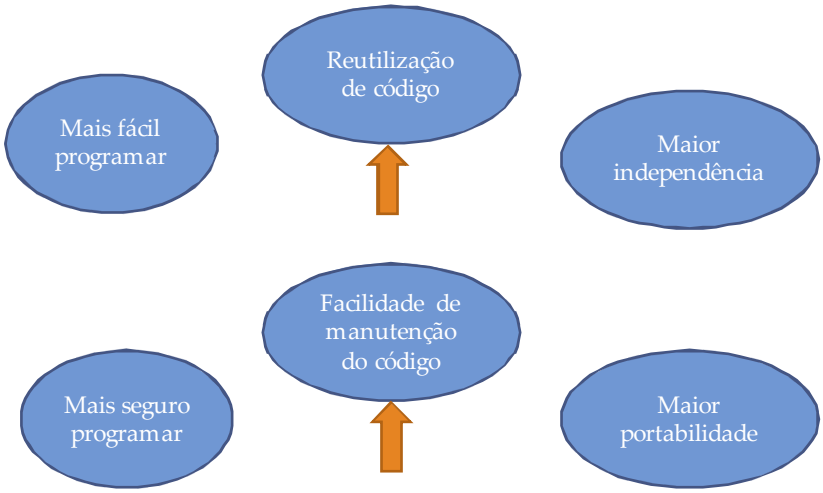
DOCUMENTAÇÃO DE CÓDIGO FONTE

Fernando Alva Manchego – estagiário PAE
email: falva@icmc.usp.br

Vantagens de usar TADs



Documentação de código fonte

- Permite entender o código fonte.
- Não suja o código com muitos comentários
- Não tenta explicar **como faz** e sim **o que faz**

3

Exemplo

```
// Este é o arquivo incluído
#include <stdio.h>
#include <stdio.h>

int main() {

    printf("Hello world\n");

    return 0;
}
}
// Final do arquivo
```

Que documentar???

4

Código Auto Documentado

- Código que se explica a sim mesmo sem a necessidade de documentação externa

```
/* Esta função calcula a seguinte posição
   dada a velocidade e o ángulo */

int calcularSeguintePosicao(int velocidade, int angulo)
```

O código é feito para humanos, não para computadores

5

Nomes significativos

- Nomes não abreviados e corretamente escritos.

```
NumeroConta
VelocidadeLuz
Radio
```

- Usar termos do domínio quando seja apropriado.

```
Max
Min
Url
```

6

Comentários significativos

- Compreensível e indispensável

```
numLines++; // incrementar o número de linhas
```

Comentários que simplesmente repetem o que o código diz não são significativos

7

Onde coloco comentários?

- Cabeçalhos de arquivos
- Cabeçalhos de funções

Comentários de documentação

- Quando o código seja incomum ou importante e não obvio

Comentários de implementação

8

Cabeçalhos de Arquivos

- Arquivos de implementação (.c) ou protótipos (.h)
- Sugestões:
  - Nome do arquivo
  - Descrição
  - Autor
  - Data de criação

```
/**
* Conjunto.h
* Estrutura de dados para representar um conjunto
* Autor: Fernando Alva Manchego
* Data: 30/09/2011 15:26
*
*/
```

9

Cabeçalhos de funções

- Antes de cada função (protótipo no .h e no .c)
- Sugestões:
  - Descrição
  - Parâmetros
  - Valor de retorno

```
/**
*
* Cria um conjunto vazio
* Parâmetros: piErro - retorna o estado final de função
*             (0 - SUCESSO, 1 - ERRO)
* Retorno: Conjunto - retorna um conjunto que será manipulado
*
*/
Conjunto* Conjunto_Criar(int *piErro)
```

10

Constantes, structs, typedef?

```c
/**Tamanho máximo de um conjunto*/
#define TAM 5

/** Tipo de dado que representa um elemento do conjunto*/
typedef int elem;

/** Estrutura de dados para representar un conjunto */
struct conjunto {
        /** Elementos do conjunto*/
        elem* v;
};

/** Tipo de dado definido com base na estrutura conjunto*/
typedef struct conjunto Conjunto;
```

11

Geradores de documentação

- Documentação de API (programadores)
- Guias de usuário final (usuários)

Doxygen

YARD
Yay! A Ruby Documentation Tool

Universal Report
State-of-the-art Documentation Solution

Javadoc

dbdesc
document your database!

