

Arquitectura hexagonal a un proyecto que contiene funciones de autenticación (login y registro) y un catálogo de productos, todo ello consumiendo APIs a través de JavaScript. En esta estructura, separaremos claramente las responsabilidades en diferentes capas: Core de la Aplicación, Adaptadores de Entrada/Salida y las Interfaces Externas.

Core de la Aplicación:

```
src/  
├─ core/  
│   ├─ auth/  
│   │   └─ AuthUseCase.js  
│   └─ product/  
│       └─ ProductUseCase.js  
└─ main.js
```

AuthUseCase.js contiene la lógica para manejar el proceso de autenticación (login y registro) utilizando llamadas a los adaptadores de entrada/salida apropiados.

ProductUseCase.js contiene la lógica para obtener y manejar los productos del catálogo utilizando llamadas a los adaptadores de entrada/salida apropiados.

Adaptadores de Entrada/Salida:

Adaptador de API:

```
src/  
├─ adapters/  
│   └─ api/  
│       ├── AuthAPI.js  
│       └── ProductAPI.js
```

AuthAPI.js contiene funciones para interactuar con las APIs de autenticación (login y registro).

ProductAPI.js contiene funciones para interactuar con las APIs relacionadas con el catálogo de productos.

Adaptador de Interfaz de Usuario (UI):

```
src/  
├─ adapters/  
│   └─ ui/  
│       ├── AuthUI.js  
│       ├── ProductUI.js  
│       └─ components/  
│           ├── LoginPage.js  
│           ├── RegistrationPage.js  
│           └─ ProductCatalogPage.js  
└─ main.js
```

AuthUI.js contiene la lógica de la interfaz de usuario para el proceso de autenticación.

ProductUI.js contiene la lógica de la interfaz de usuario para mostrar y manejar el catálogo de productos.

Los componentes en la carpeta components representan las páginas y elementos de interfaz de usuario que interactúan con los casos de uso correspondientes.

Interfaces Externas:

Interfaces de API:

```
src/  
├─ frameworks/  
│   └─ api/  
│       └─ AxiosAdapter.js
```

AxiosAdapter.js proporciona una implementación concreta para realizar solicitudes HTTP utilizando Axios u otra librería similar.

Con esta estructura, los adaptadores de API actúan como puentes entre los casos de uso en el core de la aplicación y las APIs externas. Los adaptadores de UI manejan la interacción con el usuario y utilizan los casos de uso apropiados para realizar las operaciones necesarias.