

Algoritmos

DCC 119

Informações Gerais



Informações Gerais

- Site da disciplina:
<https://sites.google.com/site/algoritmosufjf/>
(é só buscar por “**algoritmos ufjf**” no Google)

Neste site você encontrará:

- Avisos importantes;
- Material Didático;
- Data das provas;
- Exercícios adicionais;
- Horário de Monitoria e de Atendimento;
- Contato dos professores;
- Links diversos, etc.

Informações Gerais

- Disciplina ministrada juntamente com **DCC120 – Laboratório de Programação**
- Conteúdo, calendário e sistema de avaliação unificados
- Organização das aulas:
 - DCC119 – 5^{as} feiras: novo conteúdo teórico
 - DCC119 – 2^{as} feiras: continuação teoria e exercícios
 - DCC120 – 3^{as} ou 4^{as}: exercícios práticos

Avaliação do aprendizado

- Testes de Verificação de Conhecimento:
 - 1º TVC: 20 pontos
 - 2º TVC: 40 pontos
 - 3º TVC: 40 pontos
- Prova substitutiva:
 - Substitui, obrigatoriamente, a menor nota;
 - Apenas para quem entregou ao menos 2/3 (dois terços) das atividades em sala de aula na disciplina de Algoritmos e na disciplina de Laboratório.

Segunda Chamada

- **O que é ?**
É a prova que você poderá requerer no caso de perder uma das avaliações.
- **Há dois tipos de avaliação de 2ª chamada na UFJF.**

Segunda Chamada - UFJF - Caso 1

Em quaisquer das avaliações, o professor responsável pela disciplina poderá conceder segunda chamada ao aluno ausente que apresentar requerimento com justificativa julgada procedente.

O requerimento deverá ser entregue ao departamento no prazo máximo de três dias úteis, com a devida justificativa de acordo com a legislação em vigor.

Sendo **julgada procedente** a justificativa, a segunda chamada abordará os mesmos tópicos da avaliação não realizada.

Segunda Chamada - UFJF - Caso 2

O aluno que perder **uma** das avaliações tem direito, **sem a necessidade de justificativa**, a fazer segunda chamada, sobre conteúdo acumulado.

Segunda Chamada

Nessa disciplina, independente do caso:

- **é obrigatório entrar com requerimento** para a segunda chamada, mesmo que o aluno não tenha uma justificativa;
- a avaliação **será realizada uma semana após a avaliação perdida.**

Segunda Chamada

Requisição de segunda chamada:

- **Como:** Preencher formulário de requerimento.
- **Onde:** Site do DCC <http://www.ufjf.br/deptocomputacao>
em *Links* → *Requisição de segunda chamada*
- **Quando:** Até 3 dias úteis após a prova perdida

A avaliação será realizada uma semana após a avaliação perdida
(horário e sala divulgados no site da disciplina).

Critérios de aprovação

- Nota mínima: 60 pontos
- Frequência mínima: 75%
(no máximo 7 faltas no semestre)

Regulamento Acadêmico da Graduação
Capítulo IV - Art. 38:

“É vedado o abono de faltas, salvo nos casos expressos na legislação vigente. (...) ”

Metodologia

- Aulas expositivas
 - Transparências
 - Quadro negro
 - Material disponível em PDF
- Resolução de exercícios
- Leitura/Estudo do material

Objetivos da disciplina

- Apresentar os conceitos básicos da Ciência da Computação, necessários ao desenvolvimento de algoritmos.
- Introduzir uma linguagem de programação.
- Desenvolver o raciocínio lógico.

Como desenvolver o raciocínio lógico?

- Não basta estudar toda a lista de exercícios!
- Não basta copiar toda a lista de exercícios!
- *Copiar* ou *ler* ou *estudar* um algoritmo pronto não é suficiente para desenvolver o raciocínio lógico!

Para desenvolver o raciocínio lógico é necessário resolver problemas!

Como estudar algoritmos?

- É muito importante **pensar** no problema que se deseja resolver e elaborar o algoritmo de cada problema proposto **sem ajuda!**
 - Essa é a maior dificuldade de 80% dos alunos da disciplina.
 - Na maioria das vezes, estes alunos não percebem e não entendem que este é o problema.

Como estudar algoritmos?

- Para resolver problemas, é necessário:
 1. Conhecer as regras ou ações permitidas;
 2. Entender o problema proposto;
 3. Indicar quais ações devem ser realizadas e em que ordem devem ser feitas;
 4. Executar a sequência de ações e verificar se esta realmente resolve o problema proposto.

