

# Prueba desarrollador frontend

Cleverpy Machine Learning

Jordi Boix Ramos

---

## Contacto

j.boix@cleverpy.com

(+34) 622 823 911



Valencia

---

<b>1. Introducción</b>	<b>3</b>
<b>2. Descripción de la prueba</b>	<b>3</b>
2.1. Funcionalidad	3
2.2. <i>Puntos a valorar</i>	3
2.2.1. Diseño	3
2.2.2. Implementación	3
2.2.3. Conexión	4
<b>3. Ampliaciones</b>	<b>4</b>
<b>4. Entrega</b>	<b>5</b>

---

## 1. Introducción

El equipo técnico de Cleverpy Machine Learning ofrece esta prueba técnica con el objetivo de ampliar el conocimiento sobre las capacidades del entrevistado, así como su manera de trabajar y abarcar la resolución de problemas.

Cabe destacar que dicha prueba no tiene una respuesta única, ya que se espera que el entrevistado matice personalmente el planteamiento y la resolución al problema descrito en este documento.

## 2. Descripción de la prueba

La prueba consiste en diseñar, implementar y conectar una plataforma con “cards” para la visualización de posts o comentarios realizados por un usuario. Para ello, utilizaremos como ejemplo el contenido gratuito ofrecido por [jsonplaceholder.typicode.com](https://jsonplaceholder.typicode.com), y más en concreto, los resultados ofrecidos por la llamada GET de <https://jsonplaceholder.typicode.com/posts>.

### 2.1. Funcionalidad

La funcionalidad final de la solución debe permitir al usuario administrar y visualizar los posts todos de los usuarios disponibles en la herramienta.

Conceptualmente, hablamos de visualizar un post como que el usuario pueda ver el título del post (identificado por el atributo **title**), el usuario que ha creado el post (identificado por el atributo **userId**), y el contenido del texto del post (identificado por el atributo **body**). Dicha visualización debe ser resumida en una sola pantalla con una agregación de todos los post disponibles.

Dicha visualización debe permitir al usuario eliminar posts. Esta funcionalidad solamente se debe aplicar a nivel de código cliente sin necesidad de lanzar una petición para borrar el post en *backend*, aunque maquetarlo se tendrá en cuenta.

### 2.2. Puntos a valorar

#### 2.2.1. Diseño

El diseño de la interfaz de la herramienta descrita será valorado positivamente por llevar un análisis previo a la problemática y adelantarse a posibles escenarios que puedan suceder a nivel de funcionalidad.

#### 2.2.2. Implementación

Aquí se valorará todo lo relacionado a nivel de código. Esta parte es mucho más compleja de especificar y por tanto marcaremos ciertos objetivos *necesarios* y *opcionales*.

- Objetivos necesarios
  - Código totalmente en TypeScript.

- 
- *Tipado sin tipos **any**.*
  - *Código en React*
  - *Uso de componentes funcionales*
  - *Buena indentación*
  - *No usar librerías externas que realicen la función descrita en el documento. Sí se permiten las que sean un medio para conseguir la funcionalidad final. (Una librería que proporcione un dropdown para el card estaría permitido, una librería que cree el card al completo no).*
  - *Opcionales*
    - Testing unitario
    - Testing e2e
    - Estilos scss y variables css
    - BEM
    - Flexbox
    - Grid-CSS

### 2.2.3. Conexión

Para esta parte, se debe conectar directamente la plataforma de posts a los resultados de *jsonplaceholder*, de manera que se pidan los datos y se muestren dinámicamente en la plataforma.

Para este paso, se valorará la manera de implementarlo. Cualquier muestra de implementación de la funcionalidad mediante el uso del store de Redux será valorado positivamente (aunque no sea la mejor solución *a priori*).

## 3. Ampliaciones

Además de la funcionalidad descrita, se propone una serie de ampliaciones por si el entrevistado se ve interesado en la prueba y quiere extender la funcionalidad de la plataforma:

- Edición de posts
- Layout básico
  - Navbar
  - Sidebar
  - Content
  - Breadcrumbs
- Pantalla de login
- Paleta de colores y guía de estilos
- Estilos CSS y animaciones

---

#### 4. Entrega

El formato de entrega de dicho proyecto debe ser en .zip o mediante el enlace al repositorio dónde se ha implementado. Dicho repositorio debe estar preparado para lanzarse y visualizar la plataforma desarrollada al instalar las dependencias necesarias.

Att.  
Cleverpy Machine Learning