
Trabalho Final-SISOP

Fernando Elger,
Gabriel Bilhar,
Arde Junior,
Prof. Bernardo Copstein
PUCRS-2020/01

Ordem de Integração

Entidade:

- Área
- Bairro
- Roteiro
-
- Passageiro
- Viagem

CasosDeUso:

- Politicas:
- CustoViagem
- CalculoCustoViagemVerao
- CalculoCustoViagemBasico
- CalculoCustoViagemRelampago

Servicos:

ServicosPassageiro

Bugs encontrados

classe Area:

- Método pontoCentral() que só tem uma Classe de Equivalencia.
- Método classifica() quando a reta está toda fora intersectando mais de uma área (tem 4 classes de equivalencia entao de 4 falhas).

classe Roteiro:

- No construtor da classe há uma chamada para o método determinaBairrosPercorridos() que é privado, o método altera uma Collection que inclui todos os bairros que a rota do passageiro passa. No teste, quando acessamos essa Collection seu valor é nulo.

Todos os demais métodos que chamavam o determinaBairrosPercorridos() tiveram seus testes afetados, pois dependiam do retorno dessa Collection. Métodos esses que pertencem as classes CalculoCustoViagemBasico, CalculoCustoViagemVerao e CalculoCustoViagemRelampago.

A solução foi criar um mock de Roteiro que retorna a Collection necessária.

Detalhes significativos drivers

Foram testados as exceções do construtor da classe Area com valor limite, assim como outras exceções para garantir cobertura de linha;

Uma tentativa de criar um mock para a Collection de bairros percorridos já que estava retornando null:

```
Roteiro roteiro = mock(Roteiro.class);  
    Collection<Bairro> bairrosPercorridos = new ArrayList<Bairro>();  
    bairrosPercorridos.add(bairros.get(0));  
    bairrosPercorridos.add(bairros.get(1));  
    bairrosPercorridos.add(bairros.get(2));  
    bairrosPercorridos.add(bairros.get(3));  
  
    when(roteiro.bairrosPercorridos()).thenReturn(bairrosPercorridos);  
  
    ccv.defineRoteiro(roteiro);
```

Resultados obtidos

