

Proyecto Vite + React + TypeScript

Descripción

Este es un proyecto creado con Vite, React y TypeScript para desarrollo web moderno con configuración optimizada y tipado estático.

Tecnologías utilizadas

- **Vite** 5.2.3 - Build tool y servidor de desarrollo
- **React** - Biblioteca de JavaScript para interfaces de usuario
- **TypeScript** - Superset de JavaScript con tipado estático

Prerrequisitos

- Node.js (versión 16 o superior)
- npm o yarn
- Editor de código (VS Code recomendado)

Instalación

1. Crear el proyecto

Abre tu editor de código y sigue estos pasos:

1. Crear la carpeta del proyecto desde el editor de código
2. Abrir la terminal integrada de comandos
3. Ejecutar el siguiente comando:

```
bash  
  
npm create vite@5.2.3
```

4. Configurar el proyecto:

- Asignar nombre del proyecto y presionar Enter
- Confirmar el mismo nombre de paquete (Enter)
- **Select Framework:** React
- **Select a variant:** TypeScript

2. Instalar dependencias y ejecutar

Ejecutar los siguientes comandos uno por uno en la terminal:

```
bash
```

```
cd reactApp  
npm install  
npm run dev
```

3. Abrir en el navegador

- **Ctrl + Clic** en el link que aparece en la terminal para abrir el proyecto en el navegador
- Por defecto se ejecuta en: `http://localhost:5173`

Scripts disponibles

```
bash
```

```
# Servidor de desarrollo
```

```
npm run dev
```

```
# Build para producción
```

```
npm run build
```

```
# Preview del build de producción
```

```
npm run preview
```

```
# Linting
```

```
npm run lint
```

Estructura del proyecto

```
reactApp/  
├── public/  
├── src/  
│   ├── assets/  
│   ├── App.tsx  
│   ├── main.tsx  
│   └── index.css  
├── index.html  
├── package.json  
├── tsconfig.json  
├── vite.config.ts  
└── README.md
```

Desarrollo

- El servidor de desarrollo se recarga automáticamente al guardar cambios

- TypeScript proporciona verificación de tipos en tiempo real
- Vite ofrece Hot Module Replacement (HMR) para una experiencia de desarrollo rápida

Build para producción

```
bash  
  
npm run build
```

Los archivos optimizados se generan en la carpeta `dist/`

Contribución

1. Fork el proyecto
2. Crea una rama para tu feature (`git checkout -b feature/nueva-funcionalidad`)
3. Commit tus cambios (`git commit -am 'Agregar nueva funcionalidad'`)
4. Push a la rama (`git push origin feature/nueva-funcionalidad`)
5. Abre un Pull Request

Licencia

Este proyecto está bajo la Licencia MIT.

¡Feliz desarrollo! 🚀