


# Manual tecnico - Integración REST API

Este documento tendrá como proposito explicar y dar ejemplos para poder consumir las apis de manera correcta, esta se conectará a mongo para extraer los datos y manipularlos desde el backend.

## Estructura general

Cada api tendrá por defecto las siguientes rutas de consumo con sus metodos respectivos, en caso que un api determinada no tenga la estructura general o esta sea modificada se marcará con un 

Metodo	Url	Descripción
GET	/ <code>nombre_api</code>	Obtiene todos los registros del api correspondiente
GET	/ <code>nombre_api</code> / <code>:id</code>	Obtiene el registro según el id
POST	/ <code>nombre_api</code>	Registra un nuevo elemento en la base de datos
PUT	/ <code>nombre_api</code> / <code>:id</code>	Actualiza un unico registro
DELETE	/ <code>nombre_api</code> / <code>:id</code>	Elimina el registro correspondiente
POST	/ <code>nombre_api</code> / <code>search</code>	Busca los registros correspondientes a la trama enviada

 NOTA: Para los metodos, `DELETE` y `GET` no debe haber trama (cuerpo del paquete)

 NOTA 2: El metodo `PUT` tendrá parametros(id\_entidad) en la url como trama

## APIS

A continuación se listará cada una de las apis con trama(request) y sus respectivos campos obligatorios para cada metodo, el metodo GET no se tomará ya que no posee una trama(request)

✓ = Atributo obligatorio en el metodo correspondiente.

🚩 = Atributo opcional.

✗ = Se omite este campo.

## Jugador

Nombre API: `jugador`

Ruta consumo: `http://localhost:8008/jugador`

Ejemplo trama:

```
{
  "nombre": "Juan Sebastiàn",
  "apellido": "González Rivera",
  "correo": "juanmarfil9696@gmail.com",
  "contrasena": "123",
  "lider": false,
  "estado": "A",
  "descripcion": "Soy un jugador capaz..."
}
```

Obligatoriedad:

Atributo	POST	PUT
<code>nombre</code>	✓	🚩
<code>apellido</code>	✓	🚩
<code>correo</code>	✓	🚩
<code>contrasena</code>	✓	🚩
<code>lider</code>	🚩	🚩
<code>estado</code>	🚩	🚩
<code>descripcion</code>	🚩	🚩

## Equipo

---

Nombre API: `equipo`

Ruta consumo: `http://localhost:8008/equipo`

Ejemplo trama:

```
{
  "nombre": "Las aguilas",
  "descripcion": "Descripción...",
  "foto": "Base64/path",
  "categoria": "A|B|C",
  "estado": "A",
  "integrantes": [
    "idJugador1",
    "idJugador2",
    "idJugador3"
  ]
}
```

Obligatoriedad:

Atributo	POST	PUT
<code>nombre</code>	✓	🚩
<code>desrcpcion</code>	✓	🚩
<code>foto</code>	✓	🚩
<code>categoria</code>	✓	🚩
<code>estado</code>	🚩	🚩
<code>integrantes</code>	🚩	🚩

## Partido

Nombre API: `partido`

Ruta consumo: `http://localhost:8008/partido`

Ejemplo trama:

```
{
  "marcador": "1-1",
  "lugar": "Galerías cancha Sintetica",
  "estado": "A",
  "fecha": "2018/05/24 14:00:00",
  "equipos": [
    "idEquipo1",
    "idEquipo2"
  ],
  "resenas": [
    "idResena1",
    "idResena2",
    "idResena3"
  ]
}
```

Atributo	POST	PUT
marcador	✓	🚩
lugar	✓	🚩
estado	✓	🚩
fecha	✓	🚩
equipos	🚩	🚩
resenas	🚩	🚩

## Reseña

Nombre API: `resena`

Ruta consumo: `http://localhost:8008/resena`

Ejemplo trama:

```
{
  "puntaje": "5",
  "descripcion": "El partido estuvo reñido pero...",
  "jugador": "idJugador1",
  "estado": "A"
}
```

Atributo	POST	PUT
puntaje	✓	🚩
descripcion	✓	🚩
jugador	✓	🚩
estado	✓	🚩

## Common API



Metodo	Url	Descripción
GET	/ nombre_api	Obtiene todos los registros del api correspondiente
GET	/ nombre_api / :id	Obtiene el registro según el id
POST	/ nombre_api	Registra un nuevo elemento en la base de datos
PUT	/ nombre_api / :id	Actualiza un unico registro
DELETE	/ nombre_api / :id	Elimina el registro correspondiente
POST	/ nombre_api / search	Busca los registros correspondientes a la trama enviada