



Padrões de Projeto

Aplicados a rotinas do dia a dia
como NF-e

Thulio Bittencourt
#Radizeiro

Da sala de reunião à vida real

Embarcadero Conference 2019



Super Bônus da Palestra

“Como está o mundo lá fora”

Cheio de Gambiarras



Blå Band

Suklaamousse

Chokladmousse

2 x 3 dl

0

РД
РОЛ



X5

X 126 HX 190.

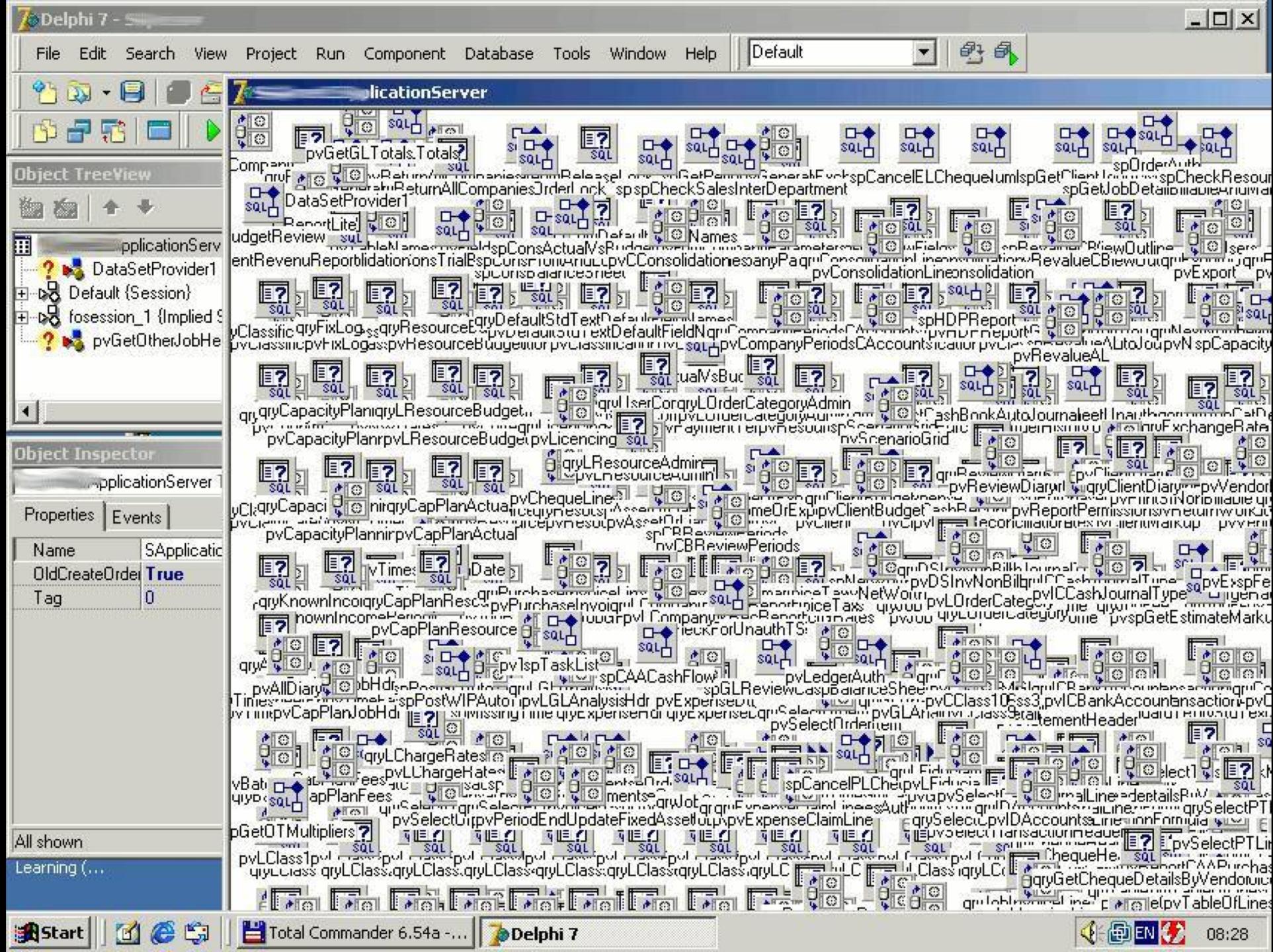
RUS

МОСКОВСКАЯ АВТОМОБИЛЬНАЯ





E no mundo Delphi



DM



MainDB



TrUniQry



UniQry



RoomQry



BStationQry



RQry



BDetailQry



BDetailQry2



ODetailQry



RRQry



UniQry2



ImportTbl



ImportExQry



DsUniQry DSRoomQDSBStationQry DsRQry



BuildTbl



DSRRQry



CustDSrc



MetroDSrc



MLineDSrc



PostDSrc



PostMDSrc



BuildDSrc



OfficeDSrc



SalesDSrc



DsCustIn



DSMetro



DSMLine



DSPost



DSPostM



DSBuild



DSOffice



DSSales



BTYPESTbl



BMaterTbl



PrettyTbl



BSecureTbl



EntrTbl



PhoneTbl



PayTpTbl



OnwerTpTbl



RentTbl



ParkingTbl



BCatTbl



OTypeTbl



RTypeTbl



PandusTbl



InsLocTbl



DSBTypes



DSBMater



DSBPretty



DSBSecure



DSEntr



DSPhone



DSPayTp



DSONwerTp



DSRent



DSParking



DSBCat



DSOTypes



DSRtypes



DSPandus



DSInsLoc

Porque os japoneses são F*****



Porque é tão difícil mudar nossa forma de pensar?



ÁRVORE BOSQUE FLORESTA



I Have a Dream.



