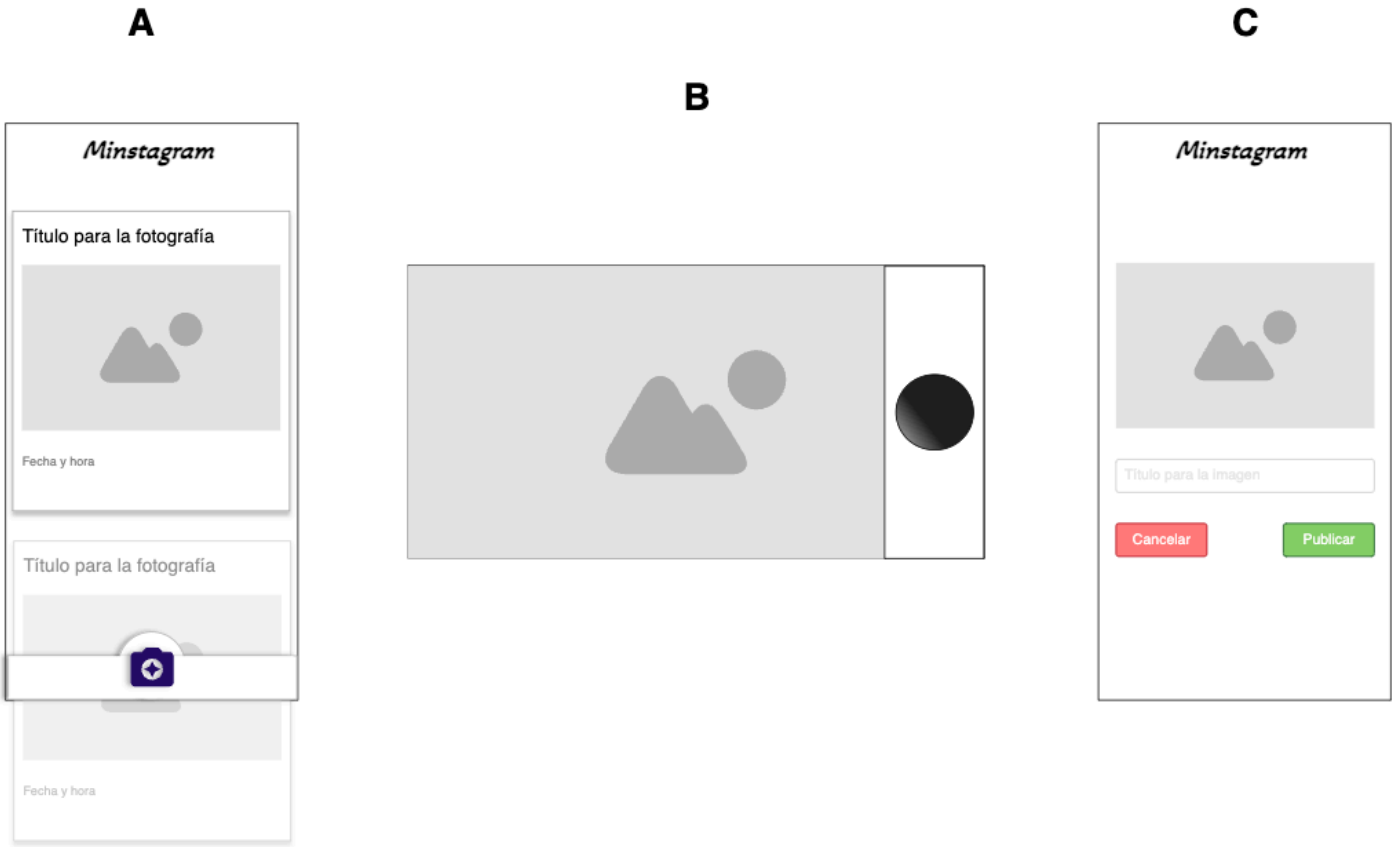
ISTEA

Consigna para el Parcial 01



Fecha tentativa de entrega: 29-04-2024 - 19 hs.

