

Igor Ortega Carmona
Estudante de Análise e Desenvolvimento de Sistemas (ADS)
RA: 00236524

Algoritmos Recursivos e Árvore Binária

Artigo sobre recursividade e árvore binária solicitado pelo professor Ivan de Andrade Santana para a matéria de *Algoritmos - Fundamentos de Programação de Computadores* grade do curso de *Análise e Desenvolvimento de Sistemas*.

UNIPAR - CIANORTE
Tecnologia em Análise e Desenvolvimento de Sistemas (ADS)

Cianorte
Setembro / 2022

RESUMO

Foram redigidos conceitos de recursividade explicando desde a sua teoria base até a implementação final sempre visando e lembrando dos cuidados a se tomar com a demonstração de exemplos na prática e detalhando o "*mind-set*" dos passos realizados para melhor entendimento.

Palavras-chave: Recursão, Árvore binária, linguagem C.

SUMÁRIO

1	Introdução	4
2	Objetivos	4
3	Fundamentação da Recursão	4
3.1	Como utilizar a Recursão	5
3.2	Cuidados a se tomar na Recursividade	5
4	Fundamentação de Árvore Binária	6
4.1	Compreensão de Nomenclaturas de Árvore comum	6
4.1.1	Nomenclaturas	6
4.1.2	Exemplos de uso na Computação	7
4.2	Conceitos sobre Árvore Binária, Árvore Estritamente Bi- nária, Árvore Estritamente Binária completa	7
4.2.1	Árvore Binária	7
4.2.2	Árvore Estritamente Binária	8
4.2.3	Árvore Binária Completa	9
5	Conclusão	10
6	Referências	11

1 INTRODUÇÃO

Durante todo o conteúdo do trabalho falaremos os principais conceitos e exemplos sobre recursividade e suas propriedades assim como a teoria básica de árvores binárias e suas formas de aplicação. Tendo em vista, sempre trazer ao leitor uma forma de escrita de fácil entendimento e transparência.

2 OBJETIVOS

O artigo tem como objetivo desmitificar o tema de recursão na programação em iniciantes em desenvolvimento *back-end*, assim como, motivar desenvolvedores junior em sua capacitação profissional de forma descomplicada e interativa.

3 FUNDAMENTAÇÃO DA RECURSÃO

Quando falamos em recursividade, é motivo de pânico para iniciantes na programação devido a complexidade de entendimento deste recurso. É preciso explicar com detalhes toda a sua ideia e praticar muito para realmente entender e se familiarizar, e com isso, passar também a criar algoritmos autênticos usando funções recursivas.

A recursão nada mais é do que você ter um grande problema e quebrar este mesmo problema em pequenas soluções para obter uma solução final. Entretanto, não é simplesmente fácil assim encontrar a solução deste grande problema, durante o artigo você entenderá melhor os cuidados a se tomar, possíveis formas de solução e também dicas na lógica para construir esse algoritmo.