In Just Dark Day They Make Little Space
Sing Doctor Deeply History Showing Stand Words Free
Made Black Wilk Valley Return To Us Brotherhood
Brotherhood Go Up Time Faith
White Day Must Every America Long Ahens When Later
God Let With Satisfied Ring Mountain Mississippi
Negro Nation Freedom One From Men Out
Our There Together Not Come Children Dream
Free Check Justice Today Back Able Years Cannot New
People John Great Hundred Injustice American Shall
Rise Great Suffering Hope Is Down Georgia
All Land Rights Physical Current Mobility

“Eu também tenho um sonho”

Good IF is Dead IF

Porque o IF's em excesso são tóxicos para seu código?

Engessam o código

Porque os IF's em excesso são **tóxicos** para seu código?

Dificultam a leitura com condicionais desnecessárias

```
1 var leak;
2 var wasHacker;
3 var noMatter;
4
5 if(leak){
6     wasHacker = true;
7 }
8 if(leak && wasHacker){
9     noMatter = tru|
10 }
11
12
```



@cafemaismais

Porque o IF's em excesso são tóxicos para seu código?

Dificultam a **reutilização** do código

Porque o IF's em excesso são tóxicos para seu código?

Prejudicam a escala da regra de negócios

Porque o IF's em excesso são tóxicos para seu código?

Fazem você perder muitas horas de sono

Porque o IF's em excesso são tóxicos para seu código?

Não deixam você tirar **férias**

O que fazer então?

