

Desenvolvimento Ágil Orientado a Testes

Uma história real de como a busca pela qualidade de software transformou uma equipe de desenvolvimento

Eder Ignatowicz

Eder Ignatowicz...



@ederign



ALL CODE IS GUILTY
UNTIL PROVEN INNOCENT

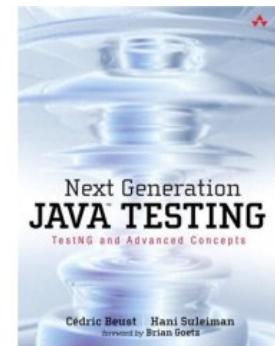
JU•org
nit



ALL CODE IS GUILTY
UNTIL PROVEN INNOCENT



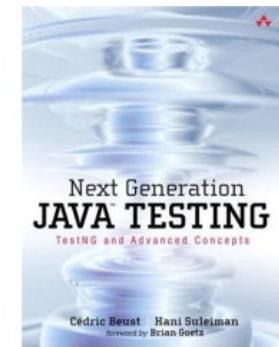
**ALL CODE IS GUILTY
UNTIL PROVEN INNOCENT**





ALL CODE IS GUILTY
UNTIL PROVEN INNOCENT

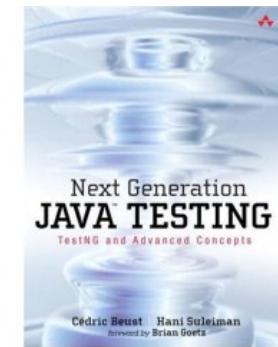
BDD





ALL CODE IS GUILTY
UNTIL PROVEN INNOCENT

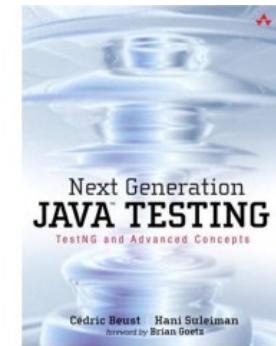
BDD



Jbehave



ALL CODE IS GUILTY
UNTIL PROVEN INNOCENT



BDD

Jbehave

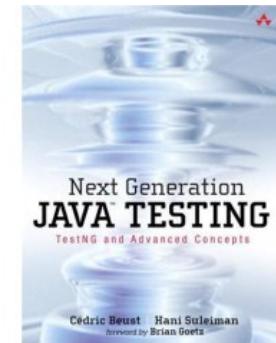


ALL CODE IS GUILTY
UNTIL PROVEN INNOCENT

BDD



Jbehave



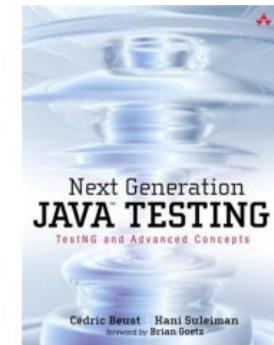


ALL CODE IS GUILTY
UNTIL PROVEN INNOCENT

BDD



Jbehave







O que te motiva no trabalho ?



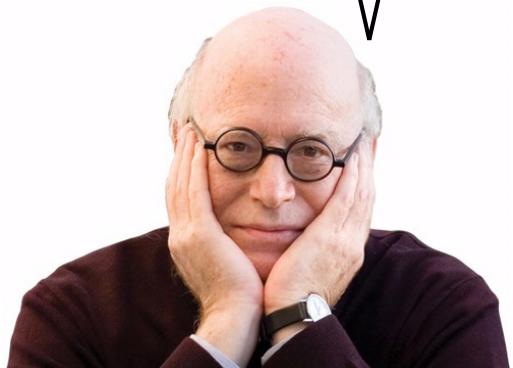


Ou um sentido de realização, um senso
de propósito, um senso de **orgulho** ?

