Descripción General

El Store es un sistema de estado global que implementa el patrón Pub/Sub (Publicador/Suscriptor) para gestionar el estado de la aplicación Reactiva.

Características Principales

- Estado global centralizado
- Patrón Pub/Sub para reactividad
- Inmutabilidad controlada
- Sistema de roles integrado
- Helpers para acciones comunes

Estructura del Estado

Estado Inicial

```javascript
const state = {
  user: null,                    // Usuario autenticado
  role: ROLES.GUEST,             // Rol del usuario actual
  currentProject: null,          // Proyecto activo (multi-proyecto)
  selectedDevice: null,          // Dispositivo seleccionado
  devices: []                    // Lista de dispositivos disponibles
};
```

Propiedades del Estado

Propiedad	Tipo	Descripción
`user`	`Object | null`	Información del usuario `{name, email}`
`role`	`String`	Rol actual (`admin`, `action`, `readonly`, `guest`)
`currentProject`	`String | null`	ID o slug del proyecto activo
`selectedDevice`	`Object | null`	Dispositivo seleccionado `{id, nombre, tipo, estado}`
`devices`	`Array`	Lista de dispositivos disponibles

Sistema de Roles

Definición de Roles

```
javascript
```

```
const ROLES = window.__CONFIG?.ROLES ?? {
  ADMIN: "admin",
  ACTION: "action",
  READONLY: "readonly",
  GUEST: "guest"
};
```

Jerarquía de Roles

- `GUEST`: Usuario no autenticado (rol por defecto)
- `READONLY`: Solo lectura
- `ACTION`: Lectura + acciones básicas
- `ADMIN`: Acceso completo + configuración

API del Store

Funciones Principales

`getState()`

Propósito: Obtiene una copia inmutable del estado actual.

```
javascript
export const getState = () => ({ ...state });
```

Uso:

```
javascript
import { getState } from '../state/store.js';
```

```
const currentState = getState();
console.log(currentState.user); // { name: "Juan", email: "juan@email.com" }
```

`subscribe(fn)`

Propósito: Suscribe una función a cambios del estado.

```
javascript
export const subscribe = fn => (subs.add(fn), () => subs.delete(fn));
```

Uso:

```
javascript
const unsubscribe = subscribe((state) => {
  console.log('Estado actualizado:', state);
});

// Para desuscribirse
unsubscribe();
```

`setState(patch)`

Propósito: Actualiza el estado parcialmente y notifica a los suscriptores.

```
javascript
export function setState(patch) {
  Object.assign(state, patch);
  emit();
```

```
}
```

Uso:

```
javascript
```

```
import { setState } from '../state/store.js';


// Actualización parcial
setState({
  user: { name: "María", email: "maria@email.com" },
  role: "admin"
});
```

Funciones Específicas (Helpers)

`login(user, role)`

Propósito: Establece el usuario autenticado y su rol.

```
javascript
```

```
export function login(user, role) {
  setState({ user, role });
}
```

Uso:

```
javascript
```

```
login(
  { name: "Carlos", email: "carlos@empresa.com" },
```

```
        ROLES_CONST.ADMIN
);
```

`logout()`

Propósito: Cierra sesión y restablece el estado inicial.

```
javascript
```

```
      export function logout() {
        setState({
          user: null,
          role: ROLES.GUEST,
          currentProject: null,
          selectedDevice: null,
          devices: []
        });
}
```

Uso:

```
javascript
logout(); // Restablece todo el estado
```

`setDevices(devices)`

Propósito: Actualiza la lista de dispositivos disponibles.

```
javascript
```

```
export function setDevices(devices) {
  setState({ devices });
}
```

Uso:

```
javascript
setDevices([
  { id: 1, nombre: "Sensor 1", tipo: "sensor", estado: "activo" },
  { id: 2, nombre: "Actuador A", tipo: "actuador", estado: "inactivo"
}
]);
```

`selectDevice(device)`

Propósito: Establece un dispositivo como seleccionado.

```
javascript
export function selectDevice(device) {
  setState({ selectedDevice: device });
}
```

Uso:

```
javascript
selectDevice({
  id: 1,
  nombre: "Termómetro Principal",
  tipo: "sensor_temperatura",
```

```
    estado: "activo"
});
```

`clearSelectedDevice()`

Propósito: Limpia la selección de dispositivo.

```
javascript

export function clearSelectedDevice() {
  setState({ selectedDevice: null });
}
```

Uso:

```
javascript

clearSelectedDevice(); // selectedDevice = null
```

Flujo de Actualización

1. Acción → `setState()`, `login()`, `logout()`, etc.
2. Actualización → `Object.assign(state, patch)`
3. Notificación → `emit()` ejecuta todos los suscriptores
4. Reacción → Componentes actualizan su UI