Código Limpo

Hugo Corbucci
hugo@agilbits.com.br

AGIL
BITS


NÃO É
Limpar seu HD
WWW.FOTOCOMEDIA.COM

NÃO É
Higiene Básica

Java:

```java
public class C {
private boolean[] p;
public C(int m){p=new boolean[m];
p[0]=p[1]=false;
  for(int n=2;n<m;n++)p[n]=true;
    for(int n=2;n<m;n++)

      if(p[n])for(int f=2;n*f < p.length; f++)

      p[n*f]=false;
}public boolean p(int n){return n>0 && n<p.length&&
  p[n];}
}
```

Python:

```python
global __p
__p= []
__p.append(  False)
__p.append(False    )
def c(n):
    if len(__p)-1<   n:
        for i in range(len(__p)  -1,n):
            __p.append(True)
        r=range(2,n)
        for i in r:
            if __p[i]:
                for f in r:
                    if(i*f<len(__p)):
                        __p[ i   *   f ]=False
def p(n):
    c(abs( n) )
    return __p[n ]
```

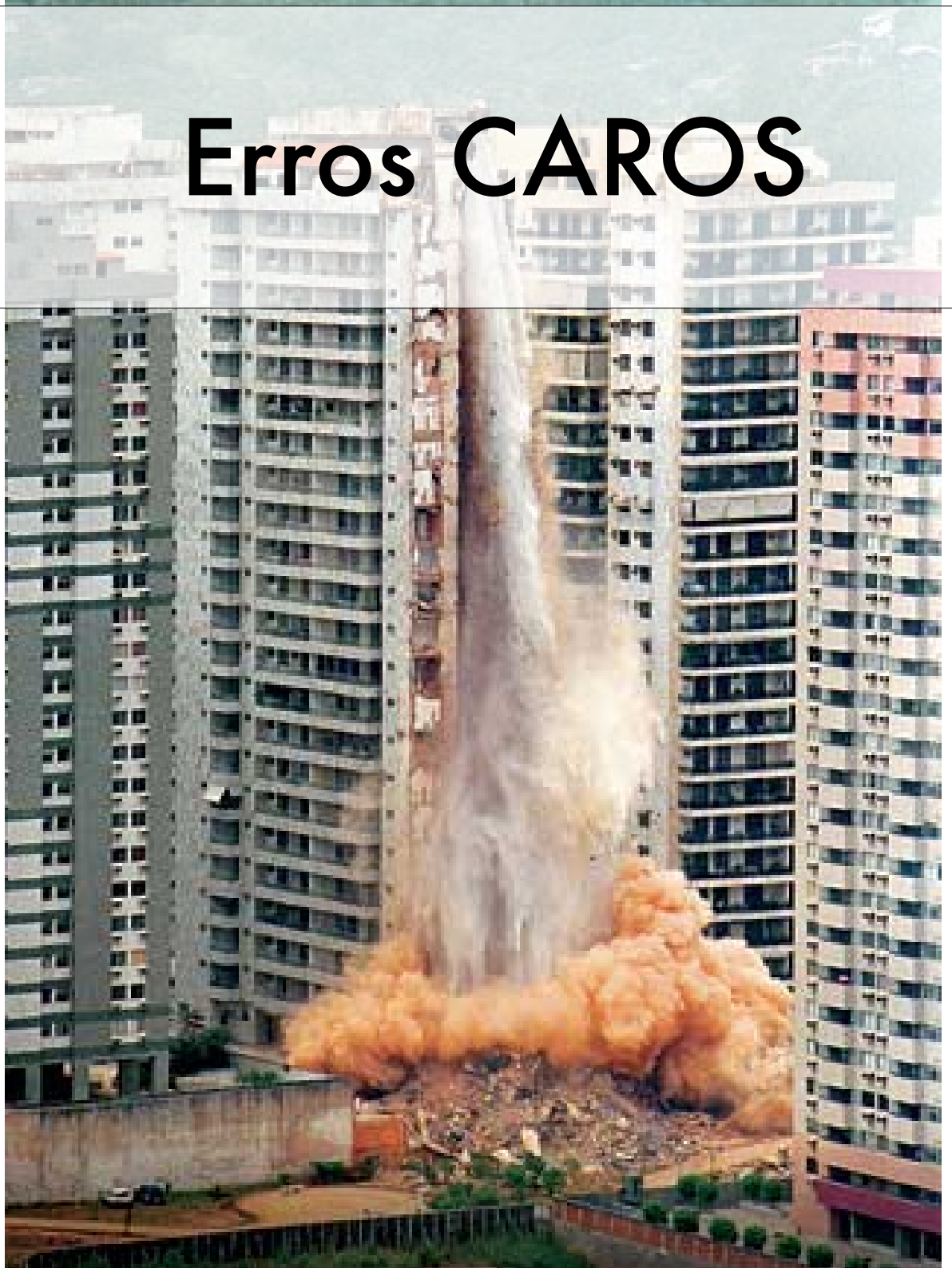
Ruby:

```ruby
$_p = []; $_p<<false
$_p << false
def c ( n ) return if $_p.length > n;
for i in ($_p.length()..n)
    $_p<<true;end
  r=2..n;r.each{|i|if($_p[i])

  r.each{|f|$_p[i*f]=false if i*f < $_p.length;
  }end}end
def p n

    nn= n.abs;c nn
$_p[nn];end
```

