

Lista de Exercícios 01

Prof. Dr. Rafael Alexandre
Algoritmos e Estruturas de Dados I.

27 de agosto de 2018

Instruções:

- i - Deve ser entregue um relatório de atividades (documento em PDF) descrevendo os passos seguidos para o desenvolvimento de cada um dos exercícios (Algoritmo Descritivo).
- ii - Deve ser utilizado o programa Code::Blocks para a compilação e testes dos algoritmos.
- iii - O arquivo deve ser entregue em formato ZIP seguindo a nomenclatura: "UFOP_EDI_TPZZ_XXXX_YYYY.zip" onde ZZ é o número identificador do trabalho, XXXX é o primeiro nome do aluno e YYYY o seu sobrenome.
- iv - Cada um dos exercícios deve criado em um diretório com o seguinte nome: Exercicio_XX onde XX é o número da questão que o algoritmo proposto está solucionando.
- v - A organização do trabalho deverá seguir as instruções definidas pelas figuras 1 a 8.
- vi - Para cada programa desenvolvido deverão ser entregues **SOMENTE** os arquivos com a extensão **".c"**, **".h"** e **".cbp"**.
- vii - O arquivo deve ser enviado via moodle limitado a data e hora de entrega definida no Plano de Ensino. Não serão aceitos trabalhos enviados por e-mail.

Questão 1. Considere a implementação de filas usando arranjos “circulares”. Escreva uma função `FuraFila(TipoFila* pFila, TipoItem x)` que insere um item na primeira posição da fila. O detalhe é que seu procedimento deve ser $O(1)$, ou seja, não pode movimentar os outros itens da fila. (observe que neste caso, estaremos desrespeitando o conceito de FILA – primeiro a entrar é o primeiro a sair).

Questão 2. Se um fila representada por arranjos (vetores) não é considerada circular, sugere-se que cada operação `Desenfileira` deve deslocar para “frente” todo elemento restante de uma fila. Um método alternativo é adiar o deslocamento até que “tras” seja igual ao último índice do vetor. Quando essa situação ocorre e faz-se uma tentativa de inserir um elemento na fila, a fila inteira é deslocada para “frente”, de modo que o primeiro elemento da fila fique na primeira posição do vetor, ou posição 0. Quais são as vantagens desse método sobre um deslocamento em cada operação `Desenfileira`? Quais as desvantagens? Reescreva as funções `Desenfileira`, `Enfileira` e `Vazia` usando esse novo método.

Questão 3. Como você implementaria uma fila de pilhas? Uma pilha de filas? Uma fila de filas? Escreva rotinas para implementar as operações corretas para cada uma destas estruturas de dados.

Questão 4. Implemente uma fila em C/C++, onde cada item da fila consista em um número variável de inteiros, ou seja, é possível que o usuário insira mais de um inteiro em uma mesma posição da fila.

Questão 5. Um deque é um conjunto de itens a partir do qual podem ser eliminados e inseridos itens em ambas as extremidades. Chame as duas extremidades de um deque `esq` e `dir`. Como um deque pode ser representado como um vetor em C/C++? Escreva quatro funções em C/C++, `RemDir`, `RemEsq`, `InsDir`, `InsEsq`, para remover e inserir elementos nas extremidades esquerda e direita de um deque. Certifique-se de que as funções funcionem corretamente para o deque vazio e detectem o estouro e o underflow (tentativa de remoção quando a fila está vazia). Quais as desvantagens dessa implementação com relação a implementação por encadeamento/alocação dinâmica?

Questão 6. Seja L uma lista simplesmente encadeada composta de números inteiros cujos nós são $l_1, l_2, l_3, \dots, l_n$. Escreva uma TAD que, per-

correndo L , uma única vez, construa uma outra lista L' formada dos seguintes elementos:

- A) $l_2, l_3, \dots, l_n, l_1$;
- B) l_n, l_{n-1}, \dots, l_1 ;
- C) $l_1 + l_n, l_2 + l_{n-1}, \dots, l_{n/2} + l_{n/2+1}$; onde n é par.

