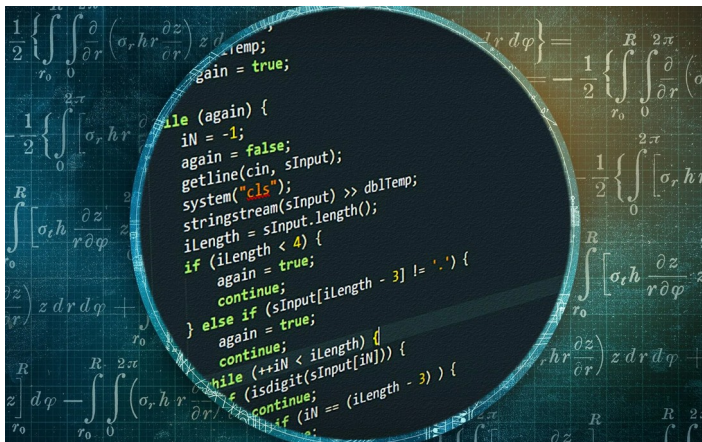


ACH2002

INTRODUÇÃO À ANÁLISE DE ALGORITMOS



Aula 1

Profa. Ariane Machado Lima
ariane.machado@usp.br

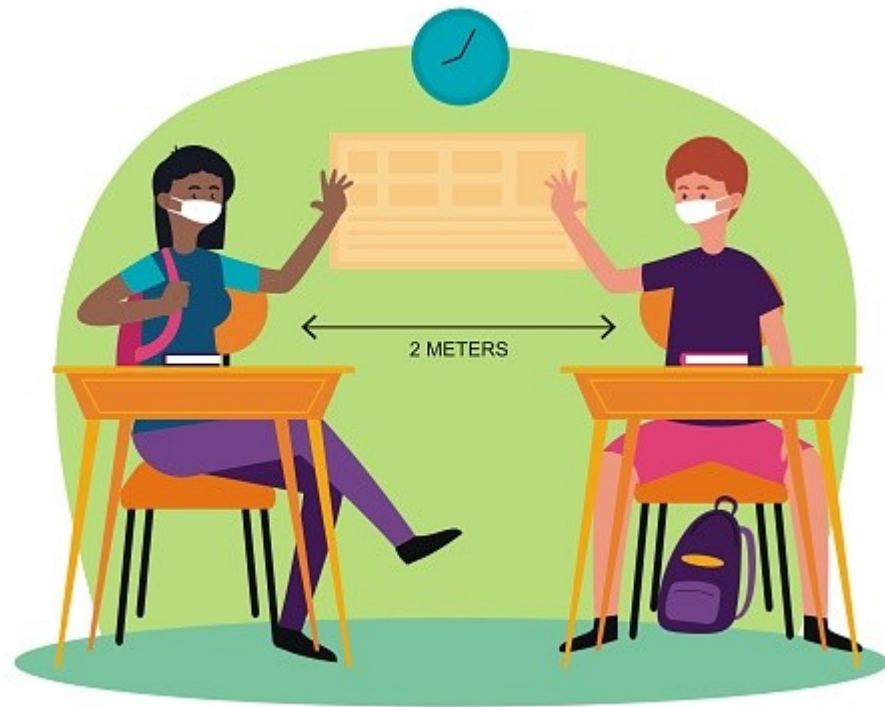
Aula 1 – Papo inicial

- Importância da disciplina e seu funcionamento
- Assuntos diversos importantes para a disciplina, vida acadêmica e além



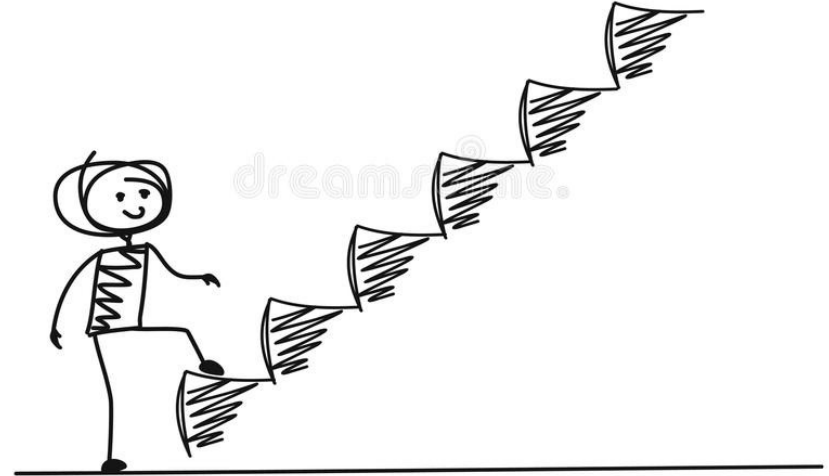
Mantendo os cuidados...

