



**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**  
**CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO**

**Trabalho Computacional – Jogo da Vida**

**Data de Entrega:** 17/08/2025 (até as 23:59).

**Pontuação:** código(cod) até 10 pontos e entrevista individual(ent) até 1 ponto.

Nota final = ent \* cod

**O TRABALHO É INDIVIDUAL**

**Forma de Entrega:** submissão no AVA

**Observações Importantes:**

- O trabalho que não estiver compilando não será corrigido.
- As saídas do seu código serão comparadas de forma automática com as do professor. Trabalhos com saídas diferentes receberão nota ZERO.
- Trabalhos evidentemente iguais receberão nota ZERO.
- O agendamento das entrevistas será realizado através de uma planilha do Google Docs que será disponibilizada no AVA após a data de entrega do trabalho.
- Os códigos devem apresentar as seguintes características, que serão consideradas na correção:
  1. Indentação correta;
  2. Modularização: Crie funções para modularizar o seu programa;
  3. Estratégias de solução visando a simplificação do código, evitando repetição de operações desnecessárias;
  4. Saída na formatação correta;
  5. Documentação: o seu programa deve ser todo comentado, explicando o que está sendo codificado. O início do código deve conter um cabeçalho seguindo o seguinte formato:

/\*\*\*\*\*

Trabalho Computacional - Programação Estruturada/Programação II - 2025/1

<nome completo do programador>

\*\*\*\*\*/

**Leia atentamente TODO o enunciado do trabalho (a especificação do problema e os detalhes sobre a confecção, submissão e avaliação do trabalho). Uma leitura inapropriada do enunciado pode ser extremamente danosa à sua nota.**



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO

## O Jogo da Vida

O jogo da vida é um autômato celular desenvolvido pelo matemático britânico John Horton Conway em 1970. É o exemplo mais bem conhecido de autômato celular. O jogo foi criado de modo a reproduzir, através de regras simples, as alterações e mudanças em grupos de seres vivos, tendo aplicações em diversas áreas da ciência. As regras definidas são aplicadas a cada nova "geração"; assim, a partir de uma imagem em um tabuleiro bidimensional, percebem-se mudanças muitas vezes inesperadas e belas a cada nova geração, variando de padrões fixos a caóticos.

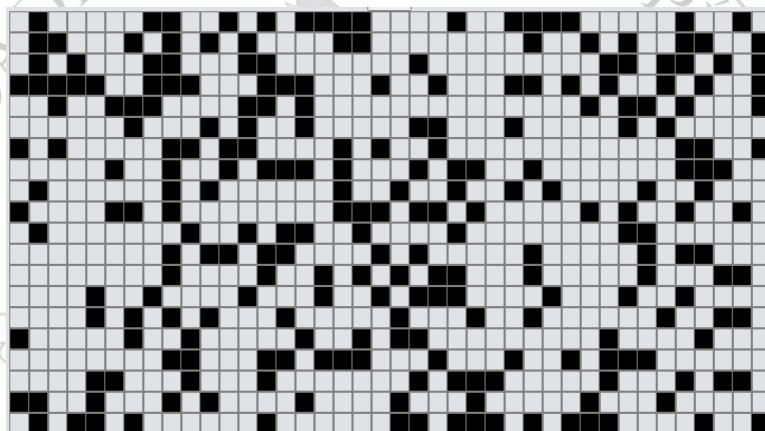


Fig. 1 - Tabuleiro do Jogo da Vida.

O jogo da vida se passa em um arranjo bidimensional de células que podem assumir dois estados: vivo ou morto (células pretas e brancas, respectivamente, na Fig. 1). Cada célula interage com suas oito vizinhas, as células adjacentes horizontal, vertical e diagonalmente. O jogo evolui em unidades de tempo discretas chamadas de gerações. A cada nova geração, o estado do jogo é atualizado pela aplicação das seguintes regras:

1. Toda célula morta com exatamente três vizinhos vivos torna-se viva (nascimento).
2. Toda célula viva com menos de dois vizinhos vivos morre por isolamento.
3. Toda célula viva com mais de três vizinhos vivos morre por superpopulação.
4. Toda célula viva com dois ou três vizinhos vivos permanece viva.

**Leia atentamente TODO o enunciado do trabalho (a especificação do problema e os detalhes sobre a confecção, submissão e avaliação do trabalho). Uma leitura inapropriada do enunciado pode ser extremamente danosa à sua nota.**



**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**  
**CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO**

As regras são aplicadas simultaneamente em todas as células para chegar ao estado da próxima geração. Supondo um cenário onde temos um tabuleiro com 5 linhas e 5 colunas e queremos ver o resultado do tabuleiro após 3 gerações, temos o seguinte cenário:

Tabela 1 - Resultado após 3 gerações para um tabuleiro 5x5. Pontos representam células mortas e zeros representam células vivas

Tabuleiro Inicial	Geração 1	Geração 2	Geração 3

### Especificações do Trabalho

Seu trabalho deverá ser desenvolvido na linguagem de programação C. Nele você irá trabalhar, no mínimo, com os seguintes conteúdos vistos ao longo da disciplina:

- Matrizes
- Strings
- Structs
- Ponteiros
- Alocação Dinâmica
- Funções
- Manipulação de Arquivos.