Questão 7. Acrescente às TADs Pilha, Fila e Lista, apresentadas em sala de aula, as seguintes operações:

- A) Verificar se um número pertence à TAD;
- B) Para a TAD Lista, inserir um novo elemento (número inteiros) mantendo a ordem;
- C) Copiar uma TAD T_1 para uma TAD T_2 ;
- D) Concatenar uma TAD T_1 com uma TAD T_2 ;
- E) Intercalar T_1 e T_2 ;
- F) Construir uma TAD que é a interseção entre T_1 e T_2 ;
- G) Construir uma TAD que é a diferença entre T_1 e T_2 ;
- H) Verifique se os elementos da TAD estão ordenados;
- I) Mescle os elementos de uma Lista L_1 com uma lista L_2 , mantendo a ordenação na lista final (L_1);

Questão 8. Considere uma coleção de nomes de sites da web e seus respectivos links na Internet armazenados através de uma lista simplesmente encadeada. Escreva a respectiva estrutura e um método que, dado o nome de um site, busque o seu link correspondente na lista e ao mesmo tempo, incremente o número de buscas pelo site. O método de busca deverá mover o nó que contém o nome buscado para o início da lista respeitando o número de buscas, de forma que ele possa ser encontrado mais rapidamente na próxima vez que for buscado.

Questão 9. Escreva um programa que remova todos os elementos com chaves pares de uma lista encadeada.

Questão 10. Escreva um programa solicite ao usuário uma sequência de caracteres sem limite de máximo de tamanho e realize as seguintes operações usando uma pilha:

- A) Imprimir o texto na ordem inversa;
- B) Verificar se o texto é um palíndromo, ou seja, se a string é escrita da mesma maneira de frente para trás e de trás para frente. Ignore espaços e pontos.

Questão 11. Suponha que um dado problema requer o uso de duas pilhas, onde cada pilha suporta no máximo 50 elementos e em nenhum momento as duas pilhas terão juntas mais do que 80 elementos. Assim, é possível implementar as duas pilhas em um único vetor usando apenas 80 posições ao invés de 100. Implemente a estrutura de dados e as de empilhar e desempilhar para estas duas pilhas.

Questão 12. Dada uma lista encadeada de caracteres formada por uma sequência alternada de letras e dígitos, construa um método que retorne uma lista na qual as letras são mantidas na sequência original e os dígitos são colocados na ordem inversa. Exemplos:

- 1. A 1 E 5 T 7 W 8 G \longrightarrow A E T W G 8 7 5 1
- 2. 3 C 9 H 4 Q 6 \longrightarrow C H Q 6 4 9 3

Como mostram os exemplos, as letras devem ser mostradas primeiro, seguidas dos dígitos. Sugestões:

- 1. usar uma fila e uma pilha;
- 2. supor um método `ehDigito()` retorna booleano que retorna verdadeiro caso um caractere seja um dígito.

Questão 13. Construa um método que recebe uma lista encadeada de números inteiros e retorna uma lista sem repetições, ou seja, uma lista onde

