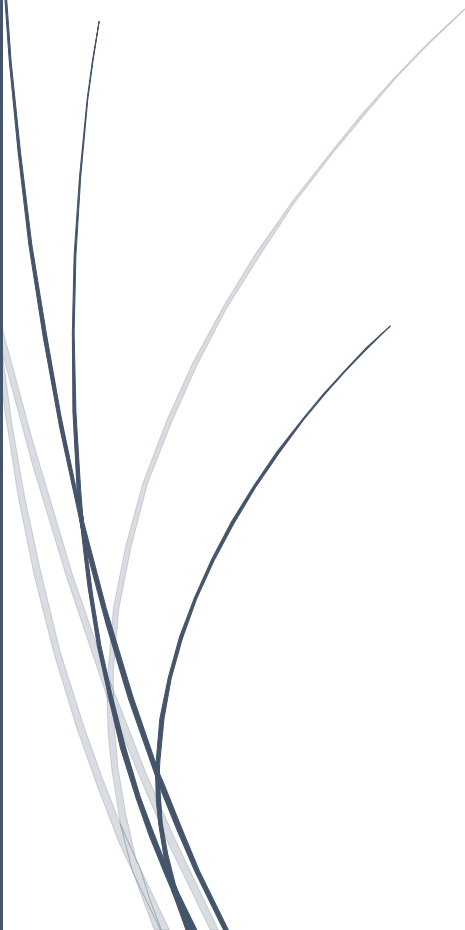


21/08/2019

# DBVenda | NF-e | NFC-e | SAT

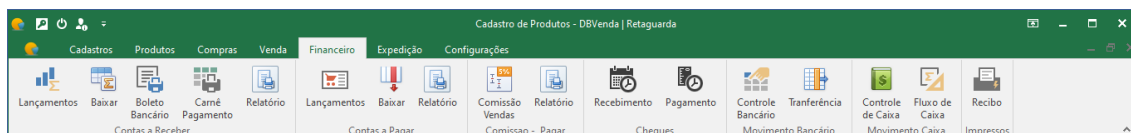
---

Apresentação do Código-Fonte



Versão 1.46 - agosto/2019

**Adriano Zanini**  
ANALISTA DE SISTEMAS



## Sumário

Autoria do código-fonte .....	3
Apresentação do código-fonte.....	4
Estrutura e organização .....	4
Exemplo de código-fonte .....	5
Nota Fiscal .....	5
Frente de Caixa (NFC-e / SAT / PDV).....	5
Uso da Classe de Parâmetros .....	6
Transações SQL no Frente de Caixa (NFC-e/SAT).....	6
Módulos/funcionalidades do sistema .....	7
Componentes necessários .....	8
Lista de componentes .....	8

## Autoria do código-fonte

Eu Adriano Zanini, sou o autor/desenvolvedor. Você poderá modificar o que desejar, mudar aparência, dar outro nome, etc. O código-fonte fica somente para você (ou sua empresa). Não autorizo vender meu código-fonte. Somente poderá vender/alugar/fornecer aos seus clientes o sistema compilado por você mesmo.

### **Requisitos Necessários**

- **Delphi 10.3 Rio (ou superior) - 32 bits**
- **Firebird 2.5 – 32 bits**

### **Não tente compilar em versão inferior ao Delphi 10.3 Rio**

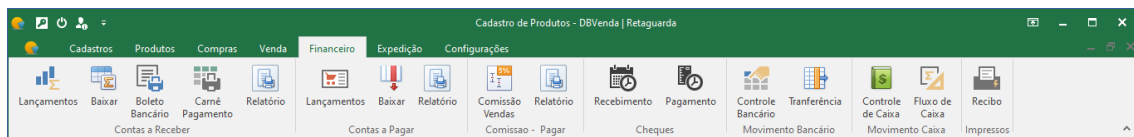
Terá êxito somente no Delphi Rio ou superior (próximas versões).

### **Caso não disponha de uma versão paga do Delphi**

Recomendo utilizar versão “*Delphi Community Edition*”. Particularmente eu utilizo e me atende muito bem. Você pode utilizar gratuitamente por 12 meses. Podendo renovar por mais 12 meses sucessivamente.