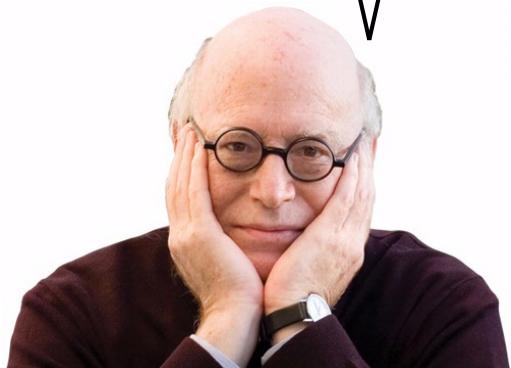
Segundo o sociólogo e historiador norte-americano Richard Sennett, autor de "L'uomo artigiano" (O homem artesão)...



É preciso restabelecer o "conceito de **ofício**, sobre o princípio do **artesão**: valorizar o **significado** do **trabalho**, em vez da remuneração que se pode esperar dele. Reabilitar o conceito de trabalho **bem desenvolvido** pelo simples **prazer de desenvolvê-lo bem**, independentemente do conceito de performance ou de retribuição. Apenas esse compromisso desinteressado dá **significado à vida**".



É preciso restabelecer o "conceito de **ofício**, sobre o princípio do **artesão**: valorizar o **significado** do **trabalho**, em vez da remuneração que se pode esperar dele. Reabilitar o conceito de trabalho **bem desenvolvido** pelo simples **prazer de desenvolvê-lo bem**, independentemente do conceito de performance ou de retribuição. Apenas esse compromisso desinteressado dá **significado à vida**".



E no desenvolvimento de Software ?

Manifesto for Software Craftsmanship

Raising the bar.

As aspiring Software Craftsmen we are raising the bar of professional software development by practicing it and helping others learn the craft. Through this work we have come to value:

Not only working software,
but also **well-crafted software**

Not only responding to change,
but also **steadily adding value**

Not only individuals and interactions,
but also **a community of professionals**

Not only customer collaboration,
but also **productive partnerships**

That is, in pursuit of the items on the left we have found the items on the right to be indispensable.

What happens when a psychologist's patients
run out of patience?

Self-Help FOR DUMMIES

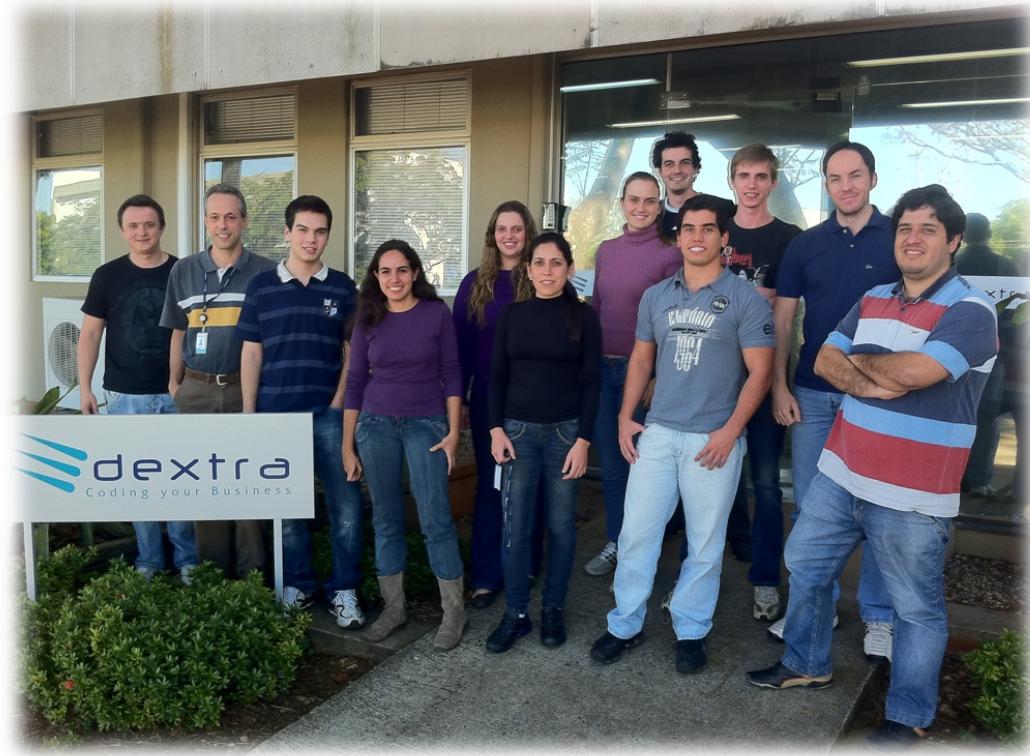
The insanely hilarious new comedy by Award-Winning Playwrights
T. Gregory Argall
and
Todd McGinnis



Mas e a minha história.... ?

