

# UNIVERSIDADE REGIONAL DE BLUMENAU

PLUGIN PARA O EA: GERAÇÃO DE INTERFACES DE  
USUÁRIO A PARTIR DE UM PROJETO DE TELAS

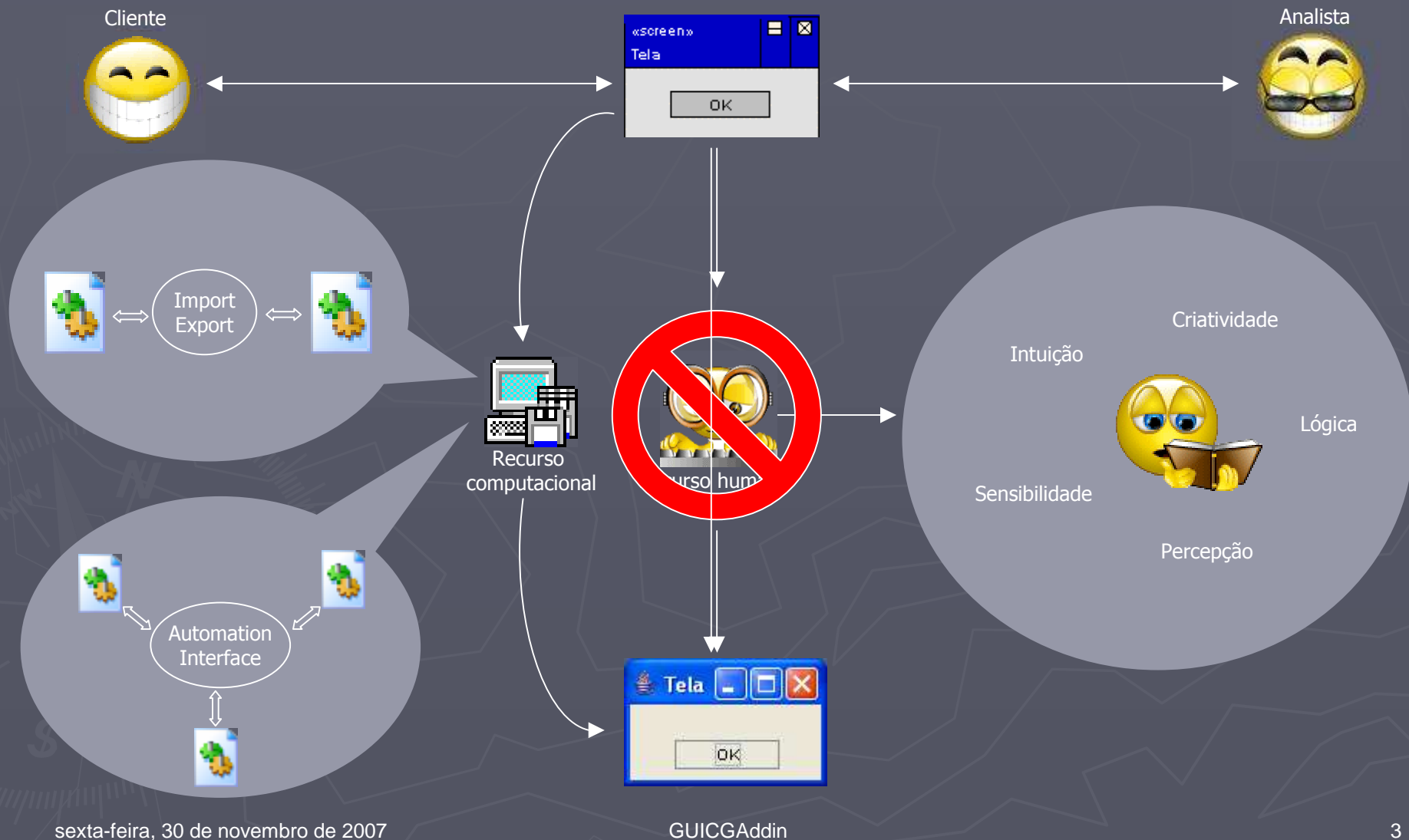
**Acadêmico: Cleiton Eduardo Saturno**

**Orientadora: Fabiane Barreto Vavassori Benitti**

# Roteiro da apresentação

- ▶ Introdução/Justificativa
- ▶ Objetivos do trabalho
- ▶ Embasamento teórico
  - I-CASE
  - Enterprise Architect
  - COM/ActiveX
  - Geração de Código
- ▶ Técnicas e ferramentas
- ▶ Especificação
- ▶ Operacionalidade
- ▶ Resultados e discussão
- ▶ Conclusão
- ▶ Extensões

# Justificativa



sexta-feira, 30 de novembro de 2007

GUICGAddin

# Objetivo

- ▶ *Plugin* para a geração de código de telas de sistema a partir de um modelo de interface de usuário projetado com o EA.
  - componentes visuais: `screen`, `button`, `checkbox`, `combobox`, `panel`, `radio` e `textbox`;
  - gerar código fonte para as linguagens Java e Delphi.

# I-CASE

O que e para quê?

- ▶ Ferramentas CASE Integradas
- ▶ Sistemas com mais qualidade
- ▶ Resposta rápida às mudanças
- ▶ Redução de custo

# I-CASE

## Modelo Arquitetural



# Enterprise Architect

Visão geral

“o EA é uma ferramenta que cobre todos os aspectos do ciclo de desenvolvimento, fornecendo suporte para testes, manutenção e alterações de requisitos.”

(LIMA 2005, p. 41-42)

- Processos de negócio
