

UNIVERSIDADE REGIONAL DE BLUMENAU  
CURSO DE SISTEMAS DE INFORMAÇÃO



# **Plugin da Ferramenta TestComplete para integração com a ferramenta TestLink**

**DOUGLAS DE OLIVEIRA WALTRICK**

Orientador: Everaldo Artur Grahl

## Apresentação

1. Introdução
2. Objetivos
3. Fundamentação Teórica
4. Desenvolvimento
5. Resultados e Discussões
6. Conclusão
7. Extensões

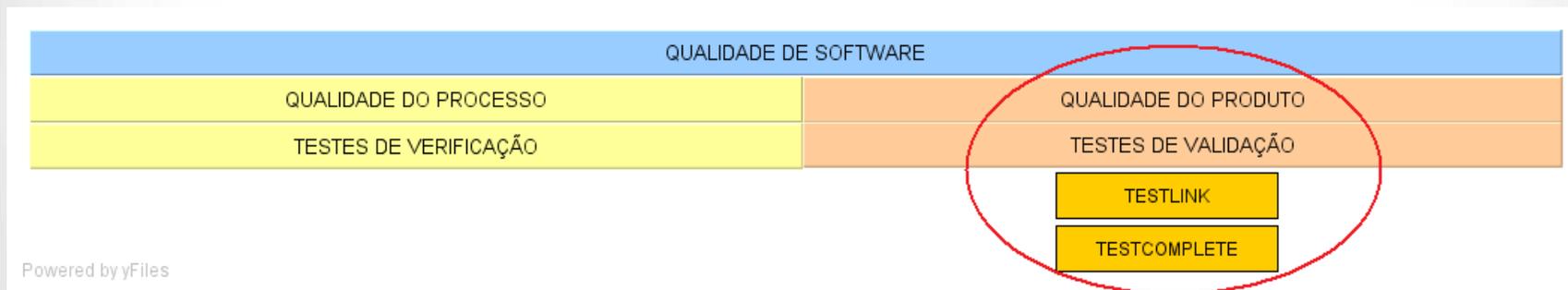
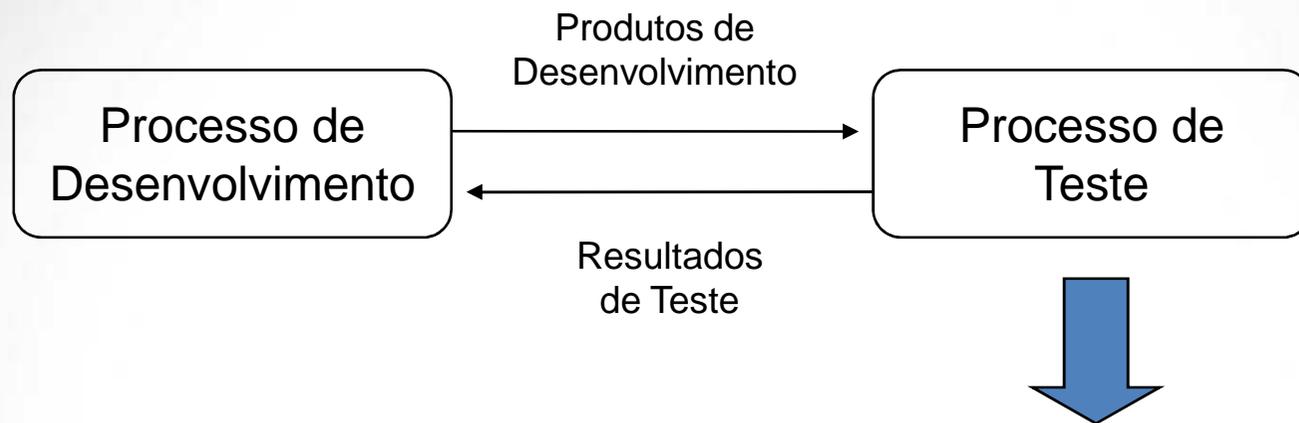