Tinhamos um grande (e longo) projeto para um cliente bancário...

E uma equipe competente...



Utilizávamos Scrum e grande parte dos preceitos ágeis...



Mas a cada mês o projeto crescia e surgiam problemas cada vez maiores (como em todos grandes projetos)....

e o incômodo ficava cada vez maior...



... dentre estes incômodos, o principal
era...

Nós podemos mais !

Precisamos construir (e garantir) uma maior qualidade do nosso trabalho...

Numa “calma” reunião técnica...



**Escutamos uns aos outros e
decidimos utilizar:**

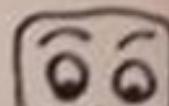
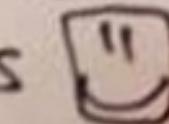
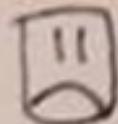
TDD e BDD para as regras de
negócio:

E como fazemos isto ?



TDD

- **RED**: test fails
- **GREEN**: test passes
- **REFACTOR**



Para mim (e para muita gente)...

TDD **não** se refere a testes...

TDD se refere a ***DESIGN*** de código.

Nunca escreva uma única linha de código a menos que falhe em um teste automatizado(**RED**). Então escreva o mínimo de código para o teste passar (**GREEN**) e depois elimine a duplicação. (**REFACTOR**)



Kent Beck em um ângulo não muito favorável ;)

**Escreveremos testes unitários para todas
as nossas classes.**



```
@Test
public void testParse() throws IOException, XmlParserException, SoaSacsWsException {
    InputStream input = XmlSoaSacsExtratoParser.class.getClassLoader().getResourceAsStream("extrato.xml");
    StringWriter writer = new StringWriter();

    IOUtils.copy(input, writer);

    String xml = writer.toString();

    XmlSoaSacsExtratoParser parser = new XmlSoaSacsExtratoParser();

    List<Map<String, String>> resultado = new ArrayList<Map<String, String>>();
    parser.parseXmlExtrato(xml, resultado);

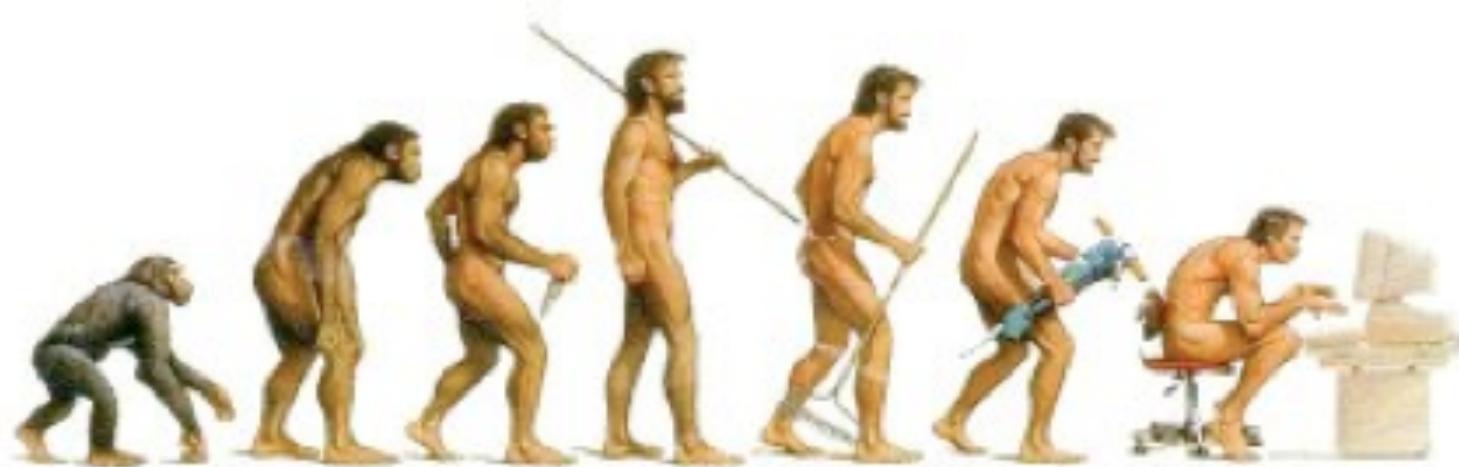
    Assert.assertEquals(4, resultado.size());

    Assert.assertEquals("20100616", resultado.get(0).get("DATA_TX"));
    Assert.assertEquals("0", resultado.get(0).get("HORA_TX"));
    Assert.assertEquals("TARIFA SMS -", resultado.get(0).get("TIPO_TRANSACAO"));
    Assert.assertEquals("0", resultado.get(0).get("VALOR_MOEDA_ORIGEM"));
    Assert.assertEquals("-", resultado.get(0).get("TIPO_OPERACAO"));

    Assert.assertEquals("20100616", resultado.get(1).get("DATA_TX"));
    Assert.assertEquals("154412", resultado.get(1).get("HORA_TX"));
    Assert.assertEquals("COMPRA A VISTA INTER - SNACK CAR", resultado.get(1).get("TIPO_TRANSACAO"));
    Assert.assertEquals("6,4", resultado.get(1).get("VALOR_MOEDA_ORIGEM"));
    Assert.assertEquals("-", resultado.get(1).get("TIPO_OPERACAO"));
```