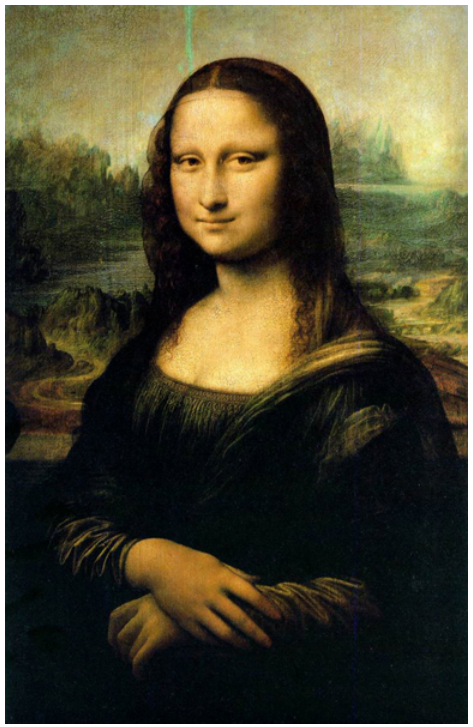
Engenharia

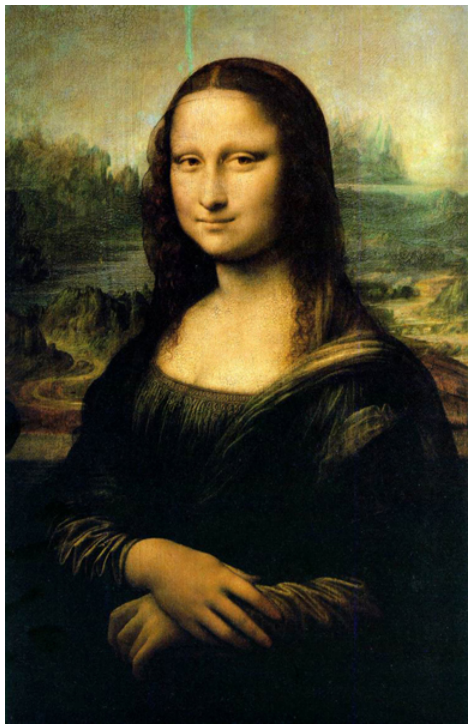
Planejamento

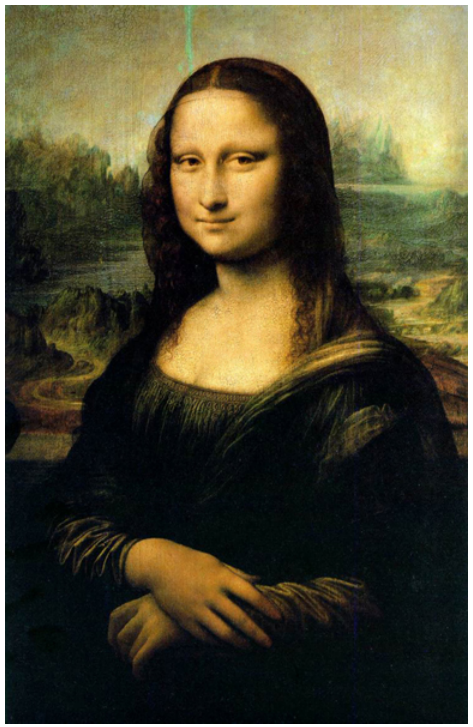
Erros CAROS

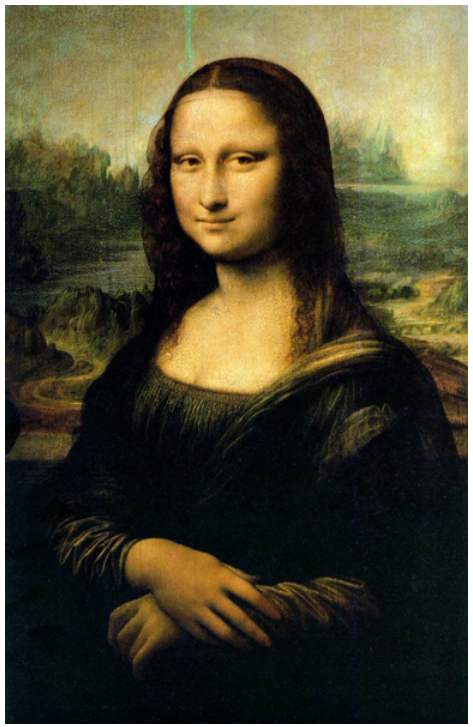



Arquitetura









LES

FLEURS DU MAL

PAR

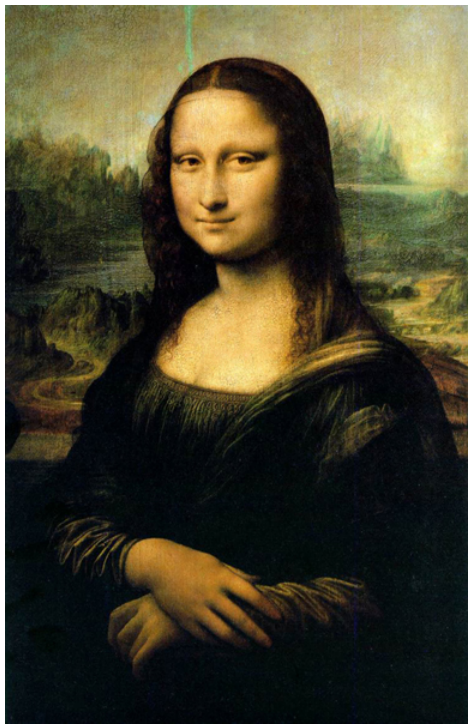
CHARLES BAUDELAIRE

On dit qu'il faut couler les exécrables choses
Dans le puits de l'oubli et au sépulchre enclos;
Et que par les écrits le mal ressuscité
Infectera les mœurs de la postérité;
Mais le vice n'a point pour mère la science,
Et la vertu n'est pas fille de l'ignorance.

(Théodore Agrippa d'Aubigné, *Les Tragiques*, liv. II

PARIS
POULET-MALASSIS ET DE BROISE
LIBRAIRES-ÉDITEURS
4, rue de Buci.

1857



LES

FLEURS DU MAL

PAR

CHARLES BAUDELAIRE

On dit qu'il faut couler les exécrables choses
Dans le puits de l'oubli et au sépulchre enclos;
Et que par les écrits le mal ressuscité
Infectera les mœurs de la postérité;
Mais le vice n'a point pour mère la science,
Et la vertu n'est pas fille de l'ignorance.

