

UNIVERSIDADE REGIONAL DE BLUMENAU  
CURSO DE SISTEMAS DE INFORMAÇÃO - BACHARELADO

# PROTÓTIPO DE FERRAMENTA PARA ANÁLISE DE LOGS DO TESTCOMPLETE

**Albino Marcos de Andrade Junior**

Prof. Jacques Robert Heckmann, Orientador

# Roteiro

- Introdução e Objetivos
- Fundamentação teórica
- Desenvolvimento e Especificação do sistema
- Técnicas e ferramentas utilizadas
- Operacionalidade
- Resultados e Discussões
- Conclusão
- Extensão
- Demonstração do sistema

# Introdução

- Melhorar a análise e comparação dos *TestLogs*.
- Importar as informações dos *TestLogs*.
- Visualizar várias execuções.
- Gerar gráficos.

# Objetivos

**Objetivo geral:** disponibilizar um protótipo que auxilie a análise e comparação de logs gerados pelas execuções dos testes através da ferramenta *TestComplete*.

## **Objetivos específicos:**

- desenvolver ao gestor da qualidade uma ferramenta para auxiliar a gestão da sua equipe de testes;
- conhecer os detalhes dos logs de testes, padrão e informações geradas;
- comparar distintas execuções dos testes;
- compor base histórica dos testes.

# Área de Testes

- Maioria das empresas realizam testes dentro do processo de desenvolvimento.
- Raramente estão livre de defeitos.
- Criar área de testes com equipe de especialistas.

# Automação de testes

- De acordo com Bartié et. al. (2002, p.63), “automação de testes é a utilização de ferramentas que simulem usuários ou atividades humanas de forma a não empregar procedimentos manuais no processo de execução dos testes.”

# Automação de testes

## Conceitos:

- gravador ou *recorder*: registrar todas as ações do usuário, gerando *script* de teste;
- *script de teste*: conjunto de ações que a ferramenta de automação irá executar;
- executor de teste ou *playback*: recurso que irá executar o *script* de teste gravado;
- *checkpoints*: ação que irá verificar se o *script* de teste foi validado com sucesso.

# *TestComplete*

- Desenvolvida pela AutomatedQA
- Amplo conjunto de funcionalidades:
  - suporte nativo para diversas tecnologias, Java, .Net, Delphi;
  - gravar testes automatizados;
  - permite edição do código gerado;
  - test logs gera registros detalhados;



# TestLogs

- Podem ser visualizados no *TestComplete* um de cada vez.
- Podem ser exportados no formato HTML, XML;
- Permitem que:
  - visualize *screenshots para cada etapa de teste*;
  - adicione textos personalizados;
  - filtre mensagens de log pelo seu texto, tipo, tempo;
  - rever os resumos de um teste automatizado;

# Resumo do Test Log

The screenshot displays a software application window titled "Financieiro 30/06/2015 20:52:57". The window contains a "Log Items" tree view on the left and a "Summary" panel on the right. The tree view shows a hierarchical structure of test items, including "Inicialização", "Procedimentos Iniciais", "Execução", and "Finalização". The "Execução" folder is expanded, showing several test cases like "CT\_Cadastro\_Portadores\_01" and "CT\_Cadastro\_TiposVencimento\_01", each with its own "Script Test Log" sub-item. The "Summary" panel provides a high-level overview of the test results.

**Summary**

**Total number of project test items:** 9 (9)  
Top-level items: 0 (0)

**Executed project test items:** 9 (9)  
Top-level items: 0 (0)

**Project test items executed successfully:** 9 (9)  
Top-level items: 0 (0)

**Failed project test items:** 0 (0)  
Top-level items: 0 (0)

numbers in parentheses include the value of test items' Count property

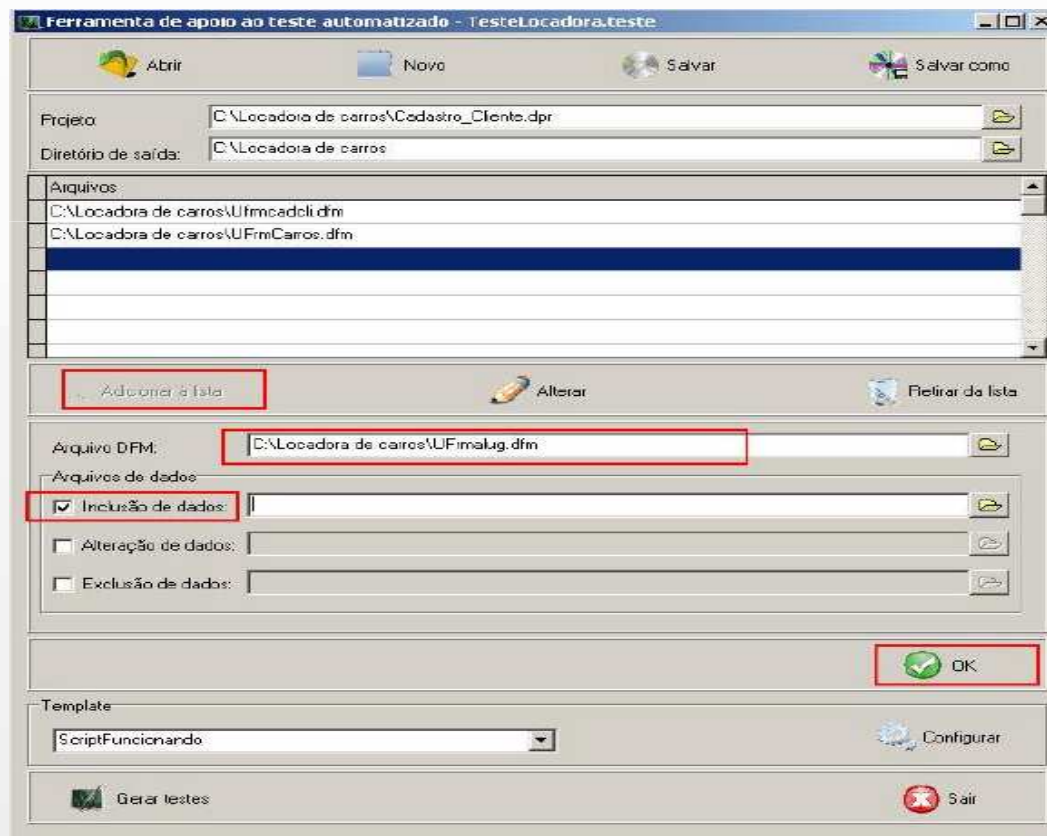
**Information**  
Errors: 0  
Warnings: 1

Start Time: 20:52 30/06/2015  
End Time: 21:18 30/06/2015  
File Name: C:\TC\TCC\_Final\_1\Financieiro\Log\30\_06\_2015\_20\_52\_57\_786\Description.td.log

Summary Project Log

# Trabalhos correlatos

- Marcos (2007): ferramenta para apoio a automatização de testes.