<?php Documentor

///<devscribe>

12

Documentando com Doxygen

- C++, C, Java, Objective-C, Python, PHP, C#, ...
- Formatos:
  - HTML
  - LaTeX
  - RTF
  - MAN (linux)
  - XML
- Licença GNU

13

Como uso Doxygen?

- Blocos de comentários com marcas especiais adicionais

```
/**Tamanho máximo de um conjunto*/
#define TAM 5

/*!Tamanho máximo de um conjunto*/
#define TAM 5
```

- Comandos especiais
  - Começam com '\' ou '@' seguidos de argumentos (em alguns casos)

http://www.stack.nl/~dimitri/doxygen/commands.html

14

Doxygen: cabeçalhos de arquivos

```
/**
* Conjunto.h
* Estrutura de dados para representar um conjunto
* Autor: Fernando Alva Manchego
* Data: 30/09/2011 15:26
*
*/

/**
* @file Conjunto.h
* @brief Estrutura de dados para representar um conjunto
* @author Fernando Alva Manchego
* @date 30/09/2011 15:26
*
*/
```

15

Doxygen: cabeçalhos de funções

```
/**
*
* Cria um conjunto vazio
* Parámetros: piErro - retorna o estado final de função
*             (0 - SUCESSO, 1 - ERRO)
* Retorno: Conjunto - retorna um conjunto que será manipulado
*
*/
Conjunto* Conjunto_Criar(int *piErro)

/**
* @brief Cria um conjunto vazio
* @param piErro - retorna o estado final da função
*        (0 - SUCESSO, 1 - ERRO)
* @return Conjunto - retorna un cojunto que será manipulado
*/
Conjunto* Conjunto_Criar(int *piErro)
```

16

Doxygen: constantes, structs, typedef

```
/**
* @brief Tamanho máximo de um conjunto
*/
#define TAM 5

/**
* @brief Tipo de dado que representa um elemento do conjunto
*/
typedef int elem;

/**
* @brief Estrutura de dados para representar un conjunto
*/
struct conjunto {
        /** Elementos do conjunto*/
        elem* v;
};

/**
* @brief Tipo de dado definido com base na estrutura conjunto
*/
typedef struct conjunto Conjunto;
```

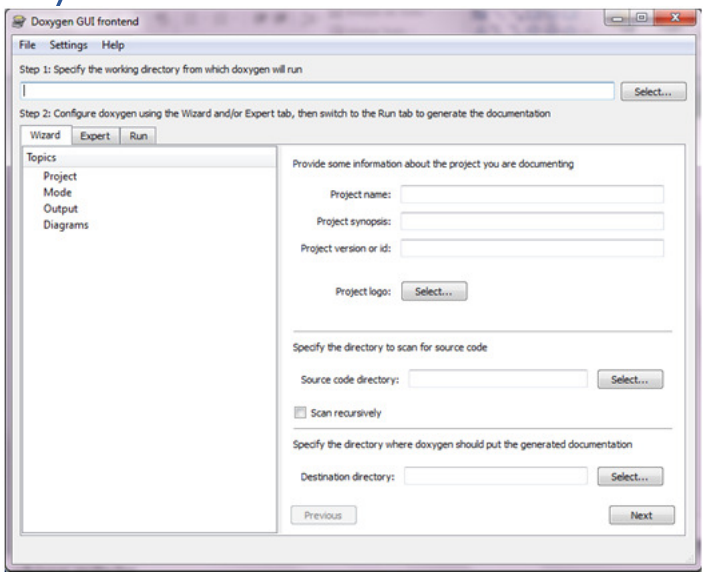
17

Doxygen: Baixar e instalar

- Para baixar:
  http://www.stack.nl/~dimitri/doxygen/download.html#latestsrc
- Há versões para Linux, Windows e MAC OS
- Manual:
  http://www.stack.nl/~dimitri/doxygen/manual.html
- **Doxywizard**

