

Algoritmos e Programação

Professor Marcus Vinícius Midená Ramos

Colegiado de Engenharia de Computação

(74)3614.1936

marcus.ramos@univasf.edu.br

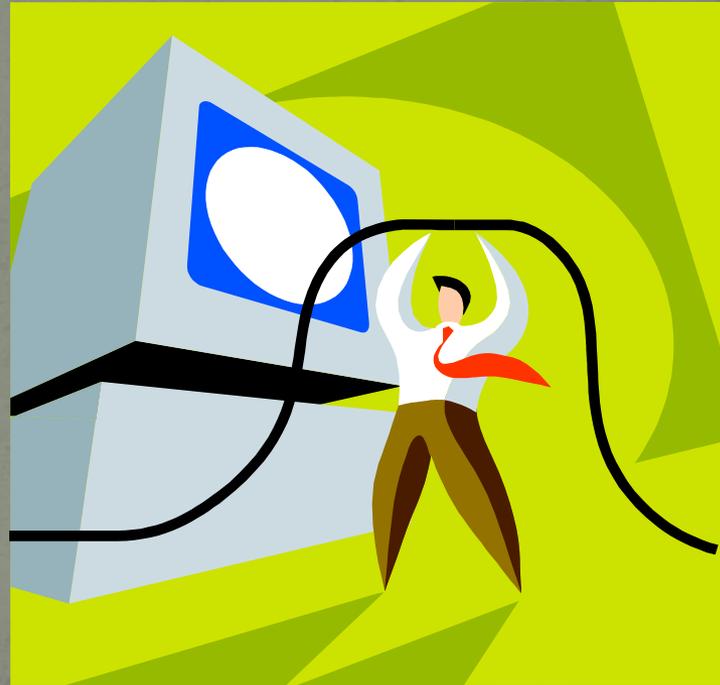
www.univasf.edu.br/~marcus.ramos

Objetivos:

- Desenvolver o raciocínio lógico e abstrato;
- Dominar o processo básico de desenvolvimento de programas;
- Compreender e saber usar o computador como ferramenta de produtividade pessoal e profissional, além dos aplicativos comerciais;
- Conseguir autonomia na organização e implementação de soluções mecanizadas para os seus próprios problemas.

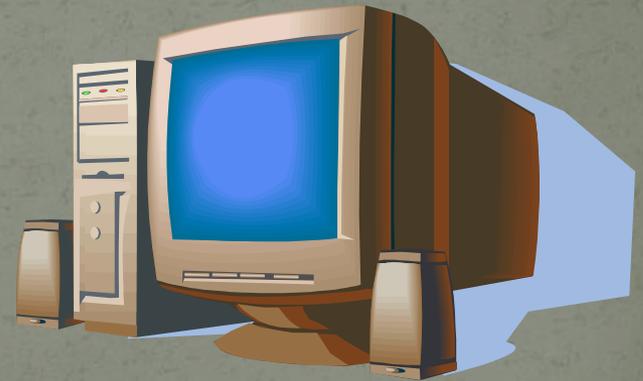
Computador

- Ferramenta indispensável;
- Faz parte das nossas vidas;
- Por si só não faz nada de útil;
- Grande capacidade de resolução de problemas;
- Necessita ser instruído.



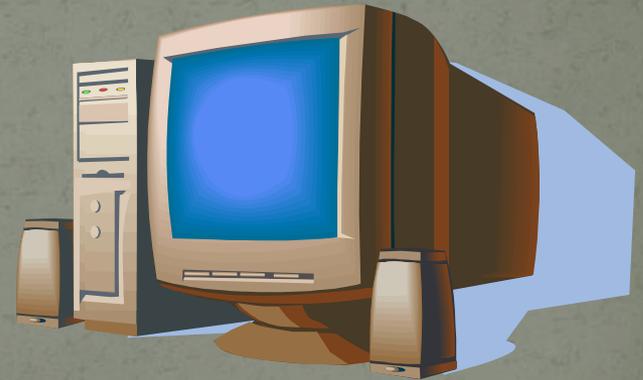
Computador

- Capaz apenas de executar poucas tarefas básicas distintas, todas muito simples;
- É extremamente rápido;
- Possui um comportamento previsível;
- É excelente para reproduzir “roteiros” pré-concebidos;
- Não se cansa e pode ser usado à exaustão.