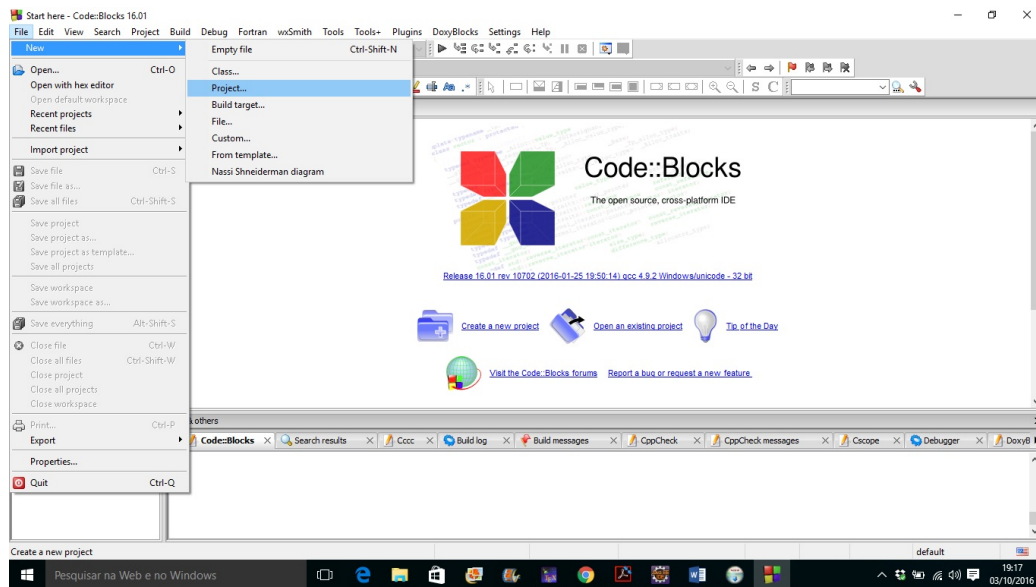


Figura 1: Tela inicial do software Code::Blocks. A figura demonstra o menu que deverá ser acessado para que seja criado um novo projeto.

cada número apareça apenas uma vez. Exemplo: 1 2 5 7 8 5 9 12 1 8 \rightarrow 1 2 5 7 8 9 1

A organização do trabalho faz parte da avaliação. As figuras a seguir auxiliam na criação de um novo projeto no Code::Blocks.

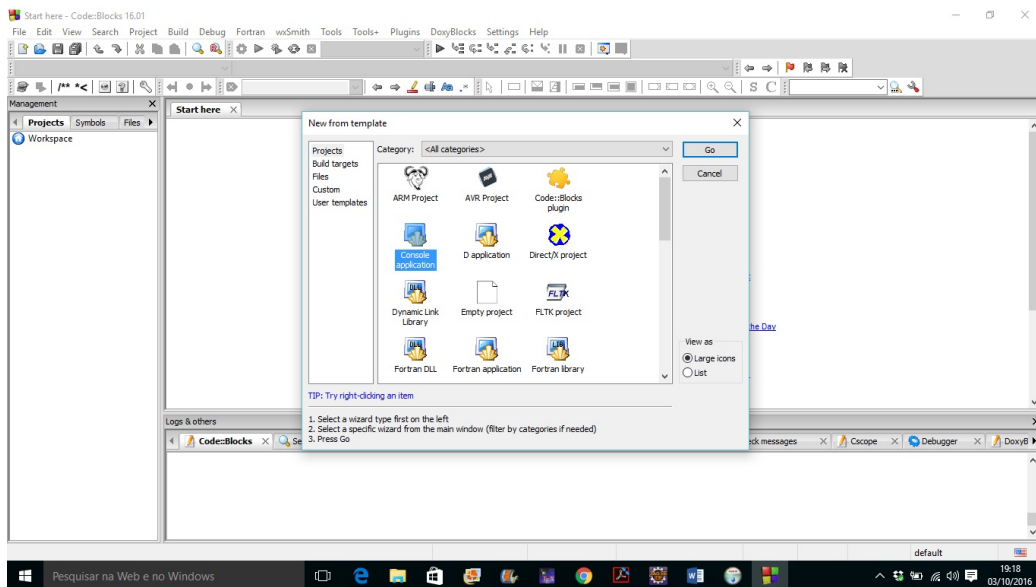


Figura 2: A opção "*Console application*" deverá ser selecionada.

