



# **UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”**

Ciencias de la Computación  
Desarrollo de Software Seguro

## **Informe Patrones de Diseño**

### **NOMBRES:**

Cadena Jeremy  
Tiamba Henry  
Valarezo Andrés

**Ingeniero:** ANGEL GEOVANNY CUDCO POMAGUALLI

**NRC:** 15589

Sangolquí, 05 de febrero del 2024

## 1. PROBLEMÁTICA

En la biblioteca central "Alejandro Segovia" de la Universidad de las Fuerzas Armadas ESPE, se requiere realizar la captura de un listado de varios libros nuevos junto al nombre de su autor que se van a agregar a su colección. Al mismo tiempo se requiere registrar los préstamos de los libros a los estudiantes.

La aplicación necesita operar en la nube, con un objetivo de nivel de servicio alto, pues varias aplicaciones clientes en distintas plataformas, tanto de dispositivos móviles como de escritorio, desean conectarse y consultar ese listado en Internet.

Además, se desea publicarlo en tiempo real en la página web de la ESPE. Esta consideración hace que la biblioteca se decante por una arquitectura de microservicios.

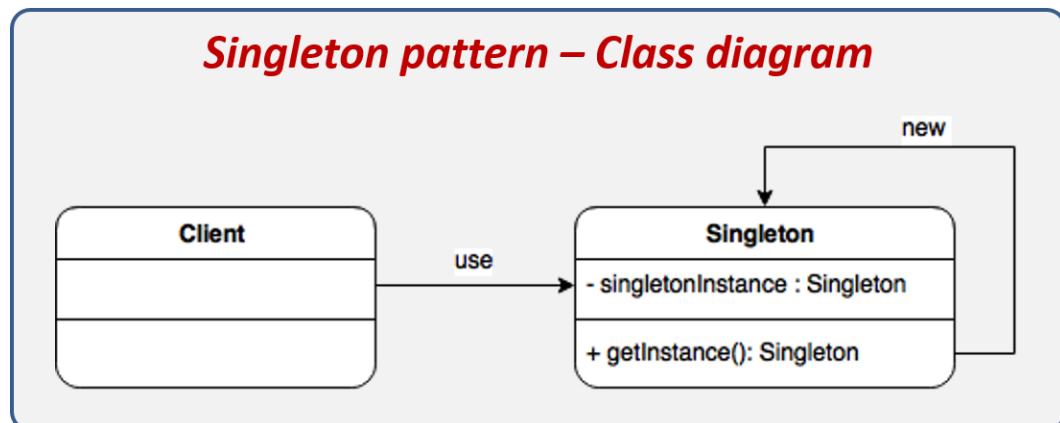
## 2. MODELADO DE ATAQUES

Patrones Creacionales

Singleton:

Explicación: Garantiza que exista solo una instancia de un objeto y proporciona un punto de acceso global a esa instancia.

Implementación en la Problemática: Puedes aplicar Singleton para asegurarte de que ciertos servicios, como la gestión de la colección de libros o el sistema de autenticación, tengan una única instancia compartida en toda la aplicación.

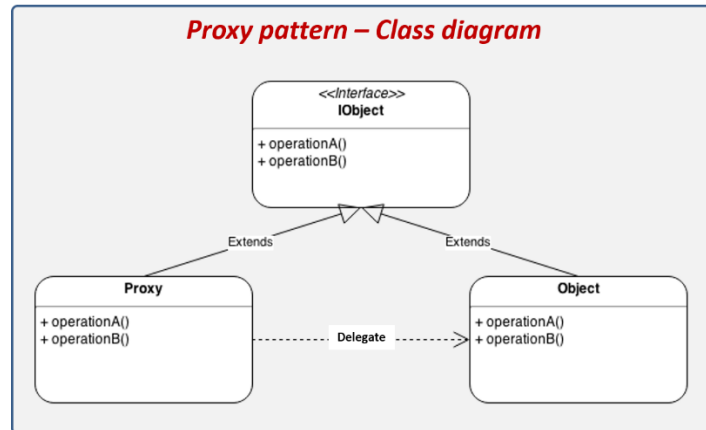


## Patrones Estructurales

### Proxy:

Explicación: Proporciona un sustituto o marcador de posición para controlar el acceso a otro objeto.

Implementación en la Problemática: Puedes usar Proxy para controlar el acceso a recursos críticos, como validar tokens de autenticación antes de permitir el acceso a la información sobre libros o préstamos.

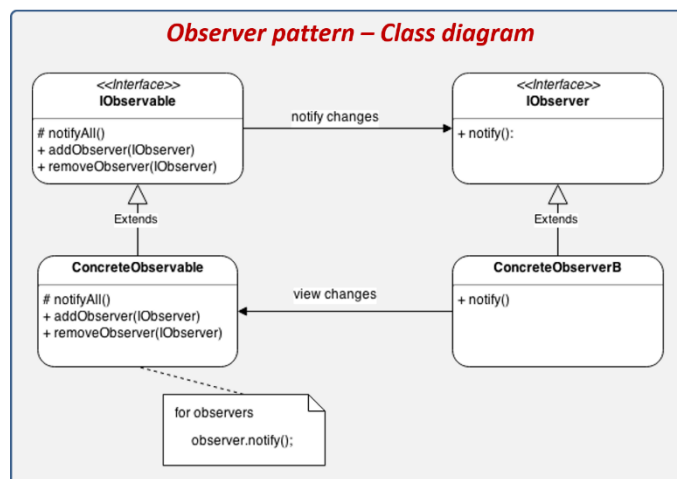


## Patrones de Comportamiento

### Observer:

Explicación: Define una dependencia uno a muchos entre objetos, de modo que cuando un objeto cambia de estado, todos sus dependientes son notificados y actualizados automáticamente.

Implementación en la Problemática: Al implementar Observer se podría notificar a los componentes interesados (interfaz de usuario, página web) sobre cambios en el listado de libros en tiempo real, manteniendo la información actualizada.



### **3. SOLUCIÓN PLANTEADA**

Para abordar la problemática en la biblioteca "Alejandro Segovia", se propone una solución que combine los patrones Singleton, Proxy y Observer. Implementaríamos un Singleton para gestionar de manera única la colección de libros y el sistema de autenticación. La clase Proxy sería responsable de controlar el acceso a recursos, utilizando la validación de roles y permisos para garantizar que solo los usuarios autorizados puedan realizar acciones como agregar o prestar libros. Además, el patrón Observer se aplicaría para notificar a los componentes de la interfaz de usuario y la página web sobre cambios en el listado de libros en tiempo real, asegurando una visualización actualizada de la información. Este enfoque proporcionaría una estructura modular y segura, garantizando la eficiencia en el control de acceso y la notificación en tiempo real de eventos relevantes en la biblioteca.