tier" : "FACIL",
      "extraData" : "extraData",
      "icon" : "icon",
      "description" : "description",
      "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
      "id" : 6,
      "type" : "PROGRESO",
      "requiredValue" : 0
    }, {
      "condition" : "WIN_N_GAMES",
      "tier" : "FACIL",
      "extraData" : "extraData",
```

```

    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  } ],
  "authority" : "authority",
  "id" : 0,
  "avatar" : "avatar",
  "matches" : [ {
    "currentTurn" : {
      "turnCount" : 1,
      "drawSource" : "NONE",
      "username" : "username"
    },
    "discard" : {
      "cards" : [ {
        "image" : "image",
        "baseValue" : 9,
        "name" : "name",
        "cardType" : "MAGO",
        "modsDT0" : [ {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        }, {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        } ],
        "id" : 7,
        "finalValue" : 3
      }, {
        "image" : "image",
        "baseValue" : 9,
        "name" : "name",
        "cardType" : "MAGO",
        "modsDT0" : [ {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        }, {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        } ],
        "id" : 7,
        "finalValue" : 3
      } ],
      "id" : 2
    },
    "creator" : "creator",
    "inScoringPhase" : true,
    "inProgress" : true,
    "endDate" : "2000-01-23T04:56:07.000+00:00",
    "deckId" : 5,
    "players" : [ {
      "score" : 1,
      "playerHand" : [ null, null ],
      "id" : 1,
      "rol" : "CREADOR",
      "username" : "username"
    }, {
      "score" : 1,
      "playerHand" : [ null, null ],
      "id" : 1,
      "rol" : "CREADOR",
      "username" : "username"
    }, {
      "score" : 1,

```

```

    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "name" : "name",
  "id" : 5,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
}, {
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
  }, {
    "image" : "image",
    "baseValue" : 9,
    "name" : "name",
    "cardType" : "MAGO",
    "modsDTO" : [ {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    } ],
    "primaryValue" : 4,
    "modType" : "PENALTY",
    "secondaryValue" : 7,
    "description" : "description",
    "id" : 2
  } ],
  "id" : 7,
  "finalValue" : 3
} ],
  "id" : 2
},
  "creator" : "creator",
  "inScoringPhase" : true,
  "inProgress" : true,

```

```

"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
} ],
"name" : "name",
"id" : 5,
"startDate" : "2000-01-23T04:56:07.000+00:00"
} ],
"email" : "email",
"username" : "username"
}, {
  "password" : "password",
  "achievements" : [ {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  }, {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  } ],
  "authority" : "authority",
  "id" : 0,
  "avatar" : "avatar",
  "matches" : [ {
    "currentTurn" : {
      "turnCount" : 1,
      "drawSource" : "NONE",
      "username" : "username"
    },
    "discard" : {
      "cards" : [ {

```

```

      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAG0",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAG0",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    } ],
    "id" : 2
  },
  "creator" : "creator",
  "inScoringPhase" : true,
  "inProgress" : true,
  "endDate" : "2000-01-23T04:56:07.000+00:00",
  "deckId" : 5,
  "players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "score" : 1,

```

```

    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "name" : "name",
  "id" : 5,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
}, {
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDT0" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDT0" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    } ],
    "id" : 2
  },
  "creator" : "creator",
  "inScoringPhase" : true,
  "inProgress" : true,
  "endDate" : "2000-01-23T04:56:07.000+00:00",
  "deckId" : 5,
  "players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,

```

