**Enunciado**

La idea de la práctica de front-end avanzado será crear un sitio similar a medium.com (https://medium.com/), una plataforma social de publicación de artículos de diferentes contenidos.

Se pretende poner en práctica las habilidades adquiridas durante los primeros módulos del Bootcamp construyendo un prototipo del front-end responsive que formará parte de la práctica final. Este front-end deberá adaptarse a terminales móviles, tablets y equipos de escritorio.

Este front-end constará de las siguientes plantillas:

* Listado de artículos - Haciendo click en cada uno de ellos se accedería al detalle del artículo.
* Detalle de un artículo

**Requisitos de todas las páginas**

Todas las páginas deberán compartir una cabecera y un pie de página común.

En la cabecera (header) se deberá mostrar:

* El nombre de la plataforma (a elección del alumno)
* Un listado de categorías temáticas (la idea es que, en un futuro, un usuario pueda hacer click en una categoría para ver sólo artículos de esa categoría) - Sin funcionalidad
* Un enlace a la plantilla de login/ registro. - Sin funcionalidad –
* Extra: Un input para poder realizar búsquedas en el listado de artículos

El menú deberá ser responsive, de manera que se adapte de la mejor manera posible a los diferentes tipos de dispositivos para los que se desea dar soporte.

En el pie de página (footer) deberá mostrar:

* El nombre de la plataforma (a elección del alumno)
* Otra información que el alumno quiera incluir
* Si el footer no está fijo a la parte inferior de la pantalla deberá incluir un enlace que permita subir a la parte superior de la página.
* Se deja a elección del alumno la posibilidad de incorporar más elementos a la cabecera o al pie de página.

**Requisitos del listado de artículos**

En la plantilla de listado de artículos se deberá mostrar un listado de 10 artículos donde al hacer click en un artículo se acceda a la plantilla de detalle de un artículo y cada artículo tenga:

* Un título
* Una imagen o vídeo. Entre los 10 artículos que debe haber en esta plantilla de listado, al menos uno de ellos deberá mostrar una imagen y otro un vídeo.
* Un texto a modo de introducción del artículo (2-5 líneas aprox.)
* Información del autor del artículo:
* Nombre del autor del artículo
* Foto de perfil: Al menos dos de los artículos deberán simular que el autor no hubiera subido ninguna imagen de perfil y mostrar una imagen de placeholder.
* El tiempo de publicación del artículo: esta fecha deberá mostrar el tiempo pasado desde la fecha y hora de publicación en un formato legible (ejemplo: Artículo publicado hace 5 minutos). Si el artículo ha sido publicado hace más de un día se mostrará la fecha de publicación
* Número de comentarios - El número de comentarios que tiene el artículo. Al hacer click en el número de comentarios, se deberá cargar la plantilla de detalle de artículo mostrando directamente la zona de comentarios del artículo.

**Requisitos de la plantilla de detalle de un artículo**

En la página de detalle de un artículo se deberán mostrar los siguientes elementos independientes para cada artículo:

* Título
* Una imagen grande o un vídeo
* Un texto de varios párrafos. Dentro de este texto podrán incluirse si se desea links, textos en negrita y cursiva, imágenes, párrafos indentados, ...
* Like: Un elemento de iteración (icono o botón) que permita un usuario indicar que le gusta el artículo.
* Comentarios: Se desea tener un sistema de comentarios en la plantilla de detalle de un artículo donde los usuarios puedan dejar sus comentarios acerca de un artículo. Este sistema de comentarios deberá mostrar: La lista de comentarios. Un formulario para que el usuario pueda dejar un comentario.

Se podrán añadir otros campos a elección del alumno

**FUNCIONALIDAD “LIKE”**

Esta funcionalidad utilizando Local Storage de manera que, cuando un usuario haga click en el botón de “me gusta” de alguno de los artículos, esta información deberá quedar almacenada en el navegador del usuario.

NOTA: En un futuro, esta funcionalidad almacenará la información de los artículos que gustan a un usuario en un backend para que, sea cual sea el navegador o dispositivo que utilice el usuario, esta información sea persistente estando almacenada en el servidor.

**SISTEMA DE COMENTARIOS**

Se desea tener un sistema de comentarios en la plantilla de detalle de un artículo donde los usuarios puedan dejar sus comentarios acerca de un artículo.

Al tratarse de un prototipo, este sistema de comentarios se implementará utilizando json-server como backend para persistir los datos (al igual que se ha hecho en las clases del módulo).

La lista de comentarios se cargará asíncronamente una vez el sistema de comentarios pasa a ser visible en la pantalla.

Se tendrán que gestionar los posibles estados de listado de comentarios:

* Vacío o sin comentarios
* Cargando comentarios
* Error en la carga de comentarios
* Éxito en la carga de comentarios - Mostrar los comentarios

Se deberá implementar la funcionalidad de dejar un comentario mediante un formulario donde se solicite:

* El nombre y apellidos del usuario que desea dejar el comentario
* El e-mail
* Un campo de texto donde el usuario podrá escribir su comentario (con un máximo de 500 caracteres)

Antes de poder enviar el comentario, se deberán validar correctamente los datos. No se deberá permitir enviar el formulario si:

Email tiene un formato incorrecto

Alguno de los campos está vacío - Todos son requeridos

El texto tiene más de 500 caracteres

Se deberá mostrar al usuario cuando ha habido un error en la comunicación con el servidor (como que el servidor no esté disponible) y a ser posible indicar en la consola el tipo de error.

Extra - Se podrá añadir una paginación de los comentarios para mostrar al usuario los comentarios de cinco en cinco

**Consideraciones generales**

Se desea que la optimización del front-end sea máxima, por lo que se espera que el sitio web cargue de una manera rápida, realizando el menor número de peticiones HTTP posible así como que todos los recursos que cargue el sitio web estén lo más optimizados posibles (esto incluye hojas de estilo, archivos JavaScript e imágenes).

Se deberán cargar las imágenes optimizadas para cada dispositivo, por lo que se deberán tener imágenes optimizadas a los diferentes tamaños de dispositivo.

Los principales valores del estilo: colores, fuente, tamaño de fuente deberán ser importados de un solo archivo y ser capaces de cambiarlos solo modificando este archivo. Ejemplo: Si queremos cambiar el color principal de toda nuestra aplicación deberemos modificar sólo una línea.

Será recomendable respetar los patrones de accesibilidad en los links, imagenes, …

El front-end deberá funcionar en los navegadores: Google Chrome, Mozilla Firefox e Internet Explorer 11 o superior.

Se permite el uso de cualquier librería así como la elección de tecnologías a utilizar por parte del alumno para la realización de la práctica.