- Modelagem para web
- Planos de teste
- Diagramas da UML
- Interfaces de usuário
- ...

# Enterprise Architect

## Diagramas de telas





# Enterprise Architect

Interface de automação

► O que é?



# Enterprise Architect

Interface de automação

- ▶ O que pode ser feito com a IA do EA?
  - Efetuar tarefas repetitivas
  - Criar relatórios personalizados
  - Efetuar consultas customizadas
  - Gerar código a partir de diagramas

# Component Object Model - COM

O que e para quê?

## ► **Padrão** para comunicação entre aplicações

```
function EA_Connect (const Repository: IDispatch): WideString; safecall;
```

```
[ id(0x000000CD) ]  
HRESULT _stdcall EA_Connect([in] IDispatch * Repository, [out, retval] BSTR * Value );
```

- Independente de plataforma e linguagem
- Trabalha por meio de interfaces
- Classes identificadas por GUID
  - {20b1cb23-6968-4eb9-b7d4-a66d00d07cee}
- Passagem de endereços de memória (obj)

# ActiveX

O que é?

Um controle ActiveX é “um documento composto que é implementado como uma *dll* de **servidor dentro de processo** que suporta **automação, edição visual e ativação de dentro para fora**”.

(CANTÙ, 2003, p. 387)

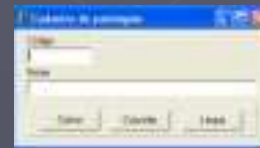
# Geração Automática de Código

Conceitos e vantagens

- O que é um gerador de código?

- Consistência

- Produtividade →



CARNEGIE MELLON S.E.I.