## Apresentação

1. Introdução
2. Objetivos
3. Fundamentação Teórica
4. Desenvolvimento
5. Resultados e Discussões
6. Conclusão
7. Extensões



# INTRODUÇÃO



## Apresentação

1. Introdução
2. **Objetivos**
3. Fundamentação Teórica
4. Desenvolvimento
5. Resultados e Discussões
6. Conclusão
7. Extensões



## OBJETIVOS

- ✓ Integração entre as ferramentas TestComplete e TestLink através de um plugin desenvolvido no TestComplete
- ✓ Melhor manipulação dos dados utilizados para input nos testes
- ✓ Facilitar a execução dos casos de testes
- ✓ Manter histórico de execução em ambas as ferramentas



## Apresentação

1. Introdução
2. Objetivos
3. **Fundamentação Teórica**
4. Desenvolvimento
5. Resultados e Discussões
6. Conclusão
7. Extensões



## FUNDAMENTAÇÃO TEÓRICA

### Teste de Software

- ✓ Teste de software é o processo que visa executar o software de forma controlada, com o objetivo de avaliar o seu comportamento, baseado no que foi especificado
- ✓ Evolução das atividades relacionadas ao Teste de Software nos anos 60, 70 e 80



## FUNDAMENTAÇÃO TEÓRICA

### Processo de Testes

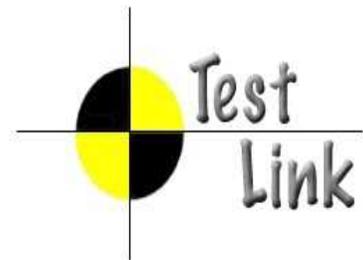
- ✓ O processo de testes de software representa uma estruturação de etapas, atividades, artefatos, papéis e responsabilidades
- ✓ Visa a padronização, ampliar a organização e controle dos projetos de testes.
- ✓ Deve basear-se em uma metodologia aderente ao processo de desenvolvimento, ter pessoal técnico qualificado, ambiente e ferramentas adequadas.



## FUNDAMENTAÇÃO TEÓRICA

### Ferramentas - TestLink

- ✓ O TestLink é uma aplicação Open Source cujo principal objetivo é gerenciar as atividades de testes de um projeto.
- ✓ Com ela, é possível criar planos de teste, atribuir casos de testes aos testadores, registrar os resultados dos testes, e acompanhar os resultados da execução dos mesmos, assim como, gerar relatórios com diversas métricas para o acompanhamento da execução dos testes.



## FUNDAMENTAÇÃO TEÓRICA

### Ferramentas - TestComplete

- ✓ O TestComplete é uma ferramenta de automatização de testes que permite criar, gerenciar e executar os mais variados tipos de testes, todos eles de forma rápida e prática, assim aumentando exponencialmente a cobertura dos testes.



## FUNDAMENTAÇÃO TEÓRICA

### Ferramentas - TestComplete

- ✓ Uma grande vantagem da ferramenta é a versatilidade de testar os mais variados tipos de produtos não se preocupando com a linguagem que este foi desenvolvido, pois o TestComplete possui uma linguagem híbrida, sendo esta, uma mescla de comandos próprios da ferramenta com comandos conhecido e largamente utilizados nas principais linguagens de programação.



## FUNDAMENTAÇÃO TEÓRICA

### Plugin no TestComplete

- ✓ Arquitetura aberta que possibilita utilização de plugin.
- ✓ Criação de plugin dentro do próprio TestComplete.
- ✓ Criação de plugin através de programa externo, TestComplete SDK.



## FUNDAMENTAÇÃO TEÓRICA

### Trabalhos Correlatos

- ✓ Ferramenta de apoio à automatização de testes através do TestComplete para programas desenvolvidos em Delphi (FURB, 2007).
- ✓ Testes de Software a partir da Ferramenta Visual Test (FURB, 2001).