A maioria dos alunos se concentra apenas no passo 1, embora o verdadeiro desafio esteja nos outros passos...

Exemplo 1



Problema: inverter a posição das peças brancas e pretas.

Ações permitidas:

- Deslizar uma peça para a posição vizinha se estiver vazia (ex: peça em 2 vai para 3);
- Pular uma única peça até a posição seguinte, se esta estiver vazia (ex: peça em 5 vai para 3).

Qual sequência de ações resolve o problema?

Exemplo 1



Sequência de ações:

1. Mover de 2 para 3;
2. Pular de 4 para 2;
3. Mover de 5 para 4;
4. Pular de 3 para 5;
5. Pular de 1 para 3;
6. Mover de 0 para 1;
7. Pular de 2 para 0;

8. Pular de 4 para 2;
9. Pular de 6 para 4;
10. Mover de 5 para 6;
11. Pular de 3 para 5;
12. Pular de 1 para 3;
13. Mover de 2 para 1;
14. Pular de 4 para 2;
15. Mover de 3 para 4.

*Esta é uma solução para o problema?
Existe apenas uma solução para este problema?*

Exemplo 1



Sequência de ações:

1. Mover de 2 para 3;
2. Pular de 4 para 2;
3. Mover de 5 para 4;
4. Pular de 3 para 5;
5. Pular de 1 para 3;
6. Mover de 0 para 1;
7. Pular de 2 para 0;

8. Pular de 4 para 2;
9. Pular de 6 para 4;
10. Mover de 5 para 6;
11. Pular de 3 para 5;
12. Pular de 1 para 3;
13. Mover de 2 para 1;
14. Pular de 4 para 2;
15. Mover de 3 para 4.

*Como saber se uma solução proposta está correta?
Executando a sequência de ações indicada e
verificando se o problema foi, de fato, resolvido.*

Como estudar algoritmos?

- Nesta disciplina:
 - Raramente um problema terá uma única solução correta;
 - Uma solução pode estar correta, mesmo sendo muito diferente de outra solução do mesmo exercício;
 - Como saber se uma solução está correta?

Executando a sequência de ações indicada e verificando se o problema foi, de fato, resolvido.

Como estudar algoritmos?

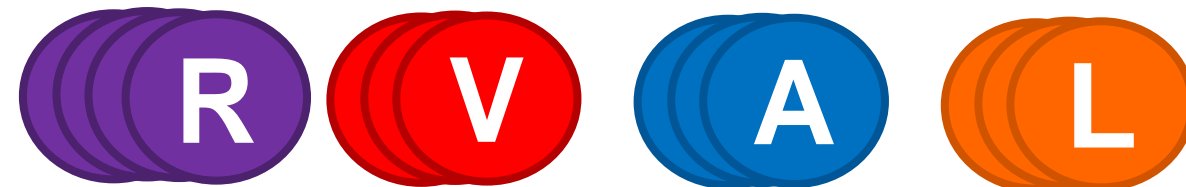
- Nesta disciplina:
 - O aluno não deve esperar que o professor corrija um exercício para saber se está correto;
 - ***O próprio aluno deve ser capaz*** de executar a sequência de instruções e verificar se o problema proposto foi resolvido.

Exemplo 2

00	01 	02	03
10	11	12	13 
20	21 	22	23
30	31	32	33

Problema: colocar as peças no tabuleiro sem que haja peças de mesma cor na mesma linha, coluna ou quadrado.

Ações permitidas:
colocar uma peça em uma posição vazia.

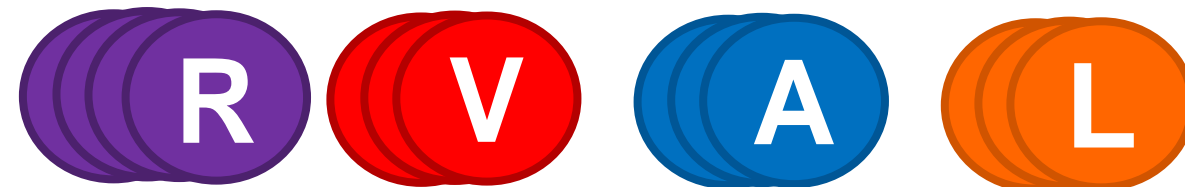


Exemplo 2

00	01 	02	03
10	11	12	13 
20	21 	22	23
30	31	32	33

A solução a seguir resolve o problema?

