

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
UNIDADE LOURDES

RAPHAEL SENA AUGUSTO DE BRITO

Substituição de Exercício:  
Projeto de Integração Multimódulo com Padrões de  
Projeto

Belo Horizonte

2025

RAPHAEL SENA AUGUSTO DE BRITO

Substituição de Exercício:

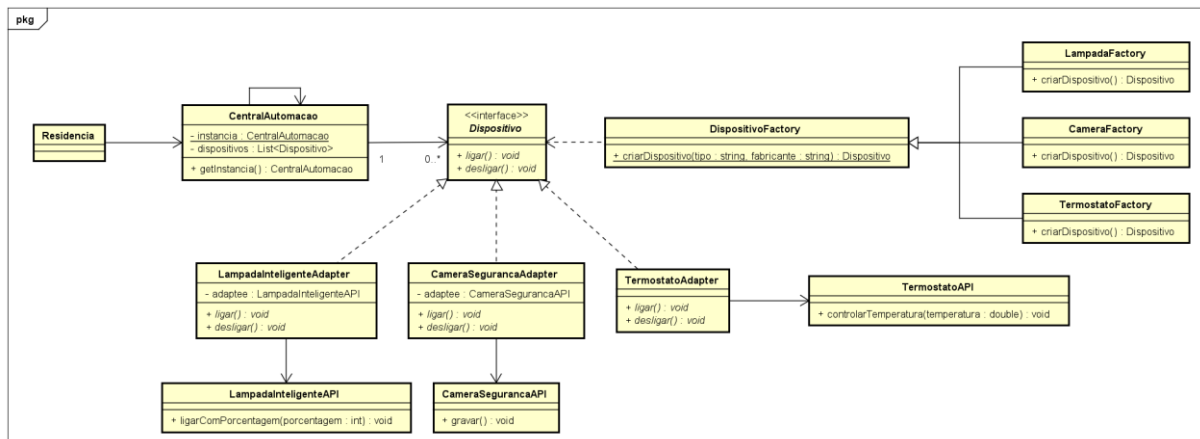
Projeto de Integração Multimódulo com Padrões de  
Projeto

Trabalho de substituição de exercício, apresentado à Disciplina Projeto de Software, lecionada pelo Prof. Dr. Rodrigo de Carvalho da Pontifícia Universidade Católica de Minas Gerais para recuperação do total de 5 pontos, perdidos na tarefa “Pré Prova”.

Belo Horizonte

2025

## I. Diagrama de Classes



## II. Código

O código do projeto está disponível em: <https://github.com/raphael-sena/substituicao-exercicio-projeto>

No repositório do código é possível encontrar o diretório /code que possui os códigos, bem como a classe Main.java para execução dos testes solicitados, conforme enunciado no exercício.

## III. Relatório

- Padrão Singleton:** O padrão Singleton foi empregado como forma de garantir apenas uma instância estática e global para a classe CentralAutomacao. Isto é vantajoso quando se deseja um e somente um ponto chave do sistema, como uma central de controle do sistema.
- Padrão Factory:** Este padrão sola a lógica de criação dos objetos Dispositivo. É vantajoso, por que flexibiliza a criação de diferentes tipos e marcas de dispositivos sem que a classe cliente, no meu exemplo a “Main.java”, precise conhecer os detalhes das classes concretas. Além disso, facilita a manutenção e extensão futura do projeto, como por exemplo, para adicionar um novo tipo de dispositivo, basta adicionar um novo case.
- Padrão Adapter:** O padrão Adapter permite que classes com interfaces incompatíveis trabalhem em conjunto. Em meu contexto, cada dispositivo inteligente tem uma API própria e diferente, como ativar() ou desligar(). Os adapters fazem a tradução dessas APIs para a interface comum Dispositivo, que a central entende. Isso promove desacoplamento entre a CentralAutomacao e as APIs específicas dos dispositivos.

## IV. Teste

Os testes foram realizados tomando como base as criações dos dispositivos e a chamada de funções, as quais estes dispositivos foram atribuídos, como exemplo vindas de diferentes APIs e se adaptando ao contexto da implementação do software desenvolvido para este presente trabalho. Na imagem abaixo é possível visualizar a execução da aplicação e a conexão de desconexão dos dispositivos.

