Netshoes

Teste para analista desenvolvedor

|  |  |
| --- | --- |
|  | Foi construído um projeto fictício com algumas regras de negócio já desenvolvidas, o teste consiste basicamente em entender o projeto e evoluir conforme as orientações citadas no decorrer deste documento.  O objetivo do teste é simular as atividades do dia-a-dia afim de avaliar as seguintes áreas:   * Desenvolvimento de novos recursos * Correções de BUG’s. * Melhorias técnicas * Analise das atividades |

Visão geral

O projeto consiste em uma tela cujo objetivo é digitar um pedido, e baseado nos dados da tela fazer a persistência dos dados.

|  |  |
| --- | --- |
|  | Dica: A base de código é bem reduzida, entenda a estrutura do projeto antes de iniciar a solução dos problemas. |

1 – Exercício (Novo recurso) – OK

Está em desenvolvimento uma tela para emissão de notas fiscais, o cálculo dos impostos já está desenvolvido, porém é necessário fazer a persistência da nota fiscal em um arquivo XML, seguindo exatamente a mesma estrutura da classe: NotaFiscal.

O diretório onde serão gerados os XML’s é configurado pelo time de infra estrutura, o usuário não pode altera-lo (o usuário tem acesso apenas ao executável).

R: Foi criado um item de configuração para o caminho de geração do XML, mas esse problema também poderia ser resolvido configurando o caminho através da base de dados

|  |  |
| --- | --- |
|  | Dica: Pode ser utilizado serialização de objetos para XML nesse caso. |

2 – Exercício (Novo recurso) - OK

|  |  |
| --- | --- |
|  | Será necessário criar um banco de dados localmente utilizando o SQL Server Express. |

Na pasta SQL foi disponibilizado um conjunto de scripts, para que seja possível criar e popular o banco de dados.

Utilizar as procedures P\_NOTA\_FISCAL e P\_NOTA\_FISCAL\_ITEM para fazer a persistência dos dados no banco.  
A persistência só pode acontecer se o XML foi gerado com sucesso.

3 – Exercício (Novo recurso) – OK

Adicionar na classe de itens da nota fiscal os campos relacionados ao imposto de IPI:

Base de cálculo de IPI: Igual ao valor total do produto.  
Alíquota de IPI: Se for brinde alíquota é igual a 0% se não brinde alíquota é igual a 10%.   
Valor de IPI: Base de cálculo \* Alíquota de IPI.

Esses campos serão persistidos na tabela de Itens de nota fiscal e também no arquivo XML.

4 – Exercício (Novo recurso) – OK

Desenvolver uma stored procedure para ser utilizada por outros sistemas, o objetivo dessa procedure é devolver a seguinte estrutura:

CFOP | Valor Total da Base de ICMS | Valor Total do ICMS | Valor Total da Base de IPI | Valor Total do IPI

Deve estar agrupado por CFOP.

5 – Exercício (Correção de BUG) - ANOTACAO

Para o estado de origem SP e destino RO o sistema deveria definir a CFOP como 6.006, corrigir o erro no sistema para que seja definido a CFOP correta.

R: O estado com problema é o SE, pois está duplicado. Mantive a opção com o valor 6.009, pois faria mais sentido no contexto do código, onde há uma validação para esse valor. O item com estado de destino para RO já está com o valor 6.006.

6 – Exercício (Correção de BUG) - OK

A tela tem alguns pontos de melhoria a serem implementados, por exemplo:

Ao salvar a tela não está limpando os campos, - OK

é possível indicar estados inválidos tanto na origem quanto no destino. -OK

Levantar os pontos de melhoria e executar as correções.

7 – Exercício (Novo recurso) - OK

Criar um campo desconto no item da nota fiscal.  
Preenchê-lo com a seguinte regra: Clientes do sudeste (campo “EstadoDestino”) tem 10% de desconto.   
Esse campo deve ser incluso no XML e no banco de dados.

|  |  |
| --- | --- |
|  | Dica: É sempre bem-vindo a criação de novas classes com apenas uma responsabilidade, utilizando os conceitos SOLID. |

8 – Exercício (Melhoria técnica) - OK

O código tem uma complexidade ciclomática muito alta, refatorar o código para que seja possível entende-lo melhor e simplificar a manutenção.

|  |  |
| --- | --- |
|  | Dica: Não tem nenhum tipo de restrição com relação a reestruturação, sinta-se à vontade para criar mais classes, projetos, métodos e etc. |

9 – Exercício (Melhoria técnica) -OK

Estão previstas muitas alterações nas regras desta aplicação, crie testes unitários para diminuir o risco e melhorar o design.

10 – Exercício (Analise)

É necessário gerar um report técnico das atividades desenvolvidas, descreva com suas palavras qual é o objetivo da tela e como foi implementado as alterações solicitadas.

R: Desenvolvi a arquitetura do sistema baseada na separação de conceitos, ou seja, cada projeto deve possuir a sua responsabilidade.

Foram criadas as camadas Business, DAL, Domain e Helpers.

A camada business é responsável pelas regras de negócios do sistema.

A camada DAL é responsável pelas operações voltadas a persistência de dados

A camada domain é responsável pelas Models utilizadas pelo sistema

A camada Helpers possui os utilitários necessários para uso no sistema e que não se enquadram em nenhuma das camadas acima citadas.

Implementei a classe Constantes para centralizar os strings utilizadas no sistema e evitar o uso de Hard-Coded, facilitando assim o entendimento e a manutenção do código. Além de agrupar cada string utilizada em seu contexto.

A geração do XML é feita a partir de Serializador, facilitando assim a transformação de uma Model já preparada em um XML.

O objetivo da tela é a geração de uma nota fiscal eletrônica, configurada devidamente para cada destino de acordo com as suas definições, assim como valores de aliquota e bases de calculo para IR baseando-se numa compra ou em entrega de brindes aos clientes

Erros Encontrados na Codificação:

* Quando o checkbox “Brinde” não é selecionado, o código original estoura exception
* Preenchimentos invertidos
  + notaFiscal.EstadoDestino = pedido.EstadoOrigem;
  + notaFiscal.EstadoOrigem = pedido. EstadoDestino;
* Validação de Estado de Destino para definição de CFOP com o estado “SE” duplicado
* A procedure P\_NOTA\_FISCAL\_ITEM deve ter o parametro pID como output
* A regra de negócio de consistir os dados somente depois da geração do XML gera o problema de não haver os IDs no XML tanto para a NotaFiscal quanto para os Itens, pois esses só são gerados após a consistência dos dados no base.