(Théodore Agrippa d'Aubigné, *Les Tragiques*, liv. II

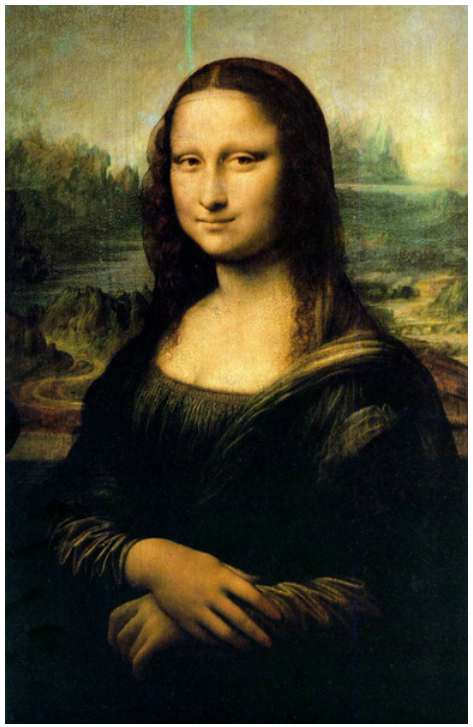
PARIS

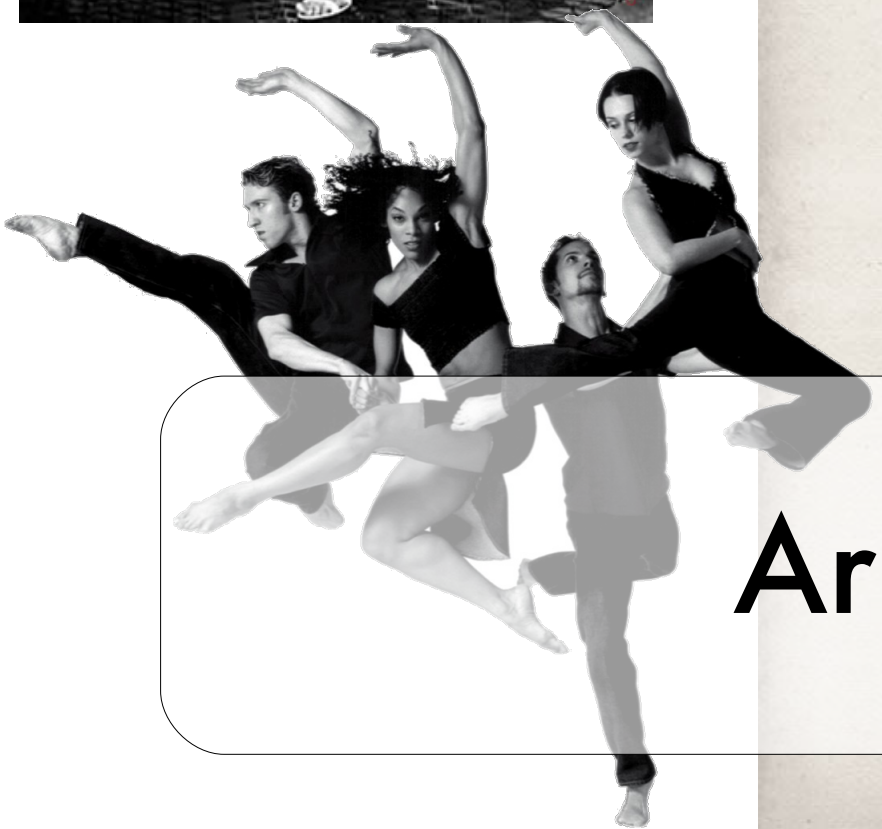
POULET-MALASSIS ET DE BROISE

LIBRAIRES-ÉDITEURS

4, rue de Buci.

1857





LES

FLEURS DU MAL

PAR

CHARLES BAUDELAIRE

On dit qu'il faut couler les exécrables choses
Dans le puits de l'oubli et au sépulchre enclos;
Et que par les écrits le mal ressuscité
Infectera les mœurs de la postérité;
Mais le vice n'a point pour mère la science,
Et la vertu n'est pas fille de l'ignorance.

(Théodore Agrippa d'Aubigné, *Les Tragiques*, liv. II)

PARIS

POULET-MALASSIS ET DE BROISE

LIBRAIRES-ÉDITEURS

4, rue de Buci.

1857

Arte

Artesanato

Fábricas VS Feito à mão


Sob Medida


Algo ÚNICO


Não sabemos

```
{
    if(m_pIO->IsDriverOn(PORT_0, D_CH1_GASLINE_PURGE))
    {
        m_pIO->GasPurgeValve(OFF);
        m_pLog->LogMessage("Manual Operation: CF4 Gas Valve OFF");
    }
    else
    {
        m_pIO->GasPurgeValve(ON);
        m_pLog->LogMessage("Manual Operation: CF4 Gas Valve ON");
    }
}

// Gasline Purge for Chamber 2
void CPTViewDlg::OnGasPurgeChamber2()
{
    if(m_pIO->IsDriverOn(PORT_0, D_CH2_GASLINE_PURGE))
    {
        m_pIO->GasPurgeValve2(OFF);
        m_pLog->LogMessage("Manual Operation: Chamber 2 Gasline Purge Valve OFF");
    }
    else
    {
        m_pIO->GasPurgeValve2(ON);
        m_pLog->LogMessage("Manual Operation: Chamber 2 Gasline Purge Valve ON");
    }
}


void CPTViewDlg::OnValveRough() //ok
{
    if(m_pIO->IsDriverOn(PORT_4, D_ROUGH))
    {
        m_pIO->RoughValve(OFF);
        m_pLog->LogMessage("Manual Operation: Rough Valve OFF");
    }
    else
    {
        m_pIO->RoughValve(ON);
        m_pLog->LogMessage("Manual Operation: Rough Valve ON");
    }
}

void CPTViewDlg::OnValveFore() //ok
```