1. V em 00
2. L em 02
3. R em 03
4. L em 10
5. R em 11
6. A em 12
7. A em 20
8. R em 22
9. A em 23
10. V em 30
11. R em 31
12. V em 32
13. L em 33



Exemplo 2

00 V	01 A	02 R	03 L
10 L	11 R	12 A	13 V
20 R	21 L	22 V	23 A
30 A	31 V	32 L	33 R

Neste exemplo, a ordem em que as ações são executadas não interfere na solução.

Mas, ao longo da disciplina, a ordem das ações normalmente será significativa na solução.

Exemplo 3



Problema: dada uma configuração inicial aleatória, fazer com que as peças do jogo fiquem na posição indicada na figura ao lado.

Ações permitidas: deslizar uma peça para a posição vizinha, se esta estiver vazia.

Exemplo 3



Para a configuração inicial ao lado, a solução a seguir resolve o problema?

1. Move o 8
2. Move o 4
3. Move o 1
4. Move o 5
5. Move o 2
6. Move o 3
7. Move o 6
8. Move o 8
9. Move o 5
10. Move o 2
11. Move o 3
12. Move o 6

A ordem dos passos é significativa?

Laboratório de Programação DCC 120

Informações gerais



Informações Gerais

- Site da disciplina:
<https://sites.google.com/site/algoritmosufjf/>
(é só buscar por “**algoritmos ufjf**” no Google)

Neste site você encontrará:

- Avisos importantes;
- Material Didático;
- Data das provas;
- Exercícios adicionais;
- Horário de Monitoria;
- Links diversos, etc.

Informações Gerais

- Disciplina ministrada juntamente com **DCC119 – Algoritmos**
- Conteúdo, calendário e sistema de avaliação unificados
- Organização das aulas:
 - DCC119 – 5^{as} feiras: novo conteúdo teórico
 - DCC119 – 2^{as} feiras: continuação teoria e exercícios
 - DCC120 – 3^{as} ou 4^{as}: exercícios práticos

Avaliação do aprendizado

- Testes de Verificação de Conhecimento:
 - 1º TVC: 20 pontos
 - 2º TVC: 40 pontos
 - 3º TVC: 40 pontos

Avaliações realizadas nas aulas de Algoritmos.

- Prova substitutiva:
 - Substitui, obrigatoriamente, a menor nota;
 - Apenas para quem entregou ao menos 2/3 (dois terços) das atividades em sala de aula na disciplina de Algoritmos e na disciplina de Laboratório.

Segunda Chamada

- **O que é ?**
É a prova que você poderá requerer no caso de perder uma das avaliações.
- **Há dois tipos de avaliação de 2ª chamada na UFJF.**

Segunda Chamada - UFJF - Caso 1

Em quaisquer das avaliações, o professor responsável pela disciplina poderá conceder segunda chamada ao aluno ausente que apresentar requerimento com justificativa julgada procedente.

O requerimento deverá ser entregue ao departamento no prazo máximo de três dias úteis, com a devida justificativa de acordo com a legislação em vigor.

Sendo **julgada procedente** a justificativa, a segunda chamada abordará os mesmos tópicos da avaliação não realizada.

Segunda Chamada - UFJF - Caso 2

O aluno que perder **uma** das avaliações tem direito, **sem a necessidade de justificativa**, a fazer segunda chamada, sobre conteúdo acumulado.

Segunda Chamada

Nessa disciplina, independente do caso:

- **é obrigatório entrar com requerimento** para a segunda chamada, mesmo que o aluno não tenha uma justificativa;
- a avaliação **será realizada uma semana após a avaliação perdida.**

Segunda Chamada

Requisição de segunda chamada:

- **Como:** Preencher formulário de requerimento.
- **Onde:** Site do DCC <http://www.ufjf.br/deptocomputacao>
em *Links* → *Requisição de segunda chamada*
- **Quando:** Até 3 dias úteis após a prova perdida

A avaliação será realizada uma semana após a avaliação perdida
(horário e sala divulgados no site da disciplina).

Critérios de aprovação

- Nota mínima: 60 pontos
- Frequência mínima: 75%
(no máximo 3 faltas no semestre)

Regulamento Acadêmico da Graduação
Capítulo IV - Art. 38:

“É vedado o abono de faltas, salvo nos casos expressos na legislação vigente. (...) ”

Metodologia

- Aulas expositivas
 - Transparências
 - Material disponível em PDF
- Resolução de exercícios práticos

Objetivos da disciplina

- Introduzir uma linguagem de programação e sua sintaxe
- Introduzir e exercitar todas as etapas do desenvolvimento de algoritmos:
 - Interpretação do problema a ser resolvido
 - Utilização de raciocínio lógico para elaborar um algoritmo
 - Implementação do algoritmo na linguagem C
 - Compilação do algoritmo
 - Execução de testes para verificar a corretude da solução proposta

Por que programação?