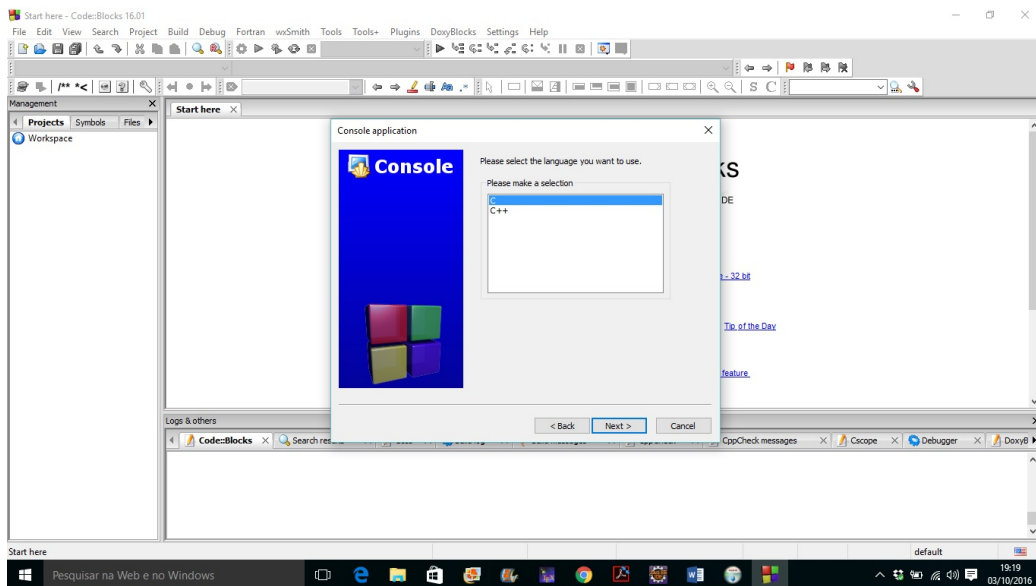


Figura 3: Nesta disciplina, CSI030, será adotada a linguagem "C". Portanto, nesta tela, deverá ser selecionado a opção "C".

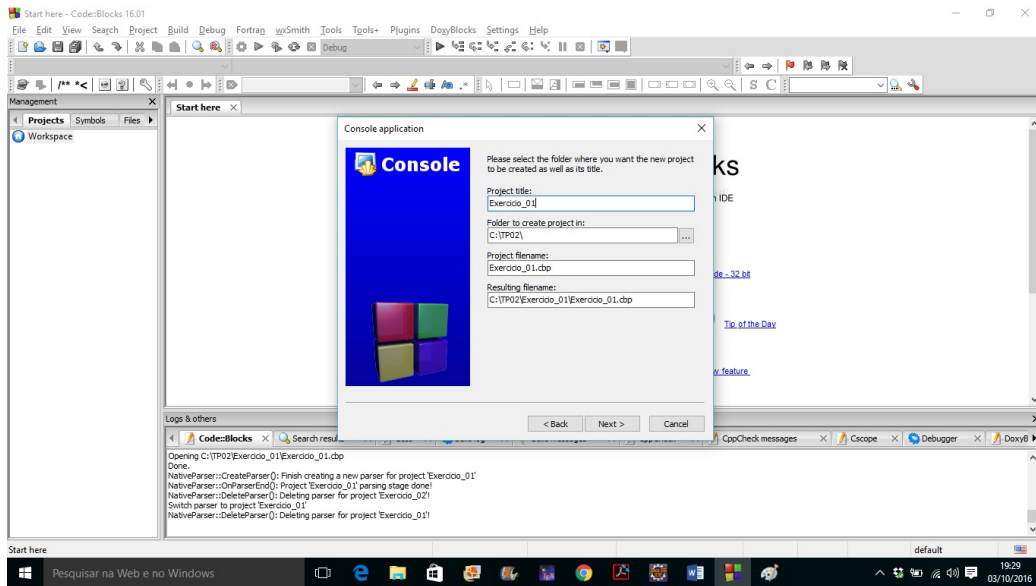


Figura 4: Nesta tela será possível definir o caminho, o nome do projeto que será criado e, em *Project filename*, o nome do arquivo que possui as referências utilizadas no projeto.

