

<Hotel Fee Calculation> Master Test Plan

Version <1.0>

Project by: Lucas Rafael Correa Abud /// 647464

Master Test Plan

1. Introdução

1.1 Propósito

O propósito do plano de testes para o software Hotel Fee Calculation é:

- Prover o controle e a organização das classes de teste.
- Prover uma visibilidade da cobertura de teste existente.

1.2 Escopo

Os testes desenvolvidos são de integração e de unidade, utilizando o framework NUnit, do C#.

1.3 Contexto do projeto

O objetivo geral do projeto é retornar o hotel mais barato, dado uma data e um tipo de cliente. O código recebe uma lista de datas e um tipo de cliente, e a partir de valores pré-definidos para cada hotel, retorna-se o hotel mais barato. Em caso de empate, o hotel com maior reputação é escolhido.

2. Itens alvo do teste

Para as classes CustomDate, Fee e Rating foram realizados testes unitários de caixa preta. Para o resultado final, foram realizados testes de caixa preta.

3. Testes planejados

3.1 Inclusão de Testes

Legenda para as tabelas:

- V(n) → casos válidos e a linha da tabela. Ex: V1: valor válido para a primeira linha.
- I(n) → casos inválidos e a linha da tabela. Ex: I1: valor inválido para primeira linha.

3.2 CustomDate

Para a classe de CustomDate:

Variáveis	Válidas	Inválidas
sun sat	valor == true	valor == false
outros valores	valor == false	valor == true

Casos de teste:

- V1 → isWeekend_Sunday_ReturnsTrue()
- V2 → isWeekend_Monday_ReturnsFalse()

3.3 Fee

Variáveis	Válidas	Inválidas
in: reward; ridgewood; thur, fri, sat out: fee	240	260
in: reward; bridgewood; thur, fri, sat out: fee	270	230
in: reward; lakewood; thur, fri, sat out: fee	240	300
in: regular; ridgewood; thur, fri, sat out: fee	590	700
in: regular; bridgewood; thur, fri, sat out: fee	380	100
in: regular; lakewood; thur, fri, sat out: fee	310	350

Casos de teste:

- V1 → Calculate_FinalPriceForClientRewardRidgewood_ReturnsFeePrice()
- I1 → Calculate_FinalPriceForClientRewardRidgewood_ReturnsInvalidFee()
- V2 → Calculate_FinalPriceForClientRewardBridgewood_ReturnsFeePrice()
- I2 → Calculate_FinalPriceForClientRewardBridgewood_ReturnsInvalid()
- V3 → Calculate_FinalPriceForClientRewardLakewood_ReturnsFeePrice()
- I3 → Calculate_FinalPriceForClientRewardLakewood_ReturnsInvalid()
- V4 → Calculate_FinalPriceForClientRegularRidgewood_ReturnsFeePrice()
- I4 → Calculate_FinalPriceForClientRegularRidgewood_ReturnsInvalid()
- V5 → Calculate_FinalPriceForClientRegularBridgewood_ReturnsFeePrice()
- I5 → Calculate_FinalPriceForClientRegularBridgewood_ReturnsInvalid()
- V6 → Calculate_FinalPriceForClientRegularLakewood_ReturnsFeePrice()
- I6 → Calculate_FinalPriceForClientRegularLakewood_ReturnsInvalid()

3.4 Rating

Variáveis	Válidas	Inválidas
hotel rating 1 >= hotel rating 2	hotel 1	hotel 2
hotel rating 1 < hotel rating 2	hotel 2	hotel 1

Casos de teste:

- V1 → Compare_Hotel1BiggerRating_ReturnsHotel1()
- I1 → Compare_Hotel1BiggerRating_ReturnsHotel2()
- V2 → Compare_Hotel2BiggerRating_ReturnsHotel2()
- I2 → Compare_Hotel2BiggerRating_ReturnsHotel1()

4. Test Approach

4.1 Extensão dos testes

Os testes cobrem todas os métodos, individualmente, de cada classe. Percorrem valores válidos e inválidos

5. Entrada e Saída

5.1 CustomDate

Entrada: uma data

Saída: se é um fim de semana ou não

5.2 Fee

Entrada: um hotel; um cliente; datas

Saída: valor da taxa à pagar

5.3 Rating

Entrada: dois hotéis

Saída: qual tem a maior classificação

6. Ambientes

6.1 Configurações de ambiente

Tanto o projeto quanto os testes foram feitos utilizando o Visual Studio 2019.

Para testes, a biblioteca usada foi: NUnit Version 3.8.1, NUnit3TestAdapter Version 3.8