Código Assassino!

```
  if(m_pIO->IsDriverOn(PORT_0, D_CH1_GASLINE_PURGE))
  {
      m_pIO->GasPurgeValve(OFF);
      m_pLog->LogMessage("Manual Operation: CF4 Gas Valve OFF");
  }
  else
  {
      m_pIO->GasPurgeValve(ON);
      m_pLog->LogMessage("Manual Operation: CF4 Gas Valve ON");
  }


 Gasline Purge for Chamber 2
id CPTViewDlg::OnGasPurgeChamber2()

  if(m_pIO->IsDriverOn(PORT_0, D_CH2_GASLINE_PURG
  {
      m_pIO->GasPurgeValve2(OFF);
      m_pLog->LogMessage("Manual Operation:  mber 2 Gasli  Purge Va  OFF");
  }
  else
  {
      m_pIO->GasPurgeValve2(ON);
      m_pLog->LogMessage("Manual Op  ion: Cha  2 Gasline  ge Valve ON");
  }



id CPTViewDlg::OnValveRou    //ok

  if(m_pIO->IsDriverOn  1_4, D_RO
  {
      m_pIO->RoughV  (OFF);
      m_pLog->LogMe  ge("Manu  Op  ion:  gh Valve OFF");
  }
  else
  {
      m_pIO->RoughValve(
      m_pLog->LogMessage("M  el Op  tion: Rough Valve ON");
  }


id CPTViewDlg::OnValveFore() //ok
```