# Trabalhos correlatos

- Waltrick (2011): *plugin* no *TestComplete* que integra-se com a ferramenta CASE *TestLink*.

Suite de Teste : Acadêmico/ Cadastros/Parâmetros Básicos Módulo Acadêmico/ Parâmetros/ Salas/

Caso de Teste ID: GA-2973 :: Versão: 2  
Cadastrar uma Sala (SalaPadrao)  
Atribuído à : adrian

Última execução (baseline qualquer) - Baseline : 01

Data : 27/01/2011 18:46:25 - Testador : adrian - Baseline : 01 - Status : Passou

Última execução (baseline atual) - Baseline : 01

Data	Baseline	Testador	Status	Versão do CT
27/01/2011 18:46:25	01	adrian	Passou	2

Tempo do Teste (em minutos): 1  
Tarefa Vinculada:

Sumário:

- 1) Possuir a base de dados CQ\_PADRAO
- 2) Possuir o executável do módulo Acadêmico
- 3) Possuir liberação do tipo Acesso Completo para Salas(Parâmetros> Salas)

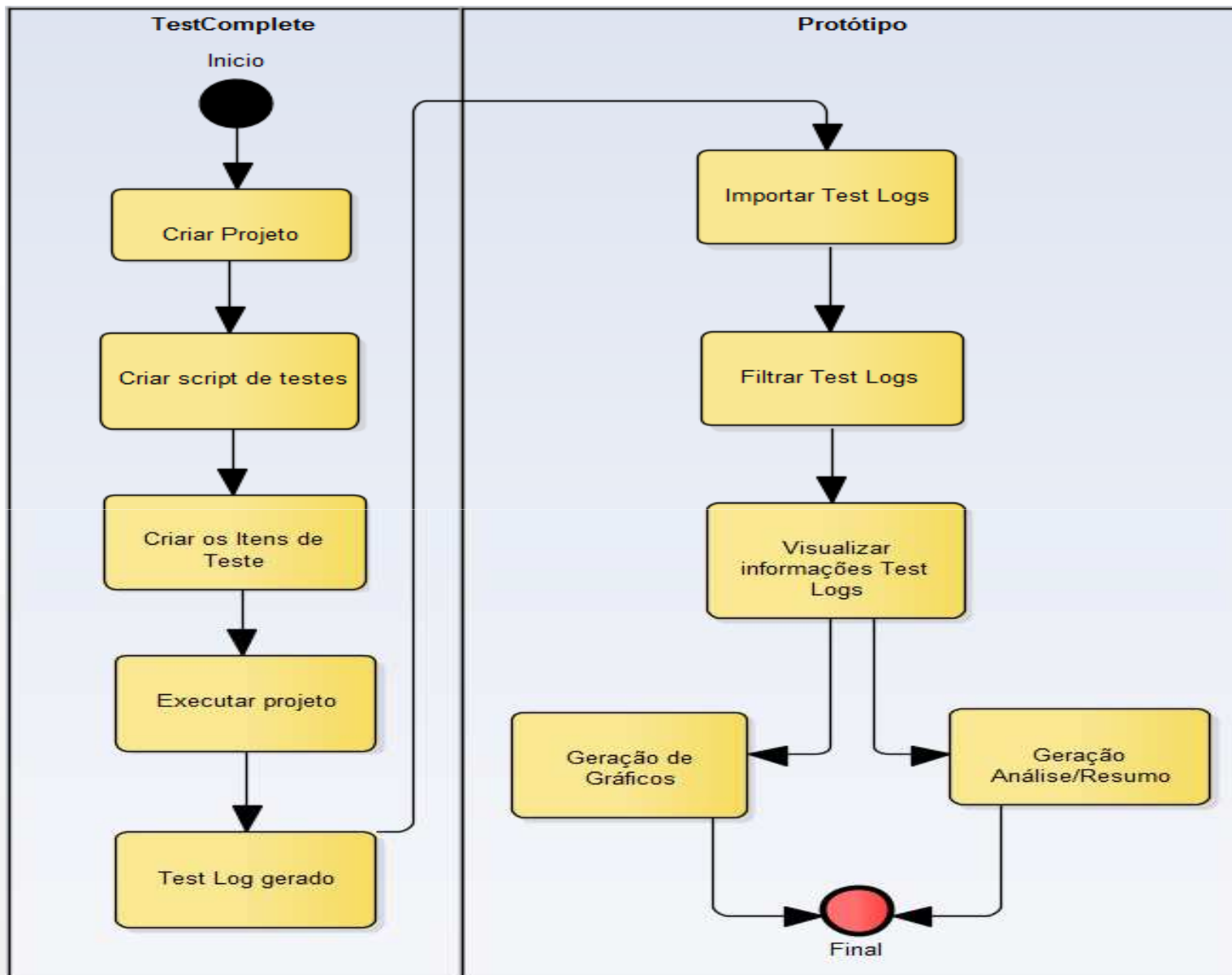
Pré-condições

Tipo de Execução : Manual

# Ações do Passo Resultados Esperados:

# Desenvolvimento e especificação

- Informações obtidas com base nas necessidades de usuários do *TestComplete*.
- Sistema para realizar a importação dos *Test Logs*.
- Visualizar as informações importadas e gerar gráficos para auxiliar.
- Casos de testes devem iniciar com a sigla CT\_.
- Plataforma *desktop*.



