

GERADOR DE DOCUMENTAÇÃO PARA LINGUAGEM C, UTILIZANDO TEMPLATES

Acadêmico: Vilmar Orsi

Orientadora: Joyce Martins



www.furb.br



Roteiro

↳ Introdução

↳ Objetivos do trabalho

↳ Fundamentação teórica

↳ Desenvolvimento do trabalho

↳ Conclusão

↳ Extensões



Introdução

- ↳ Importância da documentação
 - Evolução na indústria de software
 - Manutenção e adaptação de software

- ↳ Qualidade da documentação
 - Atualização (periodicidade)
 - Reflexão da realidade do software

- ↳ Solução
 - Combinar código fonte e documentação
 - Facilitar acesso à documentação gerada

- ↳ Escolha da linguagem C



Objetivos do trabalho

- ↳ Desenvolver uma ferramenta para a geração automática de documentação para programas escritos em C.

- ↳ Disponibilizar analisadores
 - Léxico
 - Sintático
 - Semântico

- ↳ Definir um conjunto de marcadores especiais
 - *Tags*
 - *DocComments*

- ↳ Gerar a documentação
 - HTML

- ↳ Utilizar *templates*
 - Para formatação da documentação



Fundamentação teórica

↳ Conceitos básicos

- Documentação de software
- Geradores de código e de documentação
- Analisadores de linguagens de programação
- Motores de *templates*

↳ Trabalhos correlatos

- Javadoc
- Gerador de documentação do C#
- CDoc
- Doxygen



Documentação de software

- ↳ Geralmente não é realizada por:
 - Não ser vital para o software
 - Prazos e verbas insuficientes
 - Falta de política organizacional

- ↳ Associada a qualidade do software (Pfleeger)

- ↳ Geração de forma *in-line* (Pfleeger)
 - Código fonte
 - Economia de custo e tempo



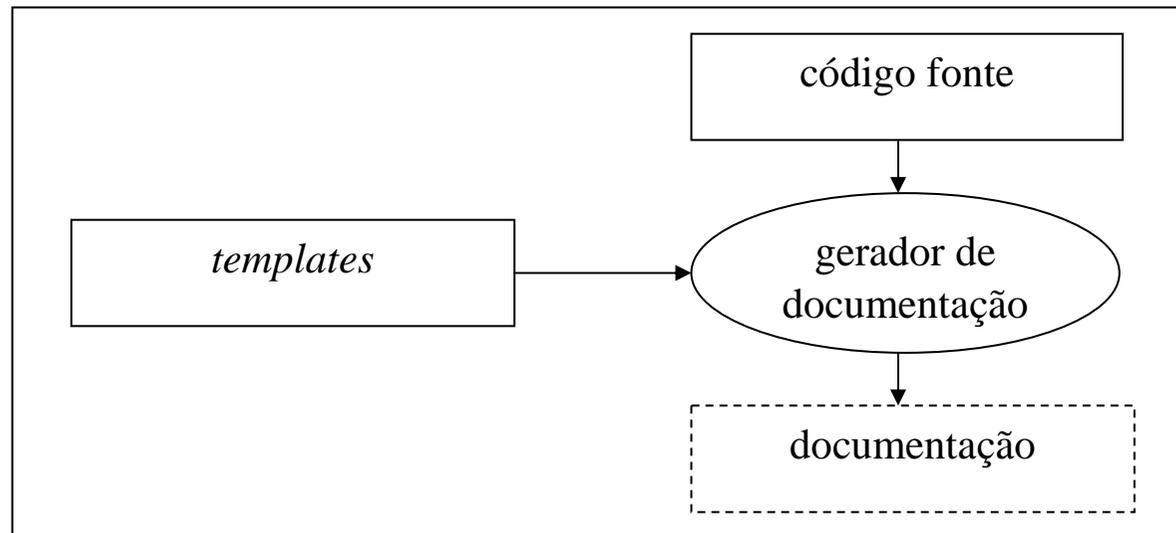
Geradores de código e de documentação

↳ Geradores são programas para escrever programas (Herrington)

↳ Os geradores de código podem ser classificados em

- Expansores de código
- Geradores mistos
- Geradores parciais de classes
- Geradores de camadas de aplicação
- Formatares de código

