

Introdução a Lógica de Programação



Pyladies Sorocaba





PyLadies Sorocaba

O que é um PyLadies?

Um grupo internacional sem fins lucrativos (<https://www.pyladies.com>) focado em aumentar a atividade e a liderança de mulheres na comunidade Python.

Nossa missão

Motivar mais meninas e mulheres a entrar no mundo da programação por meio de workshops e encontros nos quais possamos compartilhar qualquer conhecimento que tivermos.

Bora curtir o evento!

#PyLadiesSorocaba

Use a hashtag nas postagens sobre o evento :)



O que é computação desplugada?

Computação Desplugada é uma técnica de ensino que busca **abordar conceitos de lógica de programação por meio de atividades sem o computador.**

A ideia principal é expor as alunas (vocês!) a atividades divertidas e desafiadoras que ajudem a construir e materializar os conceitos que também são usados quando programamos um computador.

O que veremos hoje?

Lógica de Programação:

- Comandos condicionais
- Repetições
- Variáveis
- Um exemplo ao vivo de um código com os conceitos que aprendemos

Atividade: Como passar manteiga na bolacha?



Atividade: Como ensinar o **computador** a passar manteiga na bolacha?



Atividade: Como ensinar o **computador** a passar manteiga na bolacha?

Para “ensinar” o computador a passar manteiga no pão precisamos passar para ele as instruções nos mínimos detalhes!!!

1. Abra a manteiga
2. Pegue uma faca com uma mão
3. Raspe a manteiga com a ponta da faca de forma a acumular um pouco de manteiga na faca
4. Pegue uma bolacha com a outra mão e segure ela com uma das faces voltadas para cima
5. Passe a manteiga que está acumulada na faca na face voltada para cima da bolacha

Afinal, o que é um algoritmo?

- “Algoritmo é uma sequência de passos que visa atingir um objetivo bem definido.” (Forbellone, 1999)
- “Algoritmo é a descrição de uma sequência de passos que deve ser seguida para a realização de uma dada tarefa.” (Assencio, 1999)
- “Um algoritmo é uma sequência finita de instruções bem definidas e não ambíguas para resolver um problema ou realizar algum objetivo.

Vamos às missões?

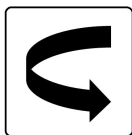
Como funciona?

- O objetivo do jogo é simples: você deve levar o robô até às luzes e acendê-las
- Você tem um número máximo de movimentos para cada exercício. Tente pensar com bastante cuidado!
- No decorrer das fases, vamos introduzindo novos conceitos e dificuldades

Legenda:



Frente



Direita



Esquerda





Luz








Acende

Exercício 1

Exercício 1 - Solução

1. Mover para a frente 
2. Mover para a frente 
3. Mover para a frente 
4. Mover para a frente 
5. Acender a luz 






[illegible]

Exercício 1 - Solução

1. Mover para a frente
2. Mover para a frente
3. Mover para a frente
4. Mover para a frente
5. Acender a luz

[illegible]

Exercício 1 - Solução

1. Mover para a frente 
2. Mover para a frente 
3. Mover para a frente 
4. Mover para a frente 
5. Acender a luz 






[illegible]

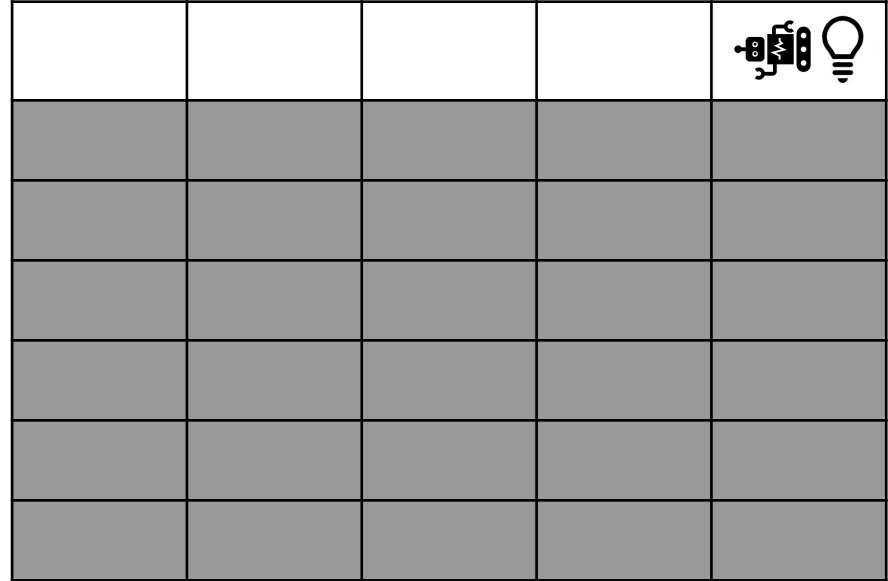
Exercício 1 - Solução

1. Mover para a frente
2. Mover para a frente
3. **Mover para a frente**
4. Mover para a frente
5. Acender a luz

[illegible]

