

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO DE COMPUTAÇÃO  
ENGENHARIA DE SOFTWARE

DANIELA RODRIGUES COUTO COSTA, RA: 2454297  
MARCELO GREGÓRIO VIEIRA, RA: 2454408

## **IMPLEMENTAÇÃO DESIGN PATTERNS**

ARQUITETURA DE SOFTWARE

CORNÉLIO PROCÓPIO

2024

## Objetivo do projeto:

Implementação de um sistema de gerenciamento de contatos em javascript, o sistema deve seguir um padrão de projeto estrutural e deve possuir as seguintes funcionalidade: Adicionar Contatos; Remover Contato; Buscar Contato; Listar Contatos. Sendo que a funcionalidade Buscar Contato deve seguir algum padrão de projeto comportamental escolhido pela equipe

## Estrutura do sistema:

O sistema utiliza o padrão de projeto Facade, visto que possibilita a divisão do sistema em camadas, sendo uma camada simples para o usuário e uma camada complexa responsável pelo funcionamento das funções, considerando o contexto de uma lista de contatos o usuário precisa apenas se preocupar e ter acesso a camada de interface, o que torna a utilização do sistema mais simples e permite correções sem alterar a interface do usuário.

O método buscarContato(valor, op) utiliza o padrão comportamental Strategy, o qual consiste em disponibilizar 3 estratégias distintas para buscar um contato: buscar por número, e-mail ou telefone, a função recebe o valor inserido e estratégia/opção escolhida pelo usuário.

O sistema é dividido em em 4 arquivos de classe e um arquivo de execução/interface:

- Classe Contato.js: Responsável por instanciar um objeto do tipo contato, que contém os atributos nome, telefone e email
- Classe ReadInput.js: Responsável por permitir a captura de dados via terminal, possui construtor, método de leitura de texto (read), método de encerramento das ações da classe (close) e um método para aguardar uma resposta de continuação do usuário (esperar). Importante destacar que a classe ReadInput utiliza funções e métodos assíncronos para realizar a captura de dados.
- Classe GerenciadorContatos.js: Responsável por realizar as principais funcionalidades de gerenciamento de contatos, possui construtor que instancia uma lista de contatos vazia, método para adicionar contatos (adicionarContato), método para remover um contato com base no nome (removerContato), método para buscar um contato por nome, e mail ou telefone utilizando o padrão de projeto comportamental strategy (buscarContato) e um método para listar todos os contatos (listarContatos)

- Classe Facade.js: Responsável por instanciar e agrupar as classes ReadInput, Contato e GerenciadorContatos por meio de exportação de módulos, possui um construtor que instancia os atributos gerContatos (classe GerenciadorContatos), readLine(classe ReadInput) e opcao. A classe utiliza métodos assíncronos com await para capturar os inputs do usuário a partir da variável readLine, a classe possui métodos async relacionados a cada funcionalidade da classe GerenciadorContatos (adicionarContato, removerContato, buscarContato e listarContatos), bem como o método async iniciar que tem como objetivo disponibilizar e gerenciar um menu com do-while e switch-case para que o usuário consiga usufruir de todas as funcionalidades do sistema.
- Arquivo Index.js responsável apenas por instanciar a classe Facade e iniciar o sistema por meio da constante AppContatos (Facade).