Recuerden que son varios cursando, y que tengo que revisar todos los proyectos y publicar una nota, antes de las 00 hs del 30-04-2024, por lo tanto, estaré destinando el tiempo de la clase para realizar esta labor.

Les pido que se pongan de acuerdo cuanto antes. Si ven la fecha complicada porque se les suman otros parciales y demás, me avisan y corremos la fecha una semana más.

ISTEA

Objetivo

Desarrollar una aplicación similar a instagram. En las imágenes está el nombre sugerido, pero por supuesto cada uno puede innovar y ponerle un nombre diferente.
La aplicación web integrará la funcionalidad de la cámara fotográfica, vista en las últimas clases. PAdemás publicará las fotografías en la plataforma **Mockapi.io** convirtiendo la imagen al formato Base64 y con algunos datos extras a publicar. También contendrá un archivo manifiesto para poder instalar la aplicación en un dispositivo o computadora.

Descripción de las funcionalidades esperadas

Al ingresar a la aplicación, se visualizará un Reel de las fotografías publicadas **(A)**. Este reel está pensado para visualizarse en un formato vertical.

- Cada fotografía publicada debe estar contenida en una Card HTML. Tendrá un título o texto que acompañe a la fotografía, la imagen en sí, y en el pié de la imagen figurará la fecha y hora.
- En algún lugar accesible del reel de fotografías se encontrará el botón para capturar una fotografía nueva.
- Este botón abrirá la cámara fotográfica del dispositivo móvil, para tomar una fotografía nueva **(B)**.

Capturada la fotografía, se visualizará la misma en un documento HTML donde se pueda hacer ajustes sobre la foto en cuestión.

- Validar su calidad en el preview de la imagen
- Agregar un título que acompañe la ilustración
- botón ACEPTAR, para publicarla, CANCELAR para volver al reel principal

Al pulsar el botón **ACEPTAR**, la imagen se guardará en MOCKAPI.IO con la siguiente estructura de datos:


```
{
    "imagen": "
     JRgABAQAAAQABAAD/4gHYSUNDX1BST0ZJTEUAAQEAAAH
     IAAAAAAQwAABtbnRyUkdCIFhZWiAH4AABAAEAAAAAAA
     BhY3NwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     AAQAA9tYAAQAAADTLQAAAAAAAAAAAAAAAAAAAAAA
     AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     AAAAAAlkZXN1psc/HWVlU/AHyf/2Q==",
    "id": "2",
    "fecha": "18/9/2023, 16:39:39",
    "titulo": "Puerto Madero"
}
```

ISTEA

Elementos básicos a construir

1. Un documento HTML llamado **index.html**
2. Un documento HTML llamado **camara.html**
3. Un archivo CSS para los estilos de los elementos HTML
4. Un archivo **Manifest.json**
5. Archivos JavaScript para la lógica. (*se recomienda uno para cada documento HTML*)
6. Cuenta gratuita en MOCKAPI.IO (*pueden crearla con el email de ISTEA*)

index.html

El manejo de estilos CSS se puede realizar con CSS puro o con algún framework CSS como **Bootstrap**, **Materialize CSS**, o similar. Mi sugerencia para no enredarse en la personalización de componentes HTML de estos frameworks CSS, es hacer un CSS puro. Pueden recurrir a **ChatGPT** o **Bard** para describirles el tipo de CSS que necesitan y que los ayude con la generación del mismo.

Deberá crearse una sola Card HTML en este documento. La misma será copiada y generada de forma dinámica desde JS, utilizando Template String + Literals. El tag **<img>** de esta Card debe contener una imagen predeterminada. No puede estar vacío inicialmente.

camara.html

Este documento deberá tener aplicados el mismo look and feel CSS que **index.html**. Contendrá un tag **<img>** para representar la captura de la imagen, un **input:text** para agregar un título o descripción a la imagen, y dos botones. *Pueden guiarse con el gráfico de la primera hoja*.
El tag **<img>** debe contener una imagen predeterminada. No puede estar vacío inicialmente.