- Janelas abertas (já abram quando chegarem) e ventiladores ligados sempre que possível
 - Venham agasalhados



Conhecendo vocês um pouco

- Estão gostando do curso de SI? O que esperam dele?
- Como foram na disciplina de Introdução a Programação?
- Qual linguagem de programação conhecem?
- Quem venho de Ensino Médio técnico e normal?



Alinhando as expectativas (do curso e da disciplina)

Disciplina: ACH2002 - Introdução à Análise de Algoritmos
Introduction to Analysis of Algorithms

Créditos Aula: 4

Créditos Trabalho: 2

Carga Horária Total: 120 h

Tipo: Semestral

Ativação: 01/01/2021 **Desativação:**

Objetivos

Familiarizar o aluno com diferentes técnicas de desenvolvimento de programas corretos e bem estruturados, com o estudo de ferramentas matemáticas úteis para a análise de algoritmos e com a análise da complexidade assintótica de algoritmos.

Alinhando as expectativas (do curso e da disciplina)

Programa

Técnicas de desenvolvimento de algoritmos baseadas em indução fraca e forte, tentativa e erro, técnicas gulosas e programação dinâmica. Elementos de análise assintótica de algoritmos (notação O , Ω e Θ) iterativos e recursivos. Desenvolvimento e análise da complexidade de diferentes algoritmos de ordenação (BubbleSort, InsertionSort, SelectionSort, QuickSort, MergeSort, HeapSort, CountingSort e RadixSort).

Plataforma de apoio



- Edisciplinas
 - Slides de aulas, materiais extras
 - Avisos
 - Fórum de discussão
 - Postagem de notas
 - Submissão de trabalhos

Plataforma de apoio



- Edisciplinas
 - Sobre os slides de aulas:
 - Coloco uma versão **preliminar** no início da aula, só para quem quiser fazer anotações no pdf durante a aula
 - Peço para não olharem slides seguintes ao que eu estiver mostrando em aula
 - e para ignorarem os slides finais que possam não ter sido abordados em aula (a não ser que eu recomende o contrário)
 - Coloco a versão final até às 14h – essa deve ser integralmente estudada até a próxima aula

Avaliação

- MP = média ponderada de 2 provas: $(2 \cdot P_1 + 3 \cdot P_2)/5$
 - Uma prova substitutiva (FECHADA) – substitui a prova em que faltou
- MT = média simples de 2 trabalhos (EP's – exercícios-programa)
 - Individuais
 - Linguagem C

- **Média 1a. Aval:**

se $(MP < 5)$ ou $(MT < 5)$ $M1 = \min(MP, MT)$

senão $M1 = (7 \cdot MP + 3 \cdot MT)/10$

Presença mínima obrigatória:

70% → no máximo 9 faltas

- **Média 2a aval:** $(M1 + REC)/2$

- **Provas Sub e Rec:**

- Cai a matéria toda

- **Super-hiper-mega difíceis**

Profa. Arianne Machado Lima

Material didático

Bibliografia

Bibliografia - Livro texto:

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. "Algoritmos - Teoria e Prática", GEN LTC, 3ª edição, 2012.
ZIVIANI, N. "Projeto de Algoritmos", Cengage Learning, 3ª edição revista e ampliada, 2011.
FEOFILOFF, P. "Algoritmos em linguagem C", Campus/Elsevier, 1ª edição, 2009.

Bibliografia Complementar:

KON, F; GOLDMAN, A. e SILVA, P.J.S. "Introdução à Ciência de Computação com Java e Orientação a Objetos", IME – USP, 2004.

Listas de exercícios

- FAZER listas de exercícios é a melhor forma de estudar, mas não haverá entrega
 - Podem tirar dúvidas comigo nos horários de aula (antes, durante e depois)

Lista de exercício de hoje ;-)

- Acessar nosso ambiente no edisciplinas
- Olhar os materiais extra-curriculares (além dos mencionados nesta aula, ver o que está no edisciplinas)

Participação



- Se você não **participar** da aula perderá o seu tempo!
 - Aprender é diferente de decorar, e para aprender é preciso raciocinar.