BDD

Evolution



Behavior **D**riven **D**evelopment pode ser visto como uma **técnica** de desenvolvimento **ágil** que **encoraja colaboração** entre **desenvolvedores, analistas, QA e o pessoal não técnico (stackholders)** para o **sucesso** de um projeto.

Behavior **D**riven **D**evelopment pode ser visto como uma **técnica** de desenvolvimento **ágil** que **encoraja** **colaboração** entre **desenvolvedores, analistas, QA** e o **pessoal não técnico (stackholders)** para o **sucesso** de um projeto.

Porque para mim **TODOS** são
desenvolvedores de software!

Multidisciplinaridade

Penso que **BDD** pode por fim a
“**GUERRA**” entre QA e
desenvolvedores (aka programadores)...

Porque a guerra acontece ?



“Testers” e “Programadores” não falam a mesma língua

E com BDD conseguimos...



**Conversar no mesmo idioma (o do
negócio)**

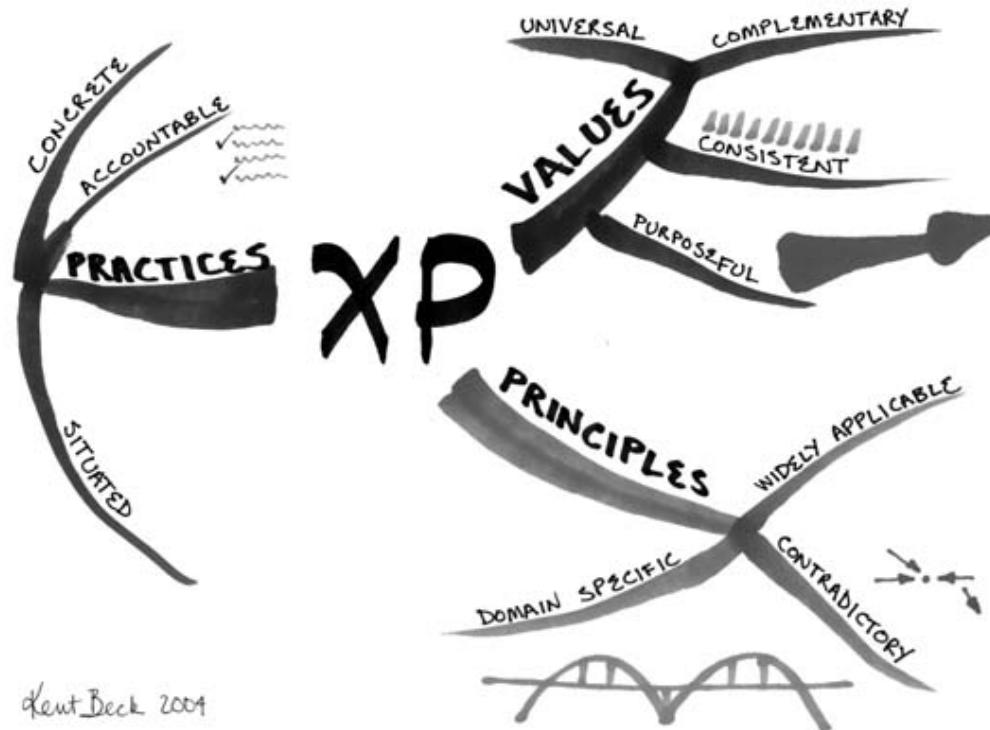
E como fazemos isto ?