↳ Geradores de documentação

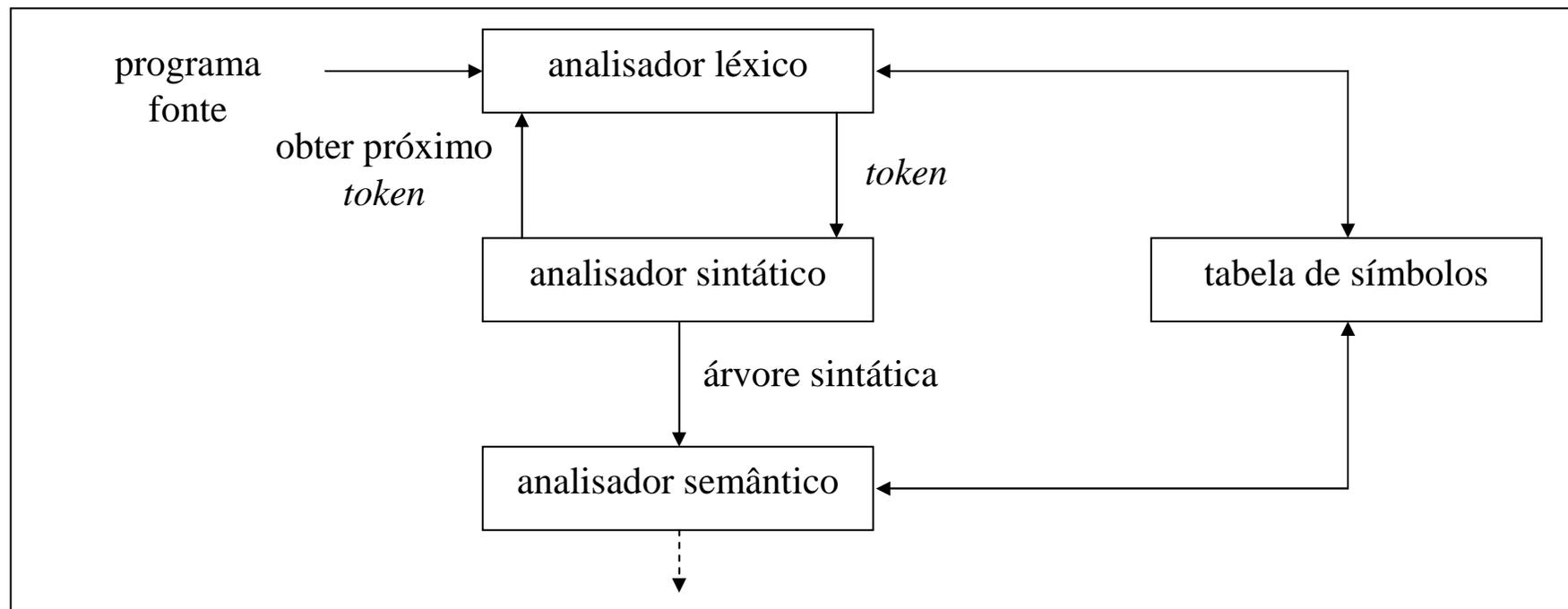




Analísadores de linguagens de programação

↳ Compiladores

- Linguagem-fonte (alto nível)
- Linguagem-alvo (código de máquina)





Motores de *templates*

↳ *Templates* podem conter código estático e dinâmico

```
#para-cada (@include)
    #se(@comentário.possui(@autor))
        <i><b>Autor: </b> @comentário.autor</i>
    #fecha
</td> </tr>
#fecha
```

Estático / Dinâmico

↳ Permitem a separação do código dinâmico e do código de interface

↳ Associam o *template* com as informações contidas em uma estrutura de dados

↳ Cada motor de *templates* define uma linguagem através da qual os *templates* deverão ser escritos

↳ FastTrac

↳ Velocity



Desenvolvimento do trabalho

- ↳ Levantamento dos requisitos
- ↳ Análise e identificação das informações necessárias para gerar a documentação
- ↳ Especificação das estruturas de dados para o armazenamento das informações
- ↳ Adaptação da gramática C ANSI
- ↳ Especificação de uma linguagem de DocComments
- ↳ Especificação de uma linguagem de *templates*
- ↳ Especificação da ferramenta através dos diagramas UML de casos de uso e classes
- ↳ Implementação da ferramenta



Requisitos da ferramenta

↳ Requisitos funcionais

- Gerar documentação, no formato HTML
- Realizar análises léxica e sintática do código fonte
- Permitir o uso de tags HTML
- Usar *templates* para a modelagem do documento a ser gerado
- Disponibilizar uma interface para permitir a elaboração dos *templates*
- Disponibilizar uma interface para a inserção de comentários no código fonte
- Disponibilizar um conjunto *tags*

↳ Requisitos não-funcionais

- Utilizar marcadores especiais no estilo dos marcadores definidos no JavaDoc
- Ser implementado utilizando o ambiente Delphi 7.0
- Ser compatível com o sistema operacional Windows 98, 2000 e XP