Seu tabuleiro deverá ser representado por uma estrutura que deve conter os seguintes campos:

- Nome do arquivo do tabuleiro
- Nome do arquivo de saída do tabuleiro
- Número de linhas do tabuleiro
- Número de colunas do tabuleiro
- Número de gerações do jogo da vida
- Um ponteiro para ponteiros de células (será a sua matriz bidimensional)

**Leia atentamente TODO o enunciado do trabalho (a especificação do problema e os detalhes sobre a confecção, submissão e avaliação do trabalho). Uma leitura inapropriada do enunciado pode ser extremamente danosa à sua nota.**



**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**  
**CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO**

Ao ser executado seu programa deverá exibir um menu, que será **REPETIDO**, até que o usuário digite -1:

```
Menu de opcoes:  
1 - Abrir arquivo de entrada  
2 - Preencher tabuleiro de entrada  
3 - Rodar geracoes  
Escolha sua opcao(-1 para sair):
```

A cada vez que o usuário escolher uma opção, seu programa deverá executar um conjunto de tarefas e ao final, exibir novamente o menu. Inclusive, posso escolher uma **MESMA OPÇÃO** várias vezes, seu programa precisará lidar com isso.

#### Especificações das Opções do Menu:

- Opção 1 – Abrir arquivo de entrada:

Criar e chamar uma função que: solicite ao usuário o nome do arquivo de entrada e guarde-o no respectivo campo da sua estrutura que representa o tabuleiro. Seu programa deve abrir este arquivo em modo leitura.

Você deve verificar se o arquivo foi aberto com sucesso, em caso negativo, imprima a seguinte mensagem para o usuário:

**Falha ao abrir o arquivo! Voltando ao menu inicial!**

Neste momento você também deve criar o nome do arquivo de saída que deverá ser **SEGUIDO À RISCA!** Você deverá guardar o nome em seu respectivo campo na estrutura do tabuleiro. O nome do arquivo de saída é composto pelo nome do arquivo de entrada (até antes de sua extensão), seguido do sufixo “\_out.txt”. Por exemplo:

Nome do arquivo de entrada: 9x9.txt

Nome do arquivo de saída: 9x9\_out.txt

**Leia atentamente TODO o enunciado do trabalho (a especificação do problema e os detalhes sobre a confecção, submissão e avaliação do trabalho). Uma leitura inapropriada do enunciado pode ser extremamente danosa à sua nota.**



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO

- Opção 2 – Preencher o tabuleiro de entrada:

<pre>5 5 3 .... ..0.. ...0. .000. .....</pre>	<p>Linha 1 – número de linhas, colunas do tabuleiro e quantidade de gerações do jogo da vida separados por espaço.</p> <p>Demais linhas – tabuleiro do jogo da vida. <b>PONTO</b> representa célula morta e <b>ZERO</b> representa uma célula viva</p>
---	--

Você deverá criar e chamar uma função que lê os valores do arquivo de entrada e aloca dinamicamente uma matriz, ou seja, ler os valores da linha 1 para os seus respectivos campos na estrutura tabuleiro e alocar a matriz definida na sua estrutura tabuleiro com o respectivo número de linhas e colunas lido do arquivo de entrada.

- Opção 3 – Rodar gerações:

Você deverá criar e chamar uma função que rode as gerações do jogo da vida. O resultado das gerações deverá ser salvo no arquivo de saída que você criou o nome na Opção 1.

**DICA IMPORTANTE:** As células da geração corrente só deverão ser atualizadas após a análise de **TODAS** as células do tabuleiro.

Seu arquivo de saída deverá seguir **EXATAMENTE** o seguinte padrão:

**Leia atentamente TODO o enunciado do trabalho (a especificação do problema e os detalhes sobre a confecção, submissão e avaliação do trabalho). Uma leitura inapropriada do enunciado pode ser extremamente danosa à sua nota.**



**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**  
**CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO**

<p>Geracao 0:</p> <pre> . . . . . . . 0 . . . . . 0 . . 0 0 0 . . . . . . Geracao 1: . . . . . . . 0 . . . . 0 0 . . . 0 . . Geracao 2: . . . . . . . 0 . . . . 0 0 . . . 0 0 . Geracao 3: . . . . . . . 0 . . . . . 0 0 . . 0 0 . </pre>	<ul style="list-style-type: none"> <li>- Geracao 0 é o tabuleiro que você leu do arquivo de entrada e as gerações seguintes são a quantidade de gerações que você leu do arquivo de entrada.</li> <li>- Os <b>PONTOS</b> e <b>ZEROS</b> estão separados por um espaço, inclusive nos elementos da última coluna.</li> <li>- No final do arquivo há uma linha em branco</li> </ul>
---	---

**Comparando Seu Arquivo de Saída com as Saídas Disponibilizadas**

Para garantir que a saída do seu programa está de acordo com as saídas dos casos de teste disponibilizadas, basta rodar o script `diff_files.py` da seguinte forma:

```
python3 diff_files.py <arquivo_caso_teste.txt> <sua_saida.txt>
```

Caso tudo esteja tudo correto, ao final da execução será impresso "Os arquivos são iguais.". Caso contrário, o script mostrará o número da linha e o que está diferente entre os arquivos.

**Leia atentamente TODO o enunciado do trabalho (a especificação do problema e os detalhes sobre a confecção, submissão e avaliação do trabalho). Uma leitura inapropriada do enunciado pode ser extremamente danosa à sua nota.**