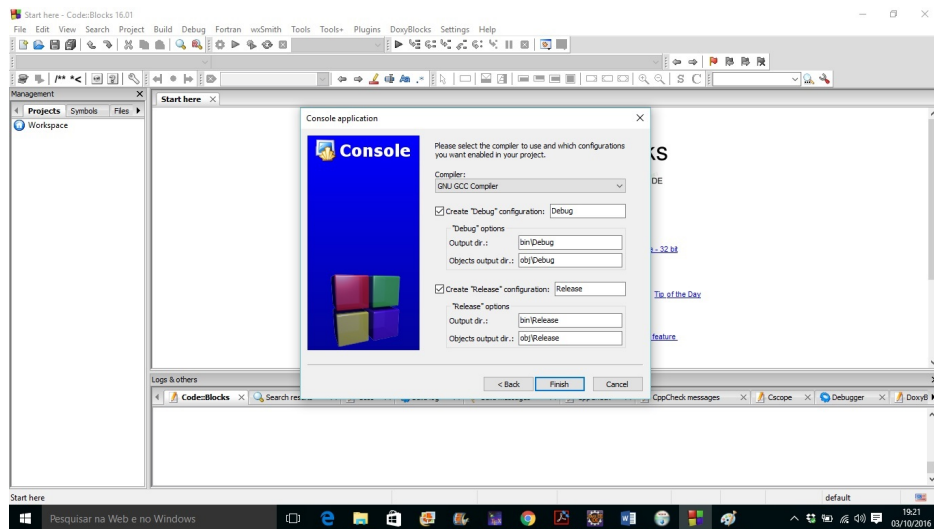


Figura 5: Nesta tela será possível definir o compilador utilizado e também saídas de *Debug* e *Release* do projeto. Normalmente a IDE já define as configurações corretas sem que seja necessário realizar alterações.

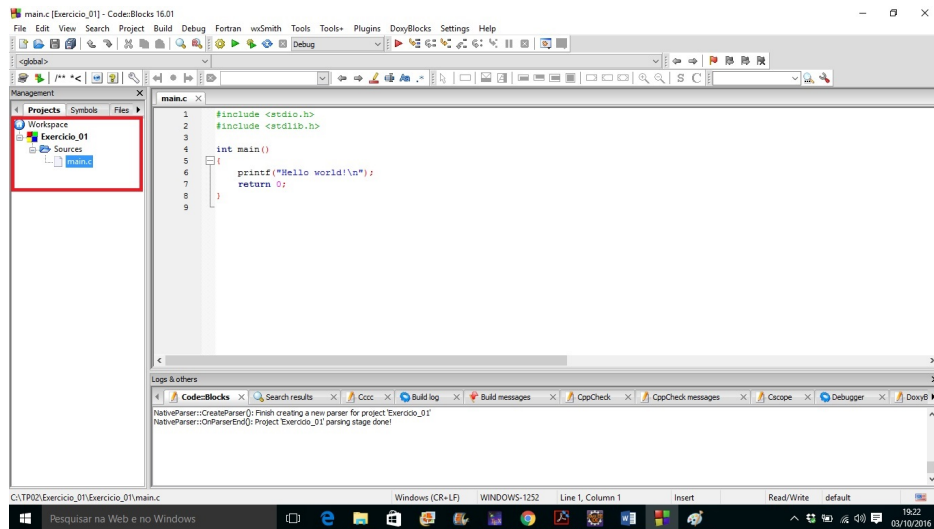


Figura 6: Destacado, em vermelho, a estrutura do projeto criado. Será criado um arquivo *main.c* com o código que possibilita a criação de um arquivo com a extensão *.exe* que escreve na tela a mensagem "Hello world!".

