

Guía de Buenas Prácticas

Proyecto Tejelas Vivi

1. Estructura de Archivos

Tu estructura actual está bien organizada. Esta es una convención recomendada:

```
1. tejelas_vivi/
2. |   backend/           # Código PHP y configuración de API
3. |   |   api/           # Endpoints de acceso a datos
4. |   |   |   productos/ # Separación lógica por entidad
5. |   |   |   config/    # Conexión a base de datos
6. |   |   |   public/    # Recursos públicos (favicon, imágenes no procesadas por Vite)
7. |   |   |   src/       # Código fuente principal de React
8. |   |   |   |   assets/ # Imágenes, logos, íconos
9. |   |   |   |   components/ # Componentes reutilizables
10. |   |   |   |   pages/    # Rutas o páginas completas
11. |   |   |   |   services/ # Servicios externos o helpers de API
12. |   |   |   |   styles/   # Archivos CSS globales
13. |   |   |   |   utils/    # Funciones reutilizables (ej: validaciones)
14. |
```

2. Convenciones de Nomenclatura

Elemento	Convención	Ejemplo
Componentes React	PascalCase	ProductCard.jsx
Funciones/variables	camelCase	handleContactClick
Hooks personalizados	use + Nombre	useProductos (si aplicara)
Archivos JS	CamelCase (modular)	ContactForm.jsx, api.js
Archivos CSS	lowercase y guiones	global.css, form-styles.css

3. Uso de Variables y Estado

- Usa useState, useEffect, y useContext de forma coherente.
- Guarda el estado persistente en localStorage solo si es necesario.
- Centraliza las llamadas fetch en un archivo api.js (ya estás usando uno).

4. Accesibilidad (a11y)

- Usa aria-label, aria-expanded, role, tabIndex en elementos interactivos.
- Asegúrate de que todos los botones e inputs tengan texto claro o aria-label.
- Verifica buen contraste de colores (ej. WebAIM).

5. Usabilidad

- Formulario de contacto con autocompletado y validación.
- Scroll automático con `scrollIntoView()`.
- Carruseles adaptativos y responsivos.
- Feedback al usuario con mensajes de carga o error.

6. Reutilización y Componentización

- Tienes componentes reutilizables: ServiceCard, ProductCard, FAQ, Navbar, Footer.
- Mantén estos componentes puros, controlados por props.

7. Seguridad y Buenas Prácticas PHP

- Usar `mysqli_set_charset($conn, "utf8")` para caracteres especiales.
- Validar y sanitizar toda entrada en scripts PHP que reciban datos del cliente.

8. Recomendaciones Adicionales

- Usa PropTypes o TypeScript para validar props.
- Centraliza tus rutas.
- Documenta tus componentes con comentarios útiles.