# Desenvolvimento e especificação

Requisitos Funcionais	Caso de Uso
RF01: A aplicação deve permitir ao gestor da qualidade/testador informar o caminho onde se encontram os <i>Test Logs</i> a serem importados.	UC01
RF02: A aplicação deve permitir ao gestor da qualidade/testador gerar uma análise dos <i>Test Logs</i> .	UC02
RF03: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com a quantidade de casos de teste por projeto.	UC03
RF04: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com a quantidade de casos de teste por responsável.	UC04
RF05: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com a quantidade total de execuções dos casos de teste.	UC05
RF06: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com a quantidade total de execuções dos casos de teste com erro.	UC06
RF07: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com a quantidade de erros e alertas dos projetos.	UC07

# Desenvolvimento e especificação

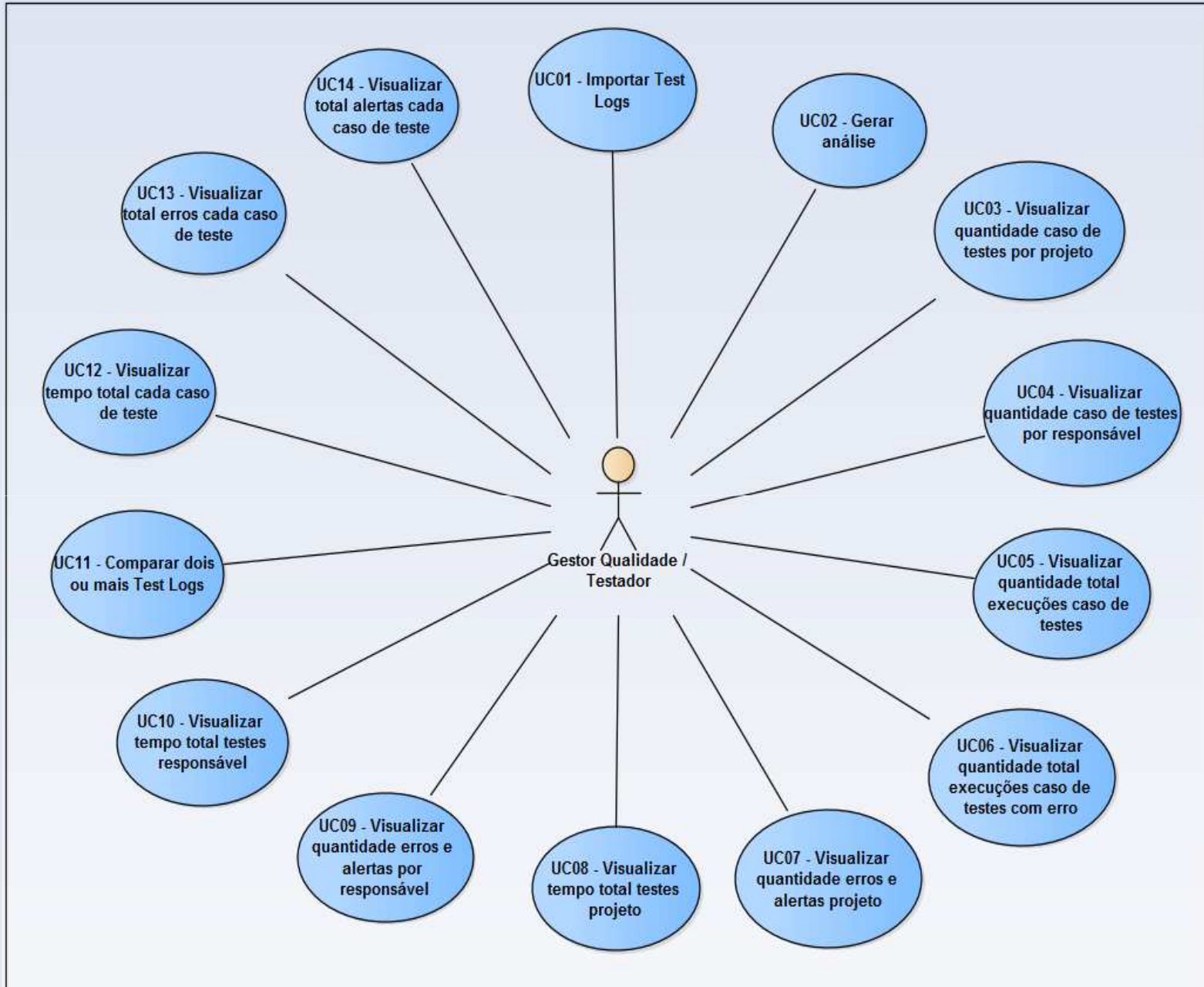
Requisitos Funcionais	Caso de Uso
RF08: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com o tempo total de testes dos projetos.	UC08
RF09: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com a quantidade de erros e alertas por responsável.	UC09
RF10: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com o tempo total de testes por responsável.	UC10
RF11: A aplicação deve permitir ao gestor da qualidade/testador comparar dois ou mais <i>Test Logs</i> do mesmo projeto.	UC11
RF12: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com o tempo total de cada caso de teste.	UC12
RF13: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com a quantidade de erros de cada caso de teste.	UC13
RF14: A aplicação deve permitir ao gestor da qualidade/testador emitir o gráfico com a quantidade de alertas de cada caso de teste.	UC13