Identificação das informações para a documentação

↳ Diretivas

↳ Declarações

- Enumeradores (`enum`)
- Estruturas (`struct` e `union`)
- Definições de tipos (`typedef`)
- Funções
- Variáveis

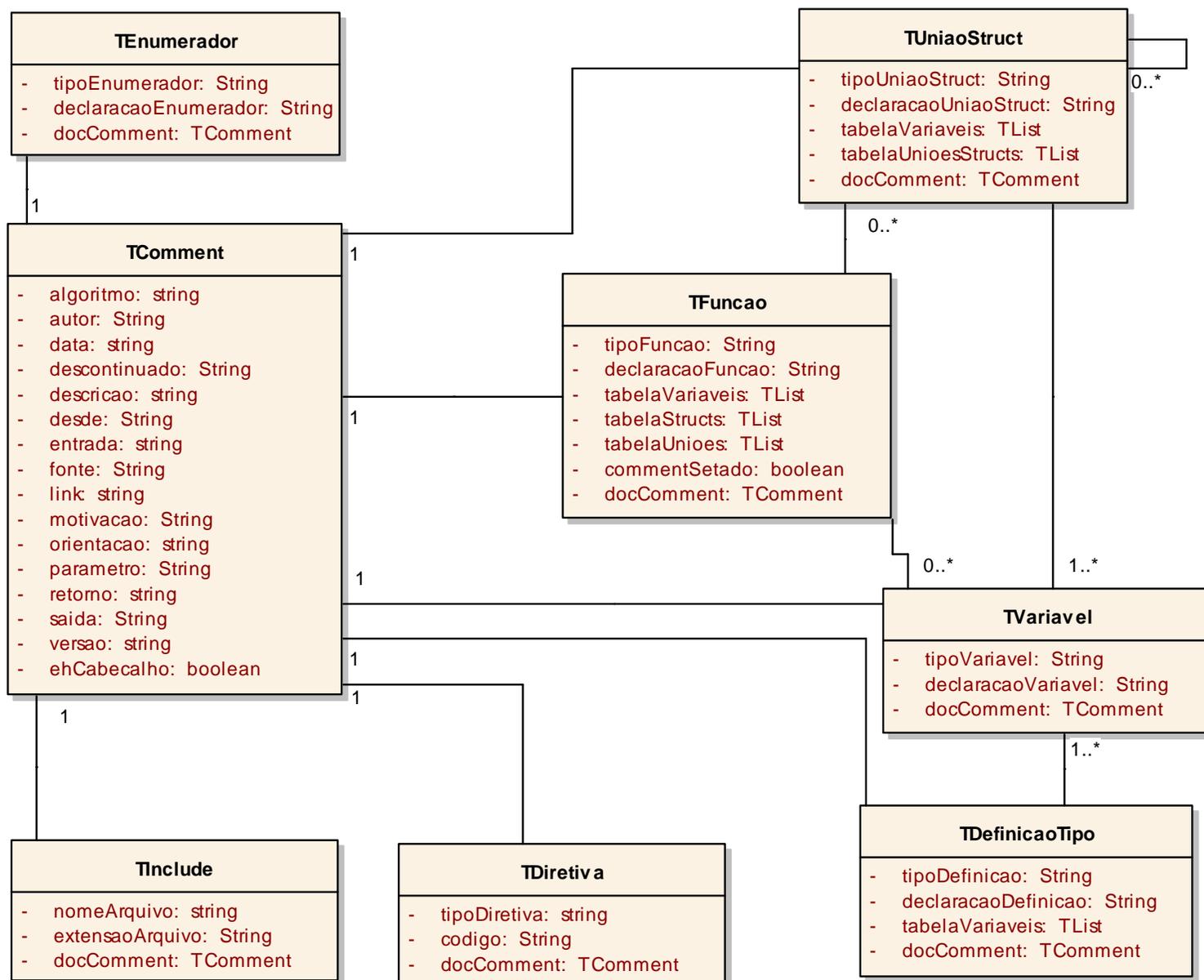
↳ Documentação

- Arquivos
- Projetos (`.PJT`)

```
/*@
  @cabecalho:
  @autor: Vilmar Orsi
  @motivação: Desenvolver uma ferramenta para desenhos
*/
#include "desenho.c"
#include "uestrutura.c"
#include "uestrutura.h"
```



Especificação das estruturas de dados





Análise e adaptação da gramática C ANSI

↳ Construções previstas pelo C ANSI, incluídas

- Declaração de tipos derivados e de variáveis a partir dos tipos derivados

```
struct RegistroProduto { int codigoProduto; };  
int main() { RegistroProduto produto; }
```