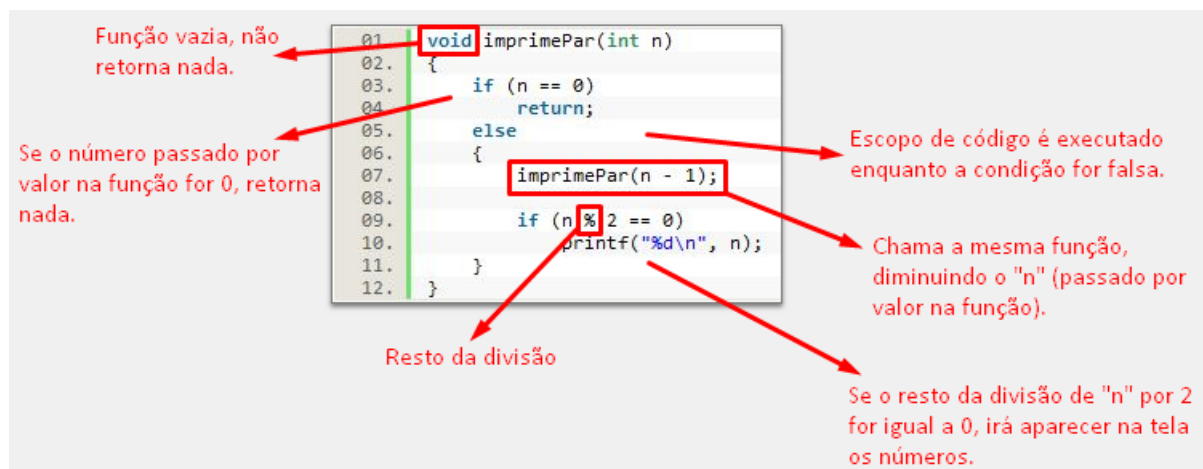


Figura 1 – Exemplo de algoritmo de impressão de número divisíveis por 2 na tela - Adaptado por Igor Carmona. (CARMONA, 2022)

Na figura acima, segue um exemplo de algoritmo recursivo em linguagem C. Vale

lembrar que independente da linguagem este recurso pode ser implementado basta seguir a sintaxe de cada linguagem.

3.1 COMO UTILIZAR A RECURSÃO

Assim como já foi dito, um algoritmo recursivo pode ser utilizado quando queremos a solução de um grande problema. Para saber a direção certa da construção do seu algoritmo deve-se pensar em duas coisas:

3.2 CUIDADOS A SE TOMAR NA RECURSIVIDADE

- O seu algoritmo deve ter **uma condição de parada**, isso porque conforme as chamadas da função forem sendo executadas elas são empilhadas na memória do computador até que um valor é retornado para a *main* e se caso não haja uma condição de pausa, você pode estourar a pilha de memória fazendo com que seu programa não execute mais. Após o retorno do valor pela função, o computador irá desempilhar as chamadas das funções e retornará o valor para cada valor chamado, resultando num valor total.
- Você deve padronizar o seu problema para uma forma genérica, um exemplo seria o fatorial de um número.