Código no Botão

Me chama de OLX e Desapega

Utilize Abstrações/Interfaces para
Deixar tudo mais maleável

Trabalho que você pode fazer

Formulário/Console



Banco de Dados



RAD

Componentes IDE

Formulário



Componente de
Conexão



FireDAC

Banco de Dados



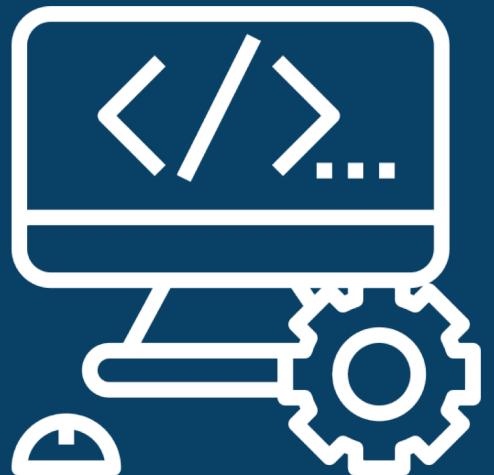
RAD

Formulário



Programador

Interface de Conexão



Componentes da IDE

Componentes de Conexão



FireDAC



Zeus



REST

Banco de Dados



MySQL



Firebird



Cloud

Trabalho que você pode fazer

PDV



Regra de Negócio

Banco de Dados

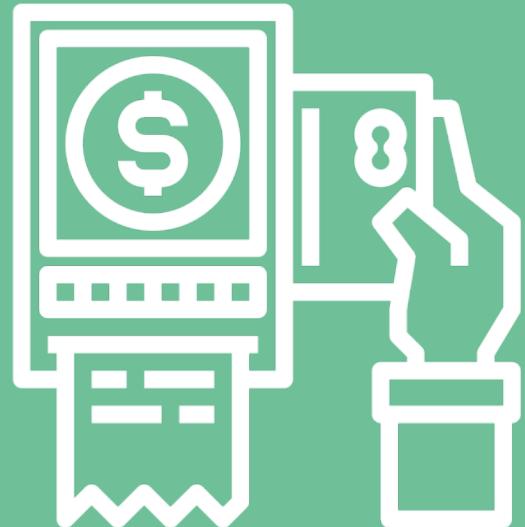


RAD

Programador

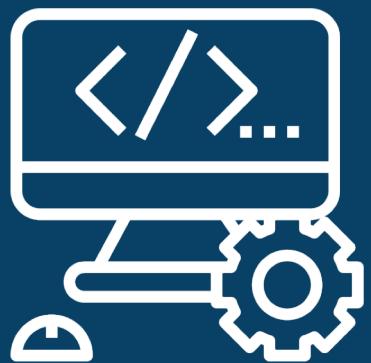
Componentes IDE

PDV



Regra de Negócio

Interface de Conexão



Componentes de Conexão



FireDAC



Zeus



