

# MC102W - Algoritmos e Programação de Computadores

## Lab09: Jogo da vida

**Prazo:** 24 de Maio de 2020.

**Peso da Atividade:** 4

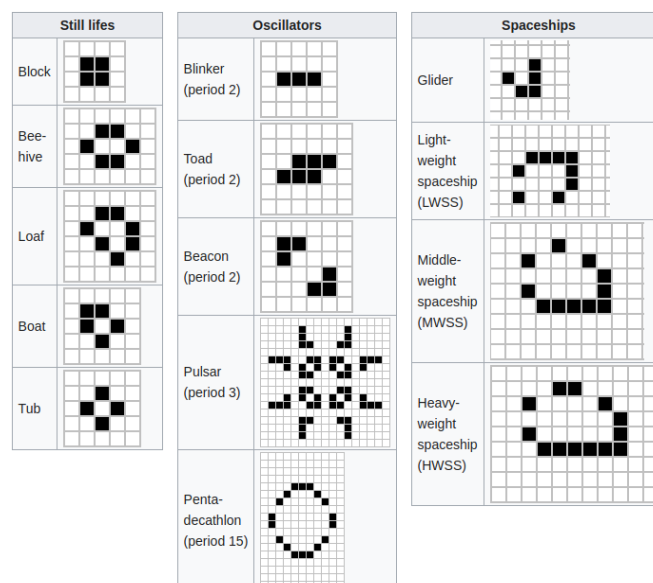
O **Jogo da vida** é um autômato celular criado para reproduzir, por meio de regras simples, modificações em grupos de seres vivos, tendo aplicações em várias áreas da ciência. O jogo foi desenvolvido pelo matemático britânico **John Horton Conway**, que infelizmente faleceu, nesse ano (2020) com 82 anos, devido ao Covid-19.

No Jogo da vida, as regras são definidas e aplicadas a cada nova geração. Assim, a partir de uma imagem em um tabuleiro bi-dimensional definida pelo jogador, percebe-se mudanças inesperadas e belas a cada nova geração, podendo ter um padrão fixo ou caótico.

**As regras do Jogo da vida são simples:**

- 1) Qualquer célula viva com menos de dois vizinhos vivos morre de solidão.
- 2) Qualquer célula viva com mais de três vizinhos vivos morre de superpopulação.
- 3) Qualquer célula morta com exatamente três vizinhos vivos se torna uma célula viva.
- 4) Qualquer célula viva com dois ou três vizinhos vivos continua no mesmo estado para a próxima geração.

A figura Abaixo, mostra padrões, gerados pelo Jogo da vida, que são fixos, oscilam e se movimentam



Acessem a Fonte para mais gifs e imagens:

[https://en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)

# Tarefa

Nessa tarefa, você deverá implementar o **Jogo da Vida**.

O programa deve ser em python e deve abranger os seguintes tópicos:

- ☐ Matrizes e objetos multidimensionais

## Entrada

O seu programa deve ter as **dimensões N e M do tabuleiro** (NxM - dimensões podem ser diferentes), **quantidades de iterações do jogo**, **quantidades de células vivas iniciais** e **as posições de cada uma no tabuleiro**. Todas as entradas serão inteiros.

Obs.: Para cada posição de células vivas iniciais, siga o padrão: **linha,coluna**

**Com isso, pode-se extrair usando a função split.**

## Saída

O seu programa deve gerar como saída **cada iteração do tabuleiro (todas as gerações), contando com inicial**.

para representar uma célula viva, imprima '+' (mais), para a morta, apenas um '.' (ponto). Deve haver um **print('-')** entre tabuleiros

- Para evitar quebra de linha, exemplo: **print('\*', end="")**
- Para quebrar linha, use: **print("\n")**

# Exemplos

A grafia da saída abaixo deve ser seguida rigorosamente por seu programa.

Exemplo 1:

## Entrada

```
8
9
2
3
2, 2
2, 3
2, 4
```

## Saída

```
. . . . .
. . . . .
..+++..
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
-
. . . . .
...+. . .
...+. . .
...+. . .
. . . . .
. . . . .
. . . . .
. . . . .
-
. . . . .
. . . . .
..+++..
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
-
```

Exemplo 2:

Entrada

7
10
2
8
2, 2
2, 3
3, 2
3, 3
4, 4
4, 5
5, 4
5, 5

Saída

.....
.....
..++.....
..++.....
....++....
....++....
.....
-
.....
.....
..++.....
..+.....
.....+
....++....
.....
-
.....
.....
..++.....
..++.....
....++....
....++....
.....
-

Exemplo 3:

Entrada

8
9
2
6
2, 2
2, 3
2, 4
3, 1
3, 2
3, 3

Saída

. . . . .
. . . . .
. . + + + . . . .
. + + + . . . . .
. . . . .
. . . . .
. . . . .
. . . . .
-
. . . . .
. . . + . . . . .
. + . . + . . . .
. + . . + . . . .
. . + . . . . .
. . . . .
. . . . .
. . . . .
-
. . . . .
. . . . .
. . + + + . . . .
. + + + . . . . .
. . . . .
. . . . .
. . . . .
. . . . .
-

## Critérios específicos

Os seguintes critérios específicos sobre o envio e implementação devem ser satisfeitos.

i. Submeter no SuSy os arquivos:

⇒ **lab09.py**: Arquivo onde deverá ser implementada a tarefa.

## Observações gerais

No decorrer do semestre haverá 3 tipos de tarefas no SuSy (descritas logo abaixo). As tarefas possuirão os mesmos casos de testes abertos e fechados, no entanto o número de submissões permitidas e prazos são diferentes. As seguintes tarefas estão disponíveis no SuSy:

- ☐ **lab09-AmbienteDeTeste**: Esta tarefa serve para testar seu programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém os arquivos submetidos aqui **não serão corrigidos**.
- ☐ **lab09-Entrega**: Esta tarefa tem limite de uma **única** submissão e serve para entregar a **versão final** dentro do prazo estabelecido para o laboratório. Não use essa tarefa para testar o seu programa e submeta aqui apenas quando não for mais fazer alterações no seu programa.
- ☐ **lab09-ForaDoPrazo**: Esta tarefa tem limite de uma **única** submissão e serve para entregar a versão final fora prazo estabelecido para o laboratório. Esta tarefa irá substituir a nota obtida na tarefa **lab09-Entrega** apenas se o aluno tiver realizado as correções sugeridas no *feedback* ou caso não tenha enviado anteriormente na tarefa **lab09-Entrega**.

## Avaliação

Este laboratório será avaliado da seguinte maneira: se o seu programa apresentar resposta correta para todos os casos de teste do SuSy, código de qualidade e comentários no código explicando o que está sendo feito, então é nota 10; caso contrário pontos serão descontados da nota final.

## Testando seu programa

Para testar se a solução do seu programa está correta, basta seguir o exemplo abaixo.

```
python lab09.py < arq01.in > arq01.out
diff arq01.out arq01.res
```

O `arq01.in` é a entrada e `arq01.res` é a saída esperada, ambos disponíveis no SuSy. O `arq01.out` é a saída gerada pelo seu programa. Após o prazo, os casos de teste fechados serão liberados e podem ser baixados e testados da mesma forma que os testes abertos.