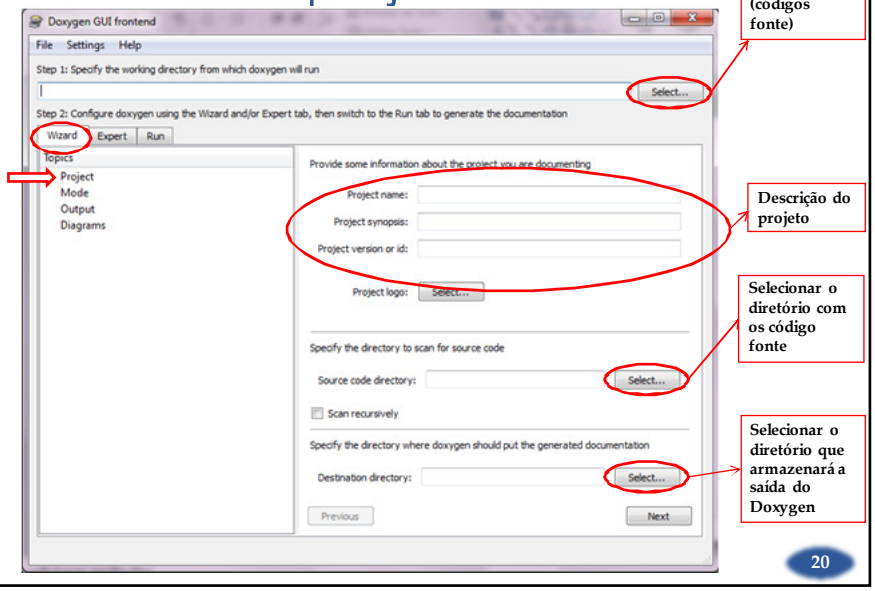
18

Doxywizard



19

Iniciando um projeto



Procurar o diretório do projeto (códigos fonte)