REST

Banco de Dados



MySQL



Firebird



Cloud

Applications

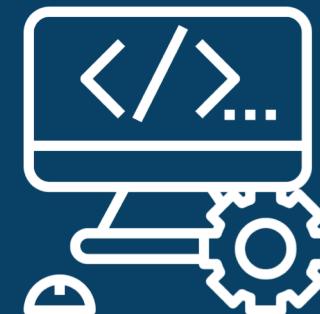


Regra de Negócios

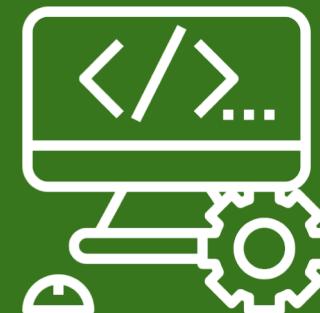


Interface

Interface de Conexão



Interface Fiscal



Componentes da IDE



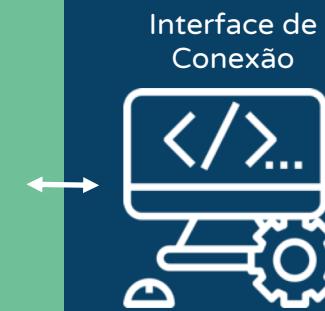
Componentes da Terceiro



Clients



Interface



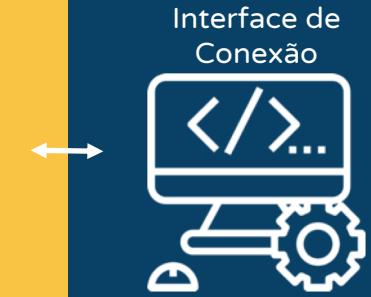
WebService



Regra de Negócios



Interface



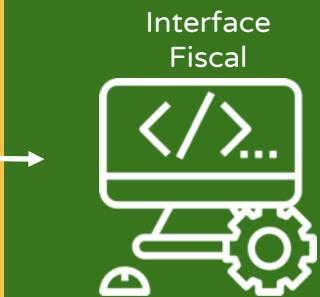
Componentes da IDE



Componentes da Terceiro



TECNOSPEED ➤



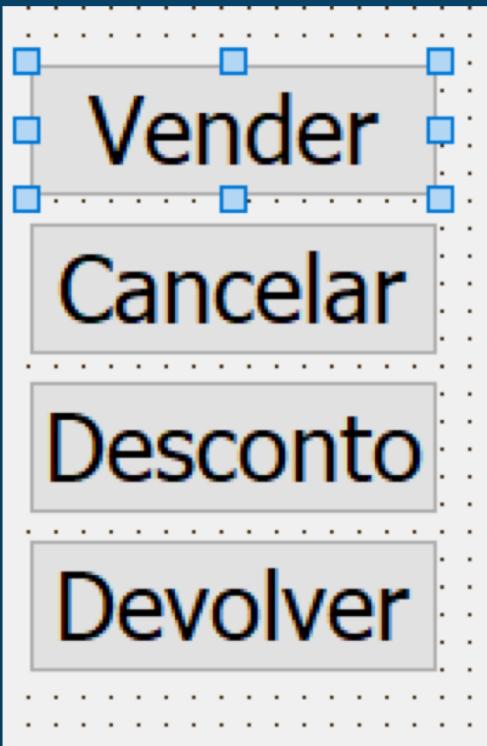
Componente

Abstract Connection

Design Pattern **State**

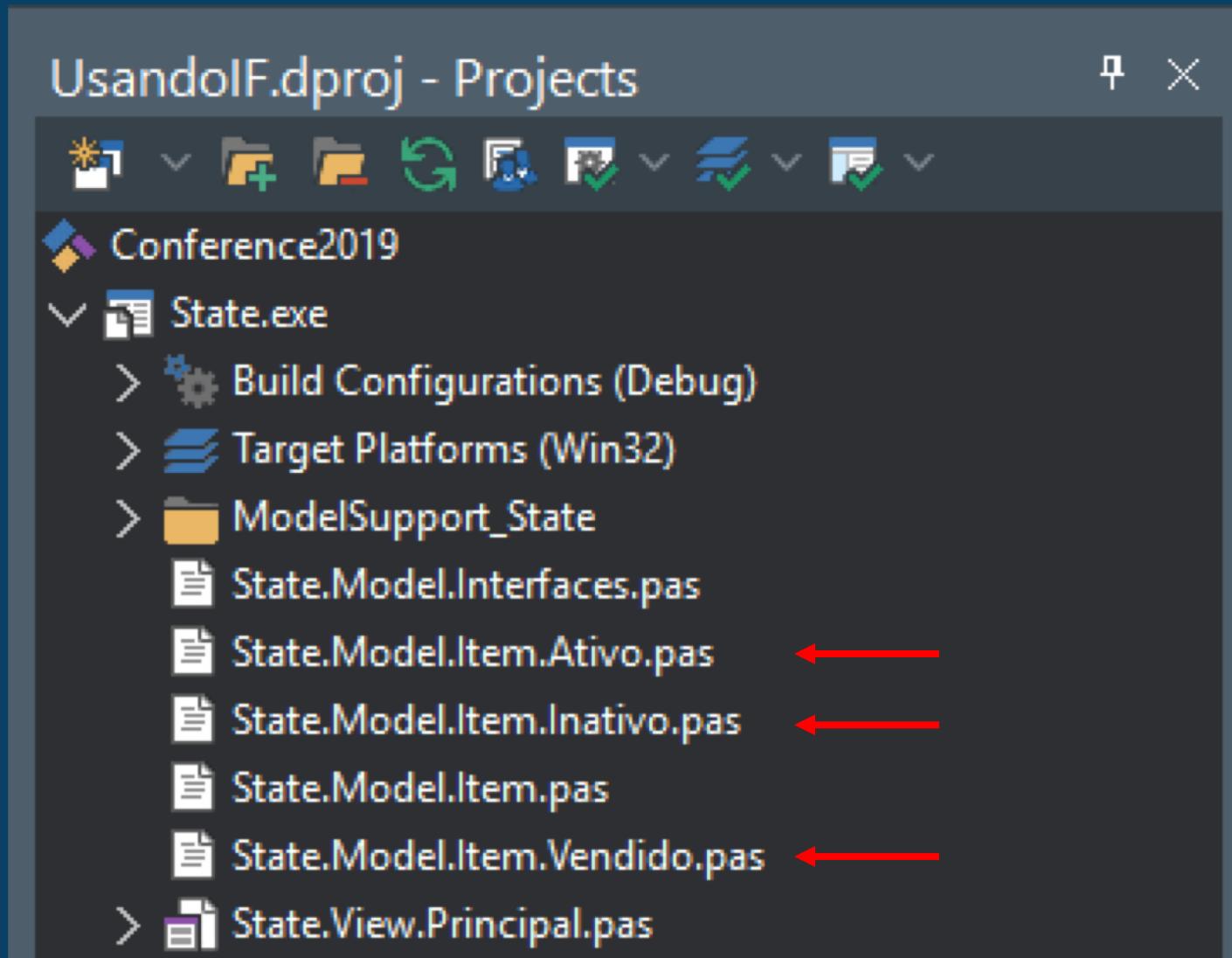
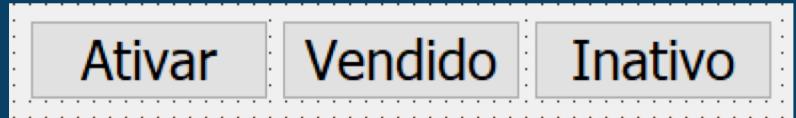
O jeito mais fácil de **eliminar IF's**

