



# BloodBowl Management API

Status In progress ▾

Owners Ibai Ramirez

## Overview

Esta API dará la funcionalidad para gestionar tanto entrenadores de BloodBowl como equipos relacionados con esos entrenadores.

**Usuarios:** En esta API el término usuario y entrenador son sinónimos. Excepto el usuario Admin que se genera por seed de migración podemos intercambiar las palabras usuario y entrenador. Entrenador la usamos simplemente por semántica del propio juego. A nivel interno se usará siempre "user" o "user\_id" dependiendo de la relacion.

## Roles

Los usuarios que accedan a la API tendrán 2 roles perfectamente diferenciados:

1. **Administrador:** Rol encargado de crear tanto entrenadores como equipos.
2. **Entrenador:** Rol encargado única y exclusivamente de modificar cualquiera de sus equipos (siendo estos creados por el admin). Evitando así posibles trampas con la cantidad de oro correspondiente a cada equipo.

## Endpoints

### REGISTRO:

**Endpoint:** `/ {base_url} /register`

**Metodo:** POST

**Descripción:** Registra un usuario en la base de datos. Esto es igual a decir "registramos un entrenador en la base de datos".

### Body (JSON):

Json

```
{
  "name": "Juan Pérez",
  "email": "juan@example.com",
  "password": "secret123",
  "password_confirmation": "secret123"
}
```

- **"name"**: obligatorio, tipo string, máximo 255 caracteres.
- **"email"**: obligatorio, formato email válido, único en la tabla users.
- **"password"**: obligatorio, mínimo 6 caracteres, confirmado (debes incluir password\_confirmation con el mismo valor).

## LOGIN:

**Endpoint:** `/ {base_url} /login`

**Metodo:** POST

**Descripción:** Hace login al usuario correspondiente y devuelve el token correspondiente.

### Body (JSON):

```
{
  "email": "usuario@example.com",
}
```

```
"password": "miclave123"  
}
```

- "email": requerido, debe ser un email válido y existente en la base de datos
- "password": requerido, string, mínimo 6 caracteres.

**Endpoint:** /{base\_url}/logout

**Metodo:** POST

**Descripción:** Hace logout del usuario correspondiente y revoca el token correspondiente.

## HABILIDADES:

**Endpoint:** /{base\_url}/skills

**Metodo:** GET

**Descripción:** Devuelve todas las habilidades de la base de datos con sus descripciones.

**Roles permitidos:** Admin y coach

**Resultado esperado:**

```
{  
  "data": [  
    {  
      "name": "{skill_name}",  
      "description": "{skill_description}"  
    }  
  ]  
}
```

```
    },  
    ....  
  ]  
}
```

### Ejemplo:

```
{ "data":  
  [  
    {  
      "name": "Block",  
      "description": "Allows a player to safely block an  
      opponent."  
    },  
    {  
      "name": "Sprint",  
      "description": "Allows extra movement when running."  
    }  
  ]  
}
```

## ROSTERS:

**Endpoint:** `/ {base_url} /rosters/` **Metodo:** GET

**Descripción:** Devuelve todos los ids de Rooster .

**Roles permitidos:** Admin y coach

**Resultado esperado:**

```
{
  "data": [
    {
      "id": Identificador del Roster. Int
      "name": nombre del roster. String
    },
  ]
}
```

**Ejemplo:**

```
{
  "data": [
    { "id": 1, "name": "Humans" },
    { "id": 2, "name": "Orcs" },
    { "id": 3, "name": "Dwarfs" },
    { "id": 4, "name": "Skaven" },
    { "id": 5, "name": "Elven Union" },
    { "id": 6, "name": "Lizardmen" },
    { "id": 7, "name": "Chaos Chosen" },
    { "id": 8, "name": "Khemri" },
    { "id": 9, "name": "Necromantic" },
  ]
}
```

```
{ "id": 10, "name": "Norse" }  
]  
}
```

**Endpoint:** `/ {base_url} /rosters/{roster_id}`

**Metodo:** GET

**Parametros:** `{roster_id}`, numérico.

**Descripción:** Devuelve los tipos de jugadores del roster especificado en `{roster_id}`

**Roles permitidos:** Admin y coach

**Resultado esperado:**

```
{  
  "data": [  
    {  
      name: nombre del jugador. String  
      ma: Movement Allowance (movimiento). String  
      st: Strength (fuerza). int  
      ag: Agility (agilidad). int  
      pa: Passing Ability (pase). int  
      av: Armor Value (armadura). int  
      skills: lista de habilidades. String[ ]
```