Todos membros do time escrevem as histórias colaborativamente nas reuniões de planning...



Cenario: Transferencia CentroCusto Cartao/Cartao Centro custo com aprovadores
Dado que o centro de custo Dolar do cliente Dextra tem saldo disponivel de U\$ 20.000,00
Quando eh criada operacao para o centro de custo Dolar do cliente Dextra
Com transferencia para atualizar o saldo do cartao 0000-0000-0000-2000 para U\$ 1.500,00
Quando o centro de custo Dolar do cliente Dextra necessita aprovacao de 2 aprovadores
Quando a operacao TransferenciasCentroCustoCartoes nao eh executada pois necessita aprovacao
Entao o saldo do cartao 0000-0000-0000-2000 eh U\$ 500,00
Quando a operacao eh aprovada uma vez pelo Eder de id 1 logado no centro de custo de id 1
Entao o saldo do cartao 0000-0000-0000-2000 eh U\$ 500,00
Quando a operacao eh aprovada uma vez pelo Luis de id 2 logado no centro de custo de id 1
Entao o saldo do cartao 0000-0000-0000-2000 eh U\$ 1.500,00



Collective Ownership entre desenvolvedores (aka testers + programadores)

(Demo BDD/JBehave)

Também decidimos...

ATDD (ou especificação por exemplo) para **TODOS** os fluxos principais do negócio

The screenshot shows a Java IDE interface with several windows:

- Top Left:** Shows a JUnit runner window with 1 run, 0 errors, and 1 failure.
- Top Right:** Shows a file tree with packages like `pedido_preenchido`, `TestCasesMock`, `src`, and `tests.behave.java.estoquedisponivel`.
- Middle:** A code editor window showing a Java class named `PedidoPreenchido` which extends `Scenario`. It contains a single method `PedidoPreenchido()` with a step definition `addstepa(new EstoqueDisponivelSteps());`.
- Bottom:** A `Console` window displaying the execution of the test. The narrative describes a user filling a pedido to finalize a venda. It includes a scenario for checking product balance and a pending step for verifying stock availability.



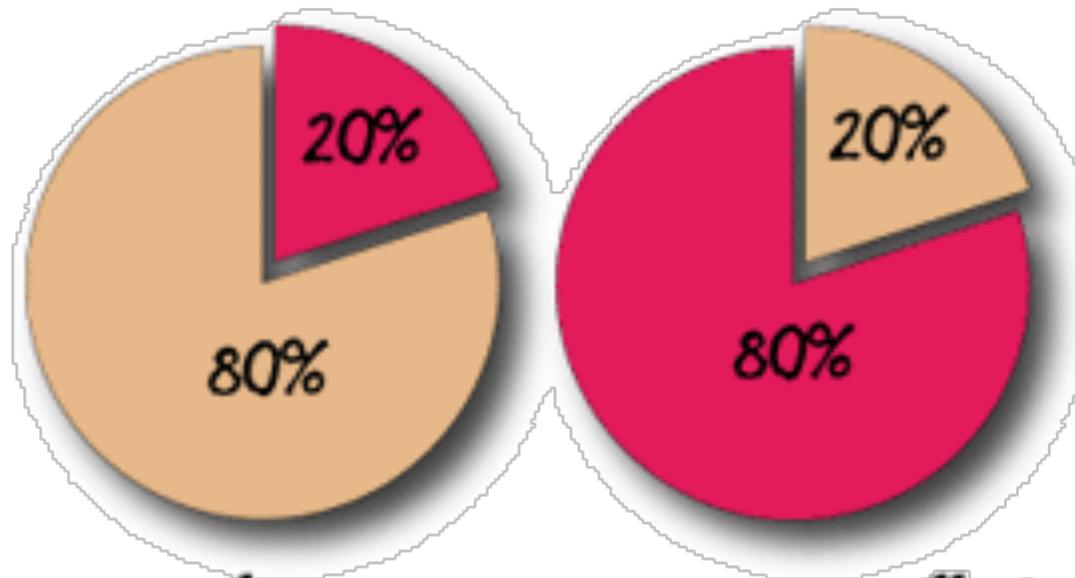
E como fazemos isto ?