## Apresentação

1. Introdução
2. Objetivos
3. Fundamentação Teórica
4. **Desenvolvimento**
5. Resultados e Discussões
6. Conclusão
7. Extensões



## DESENVOLVIMENTO

### Requisitos Funcionais

- ✓ RF01: O sistema deverá permitir configurar ambiente de testes antes de iniciar o processo de automatização.
- ✓ RF02: O sistema deverá possibilitar que um caso de teste criado no TestLink possa ser executado automaticamente no TestComplete.
- ✓ RF03: O sistema deverá possibilitar a execução de lotes de casos de testes automaticamente no TestComplete.



## DESENVOLVIMENTO

### Requisitos Funcionais

- ✓ RF04: O sistema deverá manter um histórico de cada caso de teste executado pelo TestComplete e ao final da automatização gerar automaticamente no TestLink informações referente a cada caso de teste executado.
- ✓ RF05: O sistema deverá possibilitar a codificação de script de testes a partir de funções pré-definidas.
- ✓ RF06: O sistema deverá possibilitar a utilização da massa de dados vinculada no TestLink sem a necessidade de carregá-la explicitamente no TestComplete.



## DESENVOLVIMENTO

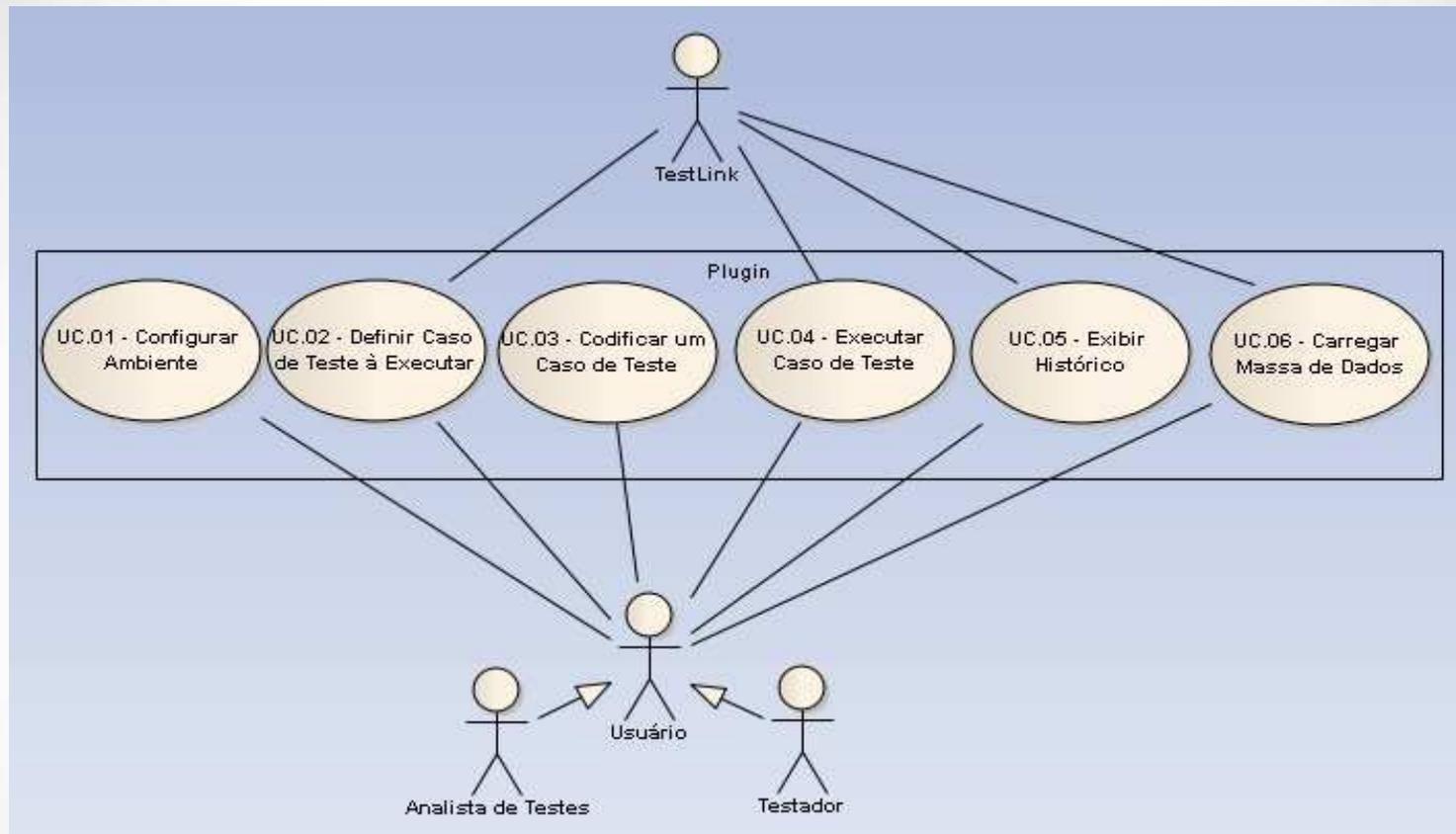
### Requisitos Não Funcionais

- ✓ RNF01: O sistema deverá ser desenvolvido utilizando as linguagens Pascal, DelphiScript e nativa do TestComplete.
- ✓ RNF02: O sistema deverá utilizar o banco de dados MySQL para manter a base de dados da ferramenta TestLink.
- ✓ RNF03: O sistema deverá ser desenvolvido para automatizar testes em ambiente Desktop.



# DESENVOLVIMENTO

## Casos de Uso



## DESENVOLVIMENTO

### Ferramentas e Técnicas

- ✓ Servidor de aplicação web Apache
- ✓ MySQL
- ✓ Linguagem Pascal
- ✓ Linguagem DelphiScript
- ✓ TestComplete
- ✓ TestLink



# DESENVOLVIMENTO

## Funcionalidades – Executar os Casos de Testes

```
unPrincipal * unCadastros x Financeiro x
5  var FModulo: OleVariant;
6  begin
7    {Inicialização do módulo a ser testado e das variáveis utilizada no Tcp}
8    FModulo := Sys.Find('ProcessName','academico');
9    if not FModulo.Exists then
10   begin
11     FModulo := TestedApps.Academico.Run();
12     setModulo(FModulo);
13     fazerLogin(FModulo);
14   end
15   else
16   begin