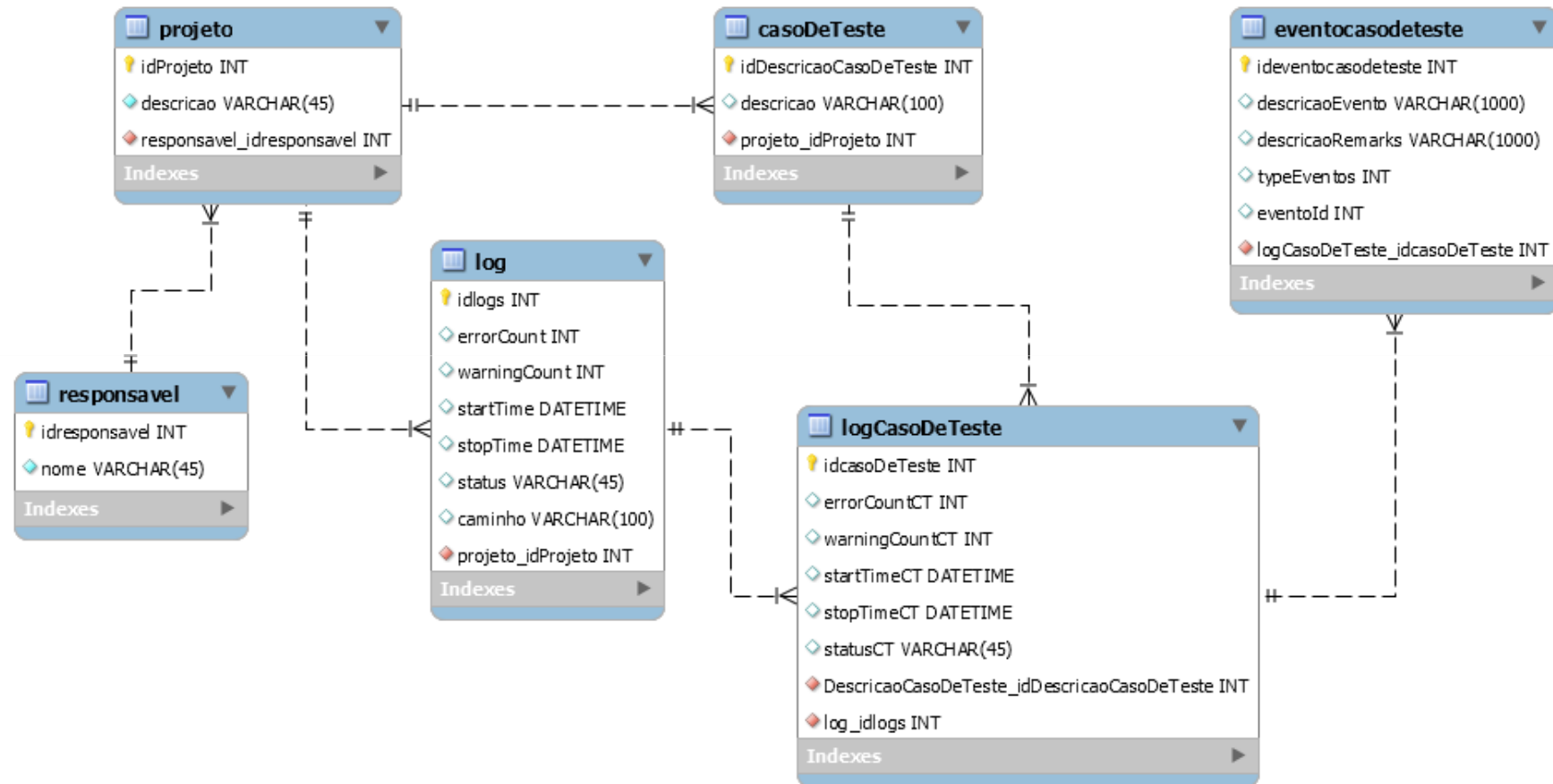
# Desenvolvimento e especificação

- Requisitos não funcionais:

Requisitos Não Funcionais
RNF01: O ambiente de desenvolvimento a ser utilizado será Netbeans.
RNF02: Para análise dos <i>Test Logs</i> será utilizado o método de pesquisa em arquivos por expressões regulares.
RNF03: O protótipo da ferramenta deverá utilizar banco de dados MySQL.
RNF04: O ambiente para geração dos gráficos será o <i>framework JFreeChart</i> .