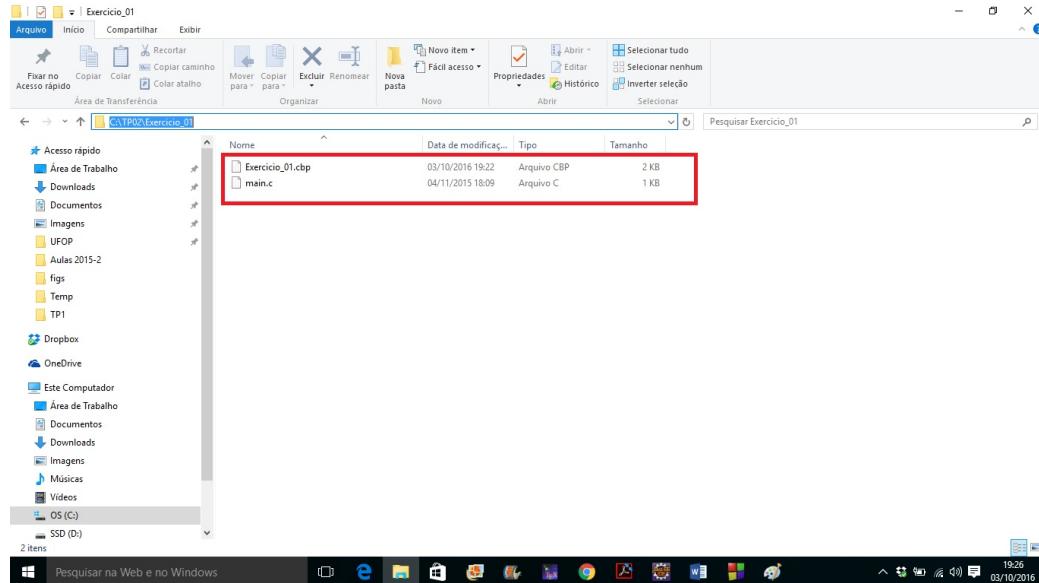


Figura 7: Dentro do diretório do exercício corrente deverá existir dois arquivos: *main.c* e *Exercicio_XX.cbp* em que o *XX* corresponde à numeração do exercício corrente.

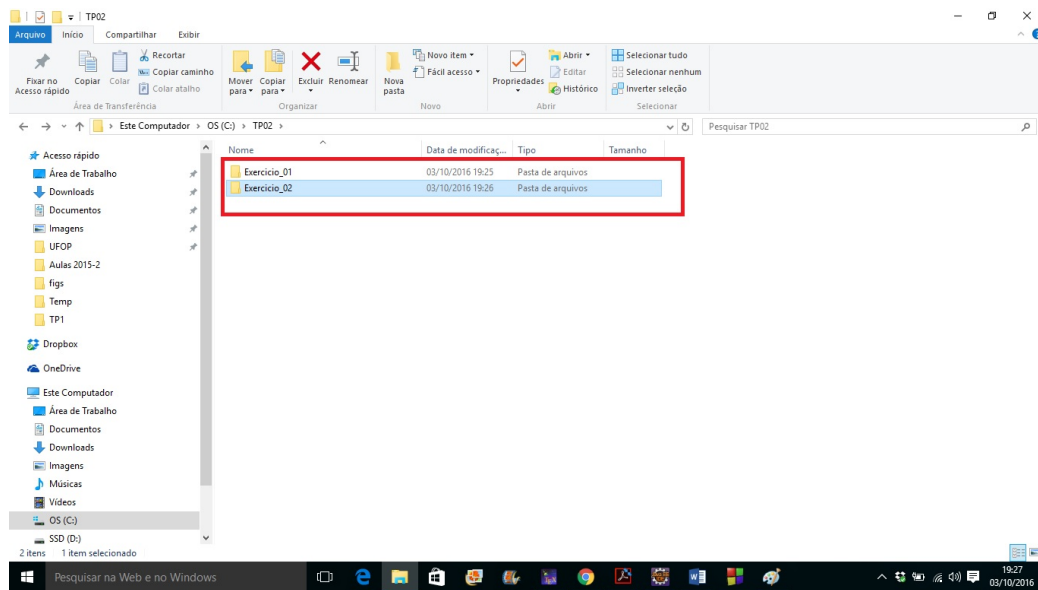


Figura 8: Estrutura de diretório dos exercícios. Dentro de cada uma das pastas deverá existir os arquivos main.c e Exercicio_XX.cbp.