- Muitos países incentivam o ensino de programação nas escolas de primeiro e segundo grau.
- Estão disponíveis no YouTube, vídeos promocionais de uma iniciativa americana cujo objetivo é disseminar o ensino de programação (procure por Hour of Code).

Como participar das aulas de Laboratório de Programação?

- A disciplina tem um caráter prático.
- Pelo cronograma, para cada assunto abordado, a aula de Laboratório é antecedida por duas aulas teóricas de Algoritmos.
- Por isso, assume-se que o aluno já possui algum conhecimento do assunto abordado em cada semana.

Como participar das aulas de Laboratório de Programação?

Antes da aula:

- Alunos que não estão matriculados em Algoritmos ou que faltarem as aulas teóricas devem, ao menos, ler o material disponibilizado no site da disciplina;
- Recomenda-se que o aluno já tenha resolvido exercícios da aula de Algoritmos, pois:
 - o aluno poderá retirar eventuais dúvidas;
 - o aluno terá maior desenvoltura na solução dos exercícios práticos.

Como serão as aulas?

- Nas aulas de Laboratório de Programação os alunos irão sempre resolver problemas.
- Para isso, é necessário:
 1. Conhecer os recursos disponíveis que podem ser utilizados para resolver o problema;
 2. Entender o problema proposto;
 3. Elaborar uma sequência de passos indicando o que deve ser feito, na ordem em que deve ser feito;
 4. Executar a sequência de ações e verificar se esta realmente resolve o problema proposto.

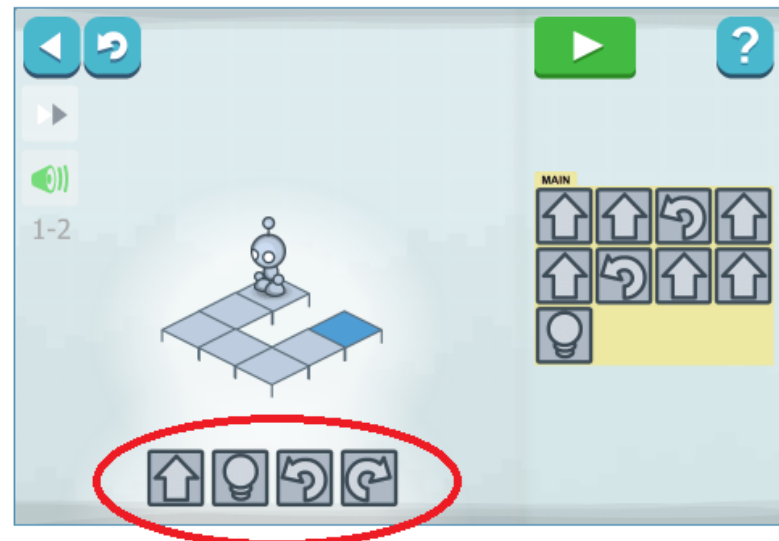
Aula de hoje

- Usaremos o jogo Light Bot para ilustrar este processo de solução de problemas.
- O jogo foi criado com o propósito de introduzir conceitos de programação para crianças.
- Consiste em um pequeno robô, situado sobre uma plataforma, que precisa se deslocar para acender os bloquinhos em azul.
- Você tem que indicar o que o robô deve fazer em cada situação.

Aula de hoje

O jogo é bastante simples, mas ilustra pontos importantes do processo de programação:

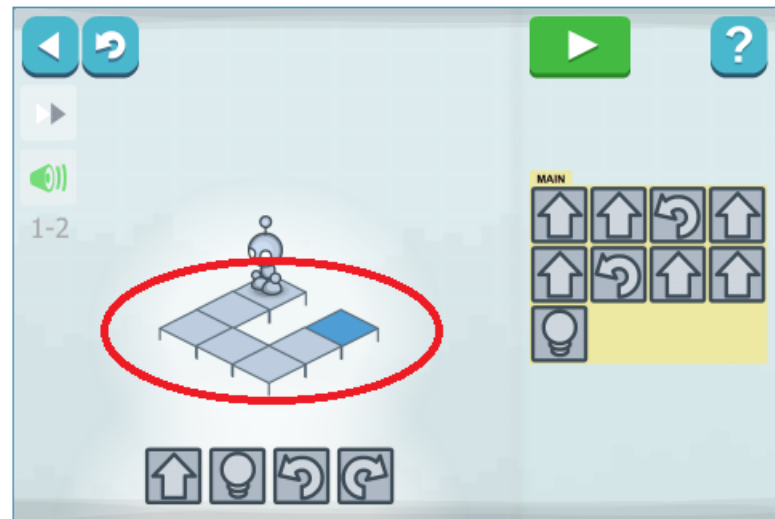
1. *É necessário conhecer os recursos disponíveis para resolver o problema.*