Exercício 1 - Solução

1. Mover para a frente 
2. Mover para a frente 
3. Mover para a frente 
4. Mover para a frente 
5. Acender a luz 



Exercício 1 - Solução

1. Mover para a frente



2. Mover para a frente



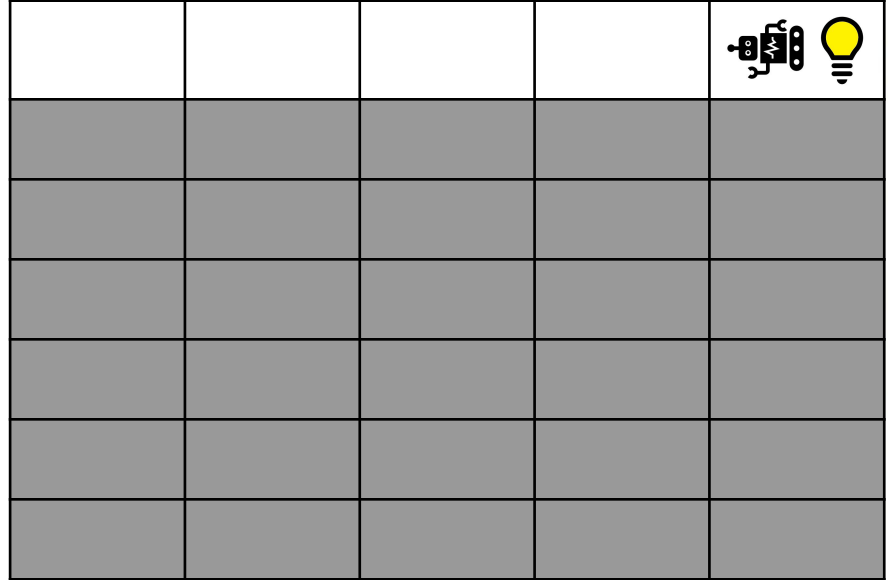
3. Mover para a frente



4. Mover para a frente



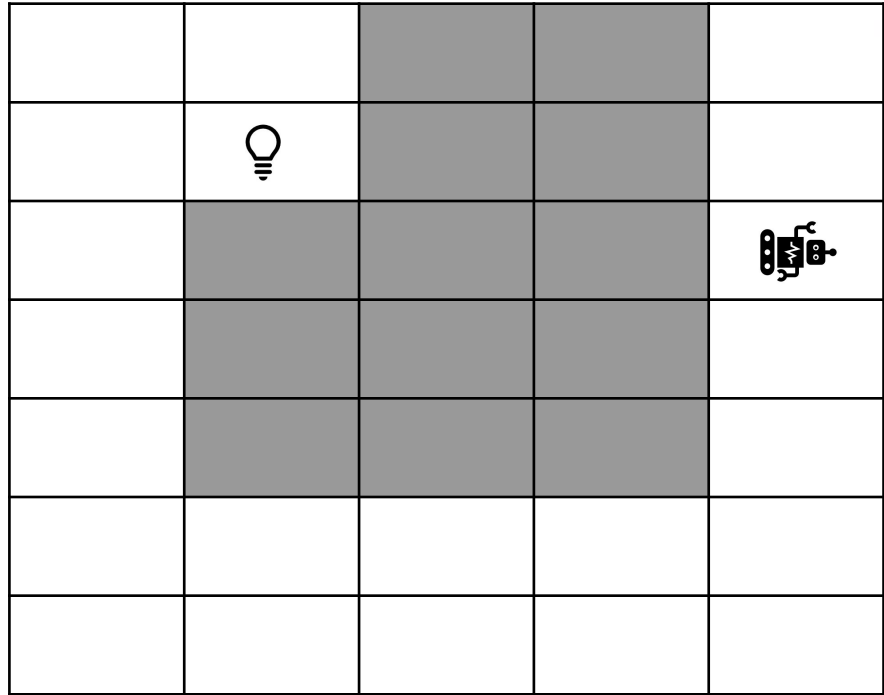
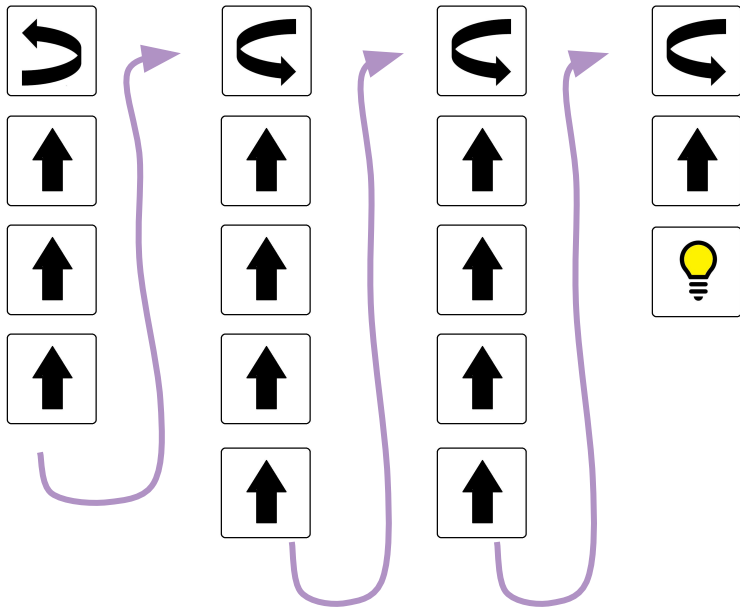
5. Acender a luz



