

# Prueba Técnica: Sistema de Seguimiento de Equipos de Fútbol

## Descripción General

Esta prueba técnica está diseñada para evaluar las habilidades técnicas de candidatos en desarrollo de software full-stack. El objetivo es desarrollar un sistema que permita a los usuarios registrarse, seleccionar equipos de fútbol de su interés y visualizar los próximos partidos de estos equipos en formato de lista, con información detallada de cada encuentro. **Como la versión gratuita no permite acceder a los datos actuales, puedes “simular” que estás en cualquier otro año.**

## Duración y Proceso de Evaluación

Los candidatos dispondrán de siete días naturales para completar la prueba técnica desde el momento en que reciban las instrucciones. Aunque la implementación básica puede realizarse en pocas horas, este tiempo adicional permite añadir mejoras y pulir detalles que demuestren las habilidades del candidato.

Una vez entregada la prueba, será revisada por nuestro equipo de desarrollo. Posteriormente, se programará una sesión donde el candidato presentará su solución al equipo técnico y responderá preguntas sobre su implementación, decisiones de diseño y posibles mejoras. Esta sesión es una parte importante del proceso de evaluación, ya que permite valorar tanto las habilidades técnicas como la capacidad de comunicación del candidato.

## Tecnologías Sugeridas

El candidato tiene libertad para elegir las tecnologías específicas, pero se sugiere:

### Backend

- Ruby/Ruby on Rails (nuestro stack por defecto)
- Laravel/PHP
- Node.js/Express
- Python/Flask/FastAPI
- Base de datos: PostgreSQL, MongoDB, MySQL, etc.
- Sistema de autenticación: JWT, OAuth, etc.

### Frontend

- React, Vue.js, Angular, etc. (React es nuestro stack por defecto)
- Componentes UI: Material-UI, Tailwind CSS, Bootstrap, etc.

# Requisitos Técnicos

## Backend

- Desarrollar un sistema de registro y autenticación de usuarios.
- Implementar un servicio que consulte la API pública de API-Football (<https://www.api-football.com/documentation-v3>) para obtener datos de equipos y partidos.
- Almacenar en base de datos la información de usuarios registrados y sus equipos seleccionados.
- Proporcionar una forma para el frontend de consultar información de los equipos del usuario y los partidos en los que van a participar

## Frontend

- Crear una interfaz web que permita:
  - Registro y login de usuarios.
  - Búsqueda y selección de equipos favoritos.
  - Visualización de los próximos partidos en formato de lista de texto.

## Elementos Opcionales (Nice to Have -- NO OBLIGATORIOS)

- Consulta de información detallada al seleccionar un partido.
- Visualización de información básica del equipo rival.
- Implementación de tests unitarios y de integración.
- Visualización de partidos en formato de calendario interactivo
- Dockerización de la aplicación (backend, frontend y base de datos).

## Posibles Arquitecturas

El candidato es libre de elegir la arquitectura que considere más adecuada para la solución. Se valorará positivamente el uso de enfoques modernos y eficientes.

## Especificaciones Técnicas

### API de API-Football

Para obtener los datos de equipos y partidos, se debe utilizar la API de API-Football. Los principales endpoints a utilizar son:

- Endpoint para buscar equipos: `/teams`
- Endpoint para obtener fixture/calendario de un equipo: `/fixtures`
- Endpoint para obtener información de equipos: `/teams`
- Endpoint para obtener información de estadios: `/venues`

### Datos a Almacenar

Se sugiere almacenar al menos los siguientes datos:

## Usuarios:

- ID
- Nombre
- Email
- Contraseña (hash)
- Fecha de registro

## Equipos seleccionados por usuario:

- ID de usuario
- ID de equipo
- Fecha de selección

## Funcionalidades del Sistema

### Registro y Autenticación:

- Los usuarios deben poder registrarse con email y contraseña.
- El sistema debe implementar autenticación segura.
- Las contraseñas deben almacenarse de forma segura (hash+salt).

### Gestión de Equipos:

- Los usuarios deben poder buscar equipos por nombre.
- Los usuarios deben poder seleccionar y deseleccionar equipos favoritos.
- El sistema debe limitar el número de equipos que un usuario puede seleccionar (por ejemplo, máximo 5).

### Visualización de Partidos:

- El sistema debe mostrar los próximos partidos de los equipos seleccionados en formato de lista de texto.
- La lista debe estar ordenada cronológicamente y mostrar claramente:
  - Fecha y hora del partido
  - Equipos enfrentados
  - Competición o liga

## Criterios de Evaluación

Se evaluará la solución considerando los siguientes aspectos:

### 1. Funcionalidad básica:

- Correcta implementación del sistema de registro y autenticación
- Conexión efectiva con la API de API-Football
- Almacenamiento adecuado en la base de datos
- Visualización de los partidos en formato de lista
- Presentación de información detallada de partidos y equipos

## **2. Calidad del código:**

- Estructura y organización
- Manejo de errores y excepciones
- Patrones de diseño utilizados
- Legibilidad y mantenibilidad

## **3. Aspectos técnicos:**

- Diseño de la base de datos
- Elección de tecnologías y arquitectura
- Seguridad (especialmente en la autenticación)
- Tests (si los implementa)

## **4. Presentación y comunicación:**

- Claridad al explicar la implementación
- Capacidad para justificar decisiones técnicas
- Conocimiento profundo de la solución desarrollada
- Identificación de posibles mejoras y limitaciones

## **Entregables**

Los candidatos deben proporcionar:

1. Código fuente completo de la solución en un repositorio Git.
2. README con instrucciones de instalación y uso.
3. Breve documentación técnica explicando la arquitectura y decisiones de diseño.

## **Consideraciones Adicionales**

Los candidatos pueden implementar funcionalidades adicionales para destacar, como:

- Visualización de partidos en formato de calendario interactivo
- Implementación de tests unitarios y de integración
- Dockerización de la aplicación (backend, frontend y base de datos)
- Diseño responsive para dispositivos móviles