# 📃Arquitectura Frontend – React (Monolito Modular)

Esta es la arquitectura del frontend en React, la cual es modular, reutilizable y fácil de escalar, organizado por módulos funcionales y con carpetas dedicadas a rutas centralizadas, hooks y componentes compartidos, layouts reutilizables y estilos organizados.

## Estructura General de Carpetas

src/  
│  
├── routes/ # Definición centralizada de rutas  
│  
├── hooks/ # Hooks genéricos y reutilizables  
│  
├── components/ # Componentes compartidos entre módulos  
│  
├── layout/ # Layouts principales de la aplicación  
│  
├── modules/ # Módulos funcionales  
│ ├── users/ # Módulo de Usuarios  
│ │ ├── pages/ # Páginas principales  
│ │ ├── components/ # Componentes específicos del módulo  
│ │ ├── styles/ # Estilos específicos  
│ │ ├── hooks/ # Hooks propios del módulo  
│ │ └── services/ # Servicios HTTP relacionados  
│ │  
│ ├── providers/ # Módulo de Prestadores  
│ │ ├── pages/  
│ │ ├── components/  
│ │ ├── styles/  
│ │ ├── hooks/  
│ │ └── services/  
│ │  
│ ├── rubros/ # Módulo de Rubros  
│ │ ├── pages/  
│ │ ├── components/  
│ │ ├── styles/  
│ │ ├── hooks/  
│ │ └── services/  
│ │  
│ ├── ratings/ # Módulo de Calificaciones  
│ │ ├── pages/  
│ │ ├── components/  
│ │ ├── styles/  
│ │ ├── hooks/  
│ │ └── services/  
│ │  
│ ├── comments/ # Módulo de Comentarios  
│ │ ├── pages/  
│ │ ├── components/  
│ │ ├── styles/  
│ │ ├── hooks/  
│ │ └── services/  
│ │  
│ └── admin/ # Módulo de Administración  
│ ├── pages/  
│ ├── components/  
│ ├── styles/  
│ ├── hooks/  
│ └── services/  
│  
└── index.js # Punto de entrada de la app

## Justificación de la Arquitectura

### 1. Organización por Módulos

Cada módulo (usuarios, prestadores, rubros, etc.) es autocontenido y tiene:  
- pages/: vistas principales.  
- components/: componentes específicos del módulo.  
- styles/: estilos propios.  
- hooks/: lógica reutilizable dentro del módulo.  
- services/: comunicación HTTP con el backend.

### 2. Carpeta routes/

Centraliza la definición de rutas con React Router.  
Facilita el control de accesos (ej. rutas privadas vs públicas).

### 3. Carpeta hooks/

Contiene hooks genéricos que no pertenecen a un módulo específico.  
Ejemplo: useAuth, useFetch.

### 4. Carpeta components/

Guarda componentes compartidos entre módulos.  
Ejemplo: Button, Modal, Navbar, Pagination.

### 5. Carpeta layout/

Define los layouts globales de la aplicación.  
Ejemplo: MainLayout (navbar, footer) y AdminLayout (sidebar + panel).

## Principios Aplicados

- SRP (Single Responsibility Principle): cada módulo y carpeta tiene un propósito único.  
- Reutilización: componentes y hooks compartidos para evitar duplicación.  
- Escalabilidad: cada módulo puede crecer sin afectar al resto.  
- Modularidad: favorece la migración futura a microfrontends si fuera necesario.  
- Mantenibilidad: estructura clara que facilita onboarding de nuevos desarrolladores.