17     setModulo(FModulo,true);
18     FModulo.Window('TApplication','academico',1).Restore;
19   end;
20   {Final Incilização ** Não chamar nada acima desse comentário}
21
22   {Inicio do Bloco de chamada para execução dos CTs}
23   //Realizado chamada de um único caso de teste
24   executaCasosDeTestes('',4998);
25
26   //Realizado chamada de todos os casos de testes automatizados
27   //a partir da suite 'Disciplinas'
28   executaCasosDeTestes('Disciplinas');
29
30   {Final do Bloco de chamada dos CTs}
31
32   {Finalização do Tcp e chamada do relatório de execução dos CTs}
33   encerraModulo;
34   Historico;
35 end;
```

# DESENVOLVIMENTO

## Funcionalidades – Monitorar a Execução

```
unPrincipal * x unIntegracao x unCadastrados x
executaCasosDeTestes (ASuite :
423
424     FNomeRotina := 'CT_'+FormatCurr('00000',FQry.FieldName('CT_ID').asString);
425     try
426         try
427             Log.Message(' Chamada da Rotina Referente TC_ID: '+FQry.FieldName('CT_ID').asString,
428                 ' Suite do Caso de Teste : '+FQry.FieldName('SUITE').asString +'#13+#13
429                 +' Id. do Caso de Teste : '+FQry.FieldName('CT_ID').asString +'#13
430                 +' Nome do Caso de Teste: '+FQry.FieldName('CT_NOME').value +'#13
431                 +' Rotina no TestComplete: procedure '+FNomeRotina );
432
433             addItemArray('CT_Executados',FQry.FieldName('CT_ID').asString,FNomeRotina);
434             FStatusCT := true;
435             FTempoIni:= time;
436             chamarRotina(FNomeRotina);
437
438             if not fecharTelaEmExecucao then
439                 begin
440                     FTempoFin := time;
441                     FStatusCT := false;
442                 end;
443             FTempoFin := time;
444
445             //Aparentemente executou corretamente
446             if FStatusCT then
447                 addItemArray('CT_OK',FQry.FieldName('CT_ID').asString,FNomeRotina)
448             else
449                 begin
450                     addItemArray('CT_Falhados',FQry.FieldName('CT_ID').asString,FNomeRotina);
451                     PrintarTela(true,FQry.FieldName('CT_ID').Integer);
452                 end;
453
```