- Declaração de estruturas (`struct`, `union`) internas a funções

```
int main() { struct RegistroProduto { int codigoProduto; }; }
```

- Inicialização de *arrays* quando da declaração

```
int valores[] = {1, 2, 3, 4, 5};
```

- Definição de tipos (`typedef`) contendo *arrays*

```
typedef char[5] nome;
```



Análise e adaptação da gramática C ANSI

- Definição de tipos (`typedef`) contendo estruturas (`struct`, `union`)

```
typedef struct nodo {      char x;      int y;};
```

- Definição de enumeradores (`enum`) acompanhados apenas de um identificador ou de uma lista de identificadores

```
enum x;  enum {x=1, y=2};
```

↳ Construções previstas pelo C ++

- Declaração e inicialização de variáveis nos argumentos do comando `for`

```
for (int lin=0; lin<n; lin++) { }
```

- Declaração de variáveis do tipo `bool`

```
bool erros;
```



Especificação da linguagem de *DocComments*

↳ Comentários para documentação

↳ *Tags* GDC

- Dezesseis *tags*
- Baseados no JavaDoc e nas sugestões do professor Maurício C. Lopes
- *Case sensitive*
- Português

↳ *Tags* HTML

```
/*@
  @cabeçalho:
  @autor: <b>Vilmar Orsi</b>
  @versão: <i>Beta 1.0</i>
  @entrada: Um número <u>inteiro</u>
*/
```

↳ Especificação

- Léxica, Sintática e Semântica



Especificação da linguagem de *templates*

↳ Motores de *templates*

- FastTrac
- Velocity

↳ Linguagem de *templates*

- *Case sensitive*
- Português
- Identificadores, Componentes, Propriedades e Comandos

```
#para-cada (@include)
  #defina (@comentário = @include.comentário)
  <tr> <td width="773">
    <b>Arquivo incluído: @include.arquivo.@include.extensão
    #se(@comentário.possui(@autor))
      <i><b>Autor: </b> @comentário.autor</i>
    #fecha
  </td> </tr>
#fecha
```

↳ Especificação

- Léxica, Sintática e Semântica



Especificação do modelo de *templates*

- ↪ Os modelos de *templates* são compostos por
- Um arquivo descritor do modelo de *templates* (.TEM)
 - Um ou mais arquivos HTML contendo código estático e dinâmico

arquivo_frame\index.html	#complemento(_index.html)
arquivo_frame\menu.html	#complemento(_menu.html)
arquivo_frame\conteudo.html	#complemento(_sobre.html)

Nome ▲	Tamanho	Tipo
 arquivo_frame		Pasta de arquivos
 arquivo_frame.tem	1 KB	Arquivo TEM

C:\templates\arquivo_frame		
Nome ▲	Taman...	Tipo
 conteudo.html	4 KB	Microsoft HTML Document 5.0
 index.html	1 KB	Microsoft HTML Document 5.0
 menu.html	3 KB	Microsoft HTML Document 5.0

Especificação da ferramenta

- ↳ Enterprise Architect
 - Conceitos de orientação a objetos
 - UML
 - Diagrama de casos de uso
 - Diagrama de classes