**Todos membros do time
colaborativamente selecionam as
principais histórias nas reuniões de
planning...**

Nem todas histórias viram testes de
aceitação...

The Pareto Principle



20% of input (time, resources, effort)
accounts for 80% of output (results, rewards)

Testes **precisam** ser organizados e
fáceis de escrever e **manter...**

Através da Selenium IDE / Badboy ?

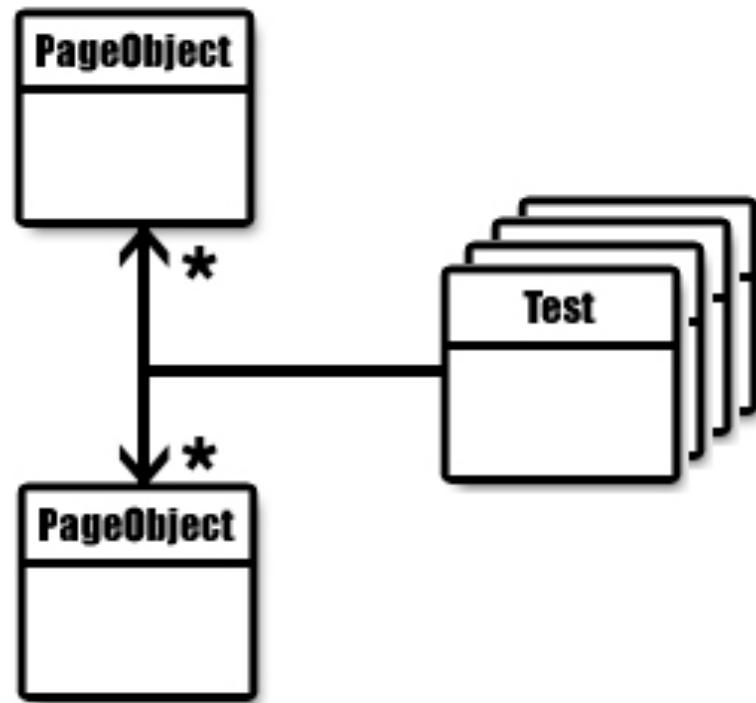


A promessa...



A realidade...

Que tal Page Objects ?