LSA = Languages and Systems Administration

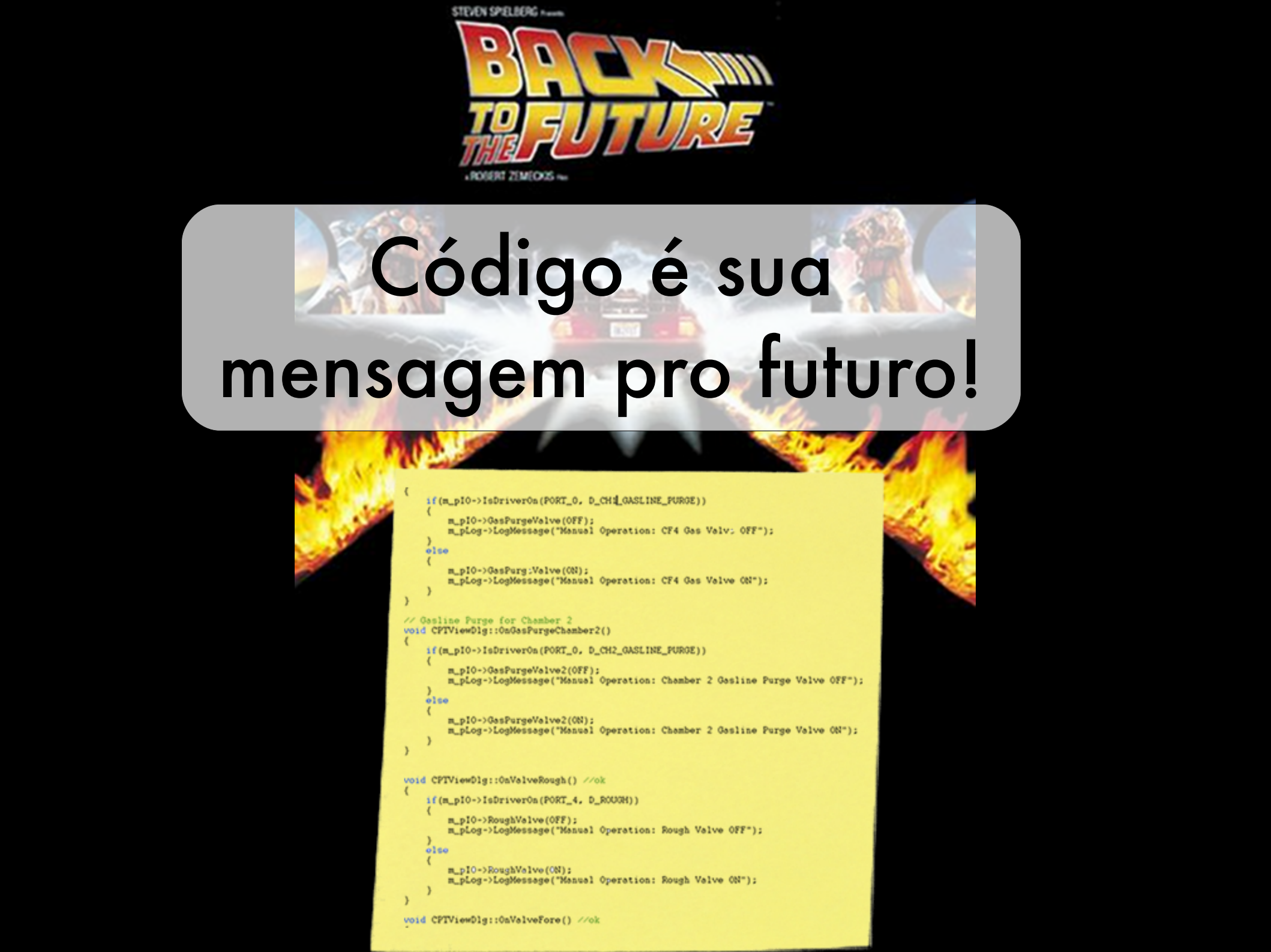

Mantenedor Psicótico

Com grandes poderes,
vem grandes responsabilidades

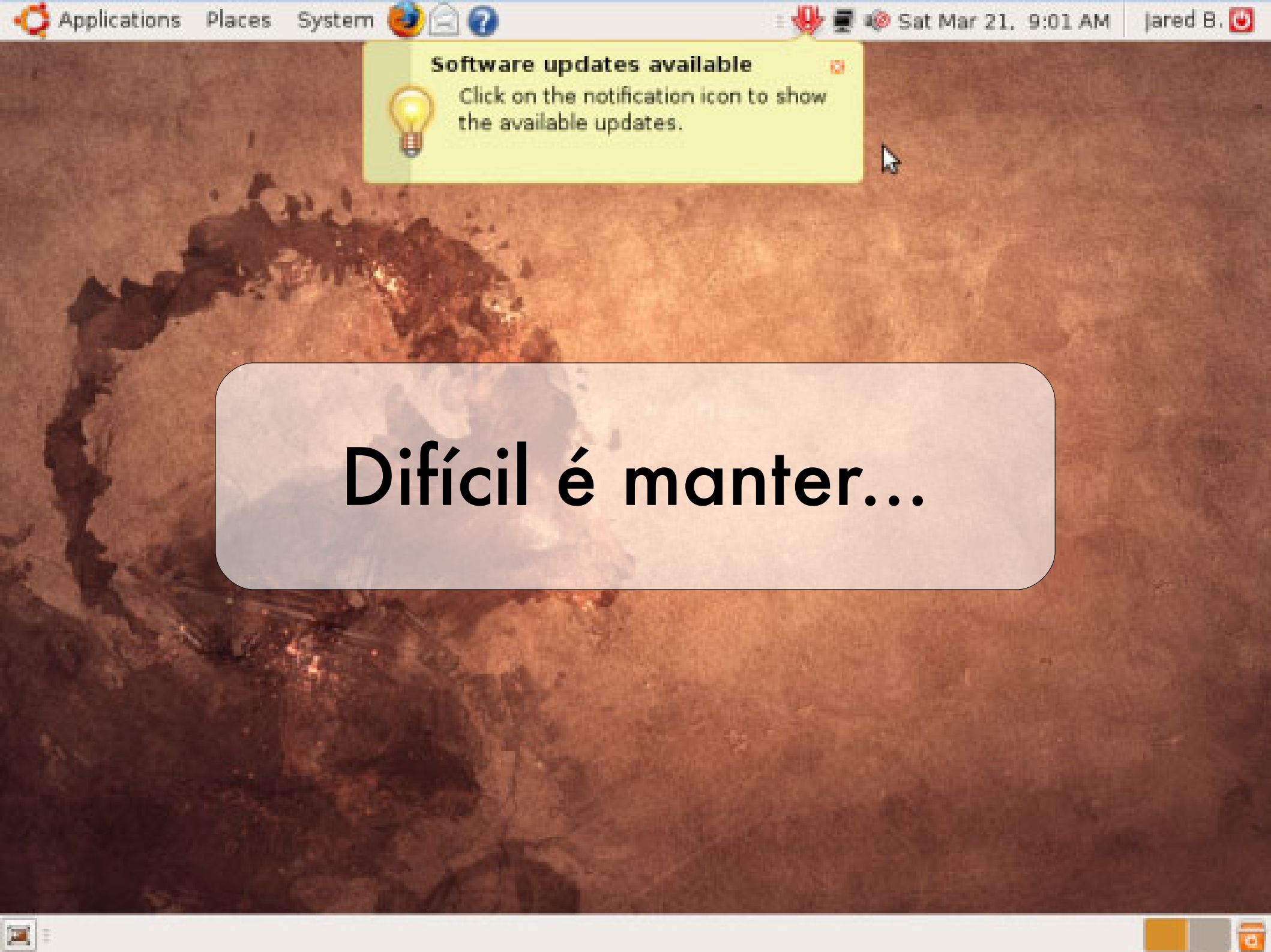
SOFTware
NÃO
deveria ser
HARD
de mudar


Código é como cozinha


Se não é cuidado,
fica nojento!


STEVEN SPIELBERG
BACK TO THE FUTURE
ROBERT ZEMECKIS
Código é sua mensagem pro futuro!
{
if(m_pIO->IsDriverOn(PORT_0, D_CH1_GASLINE_PURGE))
{
m_pIO->GasPurgeValve(OFF);
m_pLog->LogMessage("Manual Operation: CF4 Gas Valv; OFF");
}
else
{
m_pIO->GasPurg;Valve(ON);
m_pLog->LogMessage("Manual Operation: CF4 Gas Valve ON");
}
}
// Gasline Purge for Chamber 2
void CPTViewDlg::OnGasPurgeChamber2()
{
if(m_pIO->IsDriverOn(PORT_0, D_CH2_GASLINE_PURGE))
{
m_pIO->GasPurgeValve2(OFF);
m_pLog->LogMessage("Manual Operation: Chamber 2 Gasline Purge Valve OFF");
}
else
{
m_pIO->GasPurgeValve2(ON);
m_pLog->LogMessage("Manual Operation: Chamber 2 Gasline Purge Valve ON");
}
}
void CPTViewDlg::OnValveRough() //ok
{
if(m_pIO->IsDriverOn(PORT_4, D_ROUGH))
{
m_pIO->RoughValve(OFF);
m_pLog->LogMessage("Manual Operation: Rough Valve OFF");
}
else
{
m_pIO->RoughValve(ON);
m_pLog->LogMessage("Manual Operation: Rough Valve ON");
}
}
void CPTViewDlg::OnValveFore() //ok

Fazer passar é fácil!


Applications
Places
System
Sat Mar 21, 9:01 AM
Jared B.
Software updates available
Click on the notification icon to show the available updates.
Difícil é manter...