## Apresentação

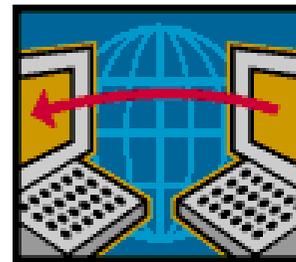
1. Introdução
2. Objetivos
3. Fundamentação Teórica
4. Desenvolvimento
5. Resultados e Discussões
6. Conclusão
7. Extensões



## RESULTADOS E DISCUSSÕES

**No uso diário, pontos fracos foram identificados...**

- ✓ Com o passar do tempo de utilização tanto da ferramenta TestLink quanto o TestComplete, naturalmente foram aparecendo problemas por motivo da falta de inter-comunicação entre estas.



## RESULTADOS E DISCUSSÕES

### Problema encontrado: execução dos Casos de Testes.

#### Problema

- ✓ Um problema que é gerado quando se possui um número considerável de casos de testes automatizados, é a execução destes na ordem correta, levando em conta que estes em sua grande maioria são dependentes entre si, como pode-se verificar no exemplo a seguir.



## RESULTADOS E DISCUSSÕES

### Problema encontrado: execução dos Casos de Testes.

#### Exemplo do problema:

- ✓ É automatizado um processo que necessita de vários cadastros para seu funcionamento, e os valores destes são gravados no script de teste. Sabendo que tem-se como prática sempre executar os lotes de automatização partindo de um database vazio, se os mesmos forem executados indiscriminadamente tens-se uma grande chance de muitos dos casos de testes falharem, por falta de informação (valor esperado no TestComplete e não encontrado no database).



## RESULTADOS E DISCUSSÕES

**Problema encontrado: execução dos Casos de Testes.**

### Solução

- ✓ O Plugin foi preparado para executar os Casos de Testes sempre por sua ordem lógica definida no TestLink, ou seja, a mesma estrutura de árvore montada no TestLink é utilizada na execução dos casos de teste automatizados.





## RESULTADOS E DISCUSSÕES

### Problema encontrado: execução dos Casos de Testes.

#### Problema

- ✓ Outro problema também decorrente quando se possui um número considerável de casos de testes automatizados, é como estruturar a execução de um lote de casos de testes sem ter que ficar sempre conferindo se um novo caso de teste criado no TestLink está contido em algum lote de execução no TestComplete, como no exemplo a seguir.



## RESULTADOS E DISCUSSÕES

### Problema encontrado: execução dos Casos de Testes.

#### Exemplo do problema

- ✓ Defini-se no TestComplete um grupo de casos de testes que será executado, e nele há uma lista de 10 casos de testes. Em seguida cria-se um novo caso de teste no TestLink e é necessário fazer o mesmo processo de ordenação destes lá no TestComplete, sendo que este trabalho já foi realizado no TestLink quando os mesmos foram concebidos.



## RESULTADOS E DISCUSSÕES

**Problema encontrado: execução dos Casos de Testes.**

### Solução

- ✓ É possível executar um ou vários casos de testes em um mesmo lote, possibilitando chamar a rotina de teste utilizando seu identificador(número do caso de teste) definido no TestLink, onde no TestComplete será executado apenas o caso de teste de forma unitária.