50% do tempo

48% do código

- Otimização de recursos humanos



Intuição

Criatividade

Lógica

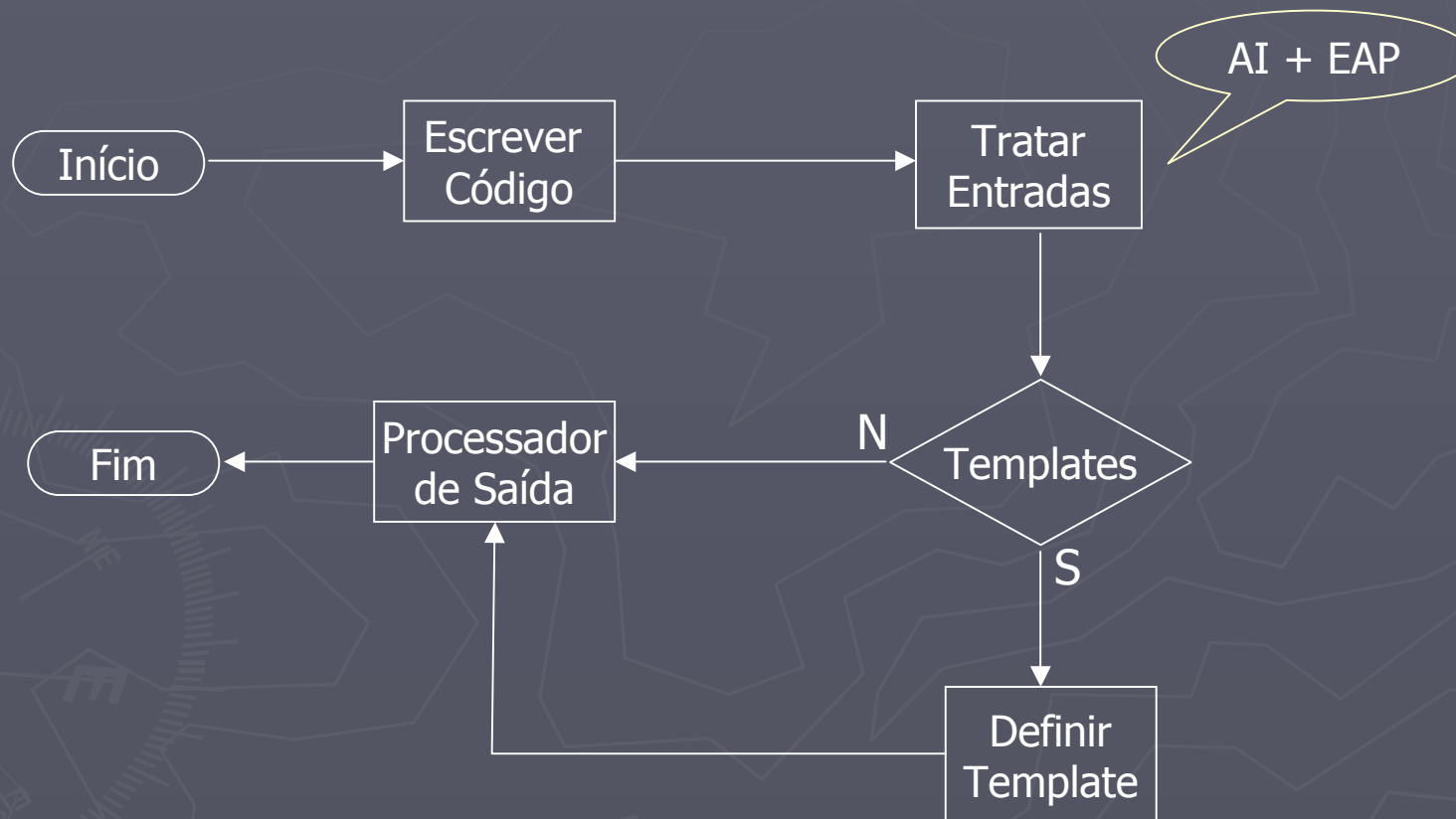
Percepção

Sensibilidade

- Padronização

# Geração Automática de Código

Desenvolvimento de um gerador



# Técnicas e Ferramentas Utilizadas

Aplicação

- ▶ UML
  - Diagramas de caso de uso
  - Diagrama de classes
  - Diagrama de componentes
  
- ▶ Ferramenta EA
- ▶ Orientação a objetos
- ▶ Ambiente Delphi 7
- ▶ Notação BNF
- ▶ GALS