```

        "rol" : "CREADOR",
        "username" : "username"
    }, {
        "score" : 1,
        "playerHand" : [ null, null ],
        "id" : 1,
        "rol" : "CREADOR",
        "username" : "username"
    }, {
        "score" : 1,
        "playerHand" : [ null, null ],
        "id" : 1,
        "rol" : "CREADOR",
        "username" : "username"
    }, {
        "score" : 1,
        "playerHand" : [ null, null ],
        "id" : 1,
        "rol" : "CREADOR",
        "username" : "username"
    } ],
    "name" : "name",
    "id" : 5,
    "startDate" : "2000-01-23T04:56:07.000+00:00"
} ],
"email" : "email",
"username" : "username"
} ],
"first" : true,
"totalElements" : 0,
"empty" : true
}

```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

##### 200

Lista de usuarios obtenida con Ã©xito [PageUserDTO](#)

##### 401

No autorizado

##### 403

Forbidden [ErrorMessage](#)

## GET /api/v1/users/authorities

Up

Listar autoridades ([findAllAuths](#))

Devuelve una lista de todas las autoridades disponibles en el sistema. Debes ser administrador para acceder a este endpoint.

#### Return type

array[[Authorities](#)]

#### Example data

Content-Type: application/json

```

[ {
    "authority" : "authority",
    "id" : 6
}, {
    "authority" : "authority",
    "id" : 6
} ]

```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

##### 200

Lista de autoridades obtenida con Ã©xito

##### 401

## GET /api/v1/users/{id}

[Up](#)

Buscar usuario por ID (`findById`)

Devuelve los detalles de un usuario específico por su ID. Debes ser administrador o el propio usuario buscado para acceder a este endpoint.

### Path parameters

**id (required)**

*Path Parameter* — format: int32

### Return type

[UserDTO](#)

### Example data

Content-Type: application/json

```
{
  "password" : "password",
  "achievements" : [ {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  }, {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  } ],
  "authority" : "authority",
  "id" : 0,
  "avatar" : "avatar",
  "matches" : [ {
    "currentTurn" : {
      "turnCount" : 1,
      "drawSource" : "NONE",
      "username" : "username"
    },
    "discard" : {
      "cards" : [ {
        "image" : "image",
        "baseValue" : 9,
        "name" : "name",
        "cardType" : "MAGO",
        "modsDTO" : [ {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        }, {
          "primaryValue" : 4,
          "modType" : "PENALTY",
          "secondaryValue" : 7,
          "description" : "description",
          "id" : 2
        } ],
        "id" : 7,
        "finalValue" : 3
      }, {
        "image" : "image",
        "baseValue" : 9,
```



```

    "name" : "name",
    "cardType" : "MAGO",
    "modsDT0" : [ {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    }, {
      "primaryValue" : 4,
      "modType" : "PENALTY",
      "secondaryValue" : 7,
      "description" : "description",
      "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
  } ],
  "id" : 2
},
"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
}, {
  "score" : 1,
  "playerHand" : [ null, null ],
  "id" : 1,
  "rol" : "CREADOR",
  "username" : "username"
} ],
"name" : "name",
"id" : 5,
"startDate" : "2000-01-23T04:56:07.000+00:00"
}, {
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDT0" : [ {
        "primaryValue" : 4,

```

```

        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
    }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
}, {
    "image" : "image",
    "baseValue" : 9,
    "name" : "name",
    "cardType" : "MAGO",
    "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
    }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
    } ],
    "id" : 7,
    "finalValue" : 3
} ],
    "id" : 2
},
"creator" : "creator",
"inScoringPhase" : true,
"inProgress" : true,
"endDate" : "2000-01-23T04:56:07.000+00:00",
"deckId" : 5,
"players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
}, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
} ],
"name" : "name",

```

```
    "id" : 5,
    "startDate" : "2000-01-23T04:56:07.000+00:00"
  } ],
  "email" : "email",
  "username" : "username"
}
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

Usuario encontrado con Ã©xito [UserDTO](#)

403

Acceso denegado: Se necesita tener permisos de administrador o ser el propio usuario para acceder a este endpoint

404

Usuario no encontrado

## PUT /api/v1/users/{userId}

[Up](#)

Actualizar usuario (**update**)

Actualiza los datos de un usuario existente en el sistema. Debes ser administrador o el propio usuario a actualizar para acceder a este endpoint.