Transforme os verbos em
Métodos



```
iItemOperacoes = interface
    function Vender : iItemOperacoes;
    function Cancelar : iItemOperacoes;
    function Desconto : iItemOperacoes;
    function Devolver : iItemOperacoes;
end;
```

Crie classes de validação para
Estados



Crie a camada de
Abstração do Item

```
iItem = interface
  ['{2FA38521-9128-49F7-87F6-8736BA74D958}']
    function State : iState<iItemOperacaoes>;
    function Operacaoes : iItemOperacaoes;
  end;
```

```
Type
  TModelItem = class(TInterfacedObject, iItem, iItemOperacaoes, iState<iItemOperacaoes>)
  private
    FState : iItemOperacaoes;
  public
    constructor Create;
    destructor Destroy; override;
    class function New: iItem;
    function Vender: iItemOperacaoes;
    function Cancelar: iItemOperacaoes;
    function Desconto: iItemOperacaoes;
    function Devolver: iItemOperacaoes;
    function SetState(Value : iItemOperacaoes) : iItemOperacaoes;
    function State : iState<iItemOperacaoes>;
    function Operacaoes : iItemOperacaoes;
  end;
```

```
function TModelItem.Vender: iItemOperacaoes;
begin
  FState.Vender;
  FState := TModelItemVendido.New;
  Result := Self;
end;
```

A implementação da ação fica
na classe de Estado

```
function TModelItemAtivo.Cancelar: iItemOperacoes;
begin
  Result := Self;
  raise Exception.Create('Este item ainda não foi vendido');
end;
```

Boas Práticas para Geração de Arquivos Fiscais

É possível validar uma regra fiscal sem usar IF?