↳ Diagrama de Casos de uso

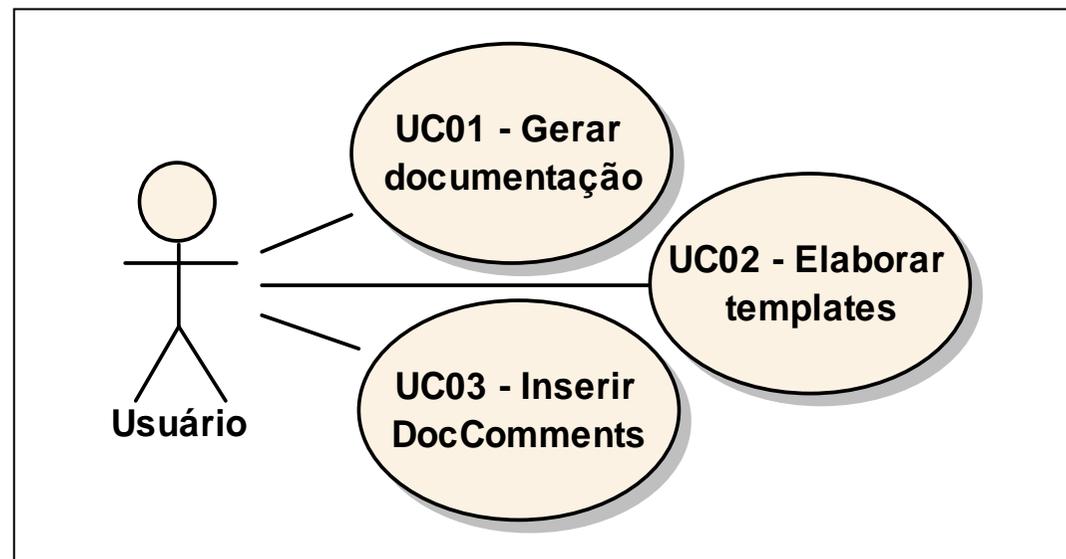
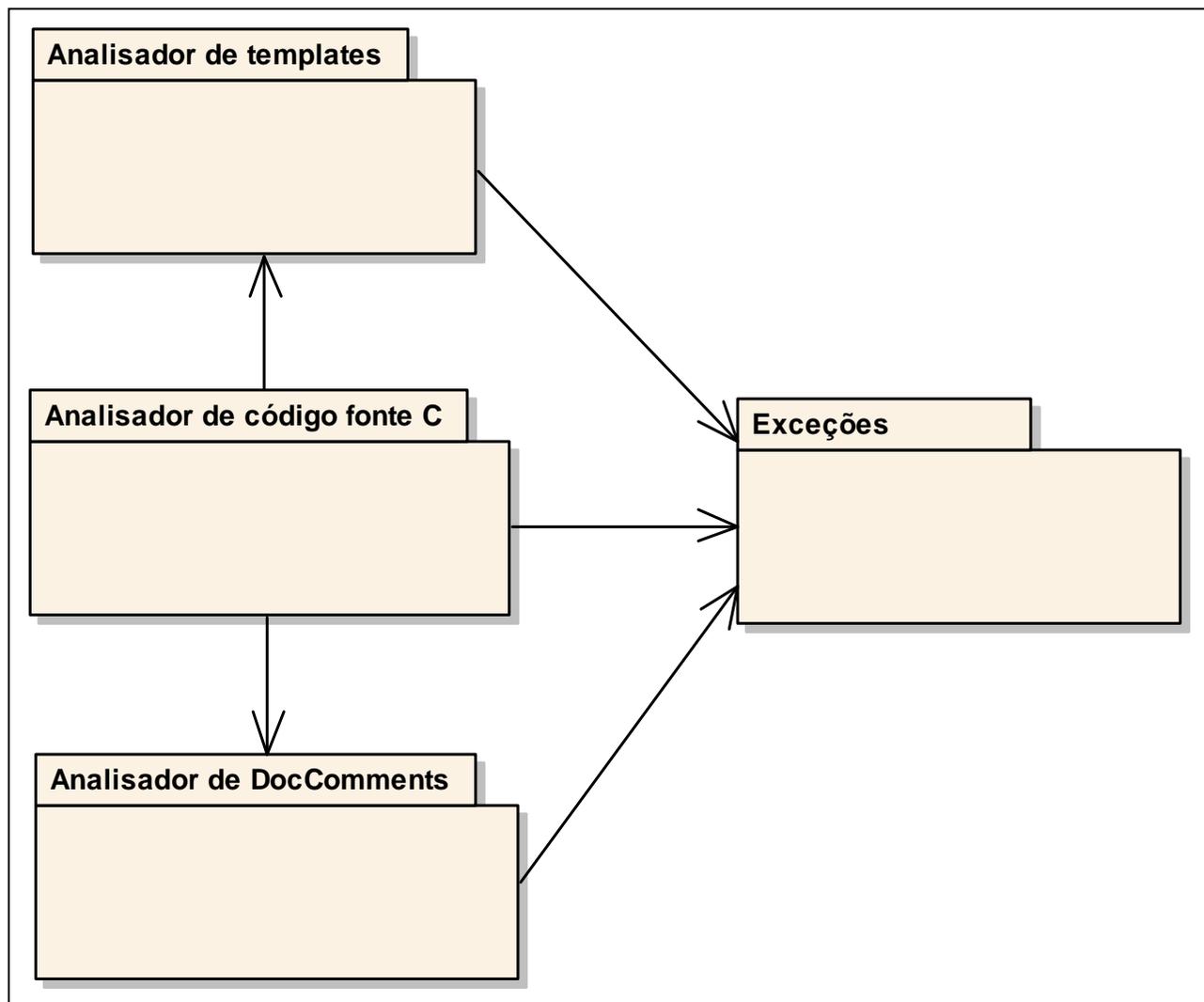




