# Descripción

*SavingThrow* es una API orientada a organizar partidas para el juego de rol *Dungeons and Dragons.* Dado que esta API es un proyecto y no es real, se han dejado muchos detalles atrás por razones de complejidad.

# Recursos

Podemos distinguir 3 recursos: usuarios, personajes y partidas.

## Usuarios

Un usuario puede crear uno o varios personajes para interpretar, partidas para jugar y grupos de usuarios. Los objetos de un usuario serían los siguientes:

* Identificación única: cada usuario debe tener una identificación única en toda la web que permita acceder a sus datos.
* Correo electrónico: un correo propio de *SavingThrow* para comunicaciones entre jugadores.
* Partidas en las que se encuentra y si es jugador o master en las mismas: un usuario puede estar asignado a varias partidas.
* Personajes que interpreta: la lista de los personajes que el usuario ha registrado como suyos.

Cada usuario constaría de estos métodos.

|  |  |  |
| --- | --- | --- |
| URL | Método HTTP | Descripción/Acción |
| /v1/users | GET | Lista todos los usuarios de la plataforma (solo accesible a administradores) |
| /v1/users/<user\_id> | GET | Lista todos los atributos de un usuario en concreto. |
| /v1/users/<user\_id>/games | GET | Lista todas las partidas en las que se encuentra el usuario actualmente. |
| /v1/users/<user\_id>/characters | GET | Lista todos los personajes de un usuario en concreto. |
| /v1/users/<user\_id>/characters | POST | Crea un personaje nuevo para un usuario en concreto. |
| /v1/users | POST | Crea un nuevo usuario. |
| /v1/users/<user\_id> | DELETE | Borra un usuario (solo accesible a administradores) |

## Personajes

Los personajes son, en esencia, fichas. Aunque puede haber varios personajes con el mismo nombre, cada uno tendrá su identificador. Los objetos de un personaje serían los siguientes:

* Identificación única: el personaje necesita tener una identificación única en toda la web que permita acceder a sus datos.
* Partida en la que se encuentra: un personaje solo puede aparecer en una partida.
* Usuario que lo interpreta: un personaje solo puede ser interpretado por un jugador, que es el usuario que lo ha creado.

Cada personaje constaría de estos métodos.

|  |  |  |
| --- | --- | --- |
| URL | Método HTTP | Descripción/Acción |
| /v1/characters/<character\_id> | GET | Lista todos los atributos de un personaje en concreto. |
| /v1/characters/<character\_id>/game | GET | Devuelve la partida en la que se encuentra el personaje. |
| /v1/characters/<character\_id> | DELETE | Borra un personaje en concreto de un usuario en concreto. |

## Partidas

Una partida une varios personajes para embarcarlos en una maravillosa aventura. Cada partida tiene un identificador único. Los objetos de una partida serían los siguientes:

* Identificación única: la partida necesita tener una identificación única en toda la web que permita acceder a sus datos.
* Jugadores envueltos en la partida: una partida debe tener más de un jugador.
* Personajes envueltos en la partida: una partida debe tener más de un personaje.

Cada partida constaría de estos métodos:

|  |  |  |
| --- | --- | --- |
| URL | Método HTTP | Descripción/Acción |
| /v1/games | GET | Lista todas las partidas activas. |
| /v1/games/<game\_id> | GET | Lista todos los atributos de una partida en concreto. |
| /v1/games | POST | Crea una nueva partida. |
| /v1/games/<game\_id> | DELETE | Borra una partida en curso (solo accesible a administradores) |