**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**

**INSTITUTO METRÓPOLE DIGITAL**

**BOAS PRÁTICAS DE PROGRAMAÇÃO**

**2024.2**

**Estudo dirigido 04 – Testes e Depuração**

**Unidade 04**

**Thaynan Paulo Fernandes Bezerra de Mendonça**

1. **TESTES E DEPURAÇÃO**

Tanto Testes quando a depuração é utilizada para verificar erros no código.

Os testes verificam a entrada e a saída de valores em um sistema. Podem ser unitário (quando são realizados nas suas unidades) ou de integração (quando se verifica se as unidades do sistema funcionam juntas). Nessa aplicação usamos o JUnit 5 para realizar os testes unitários e Mockito para testes de integração.

Já a depuração é o processo de procura e identificação de erros no sistema. Isso pode ser realizado através da criação de logs (relato de informações colocados no código ou pontos de atenção/erros) ou de breakpoints (marcadores de linhas especificas do código que permite analise das variáveis naquele momento). No código abaixo ocorreu através da criação de logs a partir da biblioteca java.util.logging e da criação de breakpoints através da IDE (netbeans)

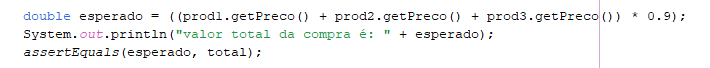
1. **APLICAÇÃO DOS PRINCÍPIOS DA DEPURAÇÃO E TESTES**

Nesse tópico será apresentado os trechos do código que tiverem ajustes em relação ao teste e depuração

2.1 Testes Unitários

Os testes unitários foram realizados através do JUnit 5. Após a inserção da dependência no arquivo pom.xml.

Wxemplo de um codigo com teste unitário é o de testCalcularTotal() que o usa o assertEquals(esperado, total) para verificar se o valor está certo (para o caso acima de 100 que tem desconto)



2.2 Testes Integração

O teste de integração foi realizado através do Mockito. Acabei conseguindo colcoar a dependecia do Mock, e a inclusão no código. Mas o discente teve dificuldade de como aplicar no trabalho

2.3 Logs

A criação de logs ocorreu através da biblioteca import java.util.logging.Logger apresentado na saída (output) do Netbeans.

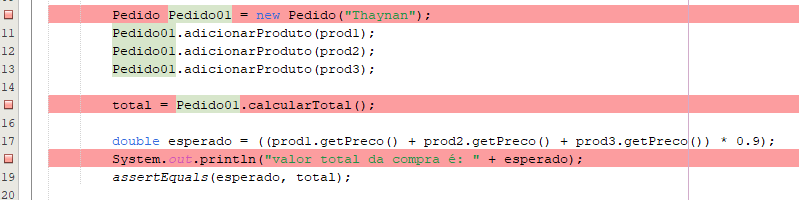
Após a inicialização da biblioteca mensagens de confirmação de algumas ações como adicionar novo item ao catalogo ou desconto realizado foram lançados para acompanhamento.

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

2.4 Breakpoints

O uso de breakpoints ocorreu em momentos importantes do código como por exemplo no momento da aplicação do desconto. No código abaixo os breakpoints foram na classe teste do pedido, mas há em alguns outros pontos do código.



1. **Referências**

<https://medium.com/rpedroni/tudo-sobre-testes-testes-unit%C3%A1rios-vs-testes-de-integra%C3%A7%C3%A3o-vs-testes-e2e-6a7cc955779> acessado em 12-12-2024 às 04:10

<https://www.techtarget.com/searchsoftwarequality/definition/debugging> acessado às 12-12-2024 as 05:04