## RESULTADOS E DISCUSSÕES

**Problema encontrado: execução dos Casos de Testes.**

### Exemplo Prático

Se for preciso realizar a chamada da rotina de automatização do caso de teste referente ao cadastro de uma disciplina, bastaria apenas utilizar o seguinte comando:

```
executaCasosDeTestes("", '<Nro_CT_TestLink>')
```



# RESULTADOS E DISCUSSÕES

## Problema encontrado: execução dos Casos de Testes.

```
unPrincipal* x unCadastrros x Financeiro x
5 var FModulo: OleVariant;
6 begin
7   {Inicialização do módulo a ser testado e das variáveis utilizada no Tcp}
8   FModulo := Sys.Find('ProcessName', 'academico');
9   if not FModulo.Exists then
10    begin
11     FModulo := TestedApps.Academico.Run();
12     setModulo(FModulo);
13     fazerLogin(FModulo);
14    end
15  else
16    begin
17     setModulo(FModulo, true);
18     FModulo.Window('TApplication', 'academico', 1).Restore;
19    end;
20  {Final Incilização ** Não chamar nada acima desse comentário}
21
22  {Inicio do Bloco de chamada para execução dos CTs}
23  //Realizado chamada de um único caso de teste
24  executaCasosDeTestes('', 4998);
25
26  //Realizado chamada de todos os casos de testes automatizados
27  //a partir da suite 'Disciplinas'
28  executaCasosDeTestes('Disciplinas');
29
30  {Final do Bloco de chamada dos CTs}
31
32  {Finalização do Tcp e chamada do relatório de execução dos CTs}
33  encerraModulo;
34  Historico;
35 end;
```

**Caso de Teste**

**GA-4998:Cadastrar uma Disciplina**

Editar Deletar Mover / Copiar Criar uma nova versão Desativar esta versão

**Versão 1**  
Criado em 2011/11/20 16:04:03 por xpto

**Objetivo do Teste:**  
O teste tem com objetivo realizar o processo de criação de um registro de disciplina.

**Pré-condições**  
Já possuir o registro de departamento "Departamento\_01" já cadastrado.

Criar um passo

**Prioridade do Teste:** Baixo

**Automatizado ?**  SIM

**Pre - Requisito p/Execução do UI atual:**

**CT(s) de Consulta Vinculado(s):**

**Brefa Vinculada:**

**Brefa origem CT:**

**Palavras-chave:** CADASTROS  
DISCIPLINA

**Requisitos:** Nenhum

**Arquivos anexados:**

Carregar novo arquivo

## RESULTADOS E DISCUSSÕES

**Problema encontrado: execução dos Casos de Testes.**

### Solução

- ✓ É possível também, e certamente muito mais utilizado, a execução de vários casos de testes, onde utiliza-se do conceito de suítes (Agrupamento de Casos de Testes) do TestLink. Ao realizar a chamada de uma suíte, o TestComplete executará todos os casos de testes vinculados a esta suíte no TestLink.



## RESULTADOS E DISCUSSÕES

**Problema encontrado: execução dos Casos de Testes.**

### Exemplo Prático

Se fosse preciso realizar a chamada de todas as rotinas automatizadas referente ao cadastro de disciplinas, bastaria apenas utilizar o seguinte comando:

```
executaCasosDeTestes('<Nome_Suite_TestLink>','')
```



# RESULTADOS E DISCUSSÕES

## Problema encontrado: execução dos Casos de Testes.

```
unPrincipal* x unCadastros x Financeiro x
5 var FModulo: OleVariant;
6 begin
7   {Inicialização do módulo a ser testado e das variáveis utilizada no Tcp}
8   FModulo := Sys.Find('ProcessName', 'academico');
9   if not FModulo.Exists then
10    begin
11     FModulo := TestedApps.Academico.Run();
12     setModulo(FModulo);
13     fazerLogin(FModulo);
14    end
15  else
16    begin
17     setModulo(FModulo, true);
18     FModulo.Window('TApplication', 'academico', 1).Restore;
19    end;
20  {Final Incilização ** Não chamar nada acima desse comentário}
21
22  {Inicio do Bloco de chamada para execução dos CTs}
23  //Realizado chamada de um único caso de teste
24  executaCasosDeTestes('', 4998);
25
26  //Realizado chamada de todos os casos de testes automatizados
27  //a partir da suite 'Disciplinas'
28  executaCasosDeTestes('Disciplinas');
29
30  {Final do Bloco de chamada dos CTs}
31
32  {Finalização do Tcp e chamada do relatório de execução dos CTs}
33  encerraModulo;
34  Historico;
35 end;
```