Participação



- Se você não **participar** da aula perderá o seu tempo!
 - Aprender é diferente de decorar, e para aprender é preciso raciocinar.

- FORTE RECOMENDAÇÃO:

LÁPIS, BORRACHA E CADERNO !!!!



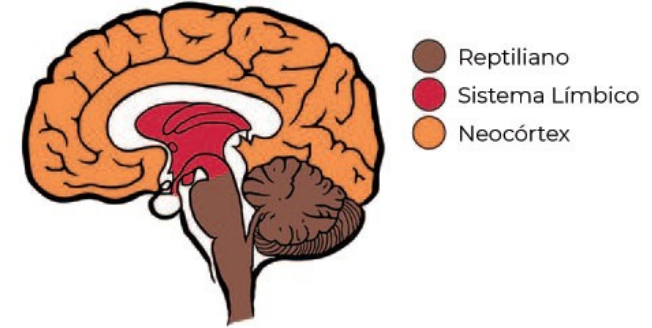


Pontualidade



- Costumo ser pontual
- No início de cada aula costumo:
 - Fazer um link com as aulas anteriores
 - Mostrar uma motivação sobre o tema a ser estudado (perguntem quando eu não for clara)
- Não percam esse tempo precioso!

FOCO (em aula e durante os estudos)



SILENCIEM CELULAR, inclusive notificações! Inclusive do whatsapp!!!

Interrupções sabotam você!!! (falta de *nobreak* / sistema límbico)

<https://open.spotify.com/episode/2bHDTwrW8euAITB74IjsGu?si=9273a6bb6dad4862>

Estipulem horários / criem uma rotina de estudos

Carreira de Sucesso



Carreira de Sucesso



**SUCESSO A LONGO PRAZO PRECISA SER
SUSTENTÁVEL....**

4 Pilares da Saúde (no mínimo 4 pratinhos, além da família, etc...)

VIDA VEDA



Atividade
Física

saúde



Alimentação

Sono



Silêncio/Presença



Tudo isso é papo sério



- Coursera – curso da Yale: The Science of Well-Being
<https://www.coursera.org/learn/the-science-of-well-being>

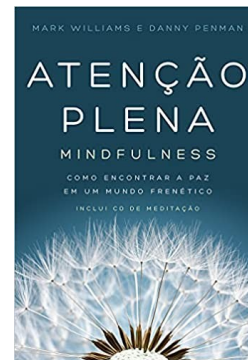
Blue zones



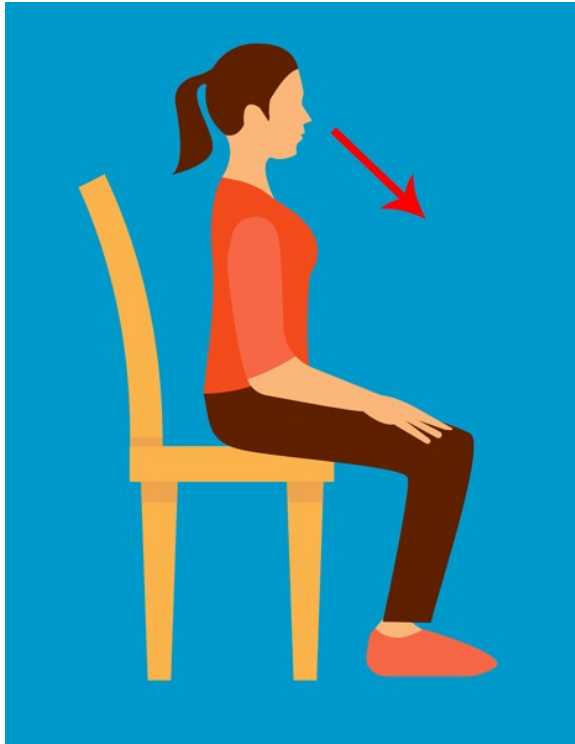
Pilar do Silêncio/Presença



- Concentração, foco, controle do stress, ansiedade, saúde mental, etc...
- Mindfulness: Práticas formais e informais
- Canal Mente Aberta no YouTube (UNIFESP):
<https://www.youtube.com/channel/UCRInguivitPAOAAMsyevfqFQ/playlists>
- Livro de Atenção Plena (pdf autorizado no edisciplinas)



