

# GUION DE LA ACTIVIDAD PRACTICA AP9-1 API Rest con Symfony

---

## Objetivos

- Entender como funcionan los APIs.
  - Aprender a montar un API Rest usando Symfony.
  - Practicar con el uso de un API mediante Bruno.
- 

## Recursos generales

- Florida Oberta
- 

## Enunciado

Eres el desarrollador de un API para la gestion de jugadores de la liga fantasy. Solo tenemos que hacer la parte de Backend y ofrecer la funcionalidad mediante un API Rest, porque el interfaz de usuario lo haran otros equipos, nosotros solo vamos a ofrecer la parte de gestion de jugadores en backend.

---

## Ejercicio 1. Crea la entidad

### Entidad PlayerCard

Campo	Tipo	Descripción
<code>id</code>	integer	Identificador único (autogenerado)
<code>name</code>	string	Nombre del jugador
<code>surname</code>	string	Apellidos del jugador
<code>age</code>	integer	Edad
<code>currentTeam</code>	string	Equipo actual
<code>goalsScored</code>	integer	Goles marcados
<code>cardsReceived</code>	integer	Tarjetas recibidas
<code>birthDate</code>	date	Fecha de nacimiento

## Ejercicio 2. Crea el API Rest

Crea el controlador necesario para ofrecer la funcionalidad CRUD para la entidad PlayerCard mediante API Rest.

### Endpoints a implementar:

Metodo	Endpoint	Descripción
GET	<code>/api/players</code>	Obtener todos los jugadores
GET	<code>/api/players/{id}</code>	Obtener un jugador por ID
POST	<code>/api/players</code>	Crear un nuevo jugador
PUT	<code>/api/players/{id}</code>	Actualizar un jugador completo
PATCH	<code>/api/players/{id}</code>	Actualizar parcialmente
DELETE	<code>/api/players/{id}</code>	Eliminar un jugador

## Ejercicio 3. Configura Bruno

Crea una colección en Bruno para poder realizar todas las acciones CRUD en el API.

**Se pide:**

- Crear una colección llamada **Fantasy League API**.
- Crear una carpeta `Players` con todas las peticiones CRUD.
- Probar que todas las operaciones funcionan correctamente.