

Padrões utilizados no projeto “Parceiros da Estrada”

1. Baixo acoplamento (Low Coupling)

No sistema, foi debatida e escolhida a opção de utilizarmos Composição no lugar de Herança.

Na procura da minimização das dependências entre classes, foi notório que herança é um tipo de acoplamento particularmente forte. Foram atribuídas responsabilidades específicas a classes que realmente tinham o dever de realizá-las.

2. Pure fabrication

No sistema, nos deparamos com a seguinte pergunta, a quem deveríamos atribuir determinada responsabilidade quando todas as opções ferem os princípios de acoplamento baixo e coesão alta?

Decidimos criar classes artificiais que tivessem alta coesão e baixo acoplamento, e que tais classes ficariam responsáveis por todo o conhecimento, apesar de que são classes altamente funcionais, que não fazem parte da realidade.

Exemplos práticos no sistema Parceiros da Estrada:

- 2.1
Classe GerenciadorDeArquivos
- 2.2
Classe GerenciadorDeSolicitacoes

3. Expert

Visando atribuir responsabilidades a classes que possuem a informação necessária para preencher certas responsabilidades, foi utilizado o padrão Expert.

Exemplos práticos no sistema Parceiros da Estrada (três exemplos):

- 3.1
A classe Sistema possui a função de realizar as seguintes funcionalidades de login e logout do sistema, entre diversas outras responsabilidades.
- 3.2
A classe GerenciadorDeSolicitacoes tem como responsabilidade fazer solicitações para caronas.
- 3.3
A classe Usuário tem como característica (responsabilidade) cadastrar novas caronas.

4. Creator

Padrão utilizado na criação de novas instâncias de uma classe. Utilizado quando precisamos atribuir à uma determinada classe a responsabilidade de criar instância de outra classe.

Exemplos práticos no sistema Parceiros da Estrada:

- 4.1
Classe Usuario: Necessidade da criação da classe PerfilDoUsuario:
A classe PerfilDoUsuario agrega atributos referentes à classe Usuario.
- 4.2
Classes SugestaoDePonto e SolicitacaoDeCarona:
A classe SugestaoDePonto possui dados usados para inicializar a classe SolicitacaoDeCarona.

5. Controller (controlador)

Padrão utilizado para determinar quem deve ser o responsável por lidar com um evento de uma interface de entrada.

Exemplos práticos no sistema Parceiros da Estrada:

- 5.1
Classe SistemaController.