Archivo CSS

Si lo crean uds. mismos, les recomiendo utilizar flex para la alineación de los elementos HTML. Se priorizará la visualización del contenido en pantalla móvil en lugar de pantalla desktop. Esto simplifica la alineación vertical de las Cards HTML para representar los posts en el reel de imágenes.
Si no tienen experiencia con CSS puro, sugiero que construyan el HTML base y luego le soliciten a una herramienta como ChatGPT o Google Gemini, que les genere las clases CSS necesarias para aplicar en un HTML (*y le comparten el código HTML*).
Alternativamente pueden recurrir también a un Framework CSS como Bootstrap, Tailwind, o el que más cómodo les parezca.

El CSS es relativo aquí. Lo que sí importa es lograr el fin de poder capturar una fotografía con el dispositivo móvil o, eventualmente, con la computadora, y que esta se grabe y se pueda recuperar desde un ENDPOINT como el de MOCKAPI.IO.

manifest.json

Este archivo deberá referenciarse en los dos documentos HTML, y debe contener las siguientes propiedades: nombre, nombre corto, scope, color del tema y color de fondo, display (*standalone*), orientation (*portrait*).
Además deberá contener un ícono representativo a la aplicación.
El mismo deberá estar referenciado en los siguientes tamaños: 512px, 192px, 144px, 64px. Agreguen este último, también, como **favicon** en los documentos HTML.

Interacción

Se ingresa a la aplicación por el documento **index.html**. El objetivo al ingresar es que se cargue en este el reel de posts publicados en MOCKAPI. Si no hay ningún dato para cargar, se debe dejar representada una Card HTML modelo; básicamente la generada en HTML de forma estática.
Si hay reels para cargar, se deben peticionar ordenando por fecha, de forma descendente.
Al pulsar el ícono para capturar una nueva fotografía éste nos lleva al documento **camara.html**.

En el documento **camara.html**, se activará la opción de capturar una fotografía, pulsando dos veces sobre la imagen modelo que esté cargada en la pantalla (*doble click o doble tap*).
Capturada y aceptada la fotografía desde la cámara del dispositivo móvil, se debe mostrar la misma en el tag **<img>** del documento HTML.
Se podrá escribir un párrafo o título para la imagen en el input type inferior para, finalmente, pulsar el botón CONFIRMAR y que la imagen y el texto cargado se publique en el endpoint de MOCKAPI.

Cómo publicar la imagen

Crear con JS un elemento **input:file**, configurar las propiedades que vimos oportunamente para que utilice la cámara del dispositivo móvil. Al capturar la imagen, la misma queda en memoria en formato **BLOB**.
Aquí es requerido utilizar el **canvas** para poder convertir la misma al formato **base64**, que es el formato alfanumérico en el cual podremos almacenar la misma en MOCKAPI.

Al utilizar el método .toDataURL() del objeto canvas, definan el tipo de imagen a generar como *“image/webp”*. Existen muchas restricciones del tamaño del payload a subir a MOCKAPI, y no tuve éxito en publicar nada que supere los 200kb. El formato webp es un formato moderno similar

al JPG pero con mejor compresión de imagen sin pérdida de calidad. Esto garantiza un menor peso de la imagen cuando se la genera en formato base64.

Recuerden que el **identificador** de cada nuevo post lo genera MOCKAPI automáticamente, por lo tanto no se debe enviar dentro de la estructura JSON la propiedad **id**.

Pruebas de funcionalidad

Les recomiendo hacer las pruebas de funcionalidad con la cámara, utilizando un dispositivo Android. También pueden realizarlas desde la computadora seleccionando imágenes para publicar, que tengan en la misma.