### Path parameters

**userId (required)**

*Path Parameter* — format: int32

### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

### Request body

body [UserUpdateProfileDTO](#) (required)

*Body Parameter* —

### Return type

[UserDTO](#)

### Example data

Content-Type: application/json

```
{
  "password" : "password",
  "achievements" : [ {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  }, {
    "condition" : "WIN_N_GAMES",
    "tier" : "FACIL",
    "extraData" : "extraData",
    "icon" : "icon",
    "description" : "description",
    "dateAchieved" : "2000-01-23T04:56:07.000+00:00",
    "id" : 6,
    "type" : "PROGRESO",
    "requiredValue" : 0
  } ],
  "authority" : "authority",
  "id" : 0,
  "avatar" : "avatar",
  "matches" : [ {
    "currentTurn" : {
```

```

    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDT0" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDT0" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      }, {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    } ],
    "id" : 2
  },
  "creator" : "creator",
  "inScoringPhase" : true,
  "inProgress" : true,
  "endDate" : "2000-01-23T04:56:07.000+00:00",
  "deckId" : 5,
  "players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,

```

```

    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "name" : "name",
  "id" : 5,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
}, {
  "currentTurn" : {
    "turnCount" : 1,
    "drawSource" : "NONE",
    "username" : "username"
  },
  "discard" : {
    "cards" : [ {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ], {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    }, {
      "image" : "image",
      "baseValue" : 9,
      "name" : "name",
      "cardType" : "MAGO",
      "modsDTO" : [ {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ], {
        "primaryValue" : 4,
        "modType" : "PENALTY",
        "secondaryValue" : 7,
        "description" : "description",
        "id" : 2
      } ],
      "id" : 7,
      "finalValue" : 3
    } ],
    "id" : 2
  },
  "creator" : "creator",
  "inScoringPhase" : true,
  "inProgress" : true,
  "endDate" : "2000-01-23T04:56:07.000+00:00",
  "deckId" : 5,
  "players" : [ {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,

```

```
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  }, {
    "score" : 1,
    "playerHand" : [ null, null ],
    "id" : 1,
    "rol" : "CREADOR",
    "username" : "username"
  } ],
  "name" : "name",
  "id" : 5,
  "startDate" : "2000-01-23T04:56:07.000+00:00"
} ],
"email" : "email",
"username" : "username"
}
```

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

**Responses**

- 200**  
Usuario actualizado con Ã©xito [UserDTO](#)
- 401**  
No autorizado
- 403**  
Acceso denegado: El usuario no tiene permisos para realizar esta acciÃ³n
- 404**  
Usuario no encontrado
- 500**  
Error al intentar actualizar el usuario

**Models**

[ [Jump to Methods](#) ]

**Table of Contents**

1. [Achievement](#)
2. [AchievementDTO](#)
3. [Authorities](#)
4. [Card](#)
5. [CardDTO](#)
6. [DecisionDTO](#)
7. [DecisionRequest](#)
8. [Deck](#)
9. [DeckDTO](#)
10. [Developer](#)
11. [Discard](#)
12. [DiscardDTO](#)
13. [ErrorMessage](#)
14. [FriendshipRequestDTO](#)
15. [GameInvitationDTO](#)
16. [LoginRequest](#)
17. [Match](#)
18. [MatchDTO](#)

- 19. [MessageResponse](#)
- 20. [Mod](#)
- 21. [ModDTO](#)
- 22. [PageMatchDTO](#)
- 23. [PageUserDTO](#)
- 24. [PageableObject](#)
- 25. [Player](#)
- 26. [PlayerDTO](#)
- 27. [SignupRequest](#)
- 28. [SortObject](#)
- 29. [Turn](#)
- 30. [TurnDTO](#)
- 31. [User](#)
- 32. [UserDTO](#)
- 33. [UserUpdateProfileDTO](#)