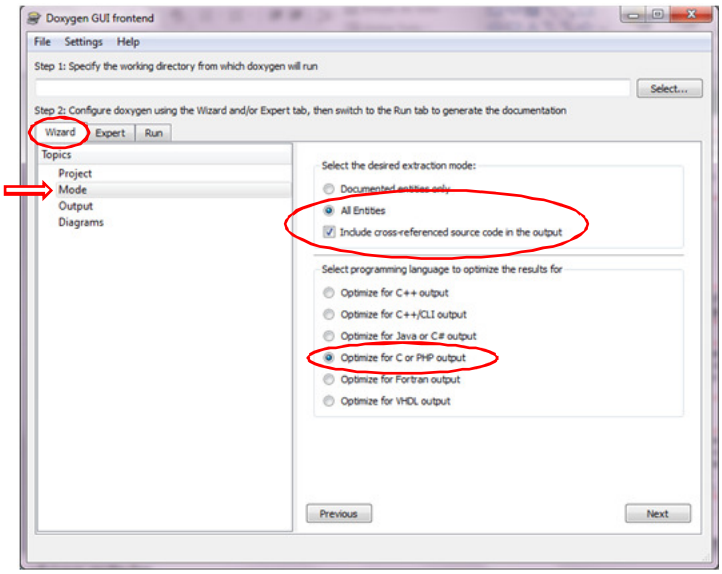
Descrição do projeto

Selecionar o diretório com os código fonte

Selecionar o diretório que armazenará a saída do Doxygen

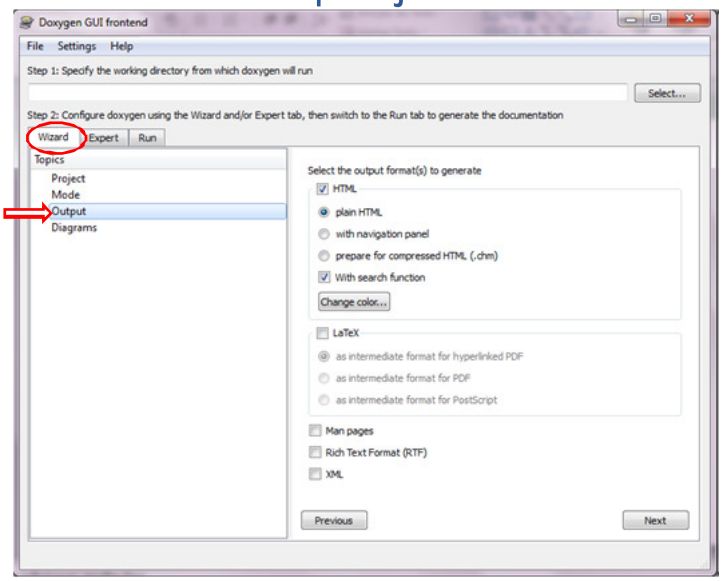
20

Iniciando um projeto



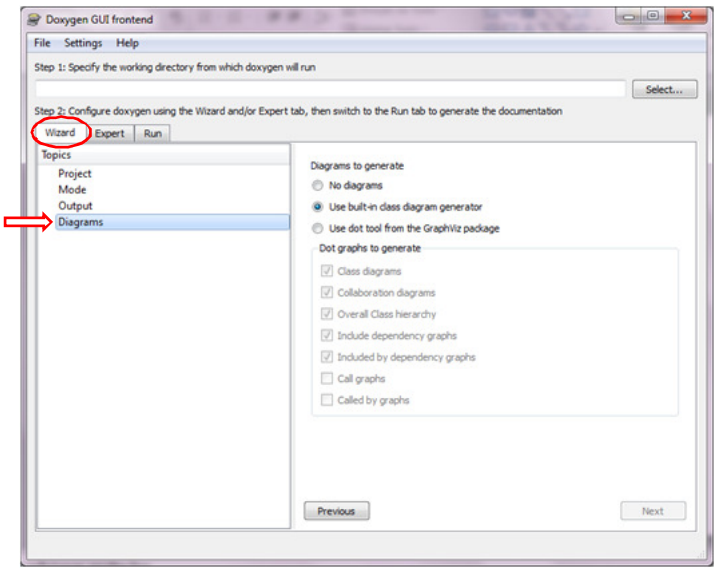
21

Iniciando um projeto

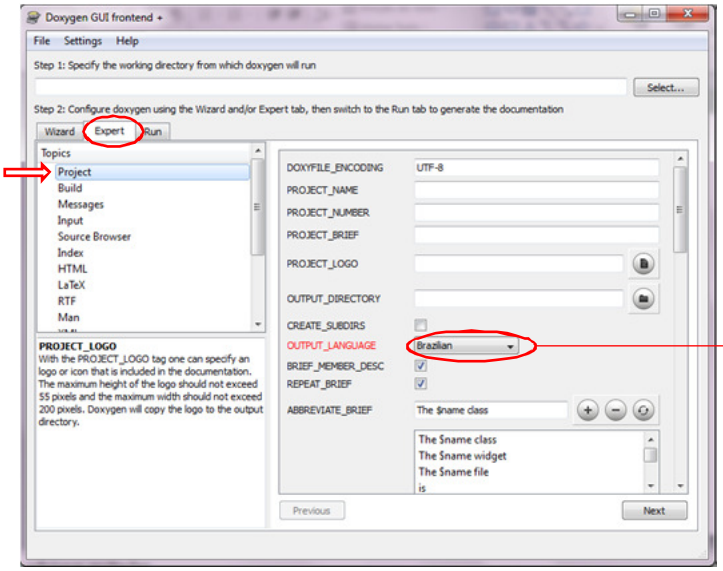


22

Iniciando um projeto

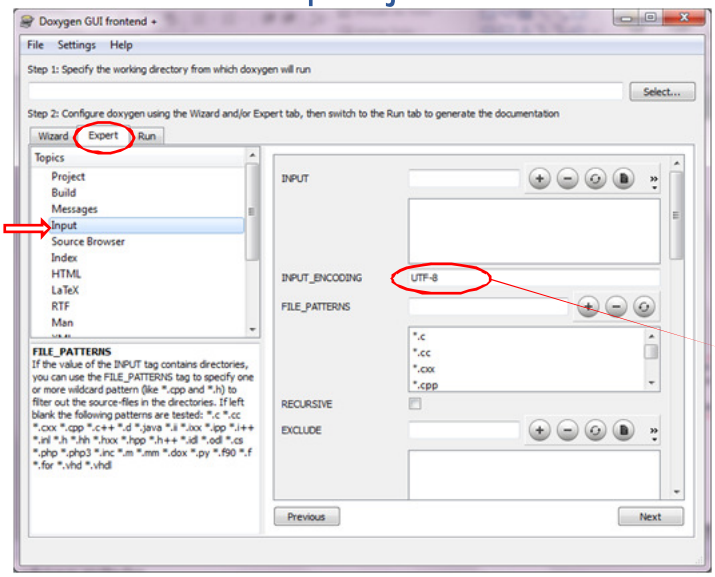


Iniciando um projeto



Para quem escolheu a língua portuguesa

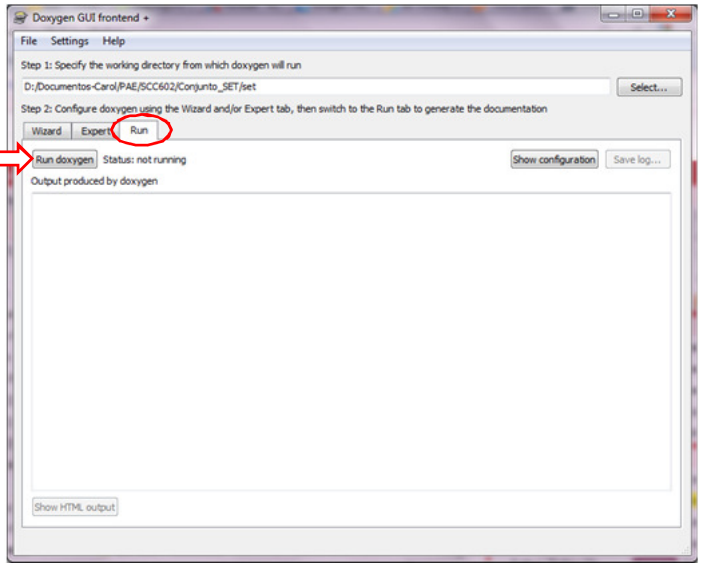
Iniciando um projeto



Para quem usa Windows: ISO-8859-1

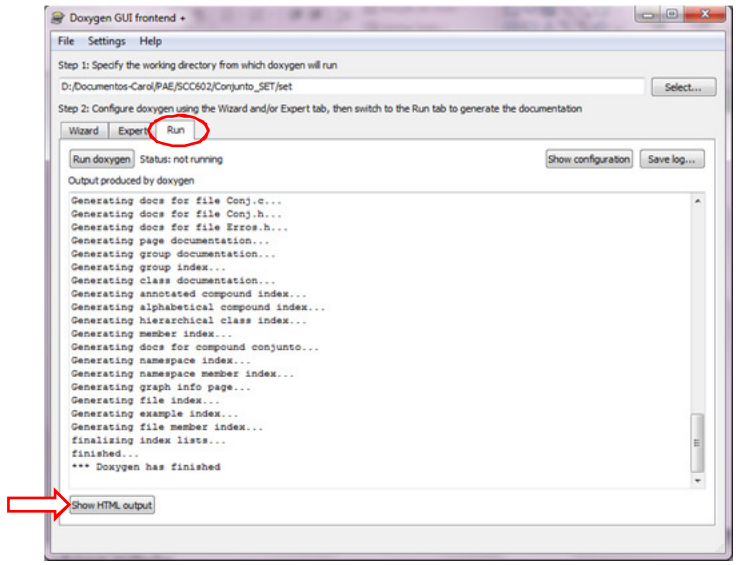
25

Iniciando um projeto



26

Iniciando um projeto



27

Doxygen: página principal

```
/**
* @mainpage
* @brief TAD Conjunto com as operações:
*        - criar um conjunto,
*        - união de dois conjuntos,
*        - intersecção de dois conjuntos,
*        - diferença de dois conjuntos,
*        - inserir um elemento em um conjunto,
*        - remover um elemento de um conjunto,
*        - verificar se um elemento é membro de um conjunto,
*        - atribuir o conteudo de um conjunto em outro,
*        - retornar o menor elemento de um conjunto,
*        - retornar o maior elemento de um conjunto,
*        - verificar se dois conjuntos sao iguais,
*        - verificar o tamanho de um conjunto,
*        - verificar se um conjunto é vazio,
*        - imprimir um conjunto na tela,
*        - liberar memória.
*
* @version 1.0
*
* @author Fernando Alva Manchego
*
* @date 30/09/2011 15:26
*
*/
```

28

Referências

- Dimitri van Heesch - Site do Doxygen: http://www.stack.nl/~dimitri/doxygen/ (acessado em 30/09/2011)

- Self Documenting Code: http://c2.com/cgi/wiki?SelfDocumentingCode (acessado em 30/09/2011)