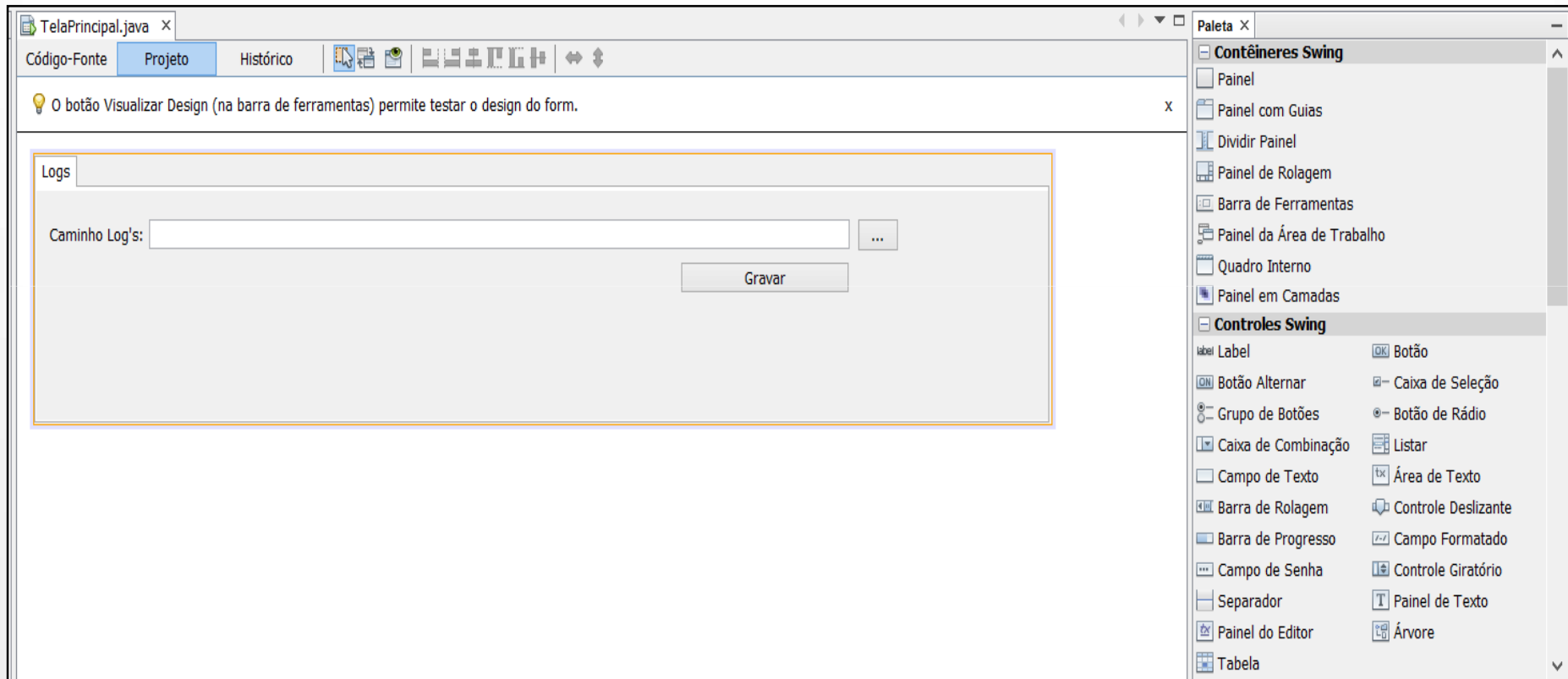


# Desenvolvimento e especificação

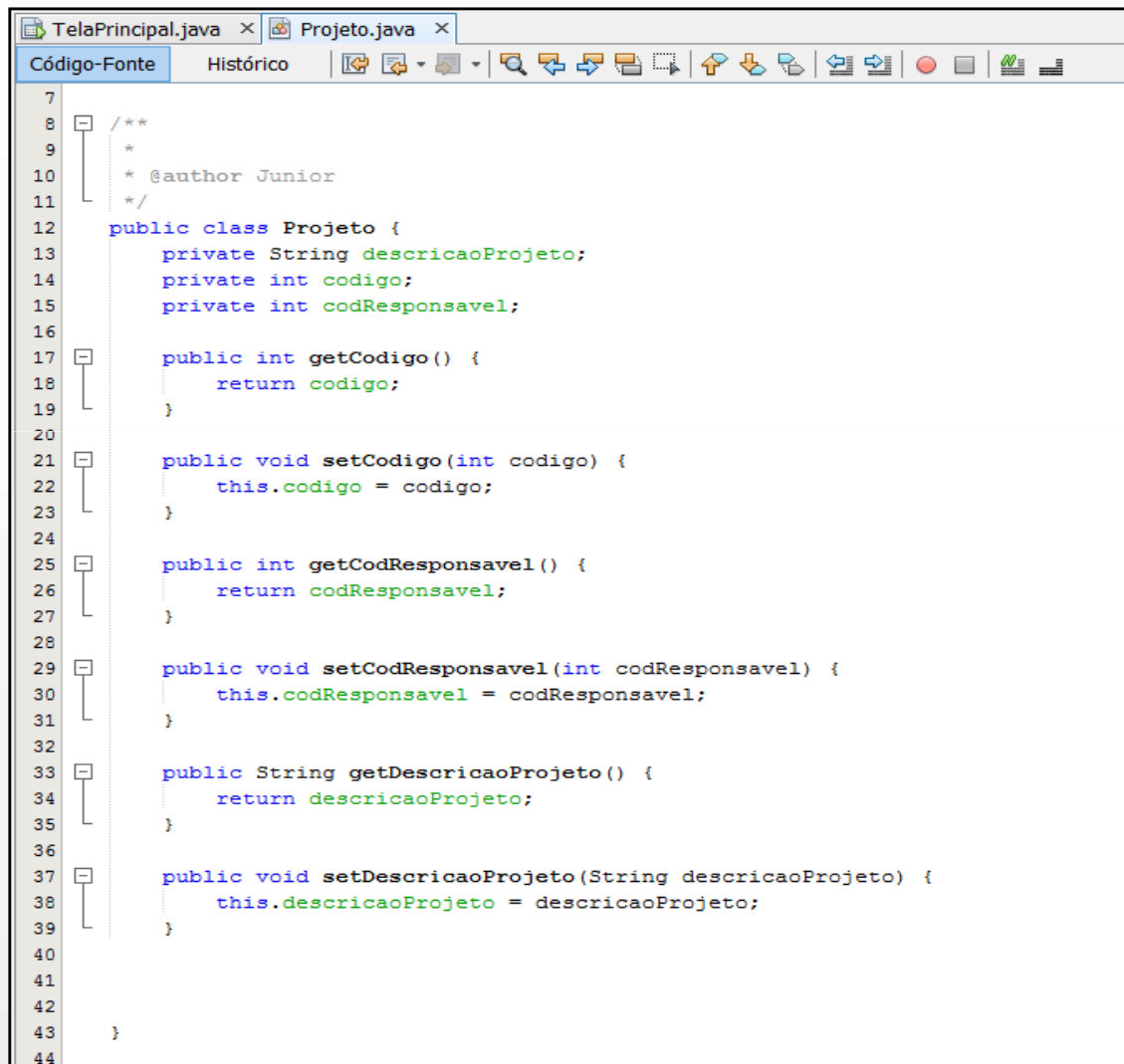


# Técnicas e ferramentas

- Netbeans:



# Técnicas e ferramentas



The image shows a screenshot of an IDE window with two tabs: 'TelaPrincipal.java' and 'Projeto.java'. The 'Código-Fonte' (Source Code) tab is active, displaying the following Java code for the 'Projeto' class:

```
7
8  /**
9   *
10  * @author Junior
11  */
12  public class Projeto {
13      private String descricaoProjeto;
14      private int codigo;
15      private int codResponsavel;
16
17      public int getCodigo() {
18          return codigo;
19      }
20
21      public void setCodigo(int codigo) {
22          this.codigo = codigo;
23      }
24
25      public int getCodResponsavel() {
26          return codResponsavel;
27      }
28
29      public void setCodResponsavel(int codResponsavel) {
30          this.codResponsavel = codResponsavel;
31      }
32
33      public String getDescricaoProjeto() {
34          return descricaoProjeto;
35      }
36
37      public void setDescricaoProjeto(String descricaoProjeto) {
38          this.descricaoProjeto = descricaoProjeto;
39      }
40
41
42
43  }
44
```