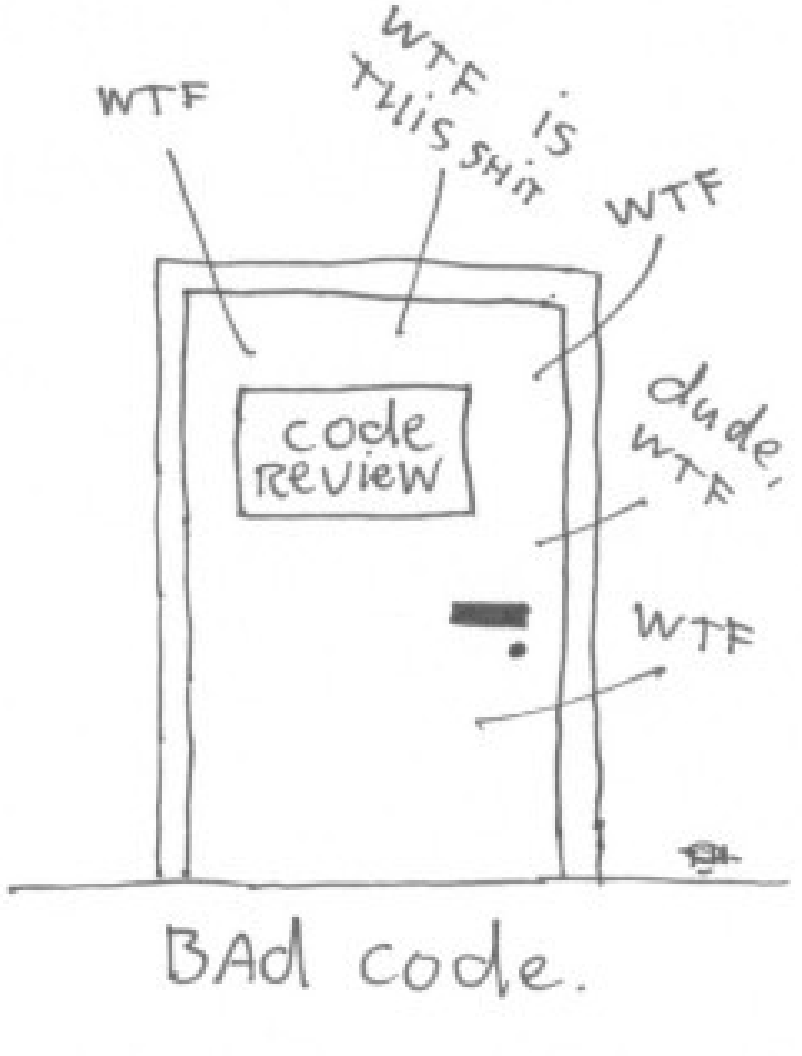
Precisa prever o futuro


Não sabemos
comparar


Sempre temos alguma
Dívida Técnica
4205 2000 3456 0012

The ONLY VALID MEASUREMENT OF Code QUALITY: WTFs/MINUTE

Good code.



BAd code.

"*Código Limpo sempre aparenta ter sido escrito por alguém que se importa. Não há nada de óbvio que você possa fazer para melhorá-lo. Todas essas coisas foram pensadas pelo autor do código, e se você tentar imaginar melhorias, você é levado de volta ao código original, apreciando o código que alguém deixou para você – código de alguém que se importa profundamente com seu trabalho.*"

Michael Feathers

Autor de
"Working Effectively with Legacy Code"

“*Você sabe que está trabalhando com código limpo quando cada rotina que você lê acaba sendo aquilo que você esperava que fosse. Você pode chamá-lo de código bonito quando o código também faz parecer que a linguagem foi feita para aquele problema.*”

Ward Cunningham

Inventor do Wiki, FIT e
co-inventor de
Programação Extrema (XP)

Mau cheiros de código fazem isso


Se feder,
troque-o


Mas como limpar?

Mantenha pequeno

```
private static void drawSmiley(final Shell shell, PaintEvent event) {
    Rectangle rect = shell.getClientArea();
    int diameter = Math.min(rect.width - 1, rect.height - 1);
    int x = (rect.width - diameter) / 2;
    int y = (rect.height - diameter) / 2;
    Rectangle smileySquare = new Rectangle(x, y, diameter, diameter);

    GC gc = event.gc;
    gc.setLineWidth(3);
    Display display = event.display;
    Color yellow = new Color(display, 255, 255, 0);
    gc.setBackground(yellow);
    gc.fillOval(smileySquare.x, smileySquare.y, smileySquare.width,
            smileySquare.height);
    gc.drawOval(smileySquare.x, smileySquare.y, smileySquare.width,
            smileySquare.height);

    Color black = new Color(display, 0, 0, 0);
    gc.setBackground(black);
    int horizontalCenter = smileySquare.x + smileySquare.height / 2;
    int leftEyeX = horizontalCenter - smileySquare.width / 8;
    int eyesStartY = smileySquare.y + smileySquare.height / 8;
    int eyesWidth = smileySquare.width / 16;
    int eyesHeight = smileySquare.height / 4;
    gc.fillOval(leftEyeX, eyesStartY, eyesWidth, eyesHeight);

    int rightEyeX = horizontalCenter + smileySquare.width / 8;
    gc.fillOval(rightEyeX, eyesStartY, eyesWidth, eyesHeight);

    gc.drawArc(smileySquare.x, smileySquare.y - smileySquare.height / 4,
            smileySquare.width, smileySquare.height, 225, 90);
}
```

```
private static void drawCenteredSmileyInShell(Shell shell, PaintEvent event) {
    drawSmiley(event, getSmileyBoundaries(shell));
}
private static Rectangle getSmileyBoundaries(Shell shell) {
    Rectangle rectangle = shell.getClientArea();
    int diameter = Math.min(rectangle.width - 1, rectangle.height - 1);
    int horizontalStart = (rectangle.width - diameter) / 2;
    int verticalStart = (rectangle.height - diameter) / 2;
    return new Rectangle(horizontalStart, verticalStart, diameter, diameter);
}
private static void drawSmiley(PaintEvent event, Rectangle smileyBoundaries) {
    event.gc.setLineWidth(3);
    drawHead(smileyBoundaries, event.gc);
    drawEyes(smileyBoundaries, event.gc);
    drawMouth(smileyBoundaries, event.gc);
}
private static void drawHead(Rectangle headBoundaries, GC gc) {
    setHeadColor(gc);
    gc.fillOval(headBoundaries.x, headBoundaries.y, headBoundaries.width,
            headBoundaries.height);
    gc.drawOval(headBoundaries.x, headBoundaries.y, headBoundaries.width,
            headBoundaries.height);
}
private static void setHeadColor(GC gc) {
    gc.setBackground(new Color(Display.getCurrent(), 255, 255, 0));
}
private static void drawEyes(Rectangle headBoundaries, GC gc) {
    setEyesColor(gc);
    Rectangle eyeBoundaries = getEyeBoundaries(headBoundaries);
    drawLeftEye(gc, headBoundaries, eyeBoundaries);
    drawRightEye(gc, headBoundaries, eyeBoundaries);
}
```

```
private static void setEyesColor(GC gc) {
    gc.setBackground(new Color(Display.getCurrent(), 0, 0, 0));
}
private static Rectangle getEyeBoundaries(Rectangle headBoundaries) {
    int headMiddle = headBoundaries.x + headBoundaries.height / 2;
    int y = headBoundaries.y + headBoundaries.height / 8;
    int width = headBoundaries.width / 16;
    int height = headBoundaries.height / 4;
    return new Rectangle(headMiddle, y, width, height);
}
private static void drawLeftEye(GC gc, Rectangle headBoundaries,
        Rectangle eyeBoundaries) {
    int leftEyeX = eyeBoundaries.x - headBoundaries.width / 8;
    gc.fillOval(leftEyeX, eyeBoundaries.y, eyeBoundaries.width,
            eyeBoundaries.height);
}
private static void drawRightEye(GC gc, Rectangle headBoundaries,
        Rectangle eyeBoundaries) {
    int rightEyeX = eyeBoundaries.x + headBoundaries.width / 8;
    gc.fillOval(rightEyeX, eyeBoundaries.y, eyeBoundaries.width,
            eyeBoundaries.height);
}
private static void drawMouth(Rectangle boundaries, GC gc) {
    gc.drawArc(boundaries.x, boundaries.y - boundaries.height / 4,
            boundaries.width, boundaries.height, 225, 90);
}
```

```
//draw paragraph top in the current page and paragraph bottom in the next
int height = paragraphBottom - paintY;
gc.setClipping(clientArea.x, paintY, clientArea.width, height);
printLine(paintX, paintY, gc, foreground, lineBackground, layout, printLayout, i);
gc.setClipping((Rectangle)null);
printDecoration(page, false, printLayout);
printer.endPage();
page++;
if (page <= endPage) {
    printer.startPage();
    printDecoration(page, true, printLayout);
    paintY = clientArea.y - height;
    int layoutHeight = layout.getBounds().height;
    gc.setClipping(clientArea.x, clientArea.y, clientArea.width, layoutHeight - height);
    printLine(paintX, paintY, gc, foreground, lineBackground, layout, printLayout, i);
    gc.setClipping((Rectangle)null);
    paintY += layoutHeight;
}
```