Exercício 2

	💡			
				🤖

Exercício 2 - Solução



Exercício 2 - Solução



	💡			
				🤖

Novos desafios...

- **O objetivo do jogo é o mesmo:** você deve levar o robô até às luzes e acendê-las
- Você tem um número máximo de movimentos para cada exercício. Tente pensar com bastante cuidado!
- No decorrer das fases, vamos introduzindo novos conceitos e dificuldades

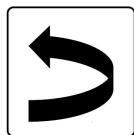
Legenda:



Frente



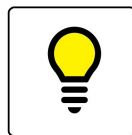
Direita



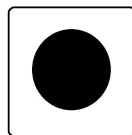
Esquerda



Luz



Acende



Buraco

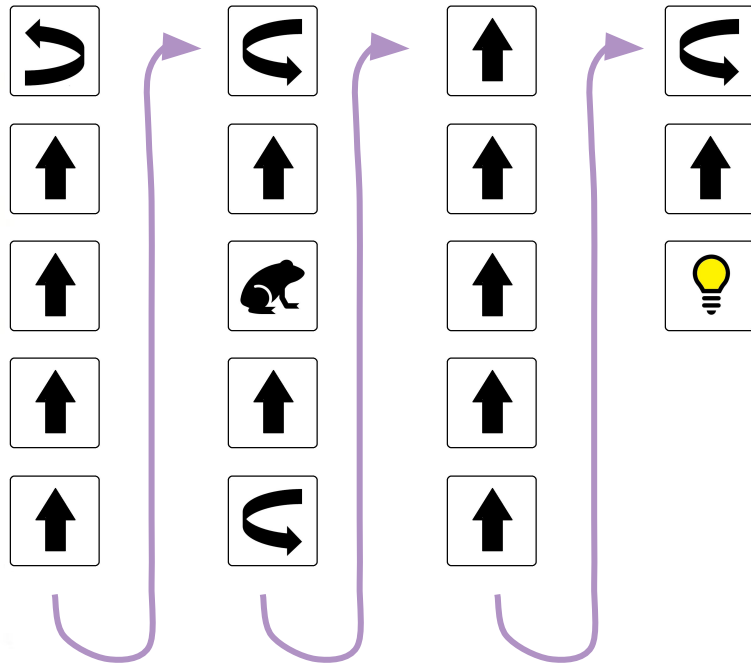


Pula

Exercício 3

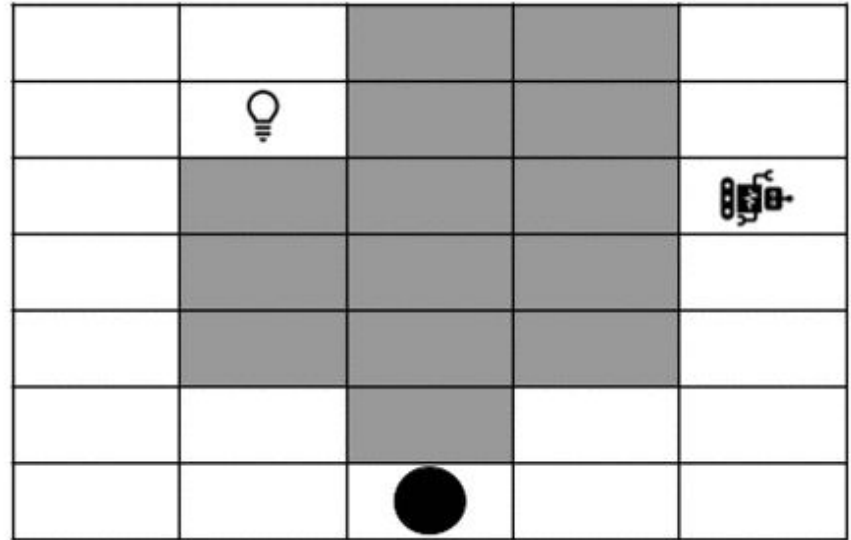
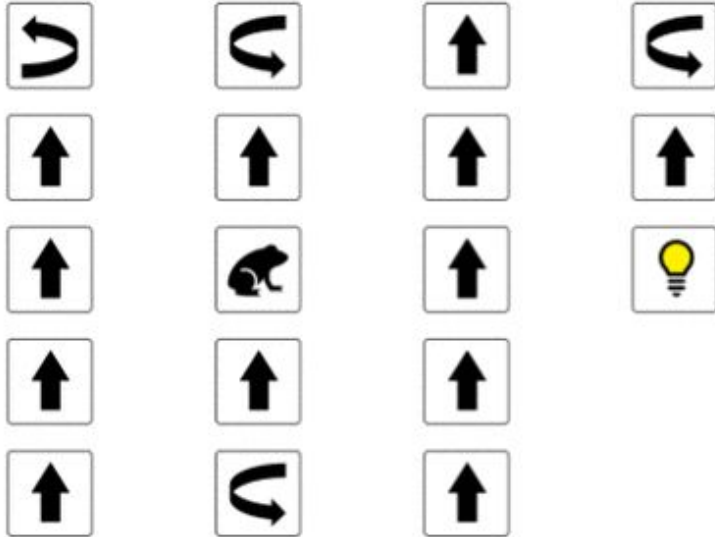
	💡			
				🤖
		●		