## Apresentação do código-fonte

### Estrutura e organização

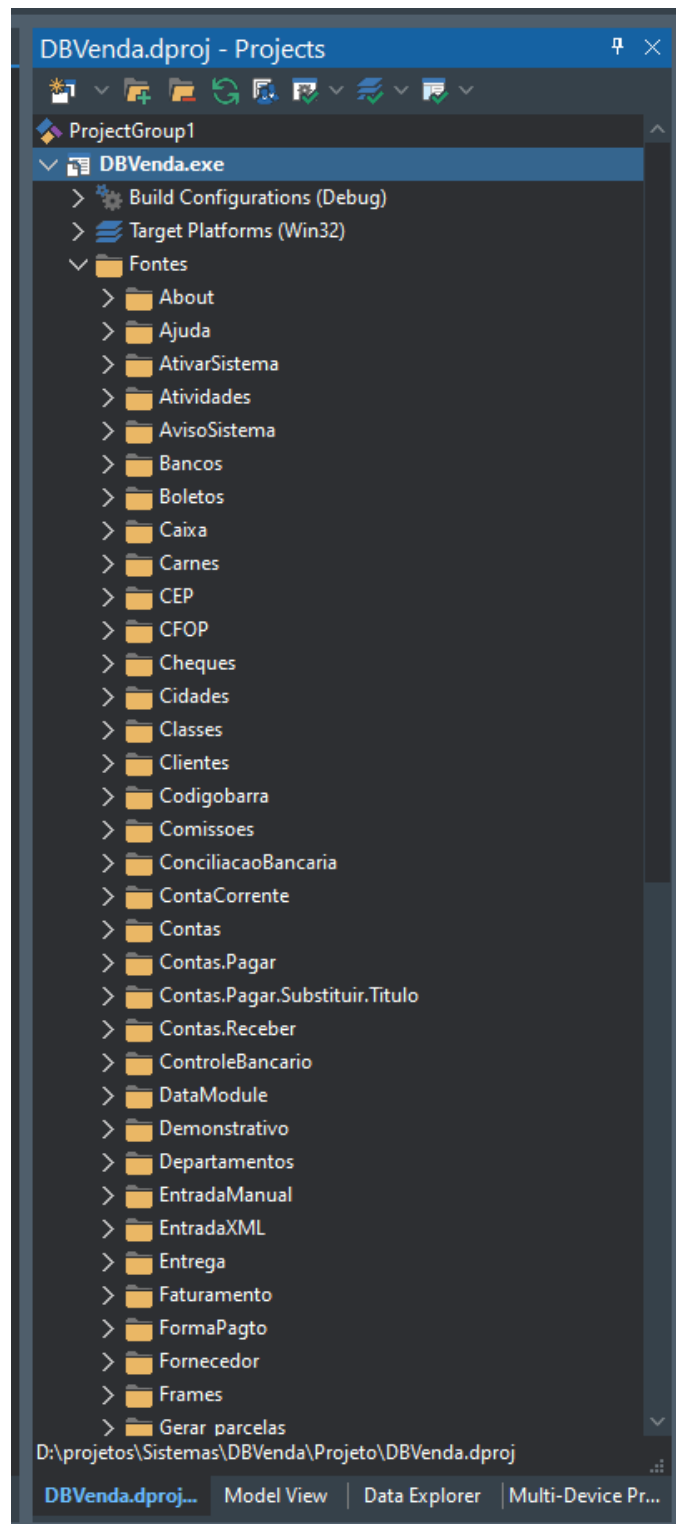


O código-fonte foi desenvolvido em Delphi (v10.3 Rio). Utiliza banco de dados Firebird 2.5.

É todo padronizado e organizado. Metodologia de programação em duas formas:

- Orientação a objetos (classes, heranças, etc).
- Procedural (funções e procedures).