Computador = Hardware (*corpo*) + Software (*alma*)

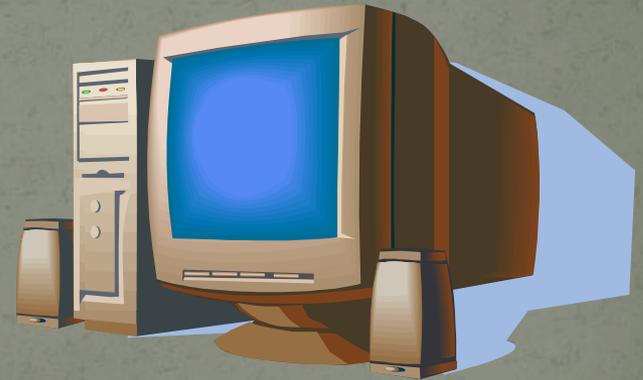
- ✓ O corpo fornece suporte para a alma.
- ✓ O corpo procura suprir as necessidades da alma.
- ✓ O corpo pode criar novas possibilidades para a alma, ou então estabelecer limitações.
- ✓ A alma se expressa através do corpo.
- ✓ A alma usa os recursos do corpo.



Computador

- **Hardware:**

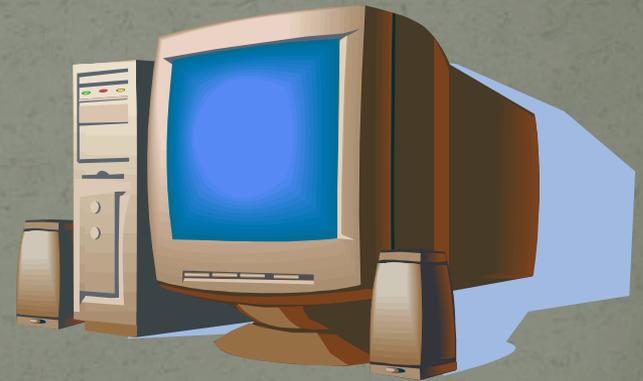
- ✓ Parte física: placas, periféricos, circuitos, cabos e componentes.
- ✓ Quanto mais usado, mais propenso à falhas.
- ✓ Sozinho, não serve para nada.
- ✓ Vem pronto da fábrica.



Computador

- **Software:**

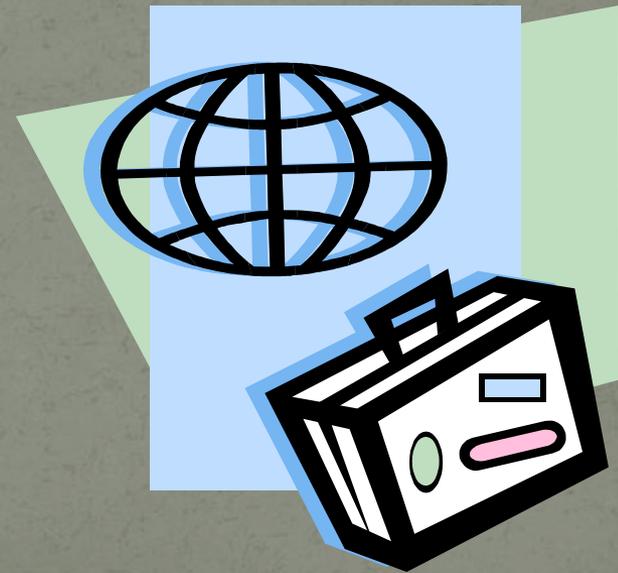
- ✓ Parte intangível:
conhecimentos e idéias que
fazem o hardware exibir
um certo comportamento.
- ✓ Quanto mais usado, menos
propenso à falhas.
- ✓ Confere funcionalidade ao
hardware.
- ✓ Pode ser adquirido ou
desenvolvido.



Software

• Comprar?

- ✓ Oferta;
- ✓ Soluções de prateleira, não personalizadas;
- ✓ Custo de aquisição;
- ✓ Vínculos (plataforma, localização, performance, funcionalidade etc.)
- ✓ Custo de personalização;



Software

- Desenvolver?

- ✓ Recursos próprios ou de terceiros (contratação).
- ✓ Custo de criação;
- ✓ Custo de manutenção;
- ✓ Conhecimento tecnológico;
- ✓ Personalização.