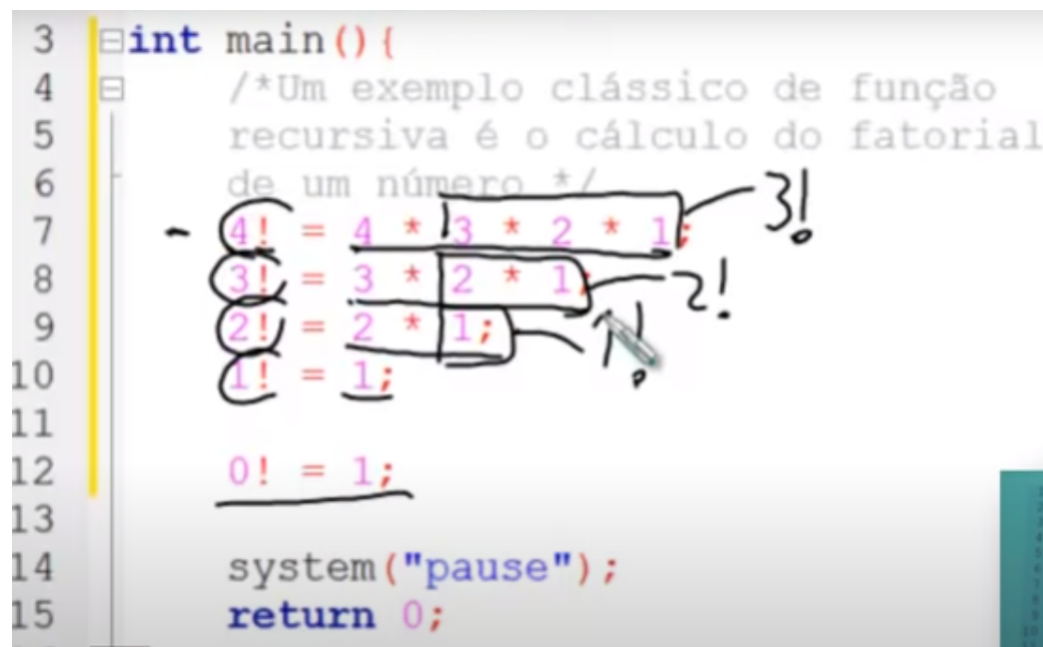
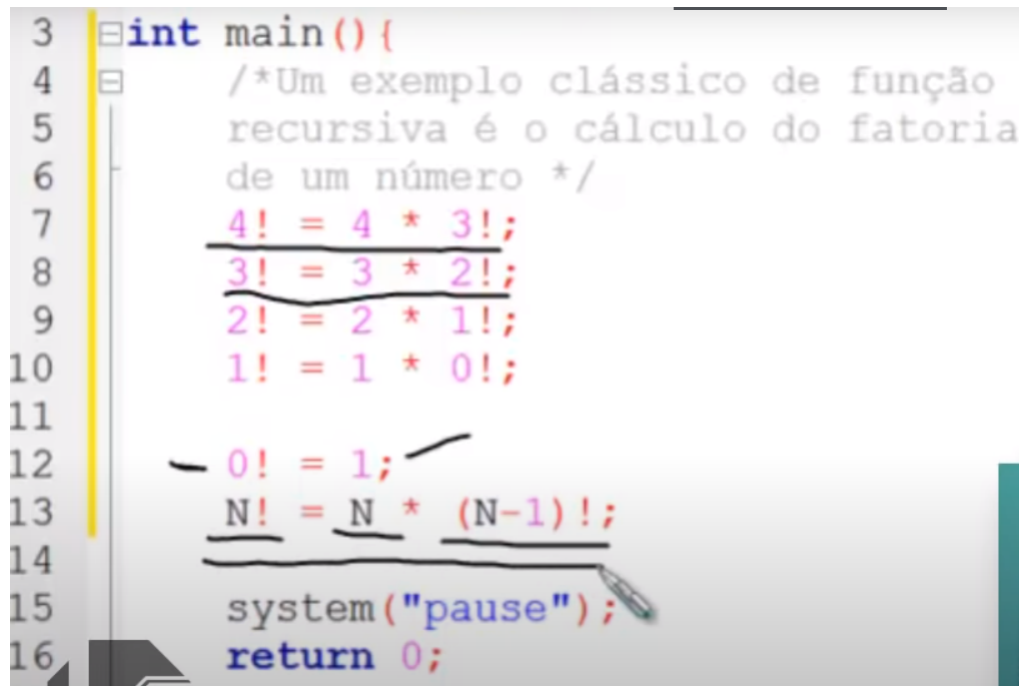


Figura 2 – Exemplo de forma genérica para um código de recursão de um fatorial (BACKES, 2012c)

Na imagem acima, temos um **caso base** e a **forma genérica** para montar um algoritmo recursivo. Observa-se que o fatorial de 4 é: 4 x **fatorial de 3**, enquanto é chamado a função passando "**n - 1**" temos: 3 x **fatorial de 2**, e assim por diante

até chegar no **caso base** de **fatorial de 0** e **retornar um valor** e a partir disso começar o **processo de desempilhamento de memória**.



```

3  int main() {
4      /*Um exemplo clássico de função
5       recursiva é o cálculo do fatorial
6       de um número */
7      4! = 4 * 3!;
8      3! = 3 * 2!;
9      2! = 2 * 1!;
10     1! = 1 * 0!;
11
12     0! = 1;
13     N! = N * (N-1)!;
14
15     system("pause");
16     return 0;

```

Figura 3 – Exemplo de forma genérica fatorial (BACKES, 2012a)

4 FUNDAMENTAÇÃO DE ÁRVORE BINÁRIA

4.1 COMPREENSÃO DE NOMENCLATURAS DE ÁRVORE COMUM

Para um melhor entendimento sobre o tipo de árvore binária, recomenda-se ter conhecimento sobre as propriedades básicas de uma árvore comum e seu conceito que estão mutuamente interligadas com a recursão. Algumas nomenclaturas frequentemente utilizadas:

4.1.1 NOMENCLATURAS

- **VÉRTICES:** É cada uma das entidades representadas na árvore. Numa árvore, os vértices são classificados em **níveis**, sendo o número de nós no caminho entre o **vértice** e a **raiz**.
- **ARESTAS:** É uma conexão entre dois vértices.
- **PAI:** É o antecessor imediato de um vértice.
- **FILHO:** É o sucessor imediato de um vértice. Dado um determinado vértice, cada **filho** seu é a **raiz** de uma nova **sub-árvore**.