Meditação 3 min



<https://www.youtube.com/watch?v=YgN1n-EgYl8&t=93s>

Pilar da Alimentação

O intestino é nosso segundo cérebro

veja

ASSINE

RADAR RADAR ECONÔMICO POLÍTICA ECONOMIA SAÚDE MUNDO CULTURA ESPORTE AGENDA VERDE

Saúde

Bactérias melhoram a memória e o desempenho cognitivo

Estudo recente revelou associação positiva entre a composição saudável da microbiota intestinal e a melhora da memória recente

Por **Giulia Vidale** 31 out 2020, 22h19



NUTRIÇÃO

O peso dos ultraprocessados

Estudos associam o consumo desse tipo de comida a 10% das mortes precoces no Brasil e à aceleração do declínio cognitivo

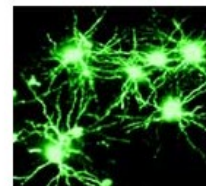


Reduzir calorias para melhorar a memória

27 de janeiro de 2009



Agência FAPESP – Um estudo feito por um grupo de cientistas da Universidade de Münster, na Alemanha, verificou que uma dieta com menos calorias resultou na melhoria da memória. A pesquisa será publicada esta semana no site e em breve na edição impressa da revista *Proceedings of the National Academy of Sciences*.



Pesquisa feita na Alemanha indica que idosos que se submeteram a dieta de restrição calórica tiveram melhoria em testes de memória

Pilar da Alimentação

O intestino é nosso segundo cérebro:

- memória, cognição, etc...
- ansiedade, depressão, ...
- doenças neurodegenerativas (Parkinson, Alzheimer, etc)
- transtornos do neurodesenvolvimento (autismo, etc)
- doenças complexas (câncer, diabetes, etc.)

Pilar da Alimentação

**COMIDA DE
VERDADE!!!**

- evite processados, ultraprocessados, conservantes, corantes, ...
- evite álcool, gorduras animais, embutidos, ...
- coma COMIDA DE VERDADE!

Mais sobre:



Denise Carreiro

12,2 mil inscritos • 38 vídeos



Vida Veda

115 mil inscritos • 1.606 vídeos

O Vida Veda é uma empresa social fundada pelo Vaidya Matheus Macêdo, primeiro brasileiro a se formar no BAMS ...

Pilar do Movimento



- De preferência todos os dias, ao menos 150 min/semana
- Aquilo que você gosta
- Temos oportunidades aqui na EACH!
- Constância é melhor que intensidade (tente fazer ao menos um pouco por dia)

Pilar do Sono

Quantidade: quanto VOCÊ PRECISA

Qualidade: ritual do sono para dormir bem

- Cuidado com cafeína
- Cuidado com atividades que te agitem antes de dormir
- Evitar telas de preferência 2h antes de dormir
- Jante cedo e leve
- Um chazinho cedo pode ajudar (não exagere nos líquidos)



Não esqueça do te faz bem para a alma...

- Família
- Amigos
- o seu Belo: 15 aos 21 anos – setênio do “Belo”

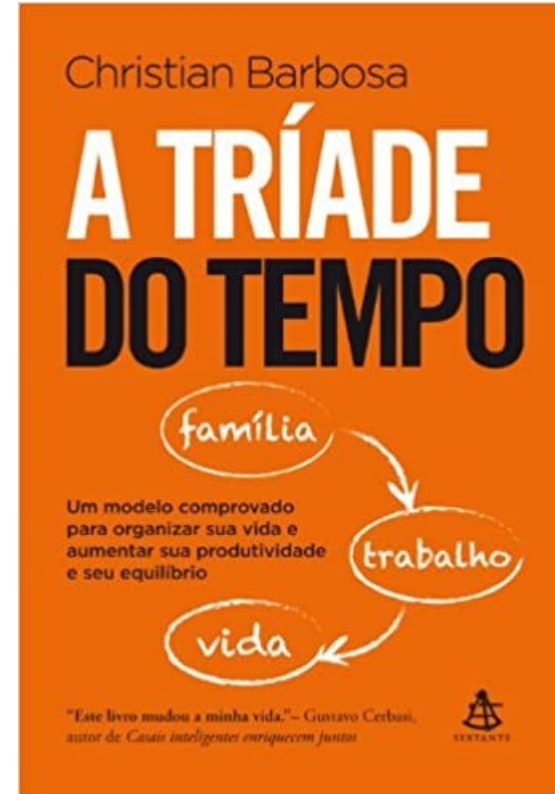


Aproveitar cada dia

Esperar para ser feliz amanhã é a receita mais certa para a infelicidade



Como fazer tudo isso?