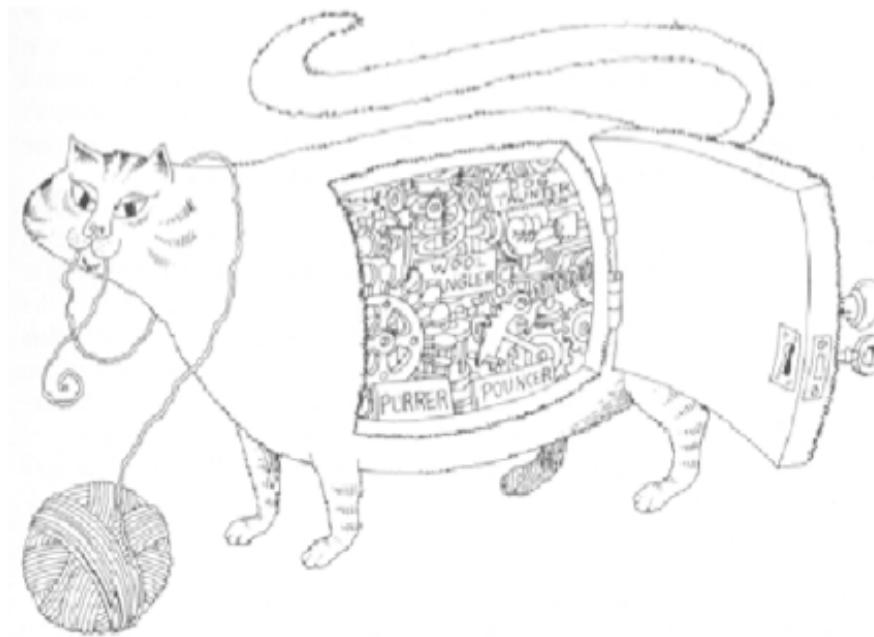
Reusáveis

**LOW
MAINTENANCE
CONFIRMATION**

No Warranty Required.



Baixa manutenção



**Escondem detalhes desnecessários
através de Orientação a Objetos**
(tornando mais fácil o seu uso)

E como fazemos isto ?





Encerrar sessão

Operador: Julio
Loja: ITALIA CURITIBA

Segurança ▾ Cadastros ▾ Tesouraria ▾ Banco ▾

Cadastro de Pessoas Físicas - Banco

CPF Passaporte/CNE

Situação



Auxílio de Preenchimento

Ao selecionar os campos para preenchimento, deixamos aqui algumas dicas para tornar sua tarefa mais fácil e rápida.

Filtrar

Novo



Encerrar sessão

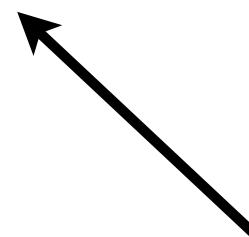
Operador: Julio
Loja: ITALIA CURITIBA

Segurança ▾ Cadastros ▾ Tesouraria ▾ Banco ▾

Cadastro de Pessoas Físicas - Banco

CPF Passaporte/CNE

Situação



Auxílio de Preenchimento

Ao selecionar os campos para preenchimento, deixamos aqui algumas dicas para tornar sua tarefa mais fácil e rápida.

Filtrar

Novo

```
public class PaginaConsultarPessoaFisicaBanco extends PaginaBase {  
  
    public void consultaPorCpf(String cpf) {  
        selecionaInputRadio("rbCPF");  
        preencheInputText("filtroCPF", cpf);  
  
        click("buscaPessoasFisicas");  
        waitForPageToLoad();  
    }  
  
    public PaginaManterPessoaFisicaBanco acessarPaginaInclusaoPessoaFisica()  
        getDriver().findElement(By.id("btnAdicionarPessoaFisica")).click();  
        waitForPageToLoad();  
        return new PaginaManterPessoaFisicaBanco();  
    }  
}
```

Animei, quero codificar meus testes...

Mas para preencher um simples
Suggestion Box Ajax no Selenium...

```
public PaginaBase preencheSuggestion(String idElemento, String valorDoElemento, String elementoASelecionar) {
    WebElement element = getDriver().findElement(By.id(idElemento));
    element.click();
    if (StringUtils.isNotEmpty(element.getValue())) {
        try {
            element.clear();
            element.sendKeys("a");
            waitForPageToLoad(1000);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    try {
        element.clear();
        element.sendKeys(valorDoElemento);
        waitForPageToLoad(1000);
    } catch (Exception e) {
        e.printStackTrace();
        getDriver().findElement(By.cssSelector(".ac_results ." + elementoASelecionar)).click();
    }
    return this;
}
```



Eu como **QA**, quero (e devo) me
preocupar com o negócio.

O meu **FOCO** é outro.

Mas...

Eu te afirmo...

Com uma API caprichada toda a
equipe é capaz de gostar de
automatizar (programar) testes
funcionais !!!