- **RAIZ:** É o vértice que não possui **pai**.
- **FOLHAS:** Qualquer vértice que não possui filhos. A raiz até uma de suas folhas denomina-se **altura da árvore**.

4.1.2 EXEMPLOS DE USO NA COMPUTAÇÃO

- Estrutura de diretórios (pastas).
- Busca de dados armazenados no computador.
- Representação de espaço de soluções como em um jogo de xadrez.
- Modelagem de algoritmos.

4.2 CONCEITOS SOBRE ÁRVORE BINÁRIA, ÁRVORE ESTRITAMENTE BINÁRIA, ÁRVORE ESTRITAMENTE BINÁRIA COMPLETA

4.2.1 ÁRVORE BINÁRIA

A árvore binária pode-se dizer que é um tipo especial de árvore, onde:

- Cada vértice pode ter duas sub-árvores sendo uma **sub-árvore esquerda** e a outra **sub-árvore direita**.
- O grau de cada um de seus vértices (número de filhos) pode ser **0, 1 ou 2** e sempre sua capacidade máxima será de 2 filhos.

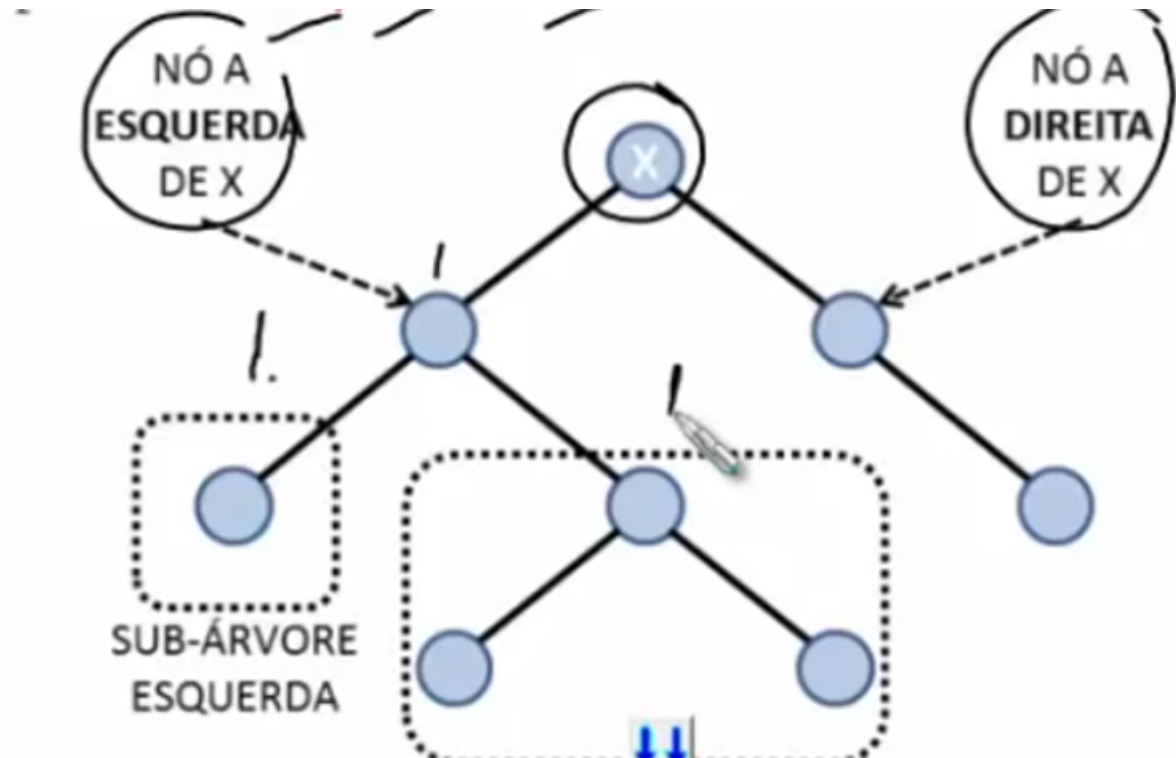


Figura 4 – Exemplo: representação em grafos da definição de árvore binária (BACKES, 2012b)

