Idea de portafolio en React.js

Mateo Castaño Castaño

Desarrollo web con nuevas tecnologías

DAVID SANTIAGO PUERTA BETANCUR

Institución Universitaria Pascual Bravo

Facultad de Ingeniería

Medellín

2022

[**Introducción 3**](#_heading=h.30j0zll)

[**1. Identificación y descripción del problema 3**](#_heading=h.1fob9te)

[1.1. Descripción 3](#_heading=h.3znysh7)

[1.2. Formulación 4](#_heading=h.2et92p0)

[**2. Justificación 5**](#_heading=h.tyjcwt)

[**3. Objetivos 6**](#_heading=h.3dy6vkm)

[3.1. Objetivo general 6](#_heading=h.1t3h5sf)

[**4. Referentes teóricos 7**](#_heading=h.4d34og8)

[**5. Descripción Técnica del Proyecto 9**](#_heading=h.2s8eyo1)

**6. Resultado de la propuesta …….…………………………………………………………...……11 7. Metodología …………..………………………………………………………………….…………11**

7.1. Tipo de proyecto………………………………………..……………………………………..**11**

7.2. método ………………………………………………………………………………..….……..12

7.3. Población y muestras …………………………………………………………...…………….14

7.4. Instrumentos de recolección de información ……………………………………………14

7.4.1. Fuentes primarias …………………………………………………………………14

7.4.2. fuentes secundarias ………………………………………………………………14

**Bibliografía 16**

### 

# instrucciones

Vamos a crear un documento y sigamos los pasos a continuación:

**Paso 1.** Estructurar por escrito y de forma detallada una idea de un portafolio (Lo más original posible) para que lo creemos con React.js, este va a contener al menos 3 pantallas donde estructuramos contenidos, funciones y el consumo de una API.

**Paso 2.** Buscar diseños para el portafolio, pueden buscar en Canva, Pinterest, Google, etc. y guardar los diseños para enseñarlos en la próxima clase.

**Paso 3.** Crear el Mockup (Dibujo) de las pantallas que tendrá el portafolio como parte del proceso de diseño y basado en el punto anterior, recuerda incluir paleta de colores y al menos 5 a 7 componentes (Botones, banners, tarjetas, títulos, etc.), Pueden usar Draw.io, Figma, canva o cualquier otra herramienta con la que se sientan cómodos.

**Paso 4.** Pensar y dejar por escrito al menos 2 funciones simples o 1 compleja que debe contener el portafolio, ejemplo: Un convertidor de moneda, una calculadora de calorías, un lienzo para dibujar con el mouse, generador de datos aleatorios, etc, busca ideas que puedas implementar.

**Paso 5.** Planear y dejar por escrito la implementación de al menos 2 o 3 clases (POO) según lo que requiera el portafolio y explicar el propósito de cada una.

**Paso 6.** Tengamos listo un repositorio en github vacío con el nombre “portfolio \_{Nombre estudiante} para comenzar a subir nuestro proyecto hecho en clase y compartirlo con los demás compañeros y como evidencia para calificar.

# Paso 1: Estructura de la Idea para el Portafolio

## Descripción

Este portafolio personal será un espacio donde los visitantes podrán conocerme y explorar mis habilidades y proyectos. La estructura será sencilla pero elegante, con una navegación fluida entre las distintas secciones. El portafolio estará dividido en tres pantallas principales: la página de inicio, la página de proyectos y la página de contacto. Además, integrará el consumo de una API externa para mostrar estadísticas relevantes, como las tendencias en tecnología o algún dato personalizado.

## Pantallas principales

1. **Página de Inicio (Home)**

**Contenido:** Una breve presentación sobre quién soy, mi rol ( desarrollador web y/o artista digital), y una imagen destacada que refleje mi personalidad o trabajo. Incluirá un menú de navegación que permita acceder a las otras secciones del portafolio.

**Funcionalidades:** Un banner rotativo con frases o proyectos recientes. Un botón CTA (Call To Action) para ver más de mis proyectos.

1. **Página de Proyectos**

**Contenido:** Galería interactiva de los proyectos en los que he trabajado, ya sea en programación, arte digital, o ambos. Cada proyecto se presentará en tarjetas con una imagen, una breve descripción y un botón que permite ver más detalles del proyecto.

**Funcionalidades:** Filtro de proyectos por categoría (Desarrollo, Diseño, etc.), y consumo de una API para mostrar proyectos destacados o una lista de tecnologías populares.

1. **Página de Contacto**

**Contenido:** Un formulario de contacto sencillo para que las personas puedan enviarme mensajes o consultas, con campos como nombre, correo y mensaje. Además, incluiré íconos que redirija a mis redes sociales profesionales, como LinkedIn, GitHub, y Behance, para que puedan ver más sobre mi trabajo y producción.