# Técnicas e ferramentas

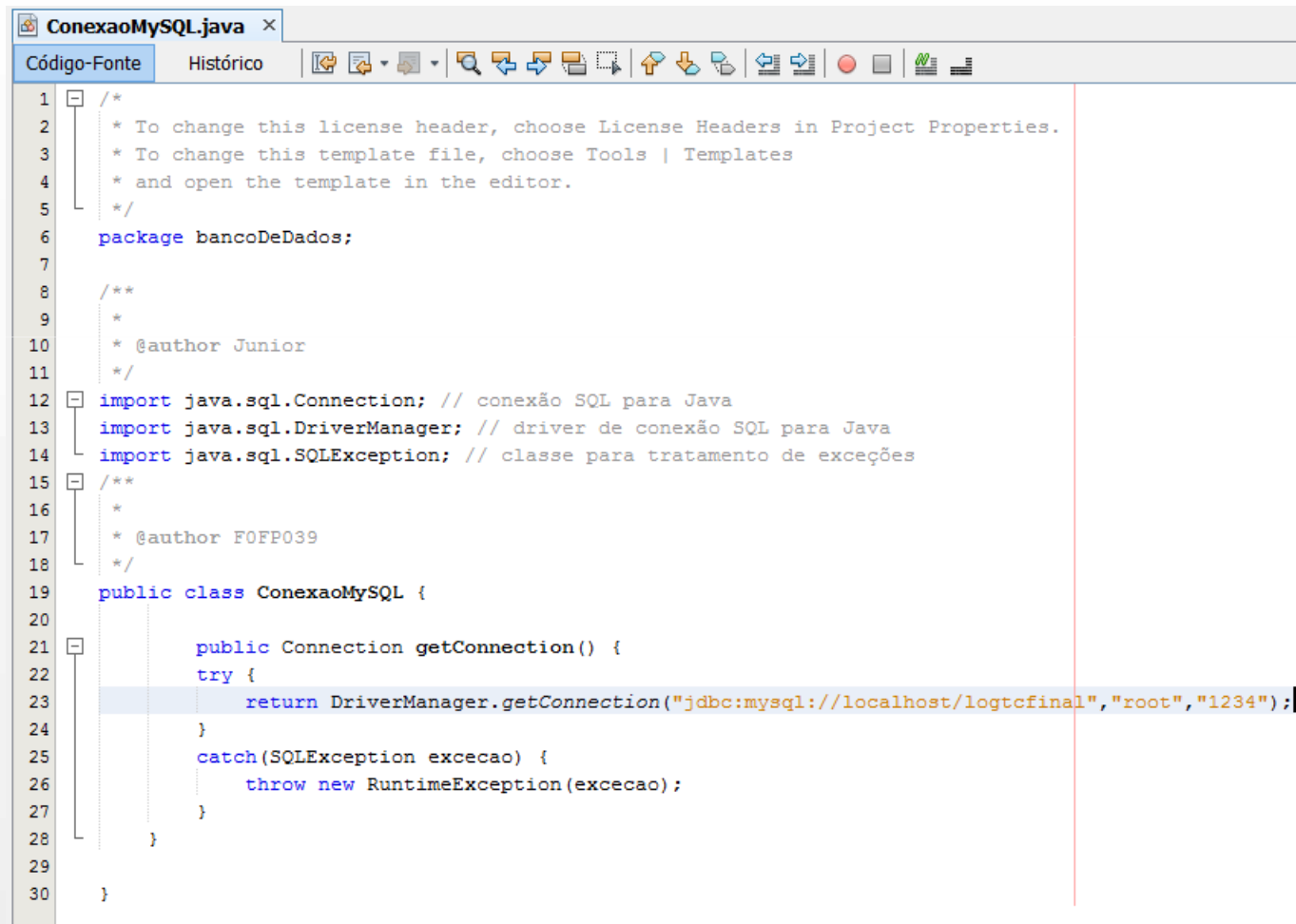
- Expressões regulares:
  - forma de identificar cadeia de caracteres;
  - utilizado na leitura dos arquivos;
  - conhecida como Regex;
  - Para utiliza-las é necessário importar o pacote de classes `java.util.regex`.

# Técnicas e ferramentas

```
TelaPrincipal.java x Projeto.java x LerArquivo.java x
Código-Fonte Histórico
25 public String lerArquivo(Reader input, String inicioPesquisa, String finalPesquisa) throws IOException {
26
27     BufferedReader rd = new BufferedReader(input);
28     String name = null;
29     boolean achouCI = false;
30     START_RE = inicioPesquisa;
31     END_RE = finalPesquisa;
32     String conteudoEncontrado = "";
33     Pattern startPattern = Pattern.compile(START_RE);
34
35     String line;
36     while ((line = rd.readLine()) != null) {
37
38         if (line.trim().isEmpty()) { // ignore linha em branco
39             continue;
40         }
41         if (name == null) { // já achou o inicio do <Node?
42             Matcher start = startPattern.matcher(line); // procure inicio do Node
43             if (start.matches()) {
44
45                 name = start.group();
46             }
47         } else {
48             if (line.matches(END_RE)) {
49                 // fim do nome (linha anterior)
50                 conteudoEncontrado = name;
51                 name = null; // procure proximo
52             } else {
53
54                 name += line;
55             }
56         }
57     }
58     return conteudoEncontrado;
59 }
```

# Técnicas e ferramentas

- MySql



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package bancoDeDados;
7
8  /**
9   *
10  * @author Junior
11  */
12  import java.sql.Connection; // conexão SQL para Java
13  import java.sql.DriverManager; // driver de conexão SQL para Java
14  import java.sql.SQLException; // classe para tratamento de exceções
15  /**
16   *
17   * @author F0FP039
18   */
19  public class ConexaoMySQL {
20
21      public Connection getConnection() {
22          try {
23              return DriverManager.getConnection("jdbc:mysql://localhost/logtcfinal", "root", "1234");
24          }
25          catch(SQLException excecao) {
26              throw new RuntimeException(excecao);
27          }
28      }
29  }
30 }
```



# Técnicas e ferramentas

- *Test Logs*