# Requisitos da aplicação

## Requisitos Funcionais

- ▶ Gerar componentes: `screen`, `button`, `checkbox`, `combobox`, `panel`, `radio` e `textbox`
- ▶ Gerar código para Delphi
  - Formulários estáticos (`dfm + pas`)
  - Formulários dinâmicos (`pas`)
- ▶ Gerar código para Java



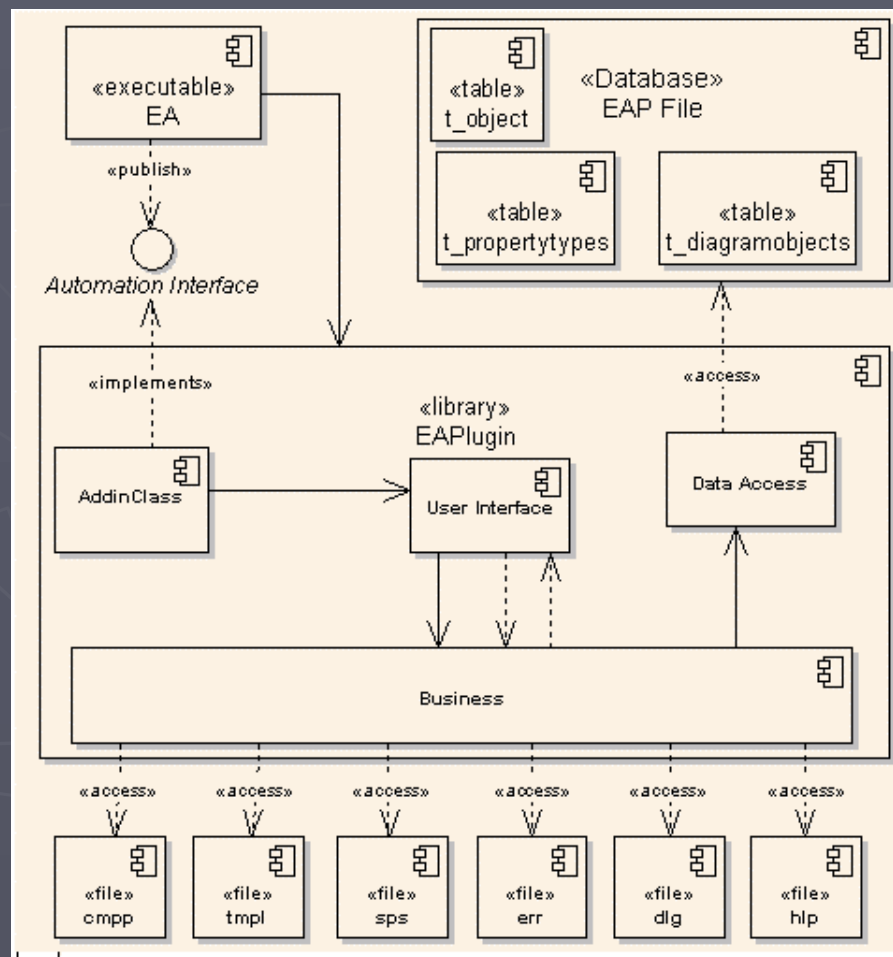
# Requisitos da aplicação

## Requisitos Não-funcionais

- ▶ Compatibilidade com EA Versão 6.5 Professional Edition para Windows
- ▶ Compatibilidade com Versão 7.0 do Ambiente Delphi
- ▶ Padrões Visuais
- ▶ Usar Delphi 7 na implementação
- ▶ Linguagem para criação de um template



# Diagrama de Componentes



# Operacionalidade da Aplicação



# Resultados e Discussão

- ▶ Flexibilidade x integração
  - Tagged Values
  - Notes
- ▶ Portabilidade das telas diagramadas
- ▶ Limitações no uso do template
- ▶ Não gera contêineres de componentes
- ▶ Não controla estrutura circular de herança
- ▶ Não implementa herança múltipla