**cost:** precio del jugador.`int`

**max:** cantidad máxima de ese tipo de jugador en el [equipo.int](#)

```
    },  
]  
}
```

### Ejemplo:

```
{ "data":  
  [  
    {  
      "name": "Lineman",  
      "ma": "5",  
      "st": "3",  
      "ag": 3,  
      "pa": 1,  
      "av": 8,  
      "skills": ["Block","Fly" ],  
      "cost": "60000k",  
      "max": "16"  
    },  
    ...  
    ...  
  ]
```



```
}
```

## Entrenadores:

**Endpoint:** `/ {base_url} /users`

**Metodo:** GET

**Descripción:** Devuelve todos los usuarios creados en la base de datos

**Roles permitidos:** Admin

**Resultado esperado:**

```
{
  "data": [
    {
      "name": "nombre del entrenador", String
      "email": "correo electrónico" String
    }
  ]
}
```

**Ejemplo:**

```
{ "data":
  [
    {
      "name": "ibai",
```

```
        "email": "ibai24@gmail.com"
    },
    {
        "name": "pepe",
        "email": "pepe23@gmail.com"
    },
]
}
```

**Endpoint:** `/ {base_url} /users/ {user_id}`

**Metodo:** GET

**Descripción:** Devuelve los datos del entrenador especificado en `{user_id}` (`numerico`)

**Roles permitidos:** Admin y Coach

**Resultado esperado:**

```
{
    "data": [
        {
            "name": "nombre del entrenador", String
            "email": "correo electronico", String
            "created_at": "fecha de creación", String
            "updated_at": "fecha de modificación", String

            "teams": [
                {
                    "team_id": "Id del equipo", String
                    "team_name": "Nombre del equipo" String
                }
            ]
        }
    ]
}
```

```
        },
        ....
    ]
}
}
```

### Ejemplo:

```
{ "data":
  [
    {
      "name": "Carlos Martínez",
      "email": "carlos.martinez@example.com",
      "created_at": "2024-01-15T10:30:00Z",
      "updated_at": "2024-06-20T14:45:00Z",
      "teams": [
        {
          "team_id": 1,
          "team_name": "Human Reapers"
        },
        {
          "team_id": 2,
          "team_name": "Orc Crushers"
        }
      ]
    }
  ]
}
```

```
        ]
    }
]
}
```

## EQUIPOS:

**Endpoint:** `/ {base_url} /teams`

**Metodo:** GET

**Descripción:** Devuelve todos los equipos creados en la base de datos

**Roles permitidos:** Admin y Coach

**Resultado esperado:**

```
{ "data":
  [
    {
      "name": "nombre del equipo", String
      "user_id": "id del entrenador", String
      "user_name": "nombre del entrenador", String
      "roster_id": "id del roster del equipo", String
      "roster_name": "nombre del roster", String
      "gold_remaining": "oro restante del equipo", int
      "team_value": "valor del equipo en puntos de bloodbowl", int
    }
  ]
}
```

**Ejemplo:**

```
{ "data":  
  [  
    {  
      "name": "Barbas de Hierro",  
      "user_id": "1",  
      "roster_id": "2",  
      "user_name": "IbaiMania",  
      "roster_name": "Dwarfs",  
      "gold_remaining": "80000",  
      "team_value": "1000"  
    }  
  ]  
}
```

**Endpoint:** `/ {base_url} / teams / {team_id}`

**Metodo:** GET

**Descripción:** Devuelve los datos y los jugadores del equipos especificado `{team_id}`

creado en la base de datos

**Roles permitidos:** Admin y Coach

**Resultado esperado:**

```
{ "data":  
  [  
    {
```

```

{
  "team": {
    "id": "ID único del equipo", String
    "name": "Nombre del equipo", String
    "user_id": "ID del entrenador que controla el equipo", String
    "user_name": "nombre del entrenador", String
    "roster_id": "ID del tipo de roster que usa el equipo ", String
    "roster_name": "nombre del roster", String
    "gold_remaining": "Oro restante del equipo disponible para gastar", Integer
    "team_value": "Valor total del equipo en puntos de Blood Bowl" Integer
  },
  "players": [
    {
      "id": "ID único del jugador dentro del sistema", String
      "name": "Nombre personalizado del jugador", String
      "player_number": "Número de camiseta del jugador en el equipo", Integer
      "spp": "Puntos de estrella ganados por el jugador", Integer
      "injuries": "Texto que describe las lesiones del jugador", String
      "player_type": {
        "id": "ID único del tipo de jugador", String
        "name": "Nombre del tipo de jugador (por ejemplo, Blitzzer, Runner, etc.)",
String
        "max_per_team": "Máximo de jugadores de este tipo permitidos por equipo",
Integer
        "movement": "Movimiento (MV) del jugador ", Integer
        "strength": "Fuerza (ST) del jugador ", Integer
        "agility": "Agilidad (AG) del jugador, con formato tipo '3' ", Integer
        "passing": "Habilidad de pase (PA) del jugador, formato tipo '4'", Integer
        "armor": "Valor de armadura (AV), formato tipo '9'", Integer
        "cost": "Coste del jugador en monedas de oro al momento de contratarlo"
Integer
      }
    }
  ]
}
]