4.2.2 ÁRVORE ESTRITAMENTE BINÁRIA

Temos também, a **Árvore Estritamente Binária** onde:

- Cada nó (vértice) possui **0 ou 2** sub-árvores.
- Nenhum nó tem **filho único**.
- Nós internos (não folhas) sempre têm 2 filhos.

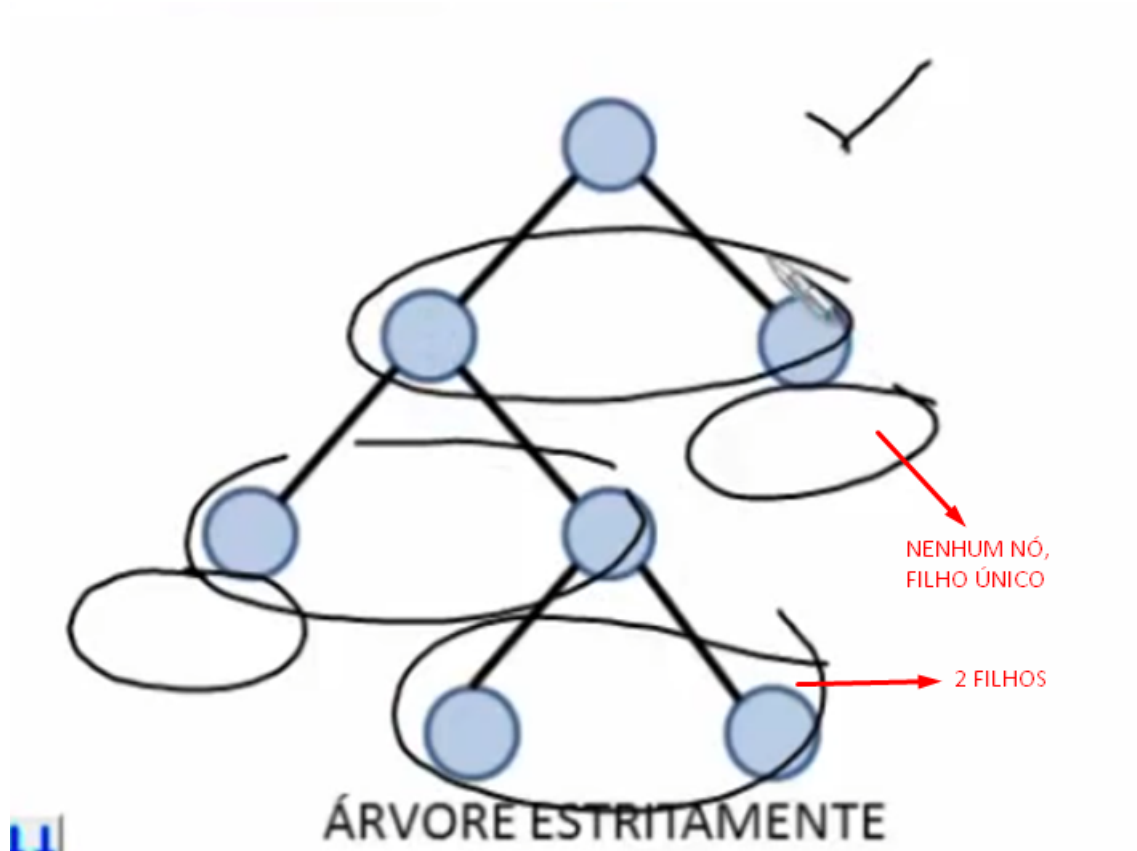


Figura 5 – Exemplo: representação em grafos de árvore estritamente binária (BACKES, 2012a)

