

- Portafolio de Investigación y Desarrollo
 - Tecnologías y Herramientas
 - Despliegue en Google Cloud Run
 - Organización del Código
 - Instrucciones de Ejecución
 - Glosario de Términos Técnicos

Portafolio de Investigación y Desarrollo

Este proyecto consiste en una página web diseñada como portafolio digital para la asignatura de Investigación y Desarrollo. La idea principal es presentar de manera interactiva las diferentes prácticas realizadas durante el ciclo, abarcando temas desde investigación de mercado hasta manufactura con corte láser e impresión 3D.

Tecnologías y Herramientas

Hemos desarrollado este sitio utilizando un stack moderno para asegurar un buen rendimiento y una experiencia de usuario fluida. A continuación detallé las herramientas principales:

React Utilizamos React como librería principal. Esto nos permitió organizar el código en componentes modulares, facilitando la creación de interfaces complejas y el mantenimiento del proyecto a largo plazo.

Vite Elegimos Vite como entorno de desarrollo por su velocidad. Acelera considerablemente el tiempo de carga mientras programamos y optimiza el código final para producción, asegurando que la página cargue rápido.

Tailwind CSS Para el diseño visual optamos por Tailwind. Es un framework de CSS que nos permite aplicar estilos directamente en el código de manera rápida y consistente. Nos ayudó a mantener un diseño coherente y totalmente responsive (adaptable a móviles) sin la complejidad de manejar múltiples hojas de estilo.

Framer Motion Queríamos que la página se sintiera dinámica y "viva", por lo que integramos Framer Motion. Esta librería maneja las animaciones de entrada, las transiciones suaves entre páginas y los efectos interactivos cuando el usuario navega por el sitio.

React Router Gestiona el enrutamiento del sitio, permitiendo navegar entre las distintas secciones (Inicio, Prácticas, Exámenes) de manera fluida sin recargas de página completas, simulando una aplicación nativa.

Despliegue en Google Cloud Run

Para publicar la página en internet y que sea accesible globalmente, utilizamos **Google Cloud Run**. Este servicio nos permite ejecutar la página web en la "nube" de Google de manera segura y escalable.

El proceso funciona de la siguiente manera:

1. **Empaquetado (Docker)**: Primero, "empaquetamos" toda la aplicación en lo que se conoce como un **contenedor Docker**. Imagina esto como meter la página web y todos los programas que necesita para funcionar en una caja cerrada y estandarizada. Para esto usamos el archivo **Dockerfile** incluido en el proyecto, el cual sigue dos pasos:
 - Construye la aplicación (transfoma el código para que el navegador lo entienda).
 - Configura un servidor ligero llamado **Nginx** (usando el archivo **nginx.conf**) para mostrar la página a los visitantes.
2. **Publicación**: Una vez creada esta "caja" (imagen de Docker), la enviamos a Google Cloud. Cloud Run toma esa imagen y la ejecuta automáticamente. Lo bueno de este servicio es que se encarga de todo el mantenimiento del servidor por nosotros y nos entrega una dirección web (URL) segura (HTTPS) para compartir el portafolio.

Organización del Código

El proyecto sigue una estructura estándar para facilitar su lectura:

- **src/pages**: Contiene las vistas principales de la aplicación. Dentro de la carpeta **practices** se encuentran las páginas individuales documentando cada trabajo práctico.
- **src/components**: Aquí se encuentran los elementos de interfaz compartidos en toda la web, como el menú lateral (**SideMenu**), las tarjetas de proyectos

([SpotlightCard](#)) y elementos visuales como el cursor personalizado.

- **public:** Almacena los recursos estáticos como imágenes y logos.

Instrucciones de Ejecución

Para visualizar este proyecto en un entorno local:

1. Asegúrate de tener Node.js instalado en tu equipo.
2. Instala las dependencias necesarias ejecutando `npm install` en la terminal dentro de la carpeta del proyecto.
3. Inicia el servidor de desarrollo con el comando `npm run dev`.
4. Abre la dirección local que aparece en la terminal (usualmente `localhost:5173`) en tu navegador web.

Glosario de Términos Técnicos

Para facilitar la comprensión de la documentación, aquí explicamos algunos de los conceptos técnicos mencionados:

- **Componente:** Es como un bloque de construcción independiente (ej. un botón o una tarjeta). En lugar de reescribir el mismo código muchas veces, creamos el componente una vez y lo reutilizamos en varias partes de la página.
- **Framework / Librería:** Conjunto de herramientas y código preescrito que usamos para no tener que empezar todo desde cero. Nos da una base sólida sobre la cual construir.
- **Responsivo (Responsive):** Capacidad de una página web para adaptarse y verse bien automáticamente en cualquier tamaño de pantalla, ya sea un monitor grande, una tableta o un teléfono celular.
- **Interfaz de Usuario (UI):** Todo aquello con lo que el usuario interactúa en la pantalla; botones, menús, textos y la disposición visual de estos elementos.
- **Enrutamiento (Routing):** El sistema que decide qué contenido mostrar según la dirección web (URL) en la que se encuentre el usuario, permitiendo "navegar" entre páginas.
- **Dependencias:** Paquetes de código externos creados por otros desarrolladores que instalamos en nuestro proyecto para añadir funcionalidades específicas (como animaciones o iconos) sin tener que programarlas nosotros mismos.

- **Deploy (Despliegue):** El proceso de publicar la página web en un servidor para que sea accesible por cualquier persona a través de internet.