OU

```
int height = drawParagraphTop(paragraphBottom, paintX, paintY, page);
if (page + 1 <= endPage)
    drawParagraphEnd(height, page + 1);
```

Revele suas intenções

```
public class C {
    private boolean[] p;

    public C(int m) {
        p = new boolean[m];
        p[0] = p[1] = false;
        for (int n = 2; n < m; n++)
            p[n] = true;

        for (int n = 2; n < m; n++)
            if (p[n])
                for (int f = 2; n * f < p.length; f++)
                    p[n * f] = false;
    }

    public boolean p(int n) {
        return n > 0 && n < p.length && p[n];
    }
}
```

```java
public class CalculadoraDePrimos {
    private boolean[] ehPrimo;

    public CalculadoraDePrimos(int numeroMaximo) {
        ehPrimo = new boolean[numeroMaximo];
        ehPrimo[0] = ehPrimo[1] = false;
        for (int numero = 2; numero < numeroMaximo; numero++)
            ehPrimo[numero] = true;

        for (int numero = 2; numero < numeroMaximo; numero++)
            if (ehPrimo[numero])
                for (int fator = 2; numero * fator < ehPrimo.length; fator++)
                    ehPrimo[numero * fator] = false;
    }

    public boolean ehPrimo(int numero) {
        return numero > 0 && numero < ehPrimo.length && ehPrimo[numero];
    }
}
```