**Suite de Teste : Disciplinas**

**Operações da Suite de Teste**

**Operações do Caso de Teste**

---

**Suite de Teste : Disciplinas**

**Detalhes**

Esta suite de testes irá agrupar todos os casos de testes, que se referem aos Cadastros de Disciplinas

**Palavras-chave :** PARAMETROS  
PROCESSO\_Cadastro

**Arquivos anexados :**



## RESULTADOS E DISCUSSÕES

**Problema encontrado: depois de executado os CT's ,  
onde registrar ?**

### Problema

- ✓ Depois de executados todos os casos de testes, é necessário manter no mínimo o registro do status da cada caso de teste executado (passou/falhou), e isso demanda um tempo considerável de uma pessoa, levando em conta que existe um volume grande casos de testes automatizados, conforme exemplo a seguir.



## RESULTADOS E DISCUSSÕES

**Problema encontrado: depois de executado os CT's ,  
onde registrar ?**

### Exemplo do problema

- ✓ Para cada caso de teste executado no TestComplete faz-se necessário varrer o log de execução deste e depois de analisado realizar manualmente o registro da execução no TestLink.



## RESULTADOS E DISCUSSÕES

**Problema encontrado: depois de executado os CT's ,  
onde registrar ?**

### Solução

- ✓ O Plugin controla os casos de testes durante e depois da execução destes, ou seja, para cada um executado é mantido um histórico com todas as informações necessárias, e ao final de todo o processo são exibidas ao usuário.



## RESULTADOS E DISCUSSÕES

**Problema encontrado: depois de executado os CT's ,  
onde registrar ?**

### Solução

- ✓ Todos os casos de testes executados no TestComplete são registrados automaticamente no TestLink sem a intervenção do testador, ou seja, cada informação armazenada durante o processo de automatização (id do caso de teste, status da execução, tempo gasto, etc) é registrado no TestLink no término do processo de automatização.



# RESULTADOS E DISCUSSÕES

Problema encontrado: depois de executado os CT's ,  
onde registrar ?

Suite de Teste : Acadêmico/ Cadastros/Parametros Básicos Módulo Acadêmico/ Parâmetros/ Disciplinas/

Caso de Teste ID GA-3699 :: Versão: 4  
Cadastrar uma Disciplina (Disciplina 00)  
Atribuído à : adrian

Última execução (baseline qualquer) - Baseline : 01

Data : 20/10/2011 19:47:34 - Testador : TestComplete - Baseline : 01 - Status : Com Falha

Última execução (baseline atual) - Baseline : 01

Data	Baseline	Testador	Status	Versão do CT	Anexo
20/10/2011 19:47:34	01	TestComplete	Com Falha	1	

Notas:

Tempo do Teste (em minutos): 00:00:11

Imagem Vinculada:

Anexo - Falha de Execução - 3699.jpg (102400 bytes, image/jpeg) 20/10/2011

Sumário:

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

## Apresentação

1. Introdução
2. Objetivos
3. Fundamentação Teórica
4. Desenvolvimento
5. Resultados e Discussões
- 6. Conclusão**
7. Extensões



## CONCLUSÃO

- ✓ Redução do retrabalho operacional
- ✓ Objetivos atingidos
- ✓ Conhecimentos pessoais agregados



## Apresentação

1. Introdução
2. Objetivos
3. Fundamentação Teórica
4. Desenvolvimento
5. Resultados e Discussões
6. Conclusão
7. Extensões



## EXTENSÕES

- ✓ Utilização do plugin para execução de testes em aplicações Web
- ✓ Utilização do plugin como uma Extension do TestComplete
- ✓ Criação de mecanismos para permitir que as próprias aplicações gerem os scripts de automatizações sem intervenção humana





## **Apresentação da utilização do Plugin**