```
Description.tcLog
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE Nodes [
3
4 <!ENTITY % NameValue "CDATA">
5 <!ENTITY % PropType "(I|S|D|L|H|B)">
6 <!ENTITY % VersionValue "CDATA">
7
8 <!ELEMENT Prp (#PCDATA)>
9 <!ELEMENT Node (Node|Prp)*>
10 <!ELEMENT Nodes (Node)+>
11
12 <!ATTLIST Nodes version CDATA "1">
13
14 <!ATTLIST Node name CDATA #REQUIRED>
15
16 <!ATTLIST Prp name CDATA #REQUIRED>
17 <!ATTLIST Prp type CDATA #REQUIRED>
18 <!ATTLIST Prp value CDATA #REQUIRED>
19 ]>
20 <Nodes version="1">
21   <Node name="root">
22     <Prp name="start time" type="D" value="42156.935309375"/>
23     <Prp name="test type" type="S" value="Orcamento"/>
24     <Prp name="stop time" type="D" value="42156.9421574653"/>
25     <Prp name="error count" type="I" value="0"/>
26     <Prp name="warning count" type="I" value="0"/>
27     <Prp name="root file name" type="S" value="RootLogData.dat"/>
28     <Prp name="root logdata name" type="S" value="Orcamento"/>
29     <Prp name="completed" type="B" value="-1"/>
30     <Prp name="computer name" type="S" value="VM-WIN7"/>
31     <Prp name="status" type="I" value="0"/>
32     <Prp name="user name" type="S" value="Goku"/>
33   </Node>
34 </Nodes>
```

# Técnicas e ferramentas

- *Framework JFreeChart*