Achievement

[Up](#)

id (optional)  
[Integer](#) format: int32

description  
[String](#)

icon (optional)  
[String](#)

condition  
[String](#)  
Enum:  
WIN\_N\_GAMES  
PLAY\_N\_GAMES  
WIN\_STREAK  
WIN\_WITH\_SPECIFIC\_CARDS  
WIN\_WITH\_MIN\_POINTS  
WIN\_WITH\_NO\_RARE\_CARDS  
WIN\_AFTER\_LAST\_PLACE  
PLAY\_N\_CARDS\_OF\_TYPE

requiredValue (optional)  
[Integer](#) format: int32

extraData (optional)  
[String](#)

tier  
[String](#)  
Enum:  
FACIL  
INTERMEDIO  
DIFICIL

dateAchieved (optional)  
[Date](#) format: date-time

type  
[String](#)  
Enum:  
PROGRESO  
HABILIDAD  
SOCIAL

AchievementDTO

[Up](#)

id (optional)  
[Integer](#) format: int32

description  
[String](#)

condition  
[String](#)  
Enum:  
WIN\_N\_GAMES  
PLAY\_N\_GAMES  
WIN\_STREAK  
WIN\_WITH\_SPECIFIC\_CARDS  
WIN\_WITH\_MIN\_POINTS

WIN\_WITH\_NO\_RARE\_CARDS  
WIN\_AFTER\_LAST\_PLACE  
PLAY\_N\_CARDS\_OF\_TYPE

**requiredValue (optional)**

[\*Integer\*](#) format: int32

**extraData (optional)**

[\*String\*](#)

**tier**

[\*String\*](#)

Enum:

*FACIL*

*INTERMEDIO*

*DIFICIL*

**type**

[\*String\*](#)

Enum:

*PROGRESO*

*HABILIDAD*

*SOCIAL*

## Authorities

[Up](#)

**id (optional)**

[\*Integer\*](#) format: int32

**authority (optional)**

[\*String\*](#)

## Card

[Up](#)

**id (optional)**

[\*Integer\*](#) format: int32

**name (optional)**

[\*String\*](#)

**cardType (optional)**

[\*String\*](#)

Enum:

*MAGO*

*BESTIA*

*EJERCITO*

*LIDER*

*INUNDACION*

*ARMA*

*LLAMA*

*SALVAJE*

*TIERRA*

*TIEMPO*

*ARTEFACTO*

**image**

[\*String\*](#)

**baseValue**

[\*Integer\*](#) format: int32

**finalValue**

[\*Integer\*](#) format: int32

**mods (optional)**

[\*array\[Mod\]\*](#)

**clone (optional)**

[\*Card\*](#)

## CardDTO

[Up](#)

**id (optional)**

[\*Integer\*](#) format: int32

**name (optional)**

[\*String\*](#)



image (optional)

[\*String\*](#)

cardType (optional)

[\*String\*](#)

Enum:

*MAGO*

*BESTIA*

*EJERCITO*

*LIDER*

*INUNDACION*

*ARMA*

*LLAMA*

*SALVAJE*

*TIERRA*

*TIEMPO*

*ARTEFACTO*

baseValue (optional)

[\*Integer\*](#) format: int32

finalValue (optional)

[\*Integer\*](#) format: int32

modsDTO (optional)

[\*array\[ModDTO\]\*](#)

## DecisionDTO

[Up](#)

modId (optional)

[\*Integer\*](#) format: int32

cardId (optional)

[\*Integer\*](#) format: int32

cardType (optional)

[\*String\*](#)

Enum:

*MAGO*

*BESTIA*

*EJERCITO*

*LIDER*

*INUNDACION*

*ARMA*

*LLAMA*

*SALVAJE*

*TIERRA*

*TIEMPO*

*ARTEFACTO*

## DecisionRequest

[Up](#)

decisions (optional)

[\*array\[DecisionDTO\]\*](#)

## Deck

[Up](#)

id (optional)

[\*Integer\*](#) format: int32

cards

[\*array\[Card\]\*](#)

initialCards

[\*array\[Card\]\*](#)

empty (optional)

[\*Boolean\*](#)

## DeckDTO

[Up](#)

cards (optional)

[\*array\[CardDTO\]\*](#)

initialCards (optional)

[array\[CardDTO\]](#)

Developer

[Up](#)