Cadastro de Pessoas Físicas > Nova Pessoa Física - Banco

Tipo	Geral	Endereço	Documento	Compliance	Banco	Relacionamento	Patrimônio	Renda	Questionário
------	--------------	----------	-----------	------------	-------	----------------	------------	-------	--------------

* Loja Matriz

Nacionalidade: Brasileiro Estrangeiro

* CPF

* Nome

* Nome reduzido

Situação: Parcial

* Tipo de documento RG Passaporte

* Número do RG

* Órgão Emissor

* Data de emissão

 **Auxílio de Preenchimento**
Nome da pessoa física (ex.: Ana Maria)

```
public void preencherAbaDadosGeraisObrigatoriosBrasileiro(String cpf, String nome,  
    TipoDocumento tipoDoc, String docIdentificacao, String orgaoEmissor, Date dat  
    Canal canal) {  
    click("abaDadosGerais");  
    waitForPageToLoad();  
  
    selecionaInputRadio(Nacionalidade.BRASILEIRO.getTipo());  
    preencheInputText("cpf", cpf);  
    preencheInputText("nome", nome);  
    preencheInputText("nomeReduzido", nomeReduzido);  
    selecionaInputRadio(tipoDoc.getTipo());  
    preencheSuggestion("indicacao", indicacao, "ac_even");  
    waitForPageToLoad();  
    selecionaComboBox("canal", canal.getId().toString()));  
}
```



Todo o time programando **testes !**

```
/**  
 * Dado que tenha usuario logado  
 * Quando usuario acessa pagina de inclusao de pessoa fisica no banco  
 * brasileiro  
 * Quando usuario acessa a aba "Geral" para o preenchimento dos dados  
 * obrigatorios do brasileiro  
 * Quando usuario acessa a aba "Endereco" para o preenchimento do  
 * endereco do brasileiro  
 * Quando usuario acessa aba "Questionario" para o preenchimento  
 * Quando usuario submete o cadastro para a analise  
 * Quando usuario salva as pendencias no banco  
 * Entao verifico que existe brasileiro com o CPF informado e com estado  
 * "Em Analise"  
 */
```

```
@Test
public void cadastrarBrasileiroNoBancoComDadosObrigatoriosTest() {
    dadoQueUsuarioEstaLogado("julio.banco", "julio.banco");
    quandoUsuarioAcessaPaginaDeInclusaoDePessoaFisicaNoBanco();
    quandoUsuarioPreencheTipoTitularPessoaFisicaNoBanco(TipoTitular.BANQUEIRO);
    quandoUsuarioPreencheDadosDeBrasileiroObrigatoriosNoBanco("801.596.134-34",
        "Joao da Silva", "Joao", TipoDocumento.PASSAPORTE_BRASILEIRO,
        "123456789", "Orgao emissor teste", new Date(),
        "Padrao Confidence", Canal.NETBANKING);
    quandoUsuarioPreencheDadosDoEnderecoNoBrasilObrigatoriosNoBanco("89456000",
        TipoEndereco.Outros, "Rua Tres", "100", Estado.SP, "Sao Paulo");
    quandoUsuarioPreencheQuestionarioPessoaFisicaNoBanco();
    quandoUsuarioSalvaPessoaFisicaNoBanco();
    quandoUsuarioSalvaPendenciasPessoaFisicaNoBanco();
    entaoExistePessoaEmAnaliseNoBancoComCpf("801.596.134-34");
}
```

```
// DADO QUE...
public void dadoQueUsuarioEstaLogado(String usuario, String senha) {
    navegarNaPaginaPadrao();
    PaginaLoginBackOffice paginaLogin = new PaginaLoginBackOffice();
    this.paginaInicial = paginaLogin.logar(usuario, senha);
    Assert.assertTrue(paginaLogin.estaLogado());
}
```



TODOS somos **desenvolvedores**
de software!

(Selenium + Page Objects +
BDD Demo)

E finalmente...

O Jenkins e o Sonar cuidando para que tudo se mantenha em ordem...



E quais foram os resultados ?

Para o cliente:



- Produto funcionando;
- Deploy “contínuo” em um sistema **bancário**;
- Velocidade e adaptação as mudanças;
- Confiança na equipe.

Para a equipe:



- Produto funcionando;
- Maximização da qualidade;
- Confiança na modificação de funcionalidades;
- Sono tranquilo e....



Eder Ignatowicz...



@ederign