```
TotalErrosAlertas.java x
Código-Fonte Histórico
19 /**
20  *
21  * @author Junior
22  */
23 public class TotalErrosAlertas extends JFrame {
24
25     DefaultCategoryDataset dataSetPronto = new DefaultCategoryDataset();
26
27     public TotalErrosAlertas(ArrayList erros, ArrayList warnings, ArrayList projeto) throws IOException {
28         super("Gráfico Total Erros Alertas");
29
30         if (projeto.size() > 0) {
31
32             for (int i = 0; i < projeto.size(); i++) {
33
34                 dataSetPronto.addValue((int) erros.get(i), "erros", projeto.get(i).toString());
35                 dataSetPronto.addValue((int) warnings.get(i), "warnings", projeto.get(i).toString());
36
37             }
38
39             JFreeChart chart = ChartFactory.createBarChart("Total Erros e Warnings", "Projetos",
40                 "Quantidades Erros e Warnings",
41                 dataSetPronto, PlotOrientation.HORIZONTAL, true, true, true);
42
43             chart.getCategoryPlot().getRenderer(0).setSeriesPaint(0, Color.RED);
44             chart.getCategoryPlot().getRenderer(0).setSeriesPaint(1, Color.YELLOW);
45             chart.getCategoryPlot().setBackgroundPaint(Color.LIGHT_GRAY);
46             chart.getCategoryPlot().setDomainGridlinePaint(Color.WHITE);
47             chart.getCategoryPlot().setRangeGridlinePaint(Color.WHITE);
48
49             this.add(new ChartPanel(chart));
50             this.pack();
51
52             setVisible(true);
53
54         } else {
55             JOptionPane.showMessageDialog(rootPane, "Não foi possível gerar o grafico. Realize o filtro.");
56         }
57
58     }
59 }
```

# Operacionalidade

Protótipo para Análise de Logs do TestComplete

Data Inicial: 23/06/2015      Responsável:

Data Final: 23/06/2015      Projeto:

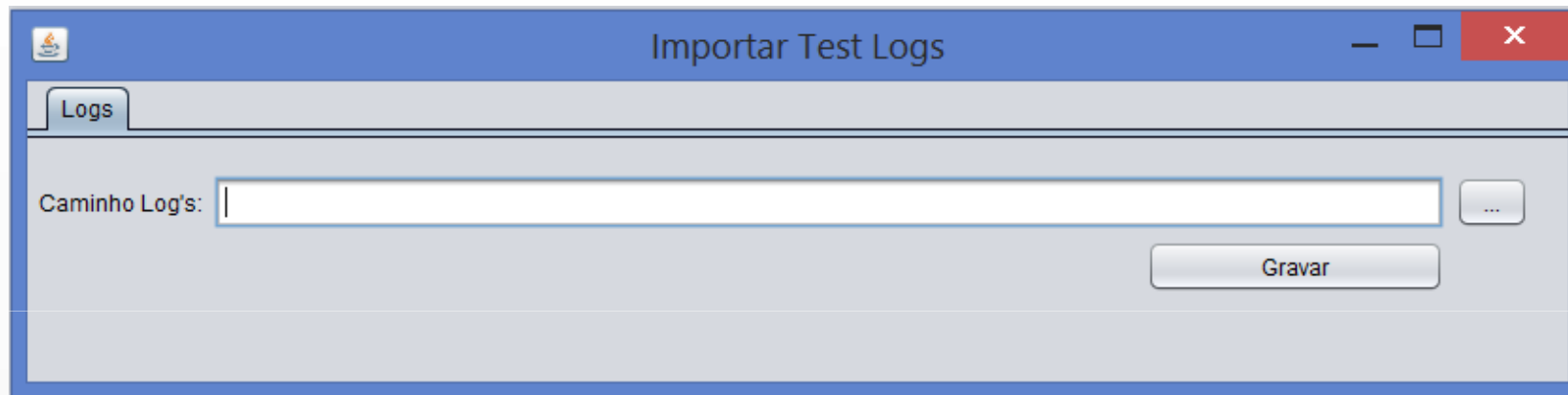
Projeto	Responsavel
Financeiro	Goku
Comercial	Chaves
Orcamento	Goku

Caso de Testes

- CT\_Cadastro\_Portadores\_01
- CT\_Cadastro\_InstrucoesBancarias\_01
- CT\_Cadastro\_TiposVencimento\_02
- CT\_Cadastro\_TiposVencimento\_01
- CT\_Cadastro\_InstrucoesBancarias\_02
- CT\_Cadastro\_Portadores\_02

# Operacionalidade



# Operacionalidade

Protótipo para Análise de Logs do TestComplete

Data Inicial: 01/01/2015      Responsável:       Importar Logs

Data Final: 23/06/2015      Projeto:       Gerar Análise / Resumo

Filtrar

Projetos    Test Logs

IdLog	Projeto	Responsavel	ErrorCount	WarningCount	StartDate	StartTime	StopDate	StopTime	Status
95	Financeiro	Goku	0	0	2015-05-29	23:09:12	2015-05-29	23:22:17	Sucesso
96	Financeiro	Goku	0	0	2015-05-30	10:28:40	2015-05-30	10:38:22	Sucesso
97	Financeiro	Goku	0	0	2015-05-31	14:22:53	2015-05-31	15:41:05	Sucesso
100	Financeiro	Goku	0	0	2015-06-01	19:27:35	2015-06-01	19:53:07	Sucesso
102	Financeiro	Goku	45	2	2015-06-15	20:00:38	2015-06-15	20:27:13	Failed
98	Comercial	Chaves	66	0	2015-05-31	17:20:31	2015-05-31	17:55:23	Failed
99	Comercial	Chaves	0	0	2015-06-01	18:39:53	2015-06-01	19:03:38	Sucesso
101	Orcamento	Goku	0	0	2015-06-01	22:26:50	2015-06-01	22:36:42	Sucesso

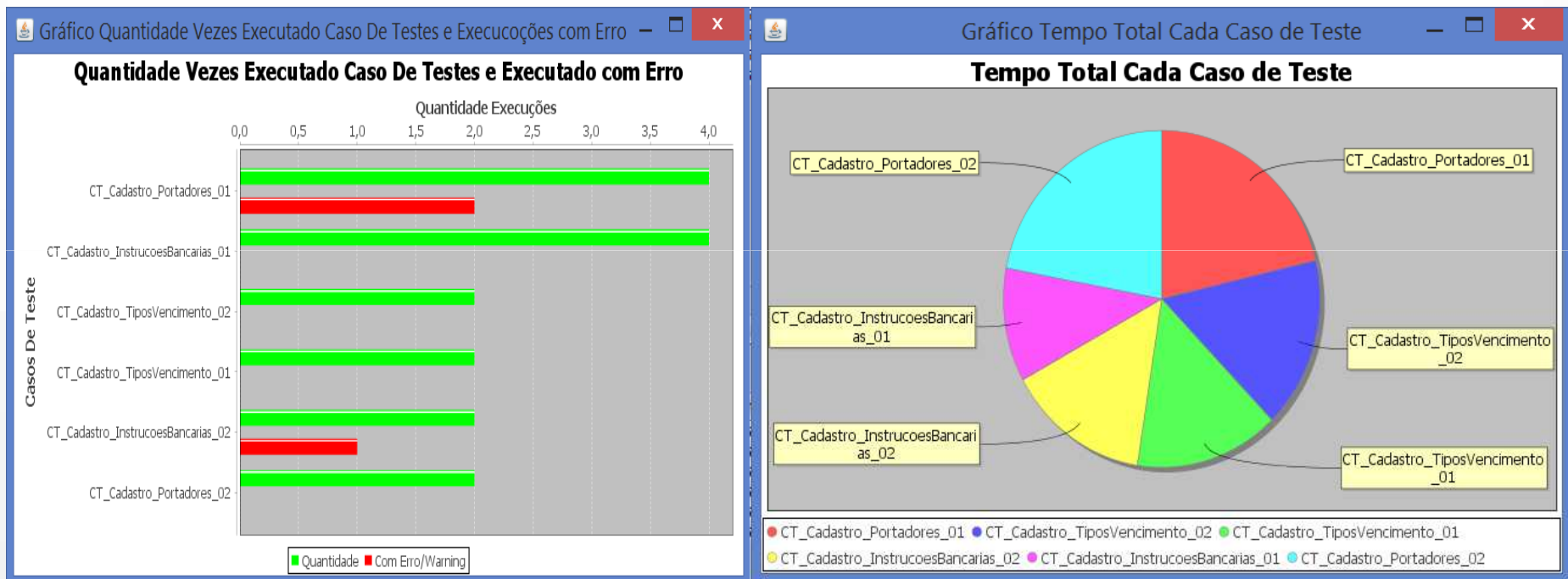
Total Erros e Alertas Projetos    Tempo total testes Projetos    Total Erros e Alertas por Responsável    Tempo Total Testes por Responsavel    Comparar

Id CasoDeTeste	Descricao	Projeto	ErrorCountCT	WarningCountCT	Start DateCT	Start TimeCT	Stop DateCT	Stop TimeCT	StatusCT
270	CT_Cadastro_P...	Financeiro	0	1	2015-06-15	20:00:52	2015-06-15	20:07:29	Failed
271	CT_Cadastro_I...	Financeiro	44	0	2015-06-15	20:13:37	2015-06-15	20:27:13	Failed
272	CT_Cadastro_P...	Financeiro	0	1	2015-06-15	20:07:29	2015-06-15	20:13:37	Failed

Tempo Total Caso de Teste    Total Erros Caso de Teste    Total Alertas Caso de Teste

Id Evento	Descrição Evento	Remarks	Tipo Evento	Sequencia Evento	Id Caso de Teste
299	⓪ Edit código deveria estar desabilitado*	==	2	477	272

# Operacionalidade



# Resultados e Discussões

- Objetivo de automatizar um processo para auxiliar usuários da ferramenta *TestComplete*.
- Foi possível realizar comparações com os trabalhos correlatos.
- Objetivos específicos foram atingidos.

# Resultados e Discussões

Características	Protótipo desenvolvido	MARCOS (2007)	WALTRICK (2011)
Linguagem de programação	Java	Delphi	Pascal / Delphi Script
Banco de dados	MySQL	Nenhum	MySQL
Ambiente (desktop/web)	Desktop	Desktop	Desktop
Ferramenta para o <i>TestComplete</i>	SIM	SIM	SIM
Automatiza um processo	SIM, visualização de vários <i>Test Logs</i>	SIM, analisa telas e gera <i>script</i> de testes.	SIM, integra resultados das execuções no <i>TestLink</i>



# Conclusão

- Ferramentas utilizadas para elaborar o trabalho mostraram-se adequadas.
- Informações das execuções sem abrir o *TestLog* no *TestComplete*.
- Agilizou as atividades de análise e comparação dos *TestLogs*.

# Extensão

- Implementação versão web.
- *Plugin* que adicione o *TestLog* selecionado na ferramenta TestComplete.
- Filtro para informar hora inicial e final da execução

# Demonstração do sistema!

Obrigado!