**Funcionalidades:**

* **Validación del formulario:** Verificar que todos los campos estén llenos correctamente antes de enviar.
* **Redes sociales:** Añadir enlaces directos a mis perfiles en LinkedIn, GitHub, Behance (u otras redes profesionales que tenga) mediante íconos interactivos, que se mostrarán en la parte inferior de la pantalla.

# Paso 2: Buscar Diseños para el Portafolio

<https://jhey.dev/>

<https://brittanychiang.com/>

<https://emikukis.wixsite.com/emixcp>

<https://rhienon.com/>

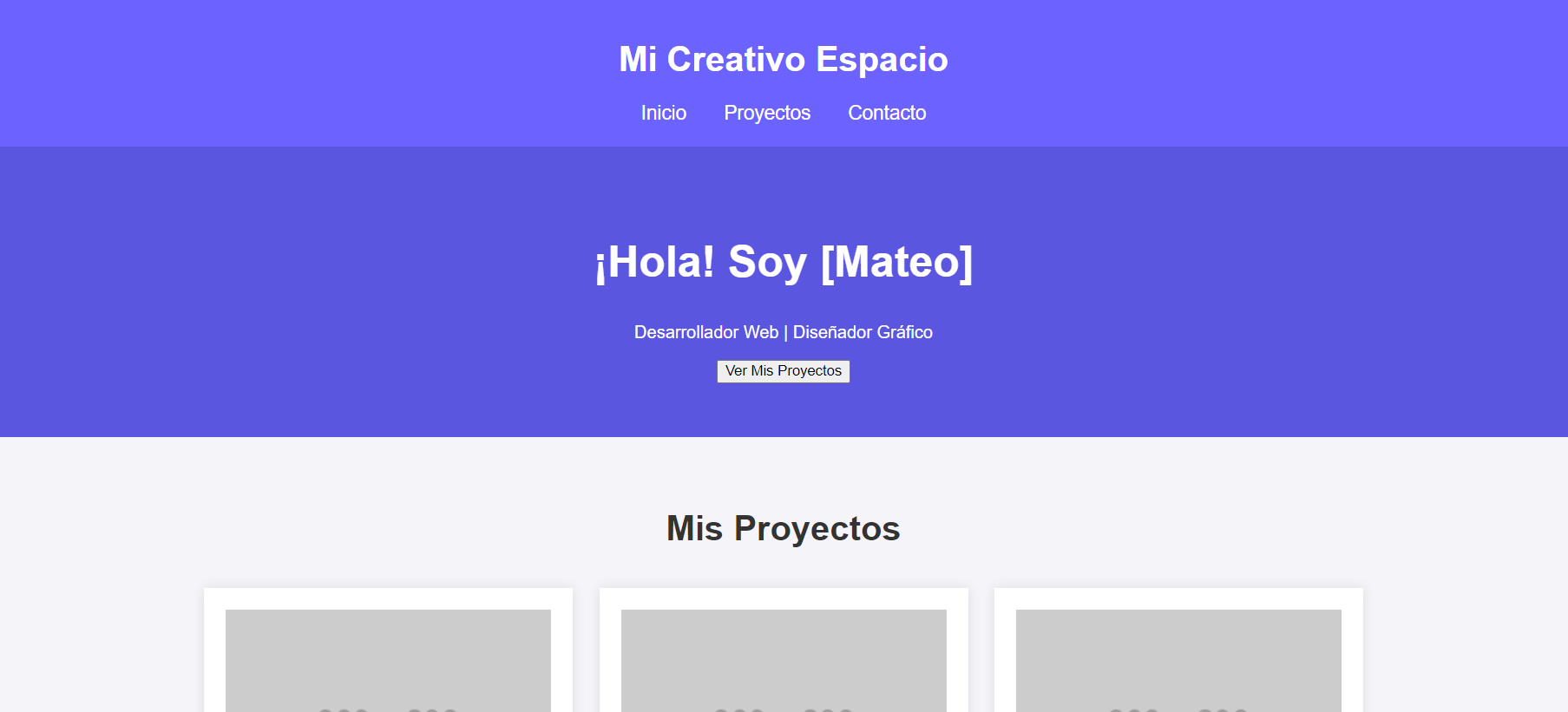
<https://oneeshi.uwu.ai/>

<https://dunks1980.com/>

<https://www.adhamdannaway.com/>

<https://tamalsen.dev/>

# Paso 3: Crear el Mockup del Portafolio



# Paso 4: Funciones para el Portafolio

**Filtro de Proyectos:** En la sección de proyectos, habrá un filtro dinámico que permitirá a los usuarios buscar proyectos por tecnología o categoría (por ejemplo, "React", "Diseño Gráfico"). Se puede crear una pequeña lógica que filtre los elementos en base a tags.