Nosso objetivo

- **Desenvolver software**
 - ✓ Organização de idéias;
 - ✓ Modelo de funcionamento do computador;
 - ✓ Conceitos básicos de programação;
 - ✓ Transcrição para linguagens apropriadas;
 - ✓ Comunicação e interação com o computador;
 - ✓ Obtenção dos resultados pretendidos;
 - ✓ Prática em laboratório.



Desenvolver software

- Roteiro

1. Problema;
2. Solução;
3. Algoritmo;
4. Programa;
5. Resultados.



Problema

- ✓ Precisa ser conhecido em todos os seus aspectos;
- ✓ É necessário ter resposta para todas as perguntas que dele possam suscitar;
- ✓ É fundamental considerar todas as situações adversas;
- ✓ Nenhuma faceta deve ser omitida.



Solução

- ✓ Existe solução para o problema?
- ✓ Qual o custo da sua implementação?
- ✓ Qual o custo da sua execução?
- ✓ Como iremos representá-la?



Algoritmo

- ✓ Representação de uma solução para um problema, com algumas características:
 - Seqüência finita de etapas;
 - Individualmente, existe realização possível para cada uma das etapas consideradas;
 - Termina após um tempo finito.



Algoritmo

- ✓ Representação :
 - Linguagem natural;
 - Pseudocódigo (linguagem textual com poucos símbolos e regras, que são simples);
 - Fluxograma (linguagem visual composta por poucos símbolos e regras)
- ✓ Um algoritmo expressa uma solução para um problema.



Terminou?

Nãoo!!!

Acontece que...

Computadores não entendem (normalmente, ou pelo menos da forma como nós precisamos):

- Linguagens naturais;
- Pseudocódigos;
- Fluxogramas.



Precisamos ir além...

Algoritmo!

Não estou
entendendo!!!!

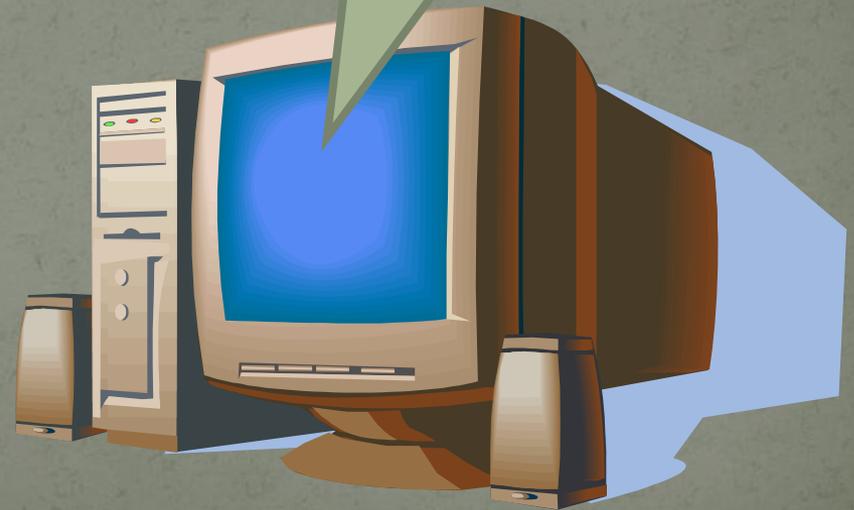


O que o
computador
entende
afinal?

Faça isso. Depois aquilo.
Se OK, então pare, senão
refaça tudo.



011010101100101011010101
00010101000101111101010
010000101111010101100111
1111?????



Temos um
problema de
comunicação

.

Solução?

Melhorar um pouco
as coisas prá ele
(computador) sem
piorar tanto prá nós
(humanos).

- ⇒ Escrever um “programa” de computador, a partir do algoritmo.
- ⇒ Para isso, vamos usar uma “linguagem de programação”.
- ⇒ Um pouco mais complexas do que as linguagens usadas para representar algoritmos;
- ⇒ Mas mais fáceis de serem entendidas pelo computador.

Java?
C?
C++?
Delphi?
Pascal?
HTML?
Perl?
Python?
Ruby?
Fortran?
Assembly?
PHP?
Cobol?
SQL?
Lisp?
Prolog?