- name (optional)  
[String](#)
- email (optional)  
[String](#)
- url (optional)  
[String](#)
- organization (optional)  
[String](#)
- organizationUrl (optional)  
[String](#)
- roles (optional)  
[array\[String\]](#)
- timezone (optional)  
[String](#)
- properties (optional)  
[map\[String, String\]](#)
- id (optional)  
[String](#)

Discard

[Up](#)

- id (optional)  
[Integer](#) format: int32
- cards  
[array\[Card\]](#)

DiscardDTO

[Up](#)

- id (optional)  
[Integer](#) format: int32
- cards (optional)  
[array\[CardDTO\]](#)

ErrorMessage

[Up](#)

- statusCode (optional)  
[Integer](#) format: int32
- timestamp (optional)  
[Date](#) format: date-time
- message (optional)  
[String](#)
- description (optional)  
[String](#)

FriendshipRequestDTO

[Up](#)

- senderName (optional)  
[String](#)
- receiverName (optional)  
[String](#)
- status (optional)  
[String](#)
- senderIsOnline (optional)  
[Boolean](#)
- receiverIsOnline (optional)  
[Boolean](#)

senderId (optional)  
[Integer](#) format: int32

receiverId (optional)  
[Integer](#) format: int32

GameInvitationDTO

[Up](#)

senderUsername (optional)  
[String](#)

receiverUsername (optional)  
[String](#)

status (optional)  
[Boolean](#)

matchId (optional)  
[Integer](#) format: int32

LoginRequest

[Up](#)

username  
[String](#)

password  
[String](#)

Match

[Up](#)

id (optional)  
[Integer](#) format: int32

name (optional)  
[String](#)

startDate (optional)  
[Date](#) format: date-time

endDate (optional)  
[Date](#) format: date-time

deck  
[Deck](#)

discard  
[Discard](#)

players  
[array\[Player\]](#)

currentTurn (optional)  
[Turn](#)

inScoringPhase (optional)  
[Boolean](#)

scoringStartTurn (optional)  
[Integer](#) format: int32

finished (optional)  
[Boolean](#)

creator (optional)  
[Player](#)

inProgress (optional)  
[Boolean](#)

maxSpecialTurn (optional)  
[Integer](#) format: int32

MatchDTO

[Up](#)

id (optional)  
[Integer](#) format: int32

name (optional)  
[String](#)

startDate (optional)  
[Date](#) format: date-time

endDate (optional)  
[Date](#) format: date-time

creator  
[String](#)

inProgress  
[Boolean](#)

deckId  
[Integer](#) format: int32

discard  
[DiscardDTO](#)

currentTurn (optional)  
[TurnDTO](#)

players  
[array\[PlayerDTO\]](#)

inScoringPhase (optional)  
[Boolean](#)

MessageResponse

[Up](#)

message (optional)  
[String](#)

Mod

[Up](#)

id (optional)  
[Integer](#) format: int32

description  
[String](#)

primaryValue (optional)  
[Integer](#) format: int32

secondaryValue (optional)  
[Integer](#) format: int32

originCard (optional)  
[Card](#)

target (optional)  
[array\[Card\]](#)

modType (optional)  
[String](#)  
Enum:  
PENALTY  
BONUS  
CLEAR  
BLANK  
STATE  
ALLHAND  
PENALTY\_AND\_BLANK  
NECROMANCER

clone (optional)  
[Mod](#)

ModDTO

[Up](#)

id (optional)  
[Integer](#) format: int32

description  
[String](#)

primaryValue (optional)  
[Integer](#) format: int32

secondaryValue (optional)  
[Integer](#) format: int32

modType (optional)

[String](#)

Enum:

*PENALTY*

*BONUS*

*CLEAR*

*BLANK*

*STATE*

*ALLHAND*

*PENALTY\_AND\_BLANK*

*NECROMANCER*

## PageMatchDTO

[Up](#)

totalElements (optional)

[Long](#) format: int64

totalPages (optional)

[Integer](#) format: int32

size (optional)

[Integer](#) format: int32

content (optional)

[array\[MatchDTO\]](#)

number (optional)