**Animación y Efectos Visuales Interactivos en Todo el Portafolio:** En mi portafolio, quiero implementar una función de animación que haga la navegación más fluida, atractiva y envolvente para los usuarios. La idea es que todos los elementos interactivos, como los botones, imágenes y secciones, respondan dinámicamente a las acciones del usuario.

Botones Interactivos: Los botones tendrán un efecto de transformación suave cuando el usuario pase el cursor sobre ellos. Al hacer hover sobre un botón, cambiará de color gradualmente, se elevará ligeramente y tendrá un efecto de sombra que lo hará resaltar. Quiero que esta animación sea rápida, pero lo suficientemente visible para crear una sensación de interactividad agradable. Además, al hacer clic, el botón mostrará un efecto de "presión", con una reducción temporal en su tamaño, volviendo a su estado normal después de la acción.

Imágenes que Cambian al Desplazarse: A medida que el usuario se desplaza por la página, las imágenes y gráficos se moverán o cambiarán ligeramente. Por ejemplo, las imágenes pueden tener un efecto de paralaje, en el que se mueven más lento que el resto del contenido para crear una sensación de profundidad. También podría implementar un cambio sutil en el brillo o contraste de las imágenes a medida que el usuario pasa sobre ellas con el mouse, lo que agregaría dinamismo.

Transiciones entre Secciones: En lugar de saltar bruscamente entre secciones al desplazarse, quiero agregar transiciones suaves. Cada nueva sección puede aparecer gradualmente con un efecto de desvanecimiento o deslizamiento desde un lado. Esto ayudará a crear una navegación más fluida y atractiva, haciendo que los cambios de contenido sean menos abruptos.

Elementos Emergentes (Pop-ups o Tooltips): Algunos textos o íconos importantes tendrán efectos emergentes cuando el usuario pase el mouse sobre ellos. Quiero que, al hover sobre ciertos íconos o palabras clave, aparezcan pequeñas ventanas emergentes con información adicional o descripciones, desvaneciéndose suavemente cuando el cursor se aleje.

Animaciones Basadas en el Scroll: A medida que el usuario se desplaza hacia abajo en la página, los elementos clave del contenido (como títulos, párrafos importantes o tarjetas de proyectos) aparecerán gradualmente mediante efectos de deslizamiento o aumento de opacidad. Esto hará que el contenido se sienta vivo y en movimiento, en lugar de estático.

Hover en Tarjetas de Proyectos: Las tarjetas que muestran mis proyectos tendrán animaciones interactivas cuando el usuario pase el cursor sobre ellas. Estas tarjetas pueden expandirse ligeramente, mostrar una sombra más pronunciada o hacer una transición suave hacia un estado donde se muestra más información, como una breve descripción o un botón de “Ver más”.

En general, mi objetivo es que la experiencia de navegar por mi portafolio sea dinámica, fluida y visualmente atractiva, haciendo que cada interacción sea gratificante y profesional, mientras demuestro mis habilidades tanto en animación como en diseño interactivo.

# Paso 5: Planificación de Clases (POO)

Para mantener el portafolio modular y organizado, seguiré aplicando principios de POO. Implementaré herencia para que las clases que compartan atributos o comportamientos comunes hereden de una clase base, lo que hará el código más reutilizable y fácil de mantener.

Clase Base: ElementoInteractivo

Propósito:

Esta clase base contendrá los atributos y métodos que comparten todos los elementos interactivos del portafolio, como botones, imágenes o tarjetas. Clases derivadas podrán heredar de ElementoInteractivo para evitar la duplicación de código.

Atributos:

elemento (DOM element): El elemento HTML que se va a manipular.

evento (string): El tipo de evento que activará la interacción (por ejemplo, "click", "hover", "scroll").

Métodos:

agregarEvento(): Vincula un evento al elemento para aplicar la funcionalidad interactiva.

removerEvento(): Permite eliminar el evento del elemento si es necesario.

Ejemplo de uso:

Las clases específicas como BotonAnimado o ImagenInteractiva heredan de esta clase y no necesitan reimplementar la lógica de agregar o remover eventos.

Clase Derivada: BotonAnimado (hereda de ElementoInteractivo)

Propósito:

Esta clase será específica para los botones animados del portafolio. Hereda de ElementoInteractivo y añade animaciones personalizadas para botones, como cambios de color, elevación o efectos de presión cuando se hace clic.