Exercício 3 - Solução







	💡			
				🤖
		●		

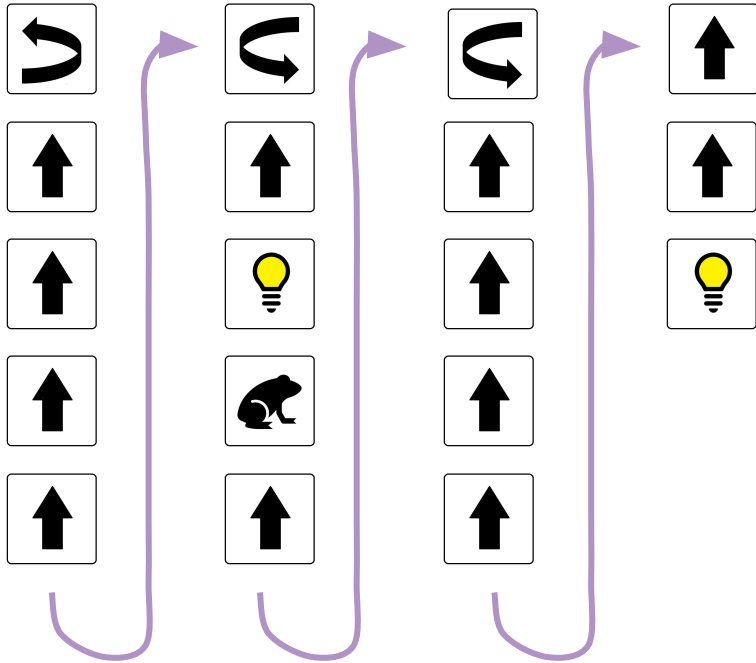
Exercício 3 - Solução



Exercício 4

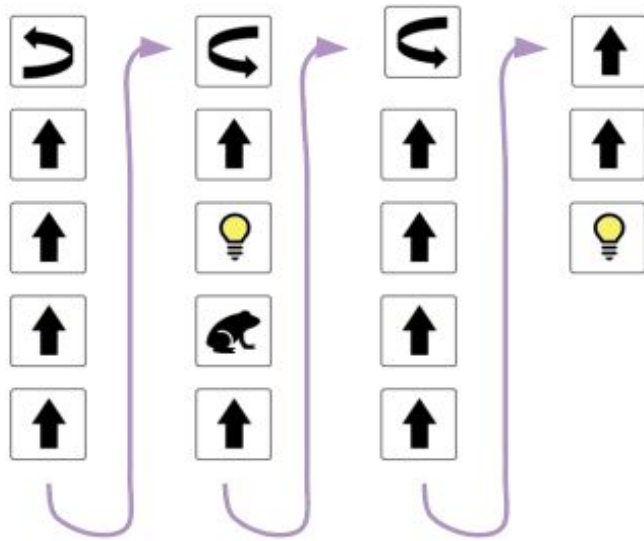
				
				
				

Exercício 4 - Solução



💡				
				🤖
		●	💡	

Exercício 4 - Solução



💡				
				🚗
		●	💡	

Tomando decisões

Precisamos tomar decisões o tempo todo!



Condições

- No início da atividade, aprendemos que **se** alcançarmos uma luz, temos que acendê-la e, da mesma forma, **se** toparmos com um buraco, temos que saltá-lo.
- Esse **se** que repetidamente aparece nas nossas ações do jogo é chamado de **“condição”** e é muito importante na programação porque nos permite criar ações para diferentes tipos de situações.
- Para exemplificar e explicitar melhor isso, vamos fazer um outro exercício...

Novos cartões para deixar tudo mais interessante...

- **O objetivo do jogo é o mesmo:** você deve levar o robô até às luzes e acendê-las
- Você tem um número máximo de movimentos para cada exercício. Tente pensar com bastante cuidado!
- No decorrer das fases, vamos introduzindo novos conceitos e dificuldades



Frente.
Condicional

Direita

Esquerda

Luz



Acende

Buraco

Pula



Mistério

Exercício 5



				
			?	
	?			
		?		

O seu algoritmo deve ser genérico.
Vamos testar?!



Exercício 5 - Executando o algoritmo: cenário 1

				
			?- Buraco	
	? - Luz			
		? - Luz		

Exercício 5 - Executando o algoritmo: cenário 2



				
			? - Buraco	
	? - Buraco			
		? - Buraco		

Exercício 5 - Executando o algoritmo: cenário 3



				
			?- Buraco!	
	? - Luz			
		? - Luz		

Exercício 5 - Solução

1. Direita
2. Frente
3. Frente
4. Direita
5. **SE LUZ**
 - a. Frente
 - b. Acende
 - c. Frente
6. **SE BURACO**
 - a. Pula
7. Frente
8. Frente
9. Acende
10. Direita
11. **SE LUZ**
 - a. Frente
 - b. Acende
 - c. frente
12. **SE BURACO**
 - a. Pular
13. Direita
14. Frente
15. **SE LUZ**
 - a. Frente
 - b. Acende
16. **SE BURACO**
 - a. Pular

				
			?	
	?			
		?		

Exercício 5 - Solução

				
			?	
	?			
		?		

Mostrando em código

```
# -*- coding: utf-8 -*-
from movimentos import *

vire_a_direita()
ande()
ande()
ande()
if proximo_movimento == "luz":
    ande()
    acenda()
    ande()
elif proximo_movimento == "buraco":
    pule()
ande()
ande()
acenda()
vire_a_direita()
```