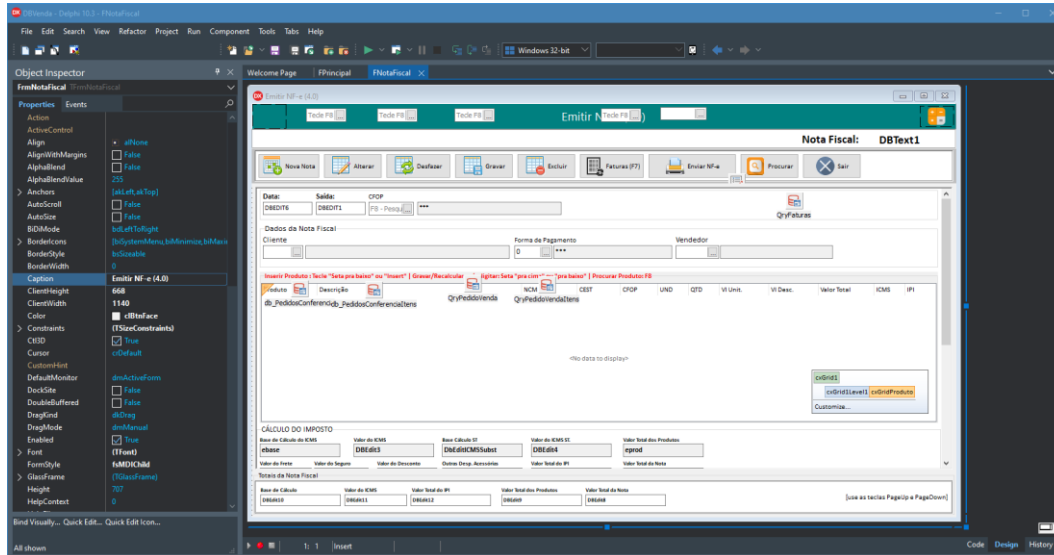
Todos arquivos do código-fonte são organizados e suas respectivas pastas. O que facilita encontrar o que desejar indo na pasta específica.



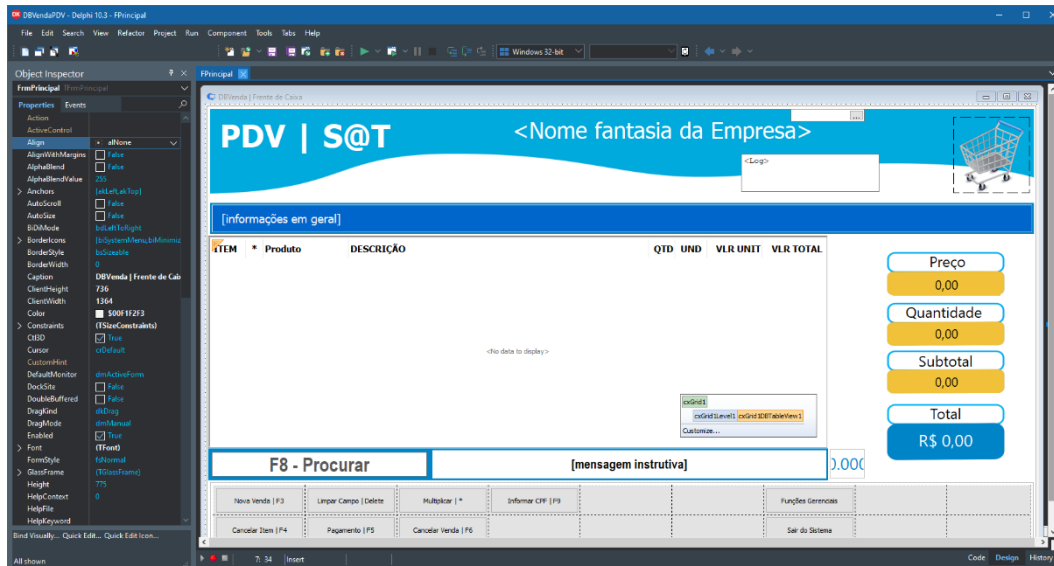
## Exemplo de código-fonte

Veja “alguns” prints abaixo.