# Conclusão

REQUISITOS FUNCIONAIS	ATENDIDO
Gerar código para Delphi e Java <b>Visual Basic e C#</b>	Sim
Gerar componentes screen, button, checkbox, combobox, panel, radio e textbox <b>Memo, LabeledEdit, Label, ListBox BitBtn, SpeedButton, MaskEdit, StringGrid, StatusBar</b>	Sim
<b>REQUISITOS FUNCIONAIS</b>	Sim
EA Versão 6.5 Professional Edition para Windows	Sim
Compatibilidade com Versão 7.0 do Ambiente Delphi	Sim
Padrões Visuais	Sim
Usar Delphi 7 na implementação	Sim
Linguagem para criação de um template	Sim

# Conclusão

- ▶ Comprovação dos benefícios das I-CASE
- ▶ Adequação das tecnologias
  - COM / ActiveX
  - GALS
- ▶ Flexibilidade dos templates

# Extensões

- ▶ Configuração de recursos
- ▶ Ampliação das funcionalidades do template
- ▶ Analisador para o arquivo .cmpp
- ▶ Implementação de contêineres de componentes
- ▶ Subdiretórios por diagrama/pacote
- ▶ Controle de estrutura circular de herança
- ▶ Controle de herança múltipla



# FIM

## Sourceforce Links

Project Info

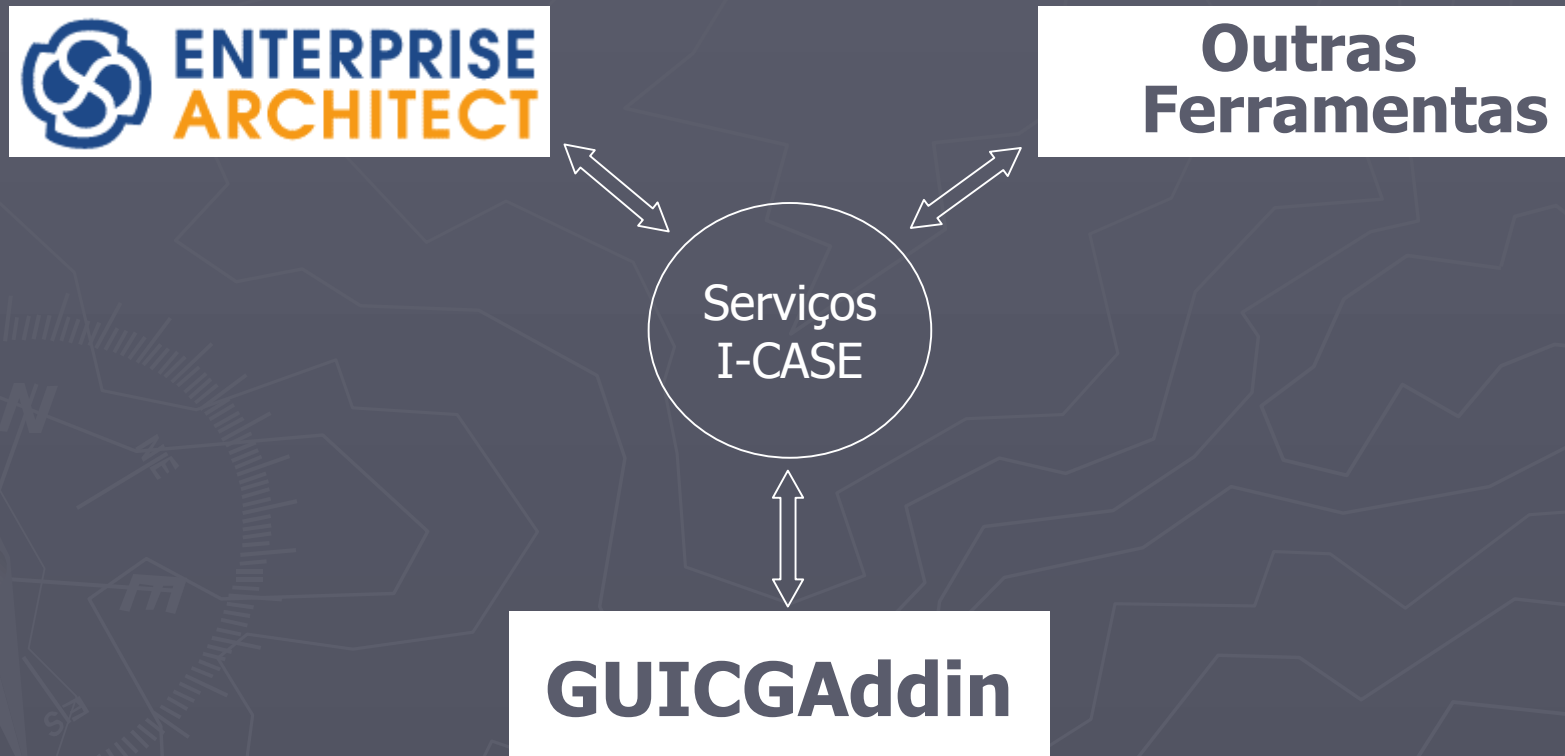
<http://sourceforge.net/projects/quicgaddin/>

