

Consigna

Actividad: Desarrollo de Estados de Sesión usando State

Temática:

Creación de un sistema de manejo de estado en memoria usando el patrón State

Instrucciones

- A continuación, se le comparte las indicaciones para el desarrollo de la actividad **Desarrollo de Estados de Sesión usando State**
- Esta actividad debe ser entregada:
 - vía commit de repositorio compartido por el profesor, un integrante por grupo por repositorio,
 - implementando los patrones y clases solicitados
 - incluye al menos un usuario admin quemado en código
 - diagrama de clases explicando la interacción de su diseño
 - actualizar README.md de su repositorio agregando a la base de su carpeta incorporando lo anterior, además de configuración para uso.
 - JavaDoc en la clase que creen integrando información necesaria de programación

Asignación

Implementar un sistema de autenticación y control de acceso seguro mediante el patrón *State*, donde cada usuario puede autenticarse una vez y, después, el sistema gestiona automáticamente el acceso mediante un *token* seguro.

Especificaciones:

1. **Control de Acceso:** El sistema debe manejar los estados de autenticación de un usuario.
2. **Autenticación Segura:** Implementar la generación y verificación de *tokens*.
3. **Estados:**
 - **EstadoNoAutenticado:** Usuario no autenticado.
 - **EstadoAutenticado:** Usuario autenticado con acceso a los servicios.
 - **EstadoSesionExpirada:** La sesión o el *token* ha expirado.
4. **Estructura inicial:**
 - **EstadoAutenticacion** (interfaz que define los métodos de los estados).
 - **ContextoAutenticacion** (mantiene el estado actual y permite cambiar entre ellos).
5. **Definir** la interfaz **EstadoAutenticacion**, que especifica los métodos **iniciarSesion()**, **accederServicio()**, y **cerrarSesion()**.
6. **Implementa** tres clases de estado que representen los posibles estados de autenticación:
 - **EstadoNoAutenticado**
 - i. Permite que el usuario se autentique y genera un *token* al cambiar a
 - **EstadoAutenticado**
 - i. Permite el acceso y verifica la validez del *token*.
 - **EstadoSesionExpirada**
 - i. Maneja el caso en que el token ha expirado y obliga al usuario a iniciar sesión nuevamente.
7. Crear la clase **ContextoAutenticacion**:
 - Esta clase mantiene el estado actual y permite cambios de estado.
8. Implementar un menú que permita ver el estado actual CON O SIN SESIÓN, iniciar sesión, salir de sesión.

Puntuación:

Esta actividad será evaluada por medio de la siguiente rúbrica:

README.md completo: 10 puntos

Componente compilando y FUNCIONAL: 80 puntos

Commits (1 por miembro de equipo): 10 puntos

Este ejercicio cuenta como el 2do Estudio de Caso del curso (10% nota total).

Total: 100 puntos