```

```
}
```

### Ejemplo:

data:

```
[  {
    "team": {
        "id": "1",
        "name": "Barbas de Hierro",
        "coach_id": "ibai2",
        "roster_id": "dwarfs",
        "gold_remaining": 80000,
        "team_value": 1000
    },
    "players": [
        {
            "id": "14",
            "name": "Thrain Ironfoot",
            "player_number": 4,
            "spp": 6,
            "injuries": "Niggling Injury",
            "player_type": {
```

```
{
  "id": "3",
  "name": "Blitzer",
  "movement": 6,
  "strength": 3,
  "agility": 3,
  "passing": 4,
  "armor": 10,
  "cost": 90000
}
}
]
}
]
}
```

**Endpoint:** `/ {base_url} /teams/ {team_id} /players`

**Metodo:** POST

**Descripción:** Crea un jugador en el equipo especificado por `{team_id}` con los datos del jugador en el cuerpo de la petición

**Roles permitidos:** Admin y Coach

**Body (JSON):**



```
{  
  
  "player_type_id": "ID del tipo de jugador al que pertenece", String  
  
  "player_number": "Número asignado al jugador dentro del equipo", Integer  
  
  "name": "Nombre personalizado del jugador", String  
  
  "injuries": "Descripción de las lesiones actuales del jugador (opcional)", String  
  
  "spp": "Puntos de estrella del jugador (integer, puede ser 0)" Integer  
  
}
```

#### Ejemplo:

```
{  
  "player_type_id": "3",  
  "player_number": 7,  
  "name": "Balin Stonebeard",  
  "injuries": "",  
  "spp": 3  
}
```

**Endpoint:** `{base_url}/players/{player_id}`

**Método:** PUT

Descripción:

Actualiza el jugador especificado por `{player_id}` dentro del equipo `{team_id}`. Solo se pueden modificar el nombre y los puntos de estrella (spp).

**Roles permitidos:**

Admin y Coach

**Body (JSON):**

json

```
{  
  "name": "Nombre personalizado del jugador", String  
  "spp": "Puntos de estrella del jugador (integer)" Integer  
}
```

**Ejemplo:**

json

```
{  
  "name": "Balin Stonebeard Updated",  
  "spp": 5  
}
```

**Endpoint:** `/ {base_url} /players / {player_id}`

**Método:** DELETE

**Descripción:**

Elimina el jugador identificado por `{player_id}` del equipo especificado por `{team_id}`.

**Roles permitidos:**

Admin y Coach

**Endpoint:** `/ {base_url} /matches / simulate`

**Método:** POST

**Descripción:**

Simula un partido entre dos equipos registrados en la base de datos. Calcula el resultado basándose en las estadísticas, skills, lesiones y otros factores aleatorios. Devuelve el resultado del partido, el número de touchdowns por equipo y los jugadores que los marcaron.

**Roles permitidos:**

Admin y Coach (solo si ambos equipos son suyos)

**Body (JSON):**

json

```
{  
  "team_a_id": "Id del primer equipo", String  
  "team_b_id": "Id del segundo equipo" String  
}
```

**Ejemplo:**

json

```
{  
  "team_a_id": 12,  
  "team_b_id": 15  
}
```

**Resultado esperado:**

```
{
  "team_a": {
    "id": 12,
    "name": "Greenfield Grasshuggers",
    "score": 2,
    "touchdowns": [
      { "player_id": 45, "player_name": "Throg Ironjaw" },
      { "player_id": 47, "player_name": "Snaga Gobbletooth" }
    ]
  },
  "team_b": {
    "id": 15,
    "name": "Skavenblight Scurries",
    "score": 1,
    "touchdowns": [
      { "player_id": 52, "player_name": "Zik-Zak Quicktail" }
    ]
  },
  "winner_team_id": 12,
  "message": "Partido simulado con éxito"
}
```

