

Princípios GRASP utilizados

Creator: O princípio *Creator* é aplicado, pois a classe “DataSource” é responsável por fornecer instâncias de objetos da aplicação, centralizando a lógica de criação e inicialização dos objetos relacionados.

Controller:

- A classe “CampusController” atua como um controlador, lidando com os eventos do sistema relacionados aos campi. Ela representa um cenário de caso de uso do sistema global.
- A classe “ConsoleController” é responsável por gerenciar a interação do sistema com o usuário, controlando a entrada e saída de dados pelo console.

Indirection: O princípio de *Indirection* é aplicado através da classe “GerenciadorReserva”, que atua como uma camada intermediária para administrar as reservas. Ela ajuda a manter o baixo acoplamento, permitindo que outras partes do sistema interajam com as reservas sem precisar conhecer os detalhes de implementação.

Pure Fabrication: Foi criada a classe “DataReserva”, que não representa nenhum conceito no domínio do sistema, mas é uma fabricação pura que auxilia na validação e manipulação de dados relacionados às reservas. Isso ajuda a manter a coesão e a responsabilidade bem definida no sistema.