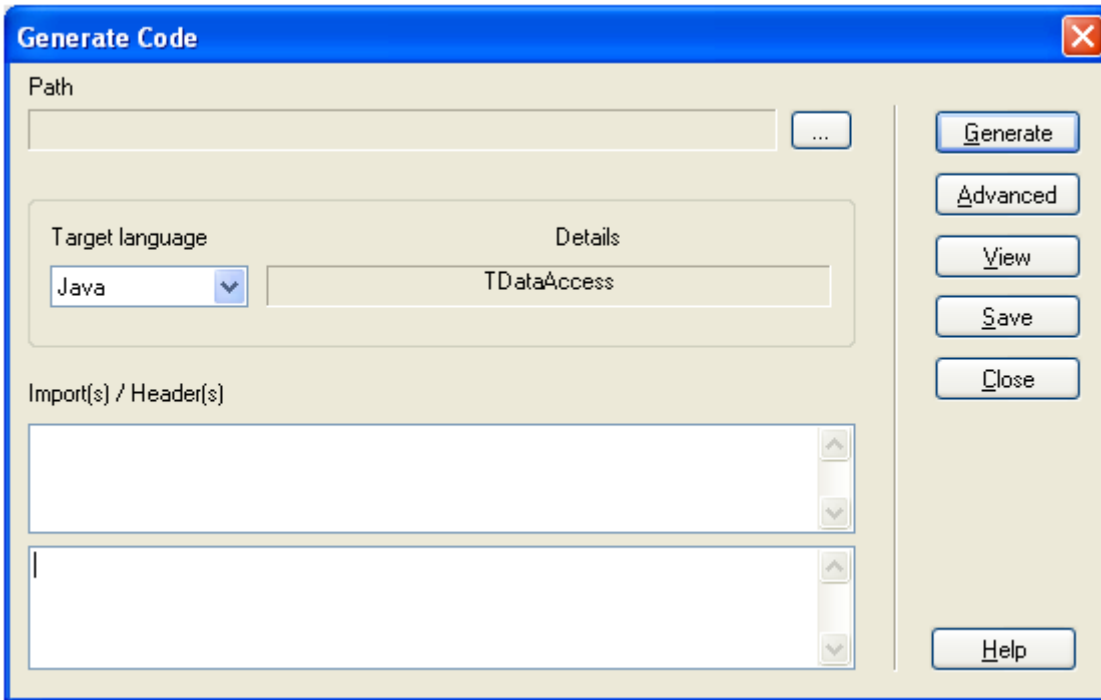
Project Files

<http://quicgaddin.cvs.sourceforge.net/quicgaddin/>

# I-CASE

Ferramentas CASE integradas





sexta-feira, 30 de novembro de 2007