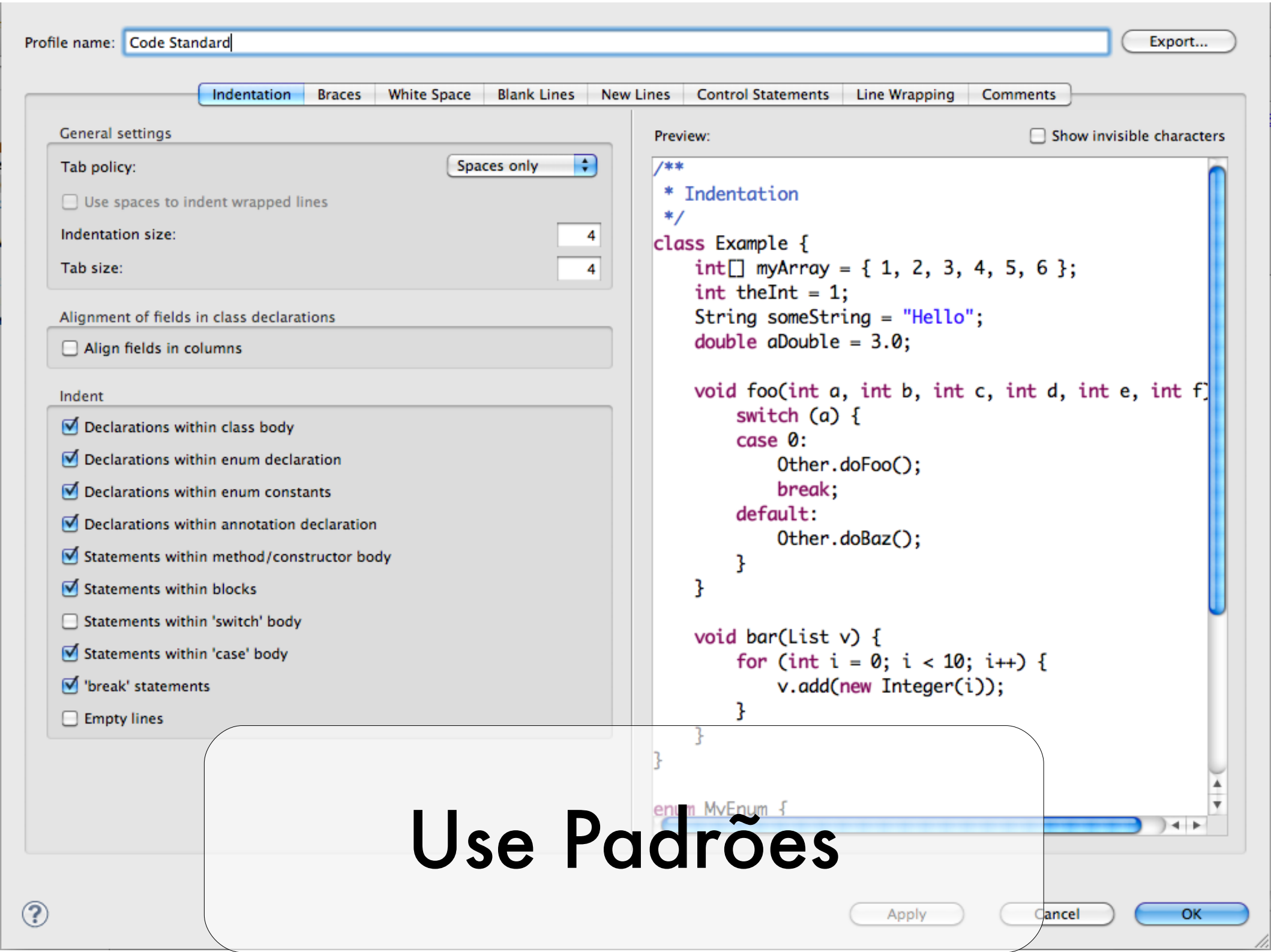
```
StyledTextContent content = printerRenderer.content;
startLine = 0;
endLine = singleLine ? 0 : content.getLineCount() - 1;
PrinterData data = printer.getPrinterData();
if (data.scope == PrinterData.PAGE_RANGE) {
    int pageSize = clientArea.height / lineHeight;//WRONG
    startLine = (startPage - 1) * pageSize;
} else if (data.scope == PrinterData.SELECTION) {
    startLine = content.getLineAtOffset(selection.x);
    if (selection.y > 0) {
        endLine = content.getLineAtOffset(selection.x + selection.y - 1);
    } else {
        endLine = startLine - 1;
    }
}
```

WTF?!?

Proteja seu código

```
DeviceHandle handler = getHandle(DEV1);
if(handler != DeviceHandle.INVALID)
    saveStatus(handler);
else
    logError(handler);
```

```
try {
    Method method = getClass().getMethod(methodName);
    method.invoke(this, methodName);
} catch (SecurityException e) {
    e.printStackTrace();
} catch (NoSuchMethodException e) {
    e.printStackTrace();
} catch (IllegalArgumentException e) {
    e.printStackTrace();
} catch (IllegalAccessException e) {
    e.printStackTrace();
} catch (InvocationTargetException e) {
    e.printStackTrace();
}
```


Profile name: Code Standard
Export...
Indentation
Braces
White Space
Blank Lines
New Lines
Control Statements
Line Wrapping
Comments
General settings
Tab policy:
Spaces only
Use spaces to indent wrapped lines
Indentation size: 4
Tab size: 4
Alignment of fields in class declarations
Align fields in columns
Indent
Declarations within class body
Declarations within enum declaration
Declarations within enum constants
Declarations within annotation declaration
Statements within method/constructor body
Statements within blocks
Statements within 'switch' body
Statements within 'case' body
'break' statements
Empty lines
Preview:
Show invisible characters
/**
* Indentation
*/
class Example {
int[] myArray = { 1, 2, 3, 4, 5, 6 };
int theInt = 1;
String someString = "Hello";
double aDouble = 3.0;
void foo(int a, int b, int c, int d, int e, int f
switch (a) {
case 0:
Other.doFoo();
break;
default:
Other.doBaz();
}
}
void bar(List v) {
for (int i = 0; i < 10; i++) {
v.add(new Integer(i));
}
}
}
enum MyEnum {
Use Padrões
Apply
Cancel
OK

```
public class CalculadoraDePrimos {
private boolean[] ehPrimo;
public CalculadoraDePrimos(int numeroMaximo) {
    ehPrimo = new boolean[numeroMaximo];
        ehPrimo[0] = ehPrimo[1] = false;
for (int numero = 2; numero < numeroMaximo; numero++)
ehPrimo[numero] = true;
        for (int numero = 2; numero < numeroMaximo; numero++)
if (ehPrimo[numero])
for (int fator = 2; numero * fator < ehPrimo.length; fator++)
                    ehPrimo[numero * fator] = false;}public boolean ehPrimo(int numero) {
                        return numero > 0 && numero < ehPrimo.length && ehPrimo[numero];
}}
```

```java
public class CalculadoraDePrimos {
    private boolean[] ehPrimo;

    public CalculadoraDePrimos(int numeroMaximo) {
        ehPrimo = new boolean[numeroMaximo];
        ehPrimo[0] = ehPrimo[1] = false;
        for (int numero = 2; numero < numeroMaximo; numero++)
            ehPrimo[numero] = true;

        for (int numero = 2; numero < numeroMaximo; numero++)
            if (ehPrimo[numero])
                for (int fator = 2; numero * fator < ehPrimo.length; fator++)
                    ehPrimo[numero * fator] = false;
    }

    public boolean ehPrimo(int numero) {
        return numero > 0 && numero < ehPrimo.length && ehPrimo[numero];
    }
}
```

Regra do Escoteiro


Robert C. Martin Series
PRENTICE HALL
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin
ARE YOU A PROFESSIONAL?
Robert C. Martin

agile tour
2010

Perguntas?
Críticas?
Comentários?

hugo@agilbits.com.br
@hugocorbucci