

Princípios GRASP e onde foram usados:

- Especialista de informação: cada classe possui delegada apenas métodos que acessam informações relacionadas à ela;
- Criador: cada classe instancia seus filhos e nada mais.

Além dos princípios que foram utilizados nas classes do projeto, citados acima, foi utilizado outros princípios, de acordo com o detalhamento a seguir.

A classe `campusControlador`, foi inserida através da utilização do princípio GRASP: controlador, pois foi atribuído a esta a responsabilidade de lidar com os eventos do sistema, onde ela representa um cenário de caso de uso do sistema global.

É perceptível que da forma como foi apresentado os casos de uso resumido, a classe `funcionário` teria muita responsabilidade. Para manter os princípios GRASP: de alta coesão e o baixo acoplamento, podemos fazer com que uma nova classe `ReservaGerenciador` tenha a responsabilidade de administrar as reservas. Com a criação desta nova classe é possível fazer a utilização de mais um princípio GRASP: que é a indireção, pois este ajuda a manter o baixo acoplamento através de delegação de responsabilidades através de uma classe mediadora, sendo este princípio essencial neste projeto, pois a não utilização dele implicaria em mais responsabilidades para a classe `SalaReunião`.

Para obter as salas livres, para fazer as reservas, para não tornar o projeto ruim, foi utilizado o princípio GRASP: acoplamento baixo, para atribuir responsabilidade de modo que o acoplamento permaneça baixo, avaliando as alternativas. Com isso o prédio fica responsável por obter as salas livres e a `salaReunião`, fica responsável por verificar a disponibilidade da sala, de acordo com o horário que se deseja reservar uma sala.

Para obter a lista de ocupação das salas, é preferível que a função de gerar essa lista, fique com a classe `ReservaGerenciador`, pois ela vai gerenciar as reservas e já vai ter as informações das reservas que são feitas, possibilitando assim selecionar as reservas por dia, semana e mês, com isso fazermos utilização do princípio GRASP: invenção pura,

A classe Reserva vai ficar encarregada da requisição de equipamentos, com isso essa mesma classe reserva pode ficar encarregada de obter os equipamentos necessários para a reserva da sala.

Os dados utilizados para os casos de teste são objetos fictícios instanciados pela classe MockData. Essa classe simula dados provenientes de uma base de dados do sistema. Em uma versão futura do projeto, essa obtenção acontecerá a partir de alguma classe repositório que acessa os dados reais. A classe MockData, entretanto, permanecerá válida para utilização em testes automatizados.