Aula de hoje

O jogo é bastante simples, mas ilustra pontos importantes do processo de programação:

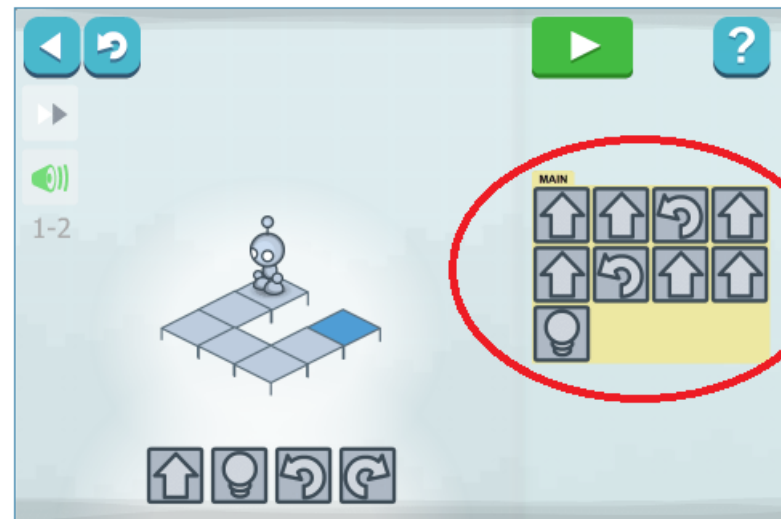
2. *É necessário entender o problema.*



Aula de hoje

O jogo é bastante simples, mas ilustra pontos importantes do processo de programação:

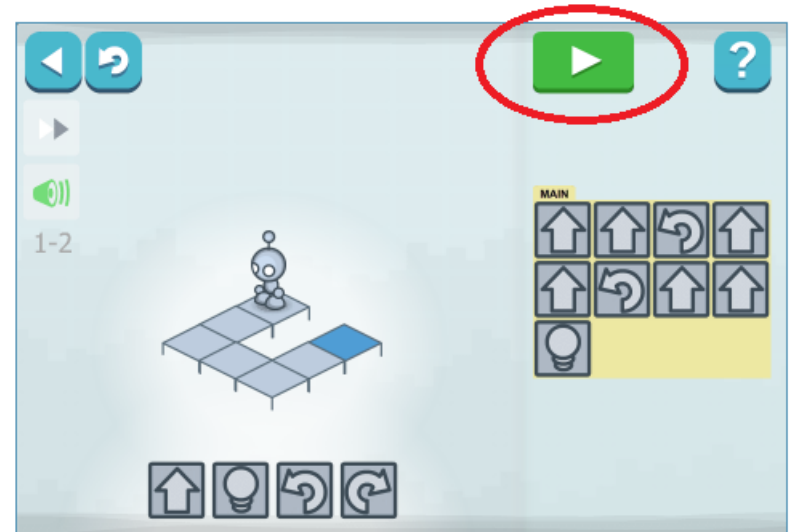
3. *É necessário combinar os recursos gerando uma sequência de instruções para o robô.*



Aula de hoje

O jogo é bastante simples, mas ilustra pontos importantes do processo de programação:

4. *É possível executar a sequência de instruções para verificar se esta, de fato, resolve o problema.*



Mãos à obra

1. Liguem o computador.
2. Para entrar nas máquinas utilizem login e senha indicados no quadro negro.
3. Abra um navegador. Sugestão:
Aplicativos → Internet → Firefox
(obs: se não tiver flash instalado no navegador, por favor, tente outro).
4. Digite o endereço:
`http://lightbot.com/hour-of-code.html`

Mãos à obra

5. Clique no local indicado pela seta para acessar a versão online.



Mãos à obra

6. Se necessário, clique no canto superior direito para mudar a língua.



7. Leia as instruções com calma e divirta-se!