Atributos:

Hereda los atributos elemento y evento de la clase ElementoInteractivo.

colorInicial (string): Color original del botón.

colorHover (string): Color del botón cuando se pasa el mouse sobre él.

animacionDuracion (number): La duración de la animación en milisegundos.

Métodos:

aplicarAnimacionHover(): Sobrescribe el método agregarEvento de la clase base para aplicar una animación de hover, cambiando el color del botón y elevándolo suavemente cuando el cursor se posiciona sobre él.

animacionClick(): Aplica un efecto de presión cuando el botón es clickeado, reduciendo su tamaño temporalmente y luego volviendo a su estado original.

Ejemplo de uso:

Crear una instancia de BotonAnimado para cada botón interactivo en la página. Cada botón reaccionará con animaciones al hover y al clic de manera dinámica, sin necesidad de reescribir la lógica de eventos.

Clase Derivada: ImagenInteractiva (hereda de ElementoInteractivo)

Propósito:

Esta clase manejará el comportamiento de las imágenes del portafolio. Al igual que BotonAnimado, hereda de ElementoInteractivo para agregar y gestionar eventos como el desplazamiento o el hover, pero con animaciones específicas para imágenes.

Atributos:

Hereda los atributos elemento y evento de la clase ElementoInteractivo.

posicionInicial (number): Posición de la imagen antes del scroll.

efectoParallax (boolean): Define si se aplica un efecto de paralaje al desplazarse.

efectoHover (boolean): Define si la imagen cambia al hacer hover.

Métodos:

aplicarEfectoParallax(): Sobrescribe agregarEvento para agregar el efecto de paralaje a la imagen cuando el usuario se desplaza por la página.

cambiarBrilloHover(): Aplica un cambio en el brillo o contraste de la imagen cuando se pasa el cursor sobre ella.

Ejemplo de uso:

Utilizo esta clase para hacer que las imágenes se desplacen suavemente con el efecto paralaje, dando una sensación de profundidad mientras el usuario navega por la página. También cambiarán de aspecto (brillo/contraste) al hacer hover.

Clase FormularioContacto

Propósito:

Heredando de la clase FormularioBase, FormularioContacto manejará la lógica específica del formulario de contacto. Se encarga de enviar mensajes a mi correo y mostrar los enlaces a mis redes sociales. Esta clase será una versión más especializada de un formulario genérico.

Atributos:

nombreRemitente (string): Nombre del remitente.

emailRemitente (string): Correo del remitente.

mensaje (string): Mensaje que enviará el remitente.

redesSociales (array): Un arreglo con enlaces a mis redes sociales.

Métodos:

validarFormulario(): Revisa que los campos no estén vacíos y que el correo sea válido.

enviarMensaje(): Envía el mensaje a mi correo electrónico o base de datos.

mostrarRedesSociales(): Genera dinámicamente los íconos y enlaces a mis redes profesionales.

Ejemplo de uso:

Esta clase me permitirá gestionar de manera eficiente el formulario de contacto en mi página. Utilizaré validarFormulario antes de enviar el mensaje y mostrarRedesSociales para mostrar dinámicamente mis enlaces a LinkedIn, GitHub, etc.

Clase Base: FormularioBase

Propósito:

Esta clase base contiene la estructura básica de cualquier formulario en el portafolio. Clases como FormularioContacto heredan de ella para no duplicar lógica común, como la validación de campos o el envío de datos.

Atributos:

campos (array): Un arreglo que contiene todos los campos del formulario.

botonEnviar (DOM element): El botón de envío del formulario.

Métodos:

validarCampos(): Verifica que todos los campos tengan valores válidos.

enviar(): Método genérico que puede ser sobreescrito por clases derivadas para enviar los datos de manera personalizada.

Ejemplo de uso:

La clase FormularioBase me permitirá crear formularios adicionales en el futuro (como un formulario de suscripción) sin duplicar la lógica de validación y envío.

Justificación del Uso de Herencia:

Reutilización de Código: Gracias a la herencia, evito repetir lógica común entre elementos interactivos y formularios. Esto hace el código más limpio y fácil de mantener.

Organización Modular: Separar la funcionalidad común en una clase base (ElementoInteractivo y FormularioBase) y crear clases derivadas permite que cada clase tenga un propósito claro, lo que facilita la comprensión y futuras expansiones del proyecto.

Escalabilidad: Si en el futuro necesito agregar más formularios o nuevos elementos interactivos, puedo simplemente crear nuevas clases derivadas que hereden de las clases base, reutilizando la lógica existente.

# Paso 6: Repositorio en GitHub

<https://github.com/TeoCast04/-portfolio_-Mateo-Casta-o-.git>