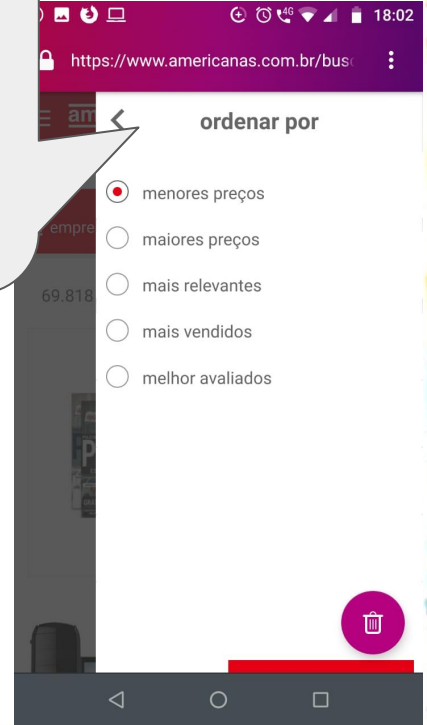
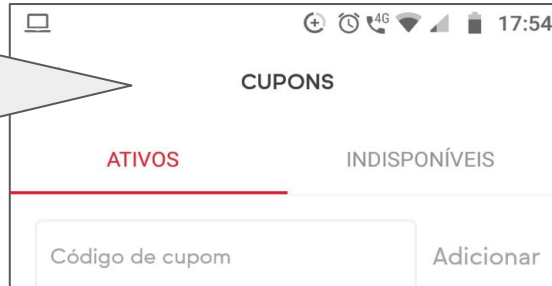
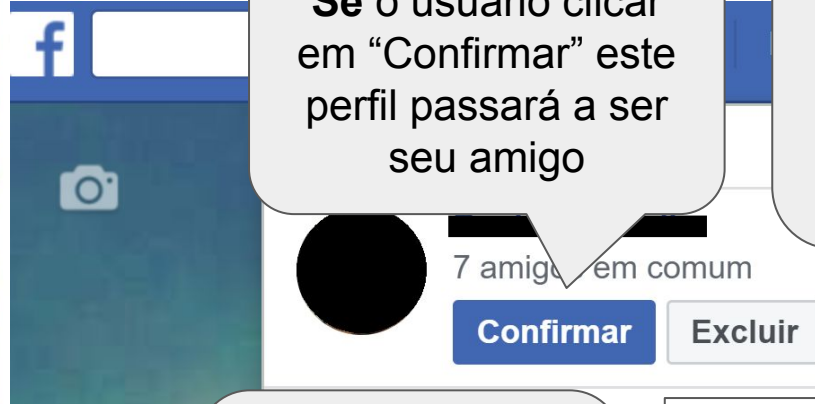
```
if proximo_movimento == "luz":
    ande()
    acenda()
    ande()
elif proximo_movimento == "buraco":
    pule()
vire_a_direita()
ande()
if proximo_movimento == "luz":
    ande()
    acenda()
    ande()
elif proximo_movimento == "buraco":
    pule()
```

Os algoritmos precisam tomar decisões o tempo






Se o usuário clicar em “Confirmar” este perfil passará a ser seu amigo

Se o usuário escolher ordenar o resultado da busca por “menor preço” então os itens mais baratos aparecerão primeiro

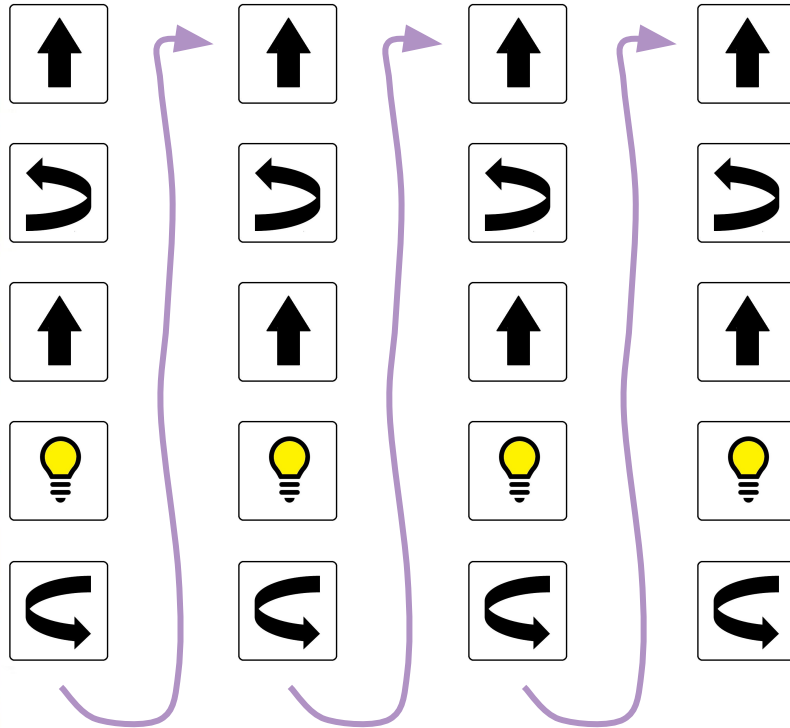
Se um cupom for adicionado, então o cliente terá um desconto no valor a ser pago








Exercício 6.1

Exercício 6.1 - Solução



Repetindo as coisas...