### Nota Fiscal



### Frete de Caixa (NFC-e / SAT / PDV)



## Uso da Classe de Parâmetros

```

Result := UpperCase ( FTBLOutrasFuncoes.FieldByName (slCampo) .AsString );
QueryObjLiberar (FTBLOutrasFuncoes);
end;

Procedure GetParametros;
begin
3278 | if FParametros=nil then
3280 |   FParametros := TParametros.Create;

QueryObjCriar (FTBLOutrasFuncoes);
Try
FTBLOutrasFuncoes.SQL.Clear;
FTBLOutrasFuncoes.SQL.Add (' SELECT * FROM CONFIG_PARAMETROS ');
FTBLOutrasFuncoes.SQL.Add (' WHERE ');
FTBLOutrasFuncoes.SQL.Add (' EMPRESA=EMPRESA ');
FTBLOutrasFuncoes.ParamByName ('EMPRESA') .AsInteger := FSistema.Empresa;
FTBLOutrasFuncoes.Open;

3290 |
3300 |
FParametros.Empresa.Codigo := FTBLOutrasFuncoes.FieldByName ('EMPRESA') .AsInteger;
FParametros.Empresa.DataAtividade := FTBLOutrasFuncoes.FieldByName ('INF_DATABASE') .AsDateTime;
FParametros.Empresa.Nome := FTBLOutrasFuncoes.FieldByName ('NOME') .AsString;
FParametros.Empresa.NomeFantasia := FTBLOutrasFuncoes.FieldByName ('NOMEFANTASIA') .AsString;
FParametros.Empresa.CNPJ := FTBLOutrasFuncoes.FieldByName ('CNPJ') .AsString;
FParametros.Empresa.IE := FTBLOutrasFuncoes.FieldByName ('IE') .AsString;
FParametros.Empresa.FMunicipal := FTBLOutrasFuncoes.FieldByName ('IMUNICIPAL') .AsString;
FParametros.Empresa.Endereco := FTBLOutrasFuncoes.FieldByName ('ENDereco') .AsString;
FParametros.Empresa.Numero := FTBLOutrasFuncoes.FieldByName ('NUMERO') .AsString;
FParametros.Empresa.Bairro := FTBLOutrasFuncoes.FieldByName ('BAIRRO') .AsString;
FParametros.Empresa.Complemento := FTBLOutrasFuncoes.FieldByName ('COMPLEMENTO') .AsString;
FParametros.Empresa.Cep := FTBLOutrasFuncoes.FieldByName ('CEP') .AsString;
FParametros.Empresa.Fone := FTBLOutrasFuncoes.FieldByName ('FONE') .AsString;
FParametros.Empresa.Fax := FTBLOutrasFuncoes.FieldByName ('FAX') .AsString;
FParametros.Empresa.Email := FTBLOutrasFuncoes.FieldByName ('EMAIL') .AsString;
FParametros.Empresa.Site := FTBLOutrasFuncoes.FieldByName ('SITE') .AsString;
FParametros.Empresa.Contato := FTBLOutrasFuncoes.FieldByName ('CONTATO') .AsString;
FParametros.Empresa.IDCIDADE := FTBLOutrasFuncoes.FieldByName ('IDCIDADE') .AsInteger;
    
```

