Relatório Técnico – Teste Netshoes para Analista de Desenvolvimento

### Jonas De Cunto Augusto

### Atividades Realizadas para resolução dos Exercícios

# GitHub: https://github.com/pokenshin/TesteImposto

# Bugs Não Documentados Corrigidos

* **Bug Não documentado 1:**
  + **Sintomas:** quando um item era adicionado ao pedido sem clicar no checkbox "Brinde", o DataColumn resultante deste DataRow ficava vazio, o que causava um erro na hora de converte-lo para boolean e adicioná-lo na propriedade "ItensDoPedido" do objeto "pedido".
  + **Correção:** O erro foi corrigido adicionando uma verificação que atribui o valor de "false" caso a DataColumn "Brinde" estiver nulo no DataRow.
* **Bub Não documentado 2:**
  + **Sintomas:** os itens do pedido não estão sendo adicionados no objeto da nota fiscal.
  + **Correção:** Corrigido ao chamar o método "ItensDaNotaFiscal.Add()" como última instrução do loop "foreach (PedidoItem itemPedido in pedido.ItensDoPedido)" dentro do método EmitirNotaFiscal da classe NotaFiscal.
* **BUG não documentado 3:**
  + **Sintomas:** originalmente no código, na função EmitirNotaFiscal da classe NotaFiscal, há duas verificações iguais que apontam EstadoOrigem == "SP" e EstadoDestino == "SE" mas atribuem valores de CFOP diferentes, um com "6.007" e outro com "6.009".
  + **Correção:** A verificação que atribui o valor "6.007" foi removido pois o valor “6.009” é utilizado em outra função do sistema.

# Exercício 1 (Geração de XML)

* Adicionado [Serializable] na classe NotaFiscal.
* Adicionado [Serializable] na classe NotaFiscalItem.
* Alteração no tipo da propriedade ItensDaNotaFiscal de IEnumerable<NotaFiscalItem> para List<NotaFiscalItem> a fim de simplificar a serialização.
* Criado método criarXML para geração de XML no serviço NotaFiscalService.

# Exercício 2 (Criação de Banco de Dados)

* Criado banco de dados Teste e populados utilizando stored procedures fornecidas.
* Importadas Stored Procedure "P\_NOTA\_FISCAL" e "P\_NOTA\_FISCAL\_ITEM"
* Criada propriedade connectionString dentro do NotaFiscalRepository
* Criado método SalvarNotaFiscal em NotaFiscalRepository
* Criado método AtualizaNotaFiscal em NotaFiscalRepository

# Exercício 3 (Campos para cálculo de IPI)

* Criados campos BaseIpi, AliquotaIpi e ValorIpi na classe NotaFiscalItem.
* Adicionado cálculo da Aliquota do Ipi em caso de Brinde no método EmitirNotaFiscal da classe NotaFiscal.
* Adicionado cálculo da Base do IPI do produto euqlizando-a ao valor do Item no método EmitirNotaFiscal da classe NotaFiscal.
* Adicionado cálculo do Valor do IPI do produto multiplicando a Base do IPI à Aliquota do IPI no método EmitirNotaFiscal da classe NotaFiscal.
* Alteração da tabela NotaFiscalItem (scriptados no arquivo CRIAR\_NOVOS\_CAMPOS\_NOTAFISCALITEM.sql)
* Novo campo: BaseIpi decimal(18,5)
* Novo campo: AliquotaIpi decimal(18,5)
* Novo campo: ValorIpi decimal (18,5)
* Alteração da Stored Procedure P\_NOTA\_FISCAL\_ITEM para adicionar os novos campos criados na tabela. Script para reproduzir alterações está no arquivo ALTER\_P\_NOTA\_FISCAL\_ITEM.
* Alteração no método SalvarNotaFiscal da classe NotaFiscalRepository para refletir as alterações da Stored Procedure P\_NOTA\_FISCAL\_ITEM

# Exercício 4 (Criação de Stored Procedure ICMS/Ipi)

* Criada nova Stored Procedure chamada P\_RELATORIO\_ICMS\_IPI. Script para criação da Stored Procedure está disponível no arquivo P\_RELATORIO\_ICMS\_IPI.

# Exercício 5 (Bug no CFOP)

* Corrigido bug que impedia a geração do CFOP para os itens inseridos no pedido. O bug era causado pela inversão dos campos EstadoOrigem e EstadoDestino no método EmitirNotaFiscal da classe NotaFiscal.

# Exercício 6 (Melhorias)

* Alteração dos campos Estado Origem e Estado Destino para Combobox:
  + Remoção da lista de estados da classe "Estado".
  + Criação de campos cbbOrigem e cbbDestino do tipo Combobox em FormImposto
  + Diminuição do tamanho dos campos para refletir seu conteúdo.
  + Criação de variável do tipo List<Estado> para popular cbbOrigem e cbbDestino
  + Criação do método geraEstados no código da janela FormImposto.
  + Criação de método populaCombosEstados no código da janela FormImposto
* Alteração da propriedade MaxLength do campo NomeCliente para 50, alinhando com o limite do banco de dados
* Criação do método formataDataGridView mesclando a funcionalidade do antigo ResizeColumns com funcionalidades para limitar os erros ao preencher a gridView.
* Diminuição do campo "Brinde" do DataGrid para evitar barras de rolagem.
* Os campos agora são limpos quando uma nota fiscal nova é gerada.

# Exercício 7 (Desconto na Nota Fiscal)

* Criação de nova propriedade na classe NotaFiscalItem chamada "Desconto"
* Alteração da tabela NotaFiscalItem (concatenado ao script CRIAR\_NOVOS\_CAMPOS\_NOTAFISCALITEM)
* Novo campo: Desconto decimal (18,5)
* Alteração da Stored Procedure P\_NOTA\_FISCAL\_ITEM para adicionar o novo campo criado na tabela. Script ALTER\_P\_NOTA\_FISCAL\_ITEM atualizado com o novo campo.
* Alteração do método SalvarNotaFiscal da classe NotaFiscalRepository para refletir as alterações da Stored Procedure P\_NOTA\_FISCAL\_ITEM

# Exercício 8 (Refatoração)

* Criação de nova classe "Estado" com todas as siglas válidas para os estados, seus respectivos nomes e regiões.
* Alteração de Propriedades EstadoDestino e EstadoOrigem da classe NotaFiscal para serem do tipo "Estado" ao invés de "string".
* Alteração de Propriedades EstadoDestino e EstadoOrigem da classe Pedido para serem do tipo "Estado" ao invés de "string"
* Criação de nova classe "Calculos" para agregar todos os calculos do sistema.
* Alteração do foreach dentro do método EmitirNotaFiscal dentro da classe NotaFiscal para utilizar os cálculos da nova classe "Calculos".
* Refatoração da lógica de cálculo do Cfop de ifs encadeados para Switch na classe "Calculos"
* Refatoração da função ResizeColumns da classe FormImposto para facilitar seu entendimento

# Exercício 9 (Testes Unitários)

* Criação de novo projeto "Imposto.Test" para abrigar os testes unitários
* Criação de classe de testes unitários chamada CalculosTeste para a classe Calculos:
  + Criação de testes para o método CalculaTipoIcms
  + Criação de testes para o método CalculaAliquotaIcms
  + Criação de testes para o método CalculaValorIcms
  + Criação de testes para o método CalculaValorIpi
  + Criação de testes para o método CalculaAliquotaIpi
  + Criação de testes para o método CalculaDesconto