Netshoes

Teste para analista desenvolvedor

1 - OK. Gerado arquivo XML, deve configurar o caminho no App.Config

2 - OK. Estou trabalhando em um Mac sem VM, sem SQL Server para rodar :/ mas efetuei tudo que foi solicitado. Difícil debugar também pois Mac não roda WindowsForms tradicional.

3 - OK. Criados campos na tabela e propriedades na classe.

4 - OK. Crianda proc P\_GET\_ALL\_VALUES\_BY\_CFOP (SQL/P\_GET\_ALL\_VALUES\_BY\_CFOP.sql)

5 - OK. No código que baixei já estava com valor informado no doc de instruções  
  
 //snippet:  
 else if ((this.EstadoOrigem == "SP") && (this.EstadoDestino == "RO"))  
 {  
 notaFiscalItem.Cfop = "6.006";  
 }

6 - Melhorias que eu sugiro por hora:

- Os IDs dos campos não estão padronizados, alguns com prefixo txt outros com prefixo text

- Cidade poderia ter intellisense

- Estado poderia ser DropDownList/ComboBox :p

- Estado poderia ser um Enum

- Validar se as informações inseridas são númericas e dar feedback ao usuário

- Ícone Netshoes

- Web Api para fornecer relatório CFOP e outros

7 - OK. Aplicada verificação de desconto para Sudeste.

8 - NoGo. Atualmente estou trabalhando em outra empresa, estou trabalhando em um projeto grande de BPMS para um grande banco, com prazo apertado, estou fazendo horas extras todos os dias, tinha apenas 3 horas da sexta-feira para efetuar este projeto teste NetShoes.  
Não deu tempo de adicionar projeto UnitTest na solution, mas conheço tudo isso.   
Eu não estava preparado para este teste, estou com um Mac sem VM e sem SQL Server, não roda Windows Forms tradicional, foi difícil adaptar em tempo hábil. Mas gostei de fazer e entreguei o solicitado. Nunca tinha escrito Windows Forms em um Mac :)

9 - NoGo. Idem ao item 8.

10 - Este app é uma interface para cálculo de impostos baseados nos estados de origem, destino e checagem de brinde. São aplicados os impostos ICMS e IPI, desconto e persistência das informações em banco de dados e arquivos XML  
  
Melhorias implementadas:

- Validação de estado de origem e destino dando feedback ao usuário

- Aplicação de IPI

- Tabela alterada para armazenar Base IPI, alíquota e valor

- Stored Procedure alterada para suportar IPI

- Classes alteradas para suportar IPI

- Aplicação de Desconto

- Tabela alterada para suportar Desconto

- Stored Procedure modificada para suportar Desconto

- Classes modificadas para suportar Desconto

- Gravação de registros em arquivo XML. Com possibilidade de configurar o diretório via App.Config

- Gravação de registros em banco de dados SQL Server. Com possibilidade de configurar ConnectionString via App.Config

- Criada Stored Procedure para retornar relatório de CFOP