Padrões Utilizados

Facade
Abstract Factory
Command
Observer
Visitor

Facade

```
iModelFiscalNFe = interface
  ['{1AE1FDBA-C25F-49A0-88C5-3F40AC801E4E}']
  function Clear : iModelFiscalNFe;
  function Gerar : iModelFiscalNFe;
  function Component : iModelFiscalNFeComponentes<TACbrNFe>;
  function Config : iNfeConfig;
  function Observer ( Value : iSubjectNfe ) : iModelFiscalNFe; overload;
  function Observer : iSubjectNfe; overload;
  function RegimeTributario( Value : Integer ) : iModelFiscalNfe;
  function PrepararNota : iModelFiscalNfe;
end;
```

Abstract Factory

```
iModelFiscalNfeFactory = interface
  ['{9AB6AAC2-B5DD-4D52-8662-BC7F4236EEA6}']
    function ACBr : iModelFiscalNFeComponentes<TACbrNFe>;
  end;
```

```
function TModelFiscalNFeComponentesFactory.ACBr: iModelFiscalNFeComponentes<TACbrNFe>;
begin
  if not Assigned(FACBr) then
    FACBr := TdmACBrNFe.Create(nil);
  Result := FACBr;
end;
```

Command

```
function TModelFiscalNFe.Gerar: iModelFiscalNFe;
begin
  Result := Self;
  TModelFiscalNFeInvoker.New
    .Add(TModelFiscalNFeCommandFactory.New(Self).Geral)
    .Add(TModelFiscalNFeCommandFactory.New(Self).Ide)
    .Add(TModelFiscalNFeCommandFactory.New(Self).Emitente)
    .Add(TModelFiscalNFeCommandFactory.New(Self).Dest)
    .Add(TModelFiscalNFeCommandFactory.New(Self).Prod)
    .Add(TModelFiscalNFeCommandFactory.New(Self).Total)
    .Add(TModelFiscalNFeCommandFactory.New(Self).Transp)
    .Add(TModelFiscalNFeCommandFactory.New(Self).TranspVol)
    .Add(TModelFiscalNFeCommandFactory.New(Self).CobrFat)
    .Add(TModelFiscalNFeCommandFactory.New(Self).CobrDup)
    .Add(TModelFiscalNFeCommandFactory.New(Self).InfAdic)
    .Add(TModelFiscalNFeCommandFactory.New(Self).InfAdicObsComp)
    .Add(TModelFiscalNFeCommandFactory.New(Self).InfAdicObsFisco)
    .Add(TModelFiscalNFeCommandFactory.New(Self).Exporta)
    .Add(TModelFiscalNFeCommandFactory.New(Self).Compra)
    .Add(TModelFiscalNFeCommandFactory.New(Self).Pagamento)
    .Add(TModelFiscalNFeCommandFactory.New(Self).GerarNFe)
    .Add(TModelFiscalNFeCommandFactory.New(Self).EnviarNFe)
  .Execute;
  FObserver.Notify('Nota foi Gerada com Sucesso');
  FComponente._this.NotasFiscais.Clear;
end;
```

```
iCommand = interface
  ['{46EDD05F-F525-488E-BF46-F006481E6162}']
  function Execute : iCommand;
end;

iInvoker = interface
  ['{F6CA4333-11C1-4768-857C-E9B867A19D86}']
  function Add( Value : iCommand ) : iInvoker;
  function Execute : iInvoker;
end;
```

Visitor

```
iModelNfeRegras = interface
  ['{3BD7420A-3D90-43B7-9EEC-CAAAC00473D1}']
  function ProdutoImpostoICMS : iModelNfeRegras;

  function Visit( Value : iModelFiscalNFe ) : iModelNfeRegras;
  function AcceptRegraCST( Value : iModelNfeRegrasImposto) : iModelNfeRegras;
end;

iModelNfeRegrasImposto = interface
  ['{AF3C874B-F0DC-4363-A41C-23EAA7725D2E}']
  function CalculoImpostoProduto : iModelNfeRegrasImposto;
  function Visit( Value : iModelFiscalNFe ) : iModelNfeRegrasImposto;
end;

iVisitable = interface
  ['{304400C1-4CDC-458F-AD54-B4CADAD2A007}']
  function Accept(Value : iModelNfeRegras) : iModelNfeRegras;
end;
```