Não devo fazer bagunça!
Não devo fazer bagunça!
Não devo fazer bagunça!
Não devo fazer bagunça!
Não devo fazer bagunça!
Não devo fazer bagunça!
Não devo fazer bagunça!
Não devo fazer bagunça!



Repetição

- Existem alguns casos em que visivelmente estamos fazendo o **mesmo conjunto** de instruções **várias vezes seguidas** e isso pode ser bem cansativo PORÉM a programação também tem algo pra facilitar a nossa vida nesse caso: Repetições!
- Sempre que notarmos um padrão nas nossas instruções, podemos “encapsular” isso em uma repetição e dizer **“Execute esse conjunto de instruções n vezes”**

repete 10 vezes:
escrever "Não devo fazer bagunça!"



Último cartão do dia!

- **O objetivo do jogo é o mesmo:** você deve levar o robô até às luzes e acendê-las
- Você tem um número máximo de movimentos para cada exercício. Tente pensar com bastante cuidado!
- No decorrer das fases, vamos introduzindo novos conceitos e dificuldades

Legenda:



Frente
Repetição



Direita



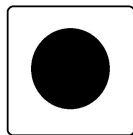
Esquerda



Luz



Acende



Buraco



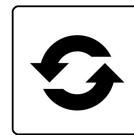
Pula








Mistério



Condicional








Exercício 6.2

Exercício 6.2 - Solução

 4 vezes








Exercício 6.2 - Solução

 4 vezes



Os algoritmos podem repetir trechos de código muitas vezes...



Em uma tela será exibido 25 itens. Ou seja, o algoritmo irá **repetir 25 vezes** operação de **mostrar item**.

Os algoritmos podem repetir trechos de código muitas vezes...

exemplo@exemplo.com

Senha

Fazer login

[Esqueceu a senha?](#)

exemplo@exemplo.com









Senha

O e-mail e a senha que você digitou não coincidem.

Fazer login

[Esqueceu a senha?](#)









Exercício 7

Exercício 7 - Solução

 3 vezes











Exercício 7 - Solução

 3 vezes



Exercício 7 - Uma outra solução possível



















3 vezes



	💡		●	🤖
	●			
				●
	💡	●		💡

Desafio

Solução - Com dois blocos de repetição



4 vezes



Desafio - Solução

Principal: direita -> repetição_1

Repetição 1: repetição_2 -> direita -> repetição_2 -> repetição_1

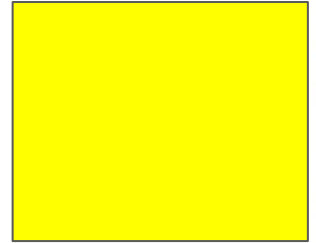
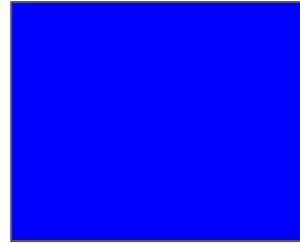
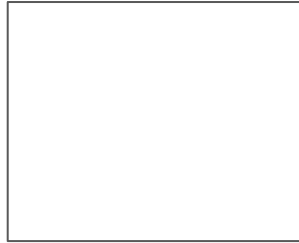
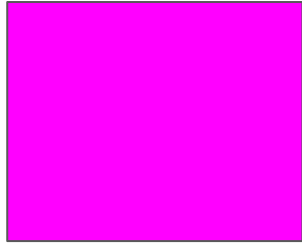
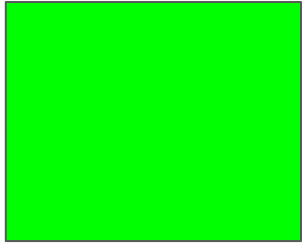
Repetição 2: pulo - acende

O mais **leve** e o mais
pesado - Algoritmo de
ordenação

Introdução

- Os computadores são muitas vezes utilizados para **colocar listas em algum tipo de ordem**, por exemplo:
 - Nomes em ordem alfabética;
 - E-mails ordenados por data;
 - Itens em ordem numérica;
 - Produtos ordenamos por preço.
- Classificar listas nos ajuda a encontrar as coisas rapidamente, e também facilita a identificação dos valores extremos.
- **Existem vários algoritmos de ordenação**, vamos tentar descobrir alguns?