[Integer](#) format: int32

sort (optional)

[SortObject](#)

numberOfElements (optional)

[Integer](#) format: int32

pageable (optional)

[PageableObject](#)

first (optional)

[Boolean](#)

last (optional)

[Boolean](#)

empty (optional)

[Boolean](#)

## PageUserDTO

[Up](#)

totalElements (optional)

[Long](#) format: int64

totalPages (optional)

[Integer](#) format: int32

size (optional)

[Integer](#) format: int32

content (optional)

[array\[UserDTO\]](#)

number (optional)

[Integer](#) format: int32

sort (optional)

[SortObject](#)

numberOfElements (optional)

[Integer](#) format: int32

pageable (optional)

[PageableObject](#)

first (optional)

[Boolean](#)

last (optional)

[Boolean](#)

empty (optional)

[Boolean](#)

offset (optional)  
[Long](#) format: int64

sort (optional)  
[SortObject](#)

unpaged (optional)  
[Boolean](#)

pageNumber (optional)  
[Integer](#) format: int32

paged (optional)  
[Boolean](#)

pageSize (optional)  
[Integer](#) format: int32

id (optional)  
[Integer](#) format: int32

score (optional)  
[Integer](#) format: int32

role  
[String](#)  
Enum:  
CREADOR  
PARTICIPANTE  
GANADOR  
ESPECTADOR

playerHand  
[array\[Card\]](#)

matchPlayed  
[Match](#)

user  
[User](#)

inLobbyOrPlaying (optional)  
[Boolean](#)

id (optional)  
[Integer](#) format: int32

username (optional)  
[String](#)

score (optional)  
[Integer](#) format: int32

playerHand (optional)  
[array\[CardDTO\]](#)

rol (optional)  
[String](#)  
Enum:  
CREADOR  
PARTICIPANTE  
GANADOR  
ESPECTADOR

username  
[String](#)

authority  
[String](#)

password

[\*String\*](#)

SortObject

[Up](#)

empty (optional)  
[\*Boolean\*](#)

unsorted (optional)  
[\*Boolean\*](#)

sorted (optional)  
[\*Boolean\*](#)

Turn

[Up](#)

id (optional)  
[\*Integer\*](#) format: int32

player (optional)  
[\*Player\*](#)

drawSource  
[\*String\*](#)  
Enum:  
    *NONE*  
    *DECK*  
    *DISCARD*

cardDrawn (optional)  
[\*Card\*](#)

turnCount (optional)  
[\*Integer\*](#) format: int32

discarded (optional)  
[\*Boolean\*](#)

TurnDTO

[Up](#)

username (optional)  
[\*String\*](#)

turnCount (optional)  
[\*Integer\*](#) format: int32

drawSource (optional)  
[\*String\*](#)  
Enum:  
    *NONE*  
    *DECK*  
    *DISCARD*

User

[Up](#)

id (optional)  
[\*Integer\*](#) format: int32

username (optional)  
[\*String\*](#)

password (optional)  
[\*String\*](#)

avatar (optional)  
[\*String\*](#)

email (optional)  
[\*String\*](#)

achievements  
[\*array\[Achievement\]\*](#)

authority  
[\*Authorities\*](#)

players (optional)  
[\*array\[Player\]\*](#)

playing (optional)  
[\*Boolean\*](#)

## UserDT0

[Up](#)

id (optional)  
[\*Integer\*](#) format: int32

username (optional)  
[\*String\*](#)

avatar (optional)  
[\*String\*](#)

email  
[\*String\*](#)

achievements (optional)  
[\*array\[Achievement\]\*](#)

matches (optional)  
[\*array\[MatchDTO\]\*](#)

authority (optional)  
[\*String\*](#)

password (optional)  
[\*String\*](#)

## UserUpdateProfileDTO

[Up](#)

username (optional)  
[\*String\*](#)

password (optional)  
[\*String\*](#)

email (optional)  
[\*String\*](#)

avatar (optional)  
[\*String\*](#)

authority (optional)  
[\*String\*](#)