Essas coisas fazem sentido para vocês?



CONTEÚDO

Introdução à Análise de Algoritmos

- Programação elegante (leiaute, documentação)
- Makefile, conceitos básicos de C (revisão – para uso em aula e EPs)
- Análise de complexidade
- Técnicas de desenvolvimento de algoritmos: recursão, programação dinâmica, tentativa e erro (backtracking), algoritmos gulosos, heurísticas, ...
- Algoritmos de ordenação (comparação dos algoritmos e suas complexidades, de tempo e espaço)


```
attachEvent("onreadystatechange",H),e.attachE
Boolean Number String Function Array Date RegEx
_={};function F(e){var t=_[e]={};return b.ea
t[1])===!1&&e.stopOnFalse){r=!1;break}n=!1,u&
?o=u.length:r&&(s=t,c(r))}return this},remove
nction(){return u=[],this},disable:function()
re:function(){return p.fireWith(this,argument
ending",r={state:function(){return n},always:
romise)?e.promise().done(n.resolve).fail(n.re
dd(function(){n=s},t[1^e][2].disable,t[2][2].
=0,n=h.call(arguments),r=n.length,i=1!==r||e&
(r),l=Array(r);r>t;t++)n[t]&&b.isFunction(n[t
/><table></table><a href='/a'>a</a><input type
```


Quais os problemas de um código assim?

```
attachEvent("onreadystatechange",H),e.attachE
boolean Number String Function Array Date RegEx
_={};function F(e){var t=_[e]={};return b.ea
t[1])===!1&&e.stopOnFalse){r=!1;break}n=!1,u&
?o=u.length:r&&(s=t,c(r))}return this},remove
nction(){return u=[],this},disable:function()
re:function(){return p.fireWith(this,argument
ending",r={state:function(){return n},always:
romise)?e.promise().done(n.resolve).fail(n.re
dd(function(){n=s},t[1^e][2].disable,t[2][2].
=0,n=h.call(arguments),r=n.length,i=1!==r||e&
(r),l=Array(r);r>t;t++)n[t]&&b.isFunction(n[t]
/><table></table><a href='/a'>a</a><input typ
```

Características visuais de programas

- Leiaute: aparência do programa:
 - Indentação, organização dos comandos pelas linhas
 - Espaçamentos
- Além disso:
 - Nomes de variáveis e funções
 - Documentação

Características visuais de programas

- Leiaute: aparência do programa:
 - Indentação, organização dos comandos pelas linhas
 - Espaçamentos
- Além disso:
 - Nomes de variáveis e funções
 - Documentação

Para quê ?

Características visuais de programas

- Leiaute: aparência do programa:
 - Indentação, organização dos comandos pelas linhas
 - Espaçamentos
- Além disso:
 - Nomes de variáveis e funções
 - Documentação

Para quê ?

Entendimento/legibilidade do código!

Para você:

- desenvolvimento (durante)
- manutenção (depois)

Para outros:

- que forem usar (e citar)
- que forem manter/evoluir
- é uma propaganda de você!!!

Características visuais de programas

- Leiaute: aparência do programa:
 - Indentação, organização dos comandos pelas linhas
 - Espaçamentos
- Além disso:
 - Nomes de variáveis e funções
 - Documentação

Leiaute – indentação

Por que são boas indentações?

Bom leiaute

```
int Funcao (int n, int v[]) {
    int i, j;
    i = 0;
    while (i < n) {
        if (v[i] != 0)
            i = i + 1;
        else {
            for (j = i + 1; j < n; j++)
                v[j-1] = v[j];
            n = n - 1;
        }
    }
    return n;
}
```

Um bom leiaute compacto

```
int Funcao (int n, int v[]) {
    int i, j;
    i = 0;
    while (i < n) {
        if (v[i] != 0) i = i + 1;
        else {
            for (j = i + 1; j < n; j++) v[j-1] = v[j];
            n = n - 1; } }
    return n; }
```

Leiaute – indentação

Por que são boas indentações? Porque evidenciam a **estrutura lógica** do código

Bom leiaute

```
int Funcao (int n, int v[]) {
    int i, j;
    i = 0;
    while (i < n) {
        if (v[i] != 0)
            i = i + 1;
        else {
            for (j = i + 1; j < n; j++)
                v[j-1] = v[j];
            n = n - 1;
        }
    }
    return n;
}
```

Um bom leiaute compacto

```
int Funcao (int n, int v[]) {
    int i, j;
    i = 0;
    while (i < n) {
        if (v[i] != 0) i = i + 1;
        else {
            for (j = i + 1; j < n; j++) v[j-1] = v[j];
            n = n - 1; } }
    return n; }
```

Características visuais de programas

- Leiaute: aparência do programa:
 - Indentação, organização dos comandos pelas linhas
 - **Espaçamentos**
- Além disso:
 - Nomes de variáveis e funções
 - Documentação

Leiaute – espaçamento

Por que são bons espaçamentos?