Diagrama de Classes

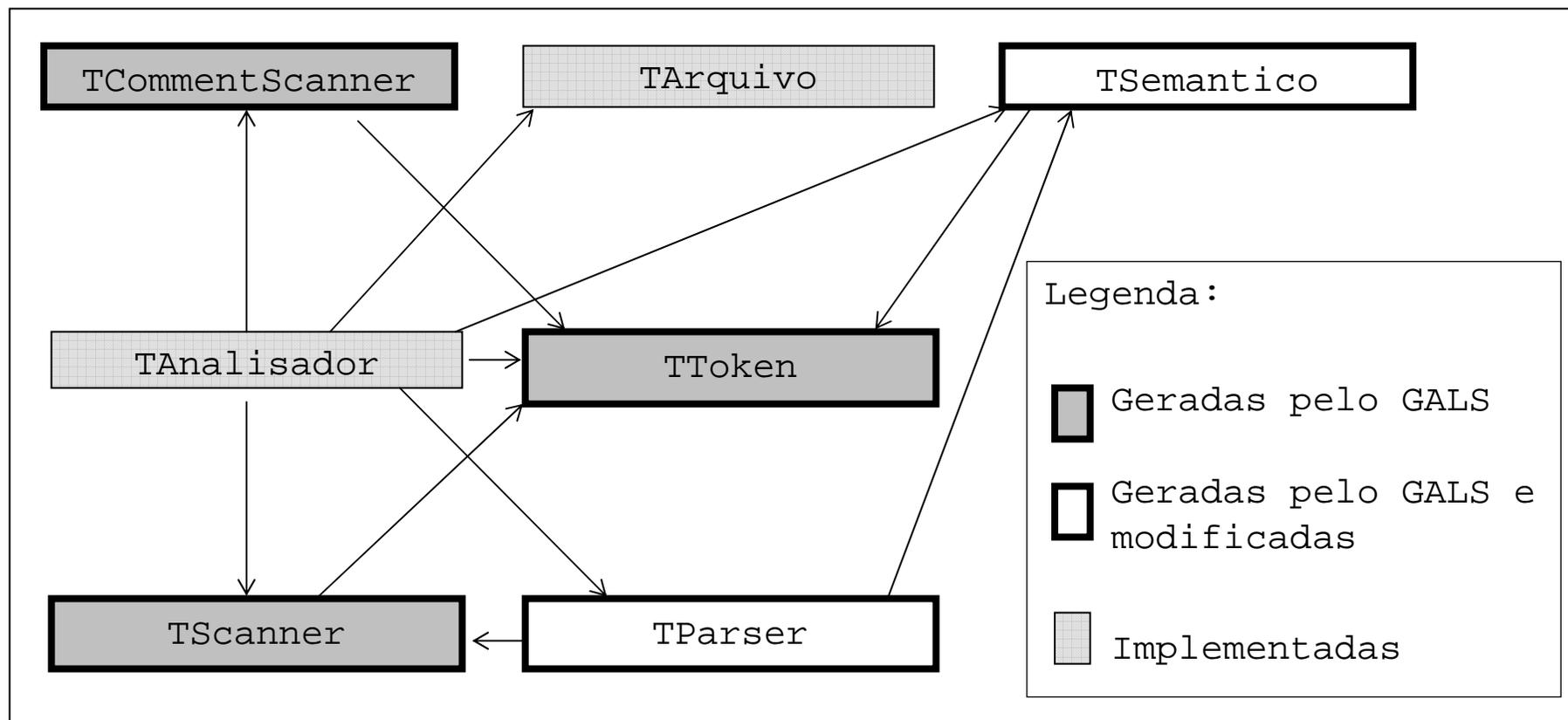
↳ Pacotes





Pacotes

↳ Analisador do código fonte C





Implementação

↳ Técnicas e ferramentas utilizadas

- *Object Pascal*