Observer

```
type
  iObserverNfe = interface
    ['{F768124E-CD62-42B4-98CB-6D18D7D95CF4}']
    function Update (Value : String) : iObserverNfe;
    function NotifyMethod (Value : TNotifyNfe) : iObserverNfe;
  end;

  iSubjectNfe = interface
    ['{D7028F24-ABE8-4F30-8A75-0134B46CFDAB}']
    function AddObserver (Value : iObserverNfe) : iSubjectNfe;
    function RemoveObserver (Value : iObserverNfe) : iSubjectNfe;
    function Notify (Value : String) : iSubjectNfe;
  end;
```

Turbinando a ACBr com

Helpers

```
type
  TpcnCRTHelper = record helper for TpcnCRT
    function Regras : iModelNfeRegras;
  end;

  TpcnCSTIcmsHelper = record helper for TpcnCSTIcms
    function Regras : iModelNfeRegrasImposto;
  end;

  TpcnCSOSNIcmsHelper = record helper for TpcnCSOSNIcms
    function Regras : iModelNfeRegrasImposto;
  end;
```

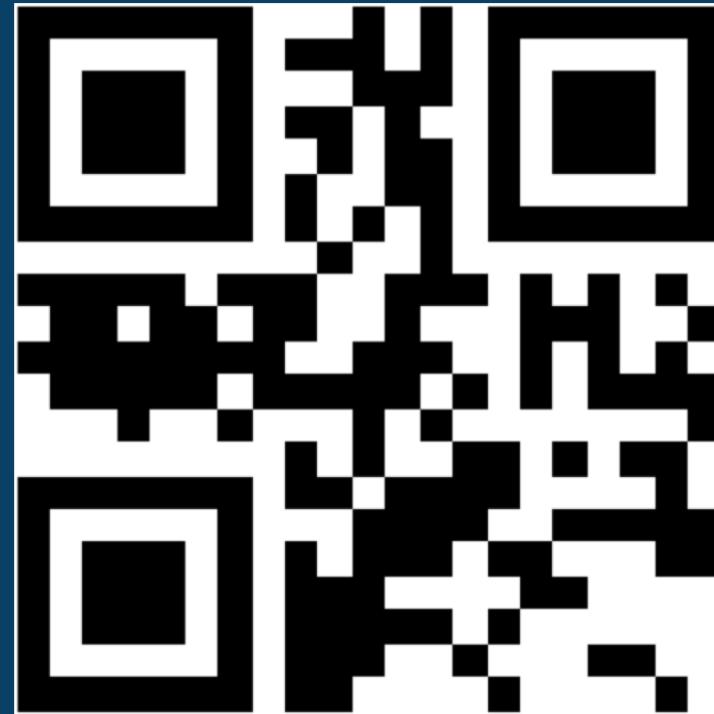
```
function TpcnCRTHelper.Regras: iModelNfeRegras;
begin
  case Self of
    crtSimplesNacional:      Result := TModelFiscaoNFeRegrasFisicaisICMSSimples.New;
    crtSimplesExcessoReceita: Result := nil;
    crtRegimeNormal:          Result := TModelFiscaoNFeRegrasFisicaisICMSRegimeNormal.New;
  end;
end;
```

```
Accept(
  FParent.Component.NotaFiscal.NFe.Emit.CRT.Regras
).ProdutoImpostoICMS;
```

E quando eu posso **usar IF**?

Quando o estado da validação for imutável e não implicar na **regra de negócio**

```
procedure TForm1.Button8Click(Sender: TObject);
begin
  if Edit1.Text = '' then
    raise Exception.Create('Preencha o nome do usuário'); ↑
end;
```



Super Bônus da Palestra



Obrigado



bittencourtthulio@gmail.com



facebook.com/bittencourtthulio



@thuliobitten



@thuliobittencourt

Da sala de reunião à vida real

Embarcadero Conference 2019