Bom leiaute

```
int Funcao (int n, int v[]) {  
    int i, j;  
    i = 0;  
    while (i < n) {  
        if (v[i] != 0)  
            i = i + 1;  
        else {  
            for (j = i + 1; j < n; j++)  
                v[j-1] = v[j];  
            n = n - 1;  
        }  
    }  
    return n;  
}
```

Um bom leiaute compacto

```
int Funcao (int n, int v[]) {  
    int i, j;  
    i = 0;  
    while (i < n) {  
        if (v[i] != 0) i = i + 1;  
        else {  
            for (j = i + 1; j < n; j++) v[j-1] = v[j];  
            n = n - 1; } }  
    return n; }
```

Leiaute – espaçamento

Por que são bons espaçamentos? Porque dão “ar” ao código, e segue um padrão (coerência)

Bom leiaute

```
int Funcao (int n, int v[]) {
    int i, j;
    i = 0;
    while (i < n) {
        if (v[i] != 0)
            i = i + 1;
        else {
            for (j = i + 1; j < n; j++)
                v[j-1] = v[j];
            n = n - 1;
        }
    }
    return n;
}
```

Um bom leiaute compacto

```
int Funcao (int n, int v[]) {
    int i, j;
    i = 0;
    while (i < n) {
        if (v[i] != 0) i = i + 1;
        else {
            for (j = i + 1; j < n; j++) v[j-1] = v[j];
            n = n - 1; } }
    return n; }
```

Leiaute – espaçamento

Por que são bons espaçamentos?

Regras

Use as regras adotadas por todos os jornais, revistas e livros:

bla bla bla

bla = bla

bla <= bla

bla; bla

bla) bla;

bla {

while (bla

if (bla

Leiaute – espaçamento

SEJA CONSISTENTE!!! Não misture...

Péssimo leiaute

```
int Funcao ( int n,int v[] ){
    int i,j;
    i=0;
    while(i<n){
        if(v[i] !=0)
            i= i +1;
        else
        {
            for (j=i+1;j<n;j++)
                v[j-1]=v[j];
            n =n- 1;
        }
    }
    return n;
}
```

Características visuais de programas

- Leiaute: aparência do programa:
 - Indentação, organização dos comandos pelas linhas
 - Espaçamentos
- Além disso:
 - Nomes de variáveis e funções
 - Documentação

Nomes de variáveis e funções

Como devem ser?

Nomes de variáveis e funções

Como devem ser?

Devem dar a entender o que são/fazem !!!

Nomes de variáveis e funções

Como devem ser?

Devem dar a entender o que são/fazem !!!

O que essa função faz?

```
int calculaIdadeAproximada(Pessoa pess)
{
    int idade;
    Data hoje = dataDeHoje();
    idade = hoje.ano - pess.dataNascimento.ano;
    return (idade);
}
```

Características visuais de programas

- Leiaute: aparência do programa:
 - Indentação, organização dos comandos pelas linhas
 - Espaçamentos
- Além disso:
 - Nomes de variáveis e funções
 - Documentação

Documentação

- ▶ documentação: **o que** um algoritmo faz
- ▶ código: **como** o algoritmo faz o que faz

Exemplo

```
/* A função abaixo recebe um número n >= 1 e um vetor v  
 * e devolve o valor de um elemento máximo de v[0..n-1].  
 *****/  
  
int Max (int v[], int n) {  
    int j, x = v[0];  
    for (j = 1; j < n; j++)  
        if (x < v[j]) x = v[j];  
    return x;  
}
```

Referências dessa parte da aula (leiaute)

Paulo Feofiloff. Algoritmos em C. Apêndice A e seção 1.1.
(lá tem exercícios! Façam!!)

<https://www.ime.usp.br/~pf/algoritmos-livro/>