- Delphi
- GALS

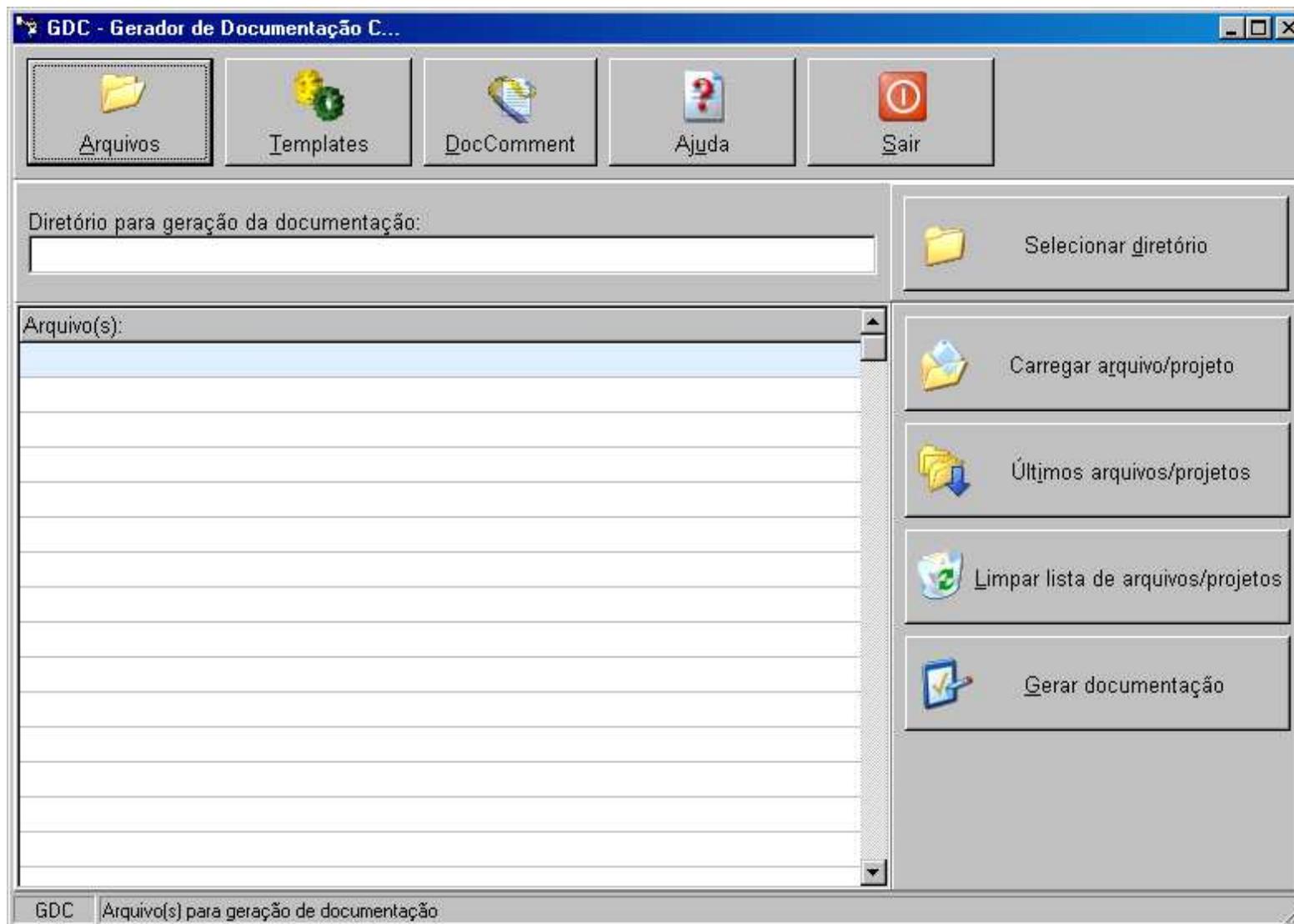
↳ Implementação

- Analisadores do código fonte C e dos *DocComments*
- Análise dos *templates* e geração da documentação



