

## Report Técnico

**Analista: Eduardo Kessler Neto**

O sistema possui o objetivo de permitir a digitação de um pedido de venda, gerando a nota fiscal com seus devidos cálculos de impostos, armazenando as informações em um arquivo XML e no banco de dados do sistema.

Nome do produto	Código do produto	Valor	Brinde
Tênis Nike	2752	R\$ 199,90	<input type="checkbox"/>
Sapato Social	1574	R\$ 131,25	<input checked="" type="checkbox"/>

Foram desenvolvidos novos recursos, melhorias e correções conforme relacionado abaixo.

### Novos Recursos

- **Geração de XML**

Ao gerar a nota fiscal, o sistema automaticamente irá gerar o arquivo XML em um diretório previamente parametrizado pela equipe de infra estrutura.

O novo parâmetro consta no arquivo App.config


```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
  <connectionStrings>
    <add name="ConexaoBD" providerName="System.Data.SqlClient" connectionString="Server=localhost;Database=Teste;Trusted_Connection=True;" />
  </connectionStrings>
  <appSettings>
    <add key="DiretorioXML" value="e:\\" />
  </appSettings>
</configuration>
```

- **Persistência na base de dados**

Apenas quando o XML for gerado com sucesso, o sistema irá armazenar as informações da nota fiscal na base de dados do sistema.

Um novo parâmetro foi incluído no arquivo App.config para informar a string de conexão com o banco de dados.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
  <connectionStrings>
    <add name="ConexaoBD" providerName="System.Data.SqlClient" connectionString="Server=localhost;Database=Teste;Trusted_Connection=True;" />
  </connectionStrings>
  <appSettings>
    <add key="DiretorioXML" value="e:\\" />
  </appSettings>
</configuration>
```



- **Imposto IPI**

Foram incluídos os campos: BaseIpi, AliquotaIpi e ValorIpi no item da nota fiscal, esses campos serão automaticamente calculados ao emitir a nota fiscal.

```
//Cálculo do Ipi
notaFiscalItem.BaseIpi = itemPedido.ValorItemPedido;
notaFiscalItem.AliquotaIpi = (itemPedido.Brinde ? 0 : 0.1);
notaFiscalItem.ValorIpi = notaFiscalItem.BaseIpi * notaFiscalItem.AliquotaIpi;
```

- **Procedure para integração com outros sistemas**

Criado um novo procedimento na base de dados do sistema para retornar o valor total dos impostos agrupados por CFOP.

```
CREATE PROCEDURE P_IMPOSTO_CFOP AS
BEGIN
  SELECT
    i.Cfop,
    SUM(i.BaseIcms) As BaseIcms,
    SUM(i.ValorIcms) As ValorIcms,
    SUM(i.BaseIpi) As BaseIpi,
    SUM(i.ValorIpi) As ValorIpi
  FROM
    NotaFiscalItem i
  GROUP BY
    i.Cfop
END
```

- **Valor de desconto**

Incluído novo campo no item da nota fiscal para calcular o percentual de desconto, clientes da região sudeste possuem 10% de desconto.


```
//Cálculo do desconto
notaFiscalItem.Desconto = new DescontoService().CalcularDesconto(notaFiscal.EstadoDestino);
```

## Correção de problemas

- **CFOP 6.006**

Quando digitado o estado de origem "SP" e destino "RO", o sistema estava carregando incorretamente a CFOP. Foi identificada uma inversão no código fonte, o problema foi corrigido.

```
this.EstadoDestino = pedido.EstadoOrigem;
this.EstadoOrigem = pedido.EstadoDestino;
```



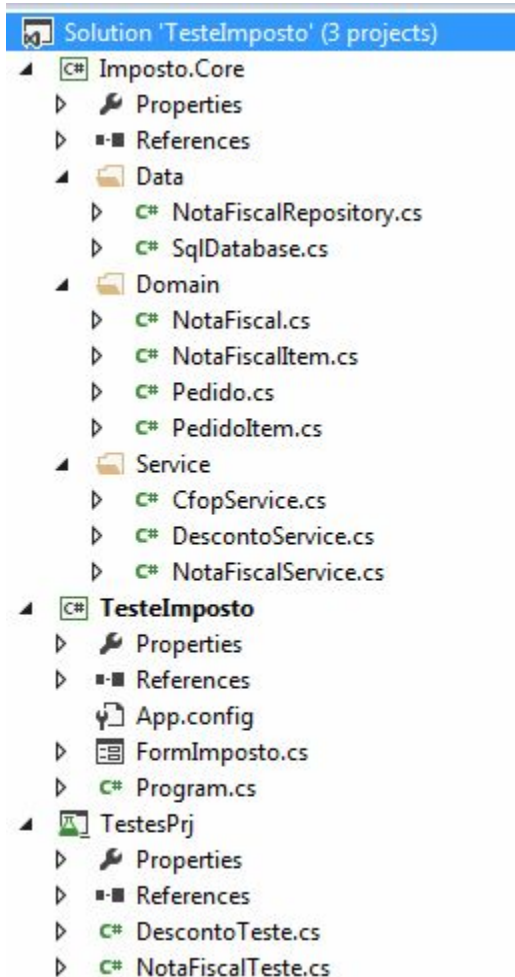
- **Tela de digitação**

Foram identificados diversos problemas de usabilidade na tela de digitação de pedidos, todos os pontos foram identificados e corrigidos (novas melhorias também foram implementadas).

## **Melhorias Técnicas**

- **Complexidade do código**

O código fonte das classes e toda sua estrutura foram refatorados afim de facilitar e agilizar o atendimento de demandas pela equipe de sustentação.



- **Testes Unitários**

Para manter a qualidade e confiabilidade da aplicação, foram criadas unidades de testes.