4.2.3 ÁRVORE BINÁRIA COMPLETA

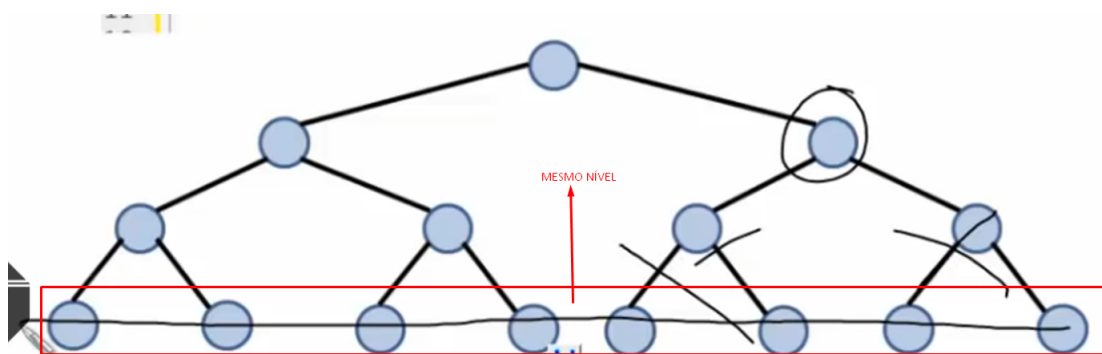


Figura 6 – Exemplo: representação em grafos de árvore binária completa (BACKES, 2012b)

Na árvore binária completa tem um diferencial que é você poder fazer o cálculo do número de nós, na qual a fórmula matemática é 2^{h-1} , sendo "h" a altura da árvore.

- É estritamente binária.
- Todos os seus **nós-folha** estão no mesmo nível.

5 CONCLUSÃO

Há diversos outros tipos de árvores e aplicações de recursão para serem demonstradas mas a didática deste trabalho em si é apresentar conceitos, exemplificar e mostrar um novo mundo a se explorar.

Vale lembrar, que foram dissertados assuntos relacionados a teoria de recursão e de árvores como uma forma de abrir o seu "mind-set". A complexidade na construção de algoritmos referente a esses temas foram abstratas para uma melhor compreensão dos iniciantes à programação.

6 REFERÊNCIAS

BACKES, A. *FIGURA 3 - Programação Descomplicada (Linguagem C) - Aula 51 - Recursão: Definição - Exemplo: Forma genérica fatorial*. 2012. <<https://youtu.be/FH5lCr-RVWE>>. Acessado em: 15 de setembro de 2022. 6

BACKES, A. *FIGURA 4 - Programação Descomplicada (Estrutura de Dados) - Aula 69 - Árvore binária: Definição - Exemplo: representação em grafo da definição de árvore binária*. 2012. <<https://youtu.be/9WxCeWX9qDs>>. Acessado em: 27 de setembro de 2022. 8

BACKES, A. *FIGURA 5 - (Adaptação Igor Carmona) - Programação Descomplicada (Estrutura de Dados) - Aula 69 - Árvore binária: Definição - Exemplo: representação em grafos de árvore estritamente binária*. 2012. Adaptação de Igor Carmona. Disponível em: <<https://youtu.be/9WxCeWX9qDs>>. 9

BACKES, A. *FIGURA 6 - (Adaptação Igor Carmona) - Programação Descomplicada (Estrutura de Dados) - Aula 69 - Árvore binária: Definição - Exemplo: representação em grafos de árvore binária completa*. 2012. Adaptação de Igor Carmona. Disponível em: <<https://youtu.be/9WxCeWX9qDs>>. 9

BACKES, A. *Programação Descomplicada (Linguagem C) - Aula 51 - Recursão: Definição - Exemplo: forma genérica de código recursivo do fatorial*. 2012. Disponível em: <<https://youtu.be/FH5lCr-RVWE>>. 5

CARMONA, I. *Responde Ai: Definindo funções recursivas - Exemplo: Algoritmo recursivo de números pares*. 2022. Adaptação Igor Carmona. Disponível em: <<https://respondeai.com.br/conteudo/definindo-funcoes-recursivas-exemplo-algoritmo-recursivo-de-numeros-pares>>. 4