# react. actividad 1

**Instrucciones:** Crear una aplicación React con TypeScript de acuerdo con una especificación exigida.

## Parte 1

* Crear una aplicación React con Vite con el nombre Actividad1.
* Eliminar todo el contenido de la carpeta src.
* Crear el archivo main.tsx en la carpeta src.
* Crear un componente con el nombre ImagenApp.tsx que cargue una imagen cualquiera.
* Renderizar el componente desde el archivo main.tsx.

## Parte 2

* Modificar el componente ImagenApp para que devuelva un h1 encima de la imagen utilizando un fragment.

## Parte 3

* Crear una función con el nombre autor que devuelva tu nombre. Ese nombre aparecerá debajo de la imagen ejecutando la función desde el código HTML.

## Parte 4

* Crear un archivo de estilos a nivel general que determine el ancho y el alto de la imagen a 200 píxeles.

## Parte 5

* Enviar desde main.tsx al componente ImagenApp.tsx una propiedad src con el valor de la imagen a visualizar y otra propiedad alt con el texto alternativo para la imagen. Crear la interface necesaria para dar un tipo a las propiedades.

## Parte 6

* Incluir en el componente ImagenApp.tsx un input para que, al cambiar de valor, ese valor pase a ser el src de la imagen. Lo haremos mediante el hook useState.

**Objetivos:** Crear una aplicación React e introducirnos al uso de componentes y Hooks.

# react. actividad 2

**Instrucciones:** Crear una aplicación React con TypeScript de acuerdo con una especificación exigida.

## Parte 1

* Crear una aplicación React con Vite con el nombre Actividad2.
* Eliminar todo el contenido de la carpeta src menos el archivo main.tsx y la carpeta assets.
* Instalar Bootstrap y añadir la importación de estilos al archivo main.tsx.
* Crear un componente llamado Edad.tsx que retorne el título Actividad 3.
* Renderizar el componente Edad desde el archivo main.tsx.

## Parte 2

* Introducir un input para teclear una edad. Mediante condicionales ternarios, se sacará un mensaje “es mayor de edad” si la edad es mayor o igual a 18 u otro mensaje “es menor de edad” si la edad es menor a 18.

## Parte 3

* Crear una interface con el nombre INombre con las propiedades nombre (string) y edad (number). Crear un array con el nombre “amigos” de tipo INombre con datos de tres amigos.
* Recorrer el array y crear una lista con los amigos. Por cada item de lista sacaremos el nombre, la edad y un mensaje “es mayor de edad” o “es menor de edad” dependiendo de la edad.

## Parte 4

* Crear una carpeta interfaces. Dentro, crear la interface *coctel.interface.ts*.
* Implementar en la interface el código necesario para recibir los datos de la API *thecocktaildb* desde la url <https://www.thecocktaildb.com/api/json/v1/1/filter.php?i=Gin>.
* Crear un componente *CoctelesGin.tsx*
* Utilizar fetch para acceder a los datos de la url anterior
* Desarrollar un componente llamado *CardCoctel.tsx* para mostrar el nombre del cóctel *(strDrink)* y su foto *(strDrinkThumb)*. Utilizar el *idDrink* como clave para identificar cada card.
* Utilizar el hook *useEffect* para sacar los cócteles al cargarse el componente.
* Controlar el error de la petición para sacar un mensaje si la carga de datos no se produce.

**Objetivos:** Desarrollar una aplicación React con condicionales, loops y acceso a web services.

# react. actividad 3

**Instrucciones:** Crear una aplicación React con TypeScript de acuerdo con una especificación exigida.

* Hacer una copia del componente *CoctelesGin.tsx* y llamarle *CoctelesForm.tsx.*
* Desarrollar un formulario para introducir el nombre del ingrediente a consultar.
* Cambiar la url de la petición para hacer la búsqueda en función del ingrediente introducido.

**Objetivos:** Desarrollar una aplicación React con un formulario básico.