Vamos ordenar papéis coloridos



- Cada papel possui um “peso”
- Podemos pedir ajuda para a **balança humana** para comparar **dois papéis por vez**. Por exemplo: Qual é o mais leve: o **verde** ou o **rosa**? E teremos como resposta: **verde** (ou **rosa**, dependendo da nossa balança).
- **Objetivo:** ordenar os papéis do mais leve para o mais pesado. E anotar os passos que foram realizados para isso.

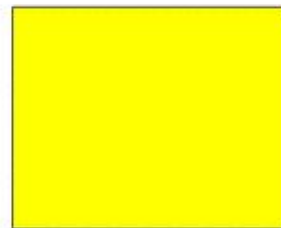
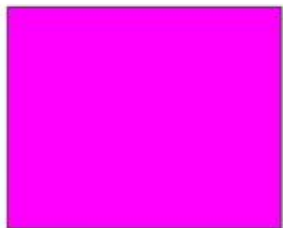
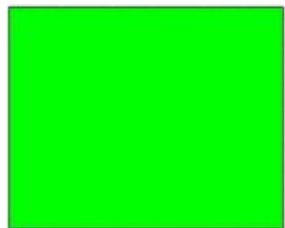
Uma possível solução...

Ordenação por seleção



1. Encontre o papel mais leve e separe ele em um canto. Como encontrar o mais leve?
 - a. Quem é mais leve: verde ou rosa? Rosa. (fico com o rosa)
 - b. Quem é mais leve: rosa ou branco? Branco. (fico com o branco)
 - c. Quem é mais leve: branco ou azul? Azul. (fico com o azul)
 - d. Quem é mais leve: azul ou amarelo? Azul. (fico com o azul) **Azul** é o primeiro mais leve.
2. Repita este processo com os papéis que sobraram até que não sobrem mais papéis.

Ordenação por seleção



Existe outra forma de
ordenar?

Os pesos dos papéis

0

10

20

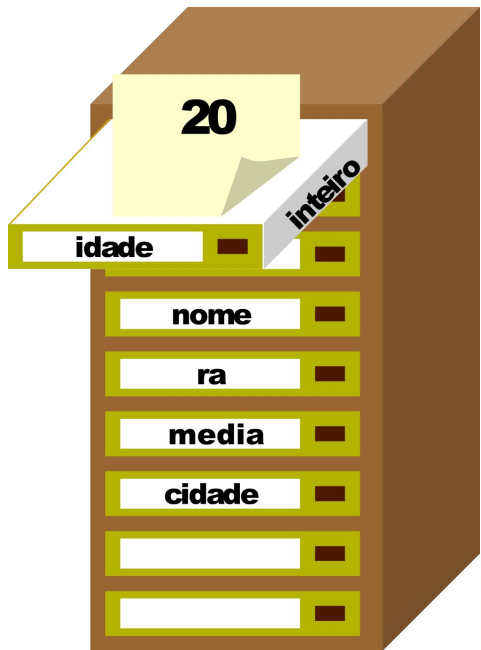
30

40

Basquete das variáveis

O que são variáveis?

- A maioria dos programas recebem e dados. Esses **dados precisam ser armazenados no computador** para serem utilizados no processamento. Esse armazenamento é feito na memória.
- Uma **variável representa uma posição de memória**.
- Uma variável possui um nome e um tipo e seu conteúdo pode variar ao longo do tempo, durante a execução do programa.
- Uma variável só pode armazenar um valor a cada instante!



Basquete de variáveis

Existem alguns tipos padrão de variáveis:

- Strings: textos
- Int: números inteiros
- Float: números racionais (com casas decimais)

Agora vamos mostrar algumas variáveis e vocês vão nos ajudar a descobrir qual o tipo de cada variável.

Calculadora simples:

Como somar dois números?

- Ok! Já sabemos de variáveis, de condições e repetições.
- Para treinarmos esses conceitos todos, vamos implementar uma calculadora.
- Essa calculadora vai primeiro receber a operação a ser feita (+, -, *, /, **) e os dois números que entrarão na conta. Se o usuário digitar “sair”, o programa deve ser encerrado, caso contrário deve continuar perguntando por operações

Esperamos que
tenham gostado de
passar a manhã junto
com nós!



Quer mais?

- Jogue online: <https://www.lightbot.com/flash.html>
- Made with code: <https://www.madewithcode.com/>
- Hora do código: <http://programae.org.br/horadocodigo/>

Vem falar com a gente!



facebook.com/PyLadiesSorocaba



twitter.com/PyLadiesSorocaba



linkedin.com/company/PyLadiesSorocaba



instagram.com/PyLadiesSorocaba



sorocaba@pyladies.com