Operacionalidade

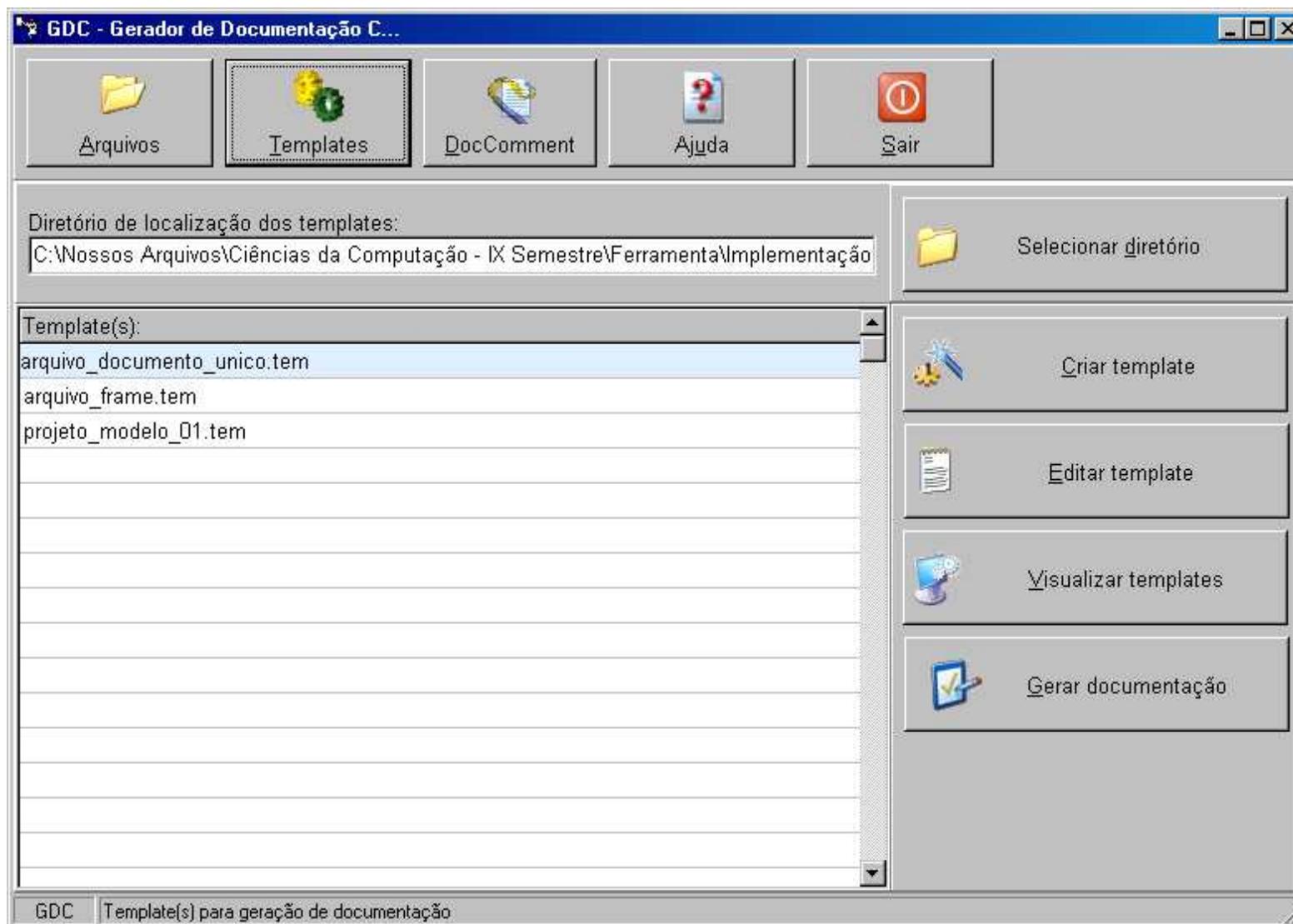
↪ Interface de trabalho com arquivos





Operacionalidade

↪ Interface de trabalho com modelos de *templates*





Resultados e discussão

↪ Três versões

↪ Testes

- Implementação
- Acadêmicos de Ciências da Computação e Sistemas de Informação

↪ Avaliado através de questionário com 16 questões

- 25 acadêmicos
- 13 de Ciências da Computação
- 12 de Sistemas de Informação

↪ A avaliação foi positiva

- Destacou a qualidade do GDC
- Promoveu a participação dos alunos

↪ Os erros reportados foram analisados e observou-se que tratavam-se de erros na escrita dos *DocComments*



Comparativo

Ferramenta	Linguagem	Uso de <i>templates</i>	Formato da saída	Tags da ferramenta	Tags HTML
JavaDoc	Java	Não	HTML	Sim	Sim
Gerador de documentação do C#	C#	Não	XML	Sim	Não, mas não há limites para os XMLs
CDoc	C C++ Java	Não	HTML	Não	Não
Doxygen	C C++ Java Objective C	Não	HTML RTF PDF XML	Sim	Não
GDC	C ANSI	Sim	HTML	Sim	Sim



Conclusão

- ↳ Marcações especiais
 - Comentários de blocos
 - Delimitam informações

- ↳ Documentação no formato HTML
 - *Templates*

- ↳ Para códigos fonte não comentados
 - Documentação com características básicas

- ↳ Para códigos fonte comentados
 - Documentos mais complexos



Conclusão (continuação)

↳ Informações para documentação

- Análise e coleta de informações no código fonte dos programas em C
- Tabela de símbolos
 - Todos os *tokens* reconhecidos no código fonte
 - Comentários
- *DocComments*
- *Templates*

↳ Modelos de *templates*

- Documentação gerada padronizada
- Liberdade de selecionar informações e ordem de apresentação



Extensões

- ↪ Melhorar a inserção de *DocComments*
- ↪ Melhorar as mensagens de erros
- ↪ Implementar um editor gráfico para os *templates*
- ↪ Informar quais funções de um include estão sendo utilizadas
- ↪ Implementar um controle de versões
- ↪ Adaptar a ferramenta permitindo a geração do algoritmo do programa
- ↪ Incluir na ferramenta a opção de seleção de idioma
- ↪ Especificar e implementar um motor de *templates* genérico para Delphi



Obrigado!!!