Java!!
C!!
C++!!
Delphi!!
Pascal!!
HTML!!
Perl!!
Python!!
Ruby!!
Fortran!!
Assembly!!
PHP!!
Cobol!!
SQL!!
Lisp!!
Prolog!!

E...?

Sim, vamos precisar traduzir algoritmos para programas.

Sim, precisaremos conhecer (pelo menos) duas linguagens.

Sim, cometeremos erros nas traduções.

C'est la vie...



Eu não existo.

Como ficamos
então?

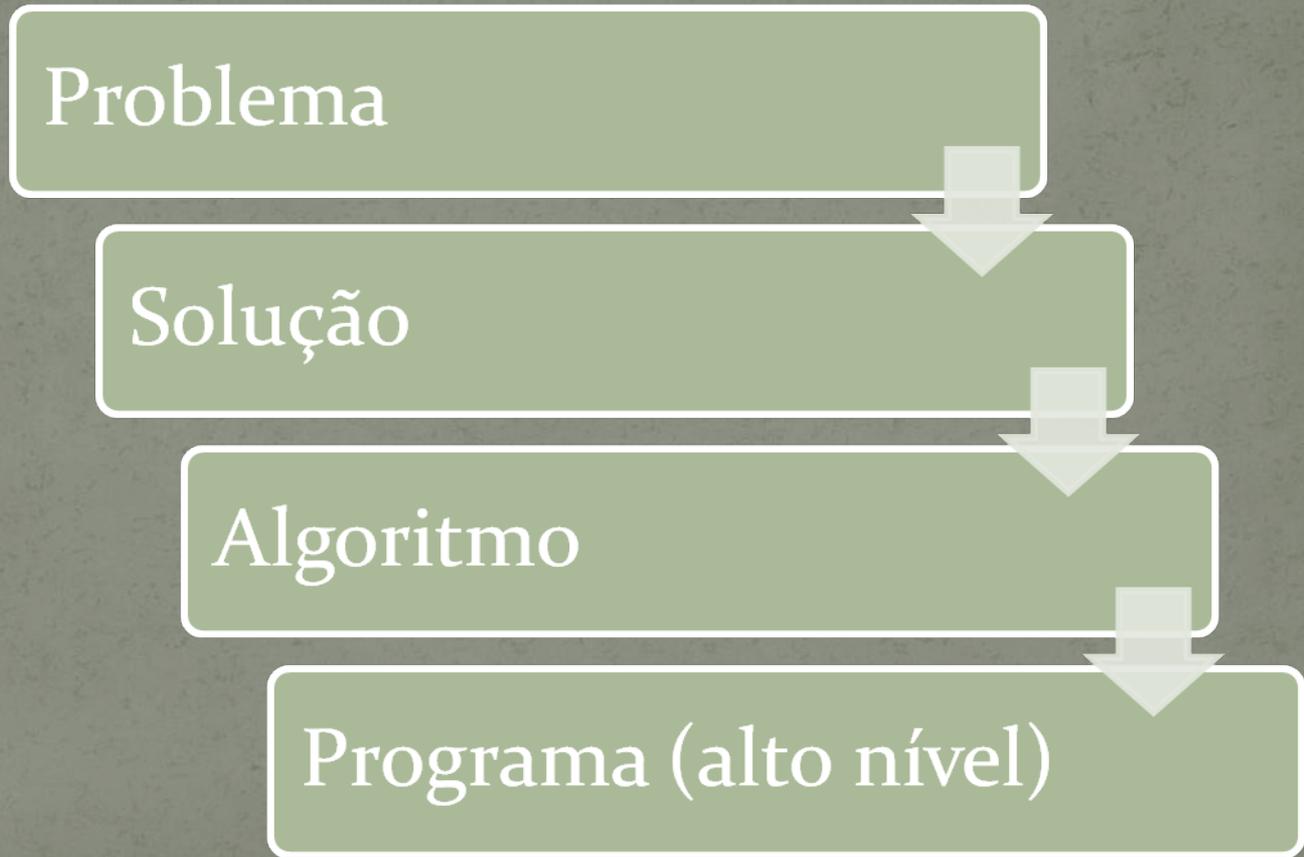
Sua parte:

Problema

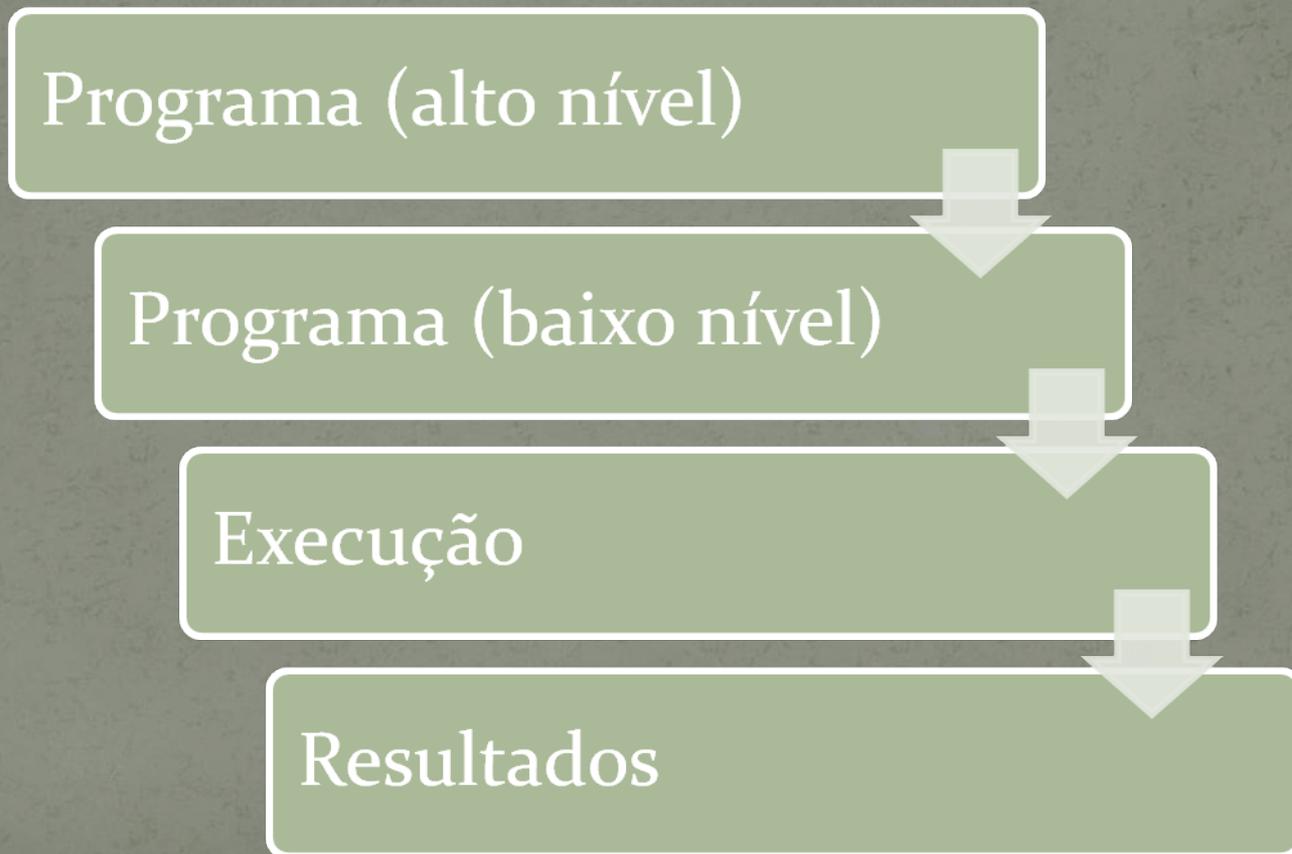
Solução

Algoritmo

Programa (alto nível)



Parte do computador (com a sua supervisão...):



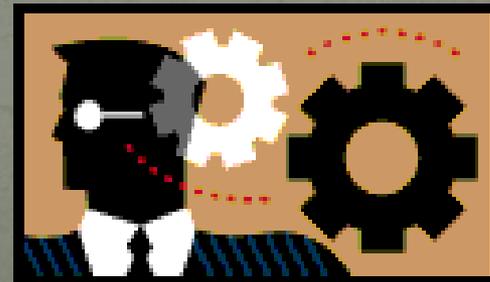
⇒ Deu errado?

⇒ Não era bem
isso que você
queria?

Não tem problema.

Volte à prancheta...

E descubra onde está o erro.



Ciclo de desenvolvimento