## Transações SQL no Frente de Caixa (NFC-e/SAT)

```

442 | //-----
443 | // Impostos: ICMS/IPI/PIS/COFINS
444 | //-----
445 | if aTblProduto.FieldByName ('ICMS') .AsFloat > 0 then
446 |   begin
447 |     mQuery.ParamByName ('ICMS') .AsFloat := aTblProduto.FieldByName ('ICMS') .AsFloat;
448 |     mQuery.ParamByName ('VLR_ICMS') .AsFloat := RoundABNT ((mValorTotal * aTblProduto.FieldByName ('ICMS') .AsFloat) / 100, 2);
449 |   end;
450 | if aTblProduto.FieldByName ('IPI') .AsFloat > 0 then
451 |   begin
452 |     mQuery.ParamByName ('IPI') .AsFloat := aTblProduto.FieldByName ('IPI') .AsFloat;
453 |     mQuery.ParamByName ('VLR_IPI') .AsFloat := RoundABNT ((mValorTotal * aTblProduto.FieldByName ('IPI') .AsFloat) / 100, 2);
454 |   end;
455 | if aTblProduto.FieldByName ('PIS') .AsFloat > 0 then
456 |   begin
457 |     mQuery.ParamByName ('PIS') .AsFloat := aTblProduto.FieldByName ('PIS') .AsFloat;
458 |     mQuery.ParamByName ('VLR_PIS') .AsFloat := RoundABNT ((mValorTotal * aTblProduto.FieldByName ('PIS') .AsFloat) / 100, 2);
459 |   end;
460 | if aTblProduto.FieldByName ('COFINS') .AsFloat > 0 then
461 |   begin
462 |     mQuery.ParamByName ('COFINS') .AsFloat := aTblProduto.FieldByName ('COFINS') .AsFloat;
463 |     mQuery.ParamByName ('VLR_COFINS') .AsFloat := RoundABNT ((mValorTotal * aTblProduto.FieldByName ('COFINS') .AsFloat) / 100, 2);
464 |   end;
465 | //-----
466 | // Impostos: IBPT
467 | //-----
468 | mQuery.ParamByName ('IN_ALIQ_MUNICIPAL') .AsFloat := mIBPT.AliquotaMunicipal;
469 | mQuery.ParamByName ('IN_VL_MUNICIPAL') .AsFloat := RoundABNT ((mValorTotal * mIBPT.AliquotaMunicipal) / 100, 2);
470 | mQuery.ParamByName ('IN_ALIQ_ESTADUAL') .AsFloat := mIBPT.AliquotaEstadual;
471 | mQuery.ParamByName ('IN_VL_ESTADUAL') .AsFloat := RoundABNT ((mValorTotal * mIBPT.AliquotaEstadual) / 100, 2);
472 | mQuery.ParamByName ('IN_ALIQ_FEDERAL') .AsFloat := mIBPT.AliquotaFederal;
473 | mQuery.ParamByName ('IN_VL_FEDERAL') .AsFloat := RoundABNT ((mValorTotal * mIBPT.AliquotaFederal) / 100, 2);
    
```

Com esses prints, você tem uma noção da forma como é a organização do código-fonte.

## Módulos/funcionalidades do sistema

### [Frente de Caixa]

- NFC-e / SAT / PDV (não fiscal)

### [VENDA/FATURAMENTO]

- NF-e 4.0
- Pedido/Aprovação de Venda.
- Etiqueta de Embalagem.
- Relatórios.

### [FINANCEIRO]

- Boleto bancário (diversos bancos) e carnê de pagamento
- Gera arquivo de remessa (.rem)
- Contas a Receber e a Pagar
- Controle de Comissões
- Relatórios em geral do financeiro

### [PRODUTOS]

- Movimento Estoque
- Grade de Cores e Tamanhos (customizáveis)
- Grupos/Subgrupos
- Tabela de Preços
- Gera Código de Barras
- Relatórios

### [CADASTROS]

- Clientes
- Representante/Vendedor
- Fornecedor
- Departamentos (Centro de Custos)
- Formas de Pagamento
- Plano de Contas
- Regiões (Área de Atuação)
- Transportadoras
- CFOP
- Relatórios

### [COMPRAS]

- Nota Fiscal de Entrada (Importa XML)
- Pedidos de Compra
- Relatórios

### [Outras funcionalidades]

- Configurações Gerais;
- Multe empresa
- Multiusuário

## Componentes necessários

### Lista de componentes

Eu forneço todos os componentes necessários (acompanham os código-fonte).

- ACBr - Conjunto de componentes operacionais e fiscais.
- DevExpress - Conjunto de componentes VCL
- FastReport - Conjunto de componentes para relatórios
- QuickReport - Conjunto de componentes para relatórios  
(Nas primeiras versões desse sistema era tudo em QuickReport. Não migrei tudo pra FastReport)
- Fortes Report - Conjunto de componentes para relatórios  
(Uso unicamente para emitir DANFE)
- EditDialog - Componente DBEdit com pesquisa em tempo real.

Eu forneço (sem custo) todos os componentes necessários (acompanham os código-fonte).  
E manual para auxiliar na instalação dos componentes.