

# Prueba técnica de Front End de LaLiga

---

Este documento comprende los requerimientos para realizar la prueba técnica de Front-End de LaLiga.

Se requiere el uso de GIT para la realización de la prueba. Esperamos que nos facilite acceso a un repositorio privado, o este mismo comprimido y enviado como un archivo adjunto.

Debes enviar el ejercicio contestando al correo desde el que has recibido la prueba, manteniendo en copia a las personas del mail original.

## Historias de usuario a implementar

Vamos a realizar una pequeña y sencilla aplicación tipo SPA con React que se conecte a una API de pruebas. El objetivo es que el usuario que te proveemos pueda autenticar en ella y acceder a un listado de usuarios. Si no está autenticado, el usuario no deberá poder acceder a la vista de listado

Puedes encontrar toda la documentación que necesitas sobre la API de pruebas en [ReqRes](#)

**IMPORTANTE:** Si tu aplicación no cumple con alguno de estos requisitos, será automáticamente descalificada.

### Vista: "Login"

Esta vista deberá mostrarse en la ruta `/login` del navegador.

Debe constar, al menos de:

- Un campo de texto para el usuario.
- Un campo de texto para la contraseña.
- Un botón de submit con el texto `Iniciar Sesión`.

El formulario deberá integrarse con el endpoint `POST https://reqres.in/api/login`, de forma que al introducirse el usuario `eve.holt@reqres.in`, la contraseña `cityslicka` y pulsar el botón de "Iniciar Sesión", se recupere un token de sesión que devuelve la API.

Este token de sesión *deberá almacenarse en el navegador* de tal forma que persista incluso ante el cierre de pestaña o ventana del navegador.

### Vista: "Listado"

Esta vista deberá mostrarse en la ruta `/users` del navegador.

- Deberá mostrarse en pantalla un listado de usuarios obtenidos a partir del endpoint `GET https://reqres.in/api/users`. En este listado, por cada usuario deberá mostrarse al menos su `avatar`, su `first_name`, su `last_name` y su `email`.
- El listado deberá estar paginado, mostrando no más de 6 elementos por página
- Un botón con el texto "Cerrar Sesión" que devuelva a la vista de "Login" y haga perder sesión al usuario.

Consideraciones adicionales de la aplicación:

- Si el usuario tiene sesión, se le mostrará la vista "Listado". De lo contrario, no podrá acceder nunca a ella.
- Un usuario con sesión en el navegador no podrá acceder a "Login" salvo que la pierda (por ejemplo, pulsando el botón "Cerrar Sesión").
- La aplicación se visualiza correctamente en dispositivos móviles a partir de una resolución de **375px x 667px** (resolución de iPhone 6).

## Requisitos técnicos

**IMPORTANTE:** Si tu aplicación no cumple con alguno de estos requisitos, será automáticamente descalificada.

- Debe estar construida con **React** y **Create React App**.
- No debe haber componentes de clase, sólo componentes funcionales.
- Para almacenar el estado, debe realizarse con **redux** y **redux-saga**. Puedes sustituir **redux-saga** por **redux-thunks**.
- Debe tener integrada alguna herramienta para inspeccionar el estado, como la extensión **redux-devtools**.
- Los estilos que introduzcas deben estar introducidos mediante **styled-components**
- Para manejar el routing de la aplicación, debes hacerlo con **react-router-dom**
- Usar **eslint** y **prettier**

Algunas de estas librerías te las proporcionamos en este starter ya integradas para evitarte el trabajo de hacerlo. Puedes añadir todas las librerías que necesites, aunque no olvides documentar las decisiones tomadas en la memoria del **README.md**.

## Valoraremos...

- Emplear los patrones estándar y aplicar las guías de estilo oficiales de las librerías que te solicitamos usar.
- Código bien estructurado y legible según los estándares de la guía Clean Code.
- Que documentes tu código pensando en que será leído por otro desarrollador.
- Que no hay código duplicado y/o copipasteado.
- Que has usado javascript moderno, declarativo y aplicando los patrones funcionales propuestos por React y Redux.
- Aderirses a los principios SOLID, al menos aquellos que pueden aplicarse en React.
- Una consola limpia de errores y warnings.
- Flujo de uso y manejo de errores (con feedback para el usuario) correctos.

Pon mucha atención y detalle a todo esto, pues serán los criterios que emplearemos para evaluarte.

De forma opcional, valoraremos muy positivamente si además...

- Usas un buen tipado estático con **Typescript**.
- Tu app tiene una UI cuidada y con un look & feel agradable.
- Has usado **redux-saga** como middleware de efectos.
- Has dado cobertura de test a almenos una de las dos vistas.
- Si has hecho test, estos son con **react testing library**.

Y cualquier otra adición (animaciones, efectos diferenciales, etc) será bienvenida. No dudes en dar rienda suelta a tu imaginación si consideras que mejora tu aplicación.