

# Código Limpo

## Definição de Clean Code

A apresentação de um código claro e organizado não consiste apenas na convenção nomes, constantes, classes, variáveis, espaçamento etc.. Um código limpo (*clean code*) deve ser:

- Simples: fácil entendimento;
- Eficiente: realizar tudo o que foi proposto;
- Único: não realizar algo que outro trecho de código já faz;
- Direto: não dar voltas para chegar no resultado;
- Feito com atenção: o código deve ser sempre feito com preocupação e revisto depois de pronto;

## Bad Smells

Em contra partida, também existem as *Bad Smells*, que como sua tradução já sugere, é algo com cheiro ruim, e representa o código com práticas que não devem ser utilizadas. Essas práticas se devem aos códigos que fogem das características de um código limpo. Abaixo são demonstrados alguns exemplos de *bad smells*.

## Encadeamento de If's

```
if (Q_Manut.FieldName('DiasDireito').AsFloat = 0) then
begin
  if (AnsiSameText(Trim(Q_Manut.FieldName('DiasDireito').AsString), '')) then
  begin
    if (MessageDialog.Show('Não foi informado os dias de Direito. Deseja' +
      'Continuar?', mtConfirmation, [mbYes, mbNo], 0) = mrNo) then
    begin
      Q_Manut.FieldName('DiasDireito').FocusControl;
      Abort;
    end;
  end;
end;
```

Esse tipo de encadeamento pode ser facilmente substituído pela palavra reservada "and".

```
if (Q_Manut.FieldName('DiasDireito').AsFloat = 0) and  
  (AnsiSameText(Trim(Q_Manut.FieldName('DiasDireito').AsString), '')) and  
  (MessageBox.Show('Não foi informado os dias de Direito. Deseja' +  
    'Continuar?', mtConfirmation, [mbYes, mbNo], 0) = mrNo) then  
begin  
  Q_Manut.FieldName('DiasDireito').FocusControl;  
  Abort;  
end;
```

## Excesso de if-else

```
if (Q_Manut.FieldName('DiasDireito').AsFloat = 0) then  
  LTipo := 1  
else if (Q_Manut.FieldName('DiasDireito').AsFloat = 1) then  
  LTipo := 2  
else if (Q_Manut.FieldName('DiasDireito').AsFloat = 2) then  
  LTipo := 3  
else if (Q_Manut.FieldName('DiasDireito').AsFloat = 3) then  
  LTipo := 5  
else  
  LTipo := 9;
```

O trecho acima pode ter sua estrutura facilitada com a utilização de um "case".

```
case Q_Manut.FieldName('DiasDireito').AsFloat of  
  0: LTipo := 1;  
  1: LTipo := 2;  
  2: LTipo := 3;  
  3: LTipo := 5;  
else  
  LTipo := 9;  
end;
```

## Atribuição Indireta

Situações com atribuições indiretas aparecem constantemente no código

```
if (Q_Manut.FieldName('DiasDireito').AsFloat = 0) then  
  LDeveIncrementar := False  
else  
  LDeveIncrementar := True;
```

```
if (RG_TipoCliente.ItemIndex = 0) then  
  LTipo := 0  
else (RG_TipoCliente.ItemIndex = 1) then  
  LTipo := 1;
```

Basta uma pequena análise por parte do desenvolvedor para perceber que podem ser simplificadas

```
LDeveIncrementar := (Q_Manut.FieldName('DiasDireito').AsFloat <> 0);
```

```
LTipo := RG_TipoCliente.ItemIndex;
```

---

Revision #3

Created 29 December 2020 16:34:46

Updated 31 December 2020 11:51:54