Decisiones de diseño

1) Se decide tener una clase abstracta de tipo usuario donde la implementen la clase administrador y cliente.

Esto es ya que suponemos que la entidad usuario va a tener el método loguearse y otros posibles métodos que van a implementar de igual manera el administrador y el usuario, además estos dos también tienen atributos en común. También tienen métodos con la misma firma pero con distinta implementación.

Podría haber sido interfaz, pero estaría repitiendo lógica y además estaría repitiendo atributos.

2) se crea solo una categoría. Ya que no cambian en comportamiento sino cambian sus atributos

3) se identifica a un dispositivo como encendido o apagado según un valor booleano, esta implementación seguramente luego haya que cambiarla y esto sea un estado para el dispositivo. Con posibles estados como apagado y encendido.

4) Se crea para futuras entregas un Repositorio de categorías. Esto se hace ya que se menciona la inicialización de las categorías por medio de un archivo (en formato JSON). Esto se realizara con el repositorio junto con otras implementaciones.

5) Se decide leer el archivo de usuarios en formato JSON con la dependencia Jackson, antes de la otra sugerida por la catedra como GSON.

Esta elección se debe a: experiencia de algunos integrantes del grupo sobre Jackson, conocimientos previos, mejor documentación.

6) Se utiliza un repositorio de usuarios donde va a contener a los clientes y a los administradores, este se encargara de cargar el archivo de json.

Se decide tener dos DAO uno para administrador y otro para cliente. Implementaran una interfaz que en un futuro podría ser una clase abstracta pero aún no lo consideramos necesario en esta entrega.