**Trabalho Final**

**Microprocessadores**

**Turno P7**

**19/12/2022**

**Alunos:**

Diogo Ventura Nº 62413

Francisco Ribeiro Nº 62695

Miguel Ferreira Nº 63064

**Índice**

# 

[**1. Introdução**](#_2svrthnp1r9u) **3**

[**2. Descrição dos requisitos**](#_1vh4gz8vrzvc) **4**

[**3. Apresentação da arquitectura implementada**](#_2svrthnp1r9u) **5**

[**4. Detalhe dos módulos do Sistema**](#_2svrthnp1r9u) **6**

[**5. Testes e Resultados**](#_2svrthnp1r9u) **7**

[**6. Conclusões**](#_2svrthnp1r9u) **8**

[**7. Referências Bibliograficas**](#_rbz9gep1vxbu) **9**

[**8. Anexos**](#_t88ym51sa2yb) **10**

# 

# 

# 1. Introdução

Na realização deste trabalho teve-se como principal objetivo desenvolver o jogo “ The Game of Life”. Jogo que consiste em uma grelha de celulas que, tendo em conta uma serie de regras matematicas, podem viver, morrer ou multiplicar. Desevolveu-se o jogo em linguagem Assembly e utilizou-se o Emu8086 como compilador. Para além disso pretendeu-se avaliar as seguintas competencias:

* Capacidade para trabalhar com a linguagem Assembly
* Manipulação de ficheiros
* Trabalho em equipa e entreajuda

O presente Relatorio encontra-se dividido em diversos capitulos:

Na Descrição dos Requisitos falamos sobre todas as funções necessárias no trabalho, tais como, “JOGAR”, “CARREGAR”, “TOP5”, “GUARDAR” e “SAIR”.

No capítulo sobre Apresentação da arquitectura implementada mostramos a interface do jogo e a sua execução.

No Detalhe dos módulos do Sistema, mostramos o funcionamento de alguns algoritmos usados no programa tal como a organização do Top5 no ficheiro.

Nos Testes e Resultados mostramos um exemplo do que se pode fazer no jogo, ou seja, o comportamento do programa.

# 2. Descrição dos requisitos

No inicio do jogo, é apresentado um menu com diversas opções em que o utilizador, com o botão direito do rato, seleciona a opção pretendida.

A opção principla é o JOGAR. Inicialmente é pedido o nome do jogador, depois o utilizador pode selecionar até 15 células e quando clicar em “Iniciar” o jogo começa. Durante o processo de verificações aparece à direita uma barra de progresso consoante o progresso das verificações. O jogador pode realizar as gerações que pretender e se o utilizador quiser deixar de jogar pode clicar em “Sair”. À medida que o jogador vai jogando o número de células e gerações vai atualizando no ecrã. O Top5 e log.txt vão sendo atualizados quando o utilizador sai de cada jogo.

Outra opção é GUARDAR, onde é guardado num ficheiro binário a informação à cerca do jogo e a posição das células.

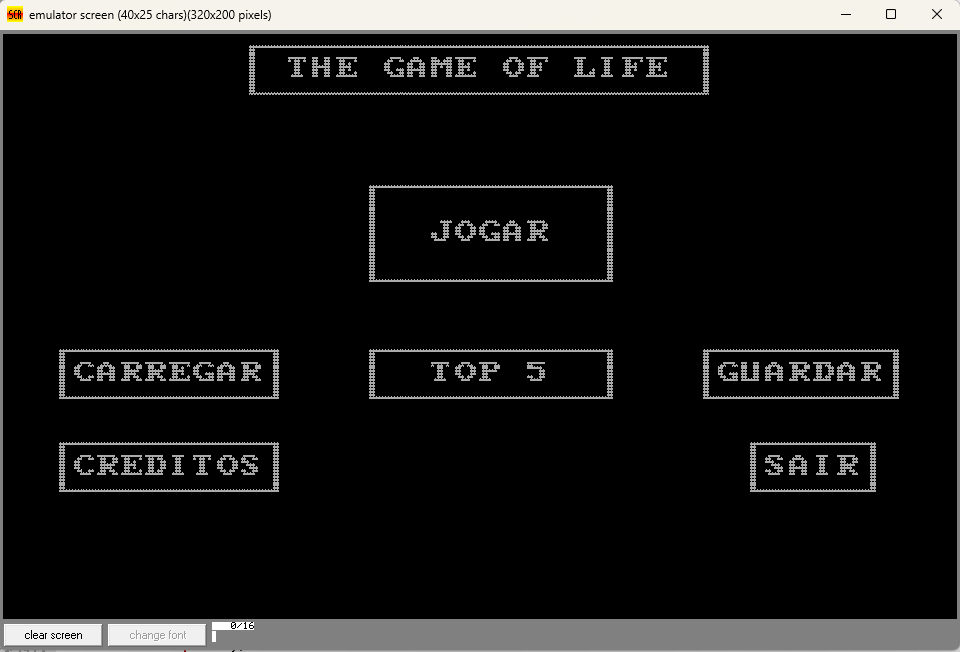
Outra opção é CARREGAR, onde é pedido ao utilizador para introduzir o nome do ficheiro e é passado para o ecrã do jogo o estado previamente guardado no ficheiro. Aqui o utilizador pode alterar algumas células e continuar numa nova geração.

Outra opção é TOP5, onde é carregado um ficheiro TOP5.txt os dados e apresenta no ecrã de forma a mostrar ao utilizador. Aguarda que o utlizador pressione uma tecla ou se durante 10 segundos nenhuma tecla for pressionada volta ao Menu.

Outra opção é CREDITOS, onde é apresentado o nome e os números dos alunos. Aguarda que o utlizador pressione uma tecla ou se durante 10 segundos nenhuma tecla for pressionada volta ao Menu.

Por último, existe a opção SAIR que sai do programa deixando uma mensagem de despedida.

# 3. Apresentação da arquitectura implementada

* Ao iniciar o programa é apresentado o Menu:

É nesta interface que todo o jogo se desenvolve, o jogador pode selecionar qualquer opção que o jogo lhe vai apresentando, quando uma opção termina o utilizador é redirecionado novamente para este menu onde pode escolher outra opção, este afirmação é verdadeira para todas as opções do presente jogo excepto a opção Sair onde o programa termina.

Durante a execução do jogo o número de células e gerações vai alterando consoante o conteúdo dentro do terreno do jogo. No lado direito existe uma barra de progresso que vai evoluindo consoante o decorrer do jogo. O utilizador pode realizar as gerações que desejar.

A screenshot of a computer

Description automatically generated with medium confidence

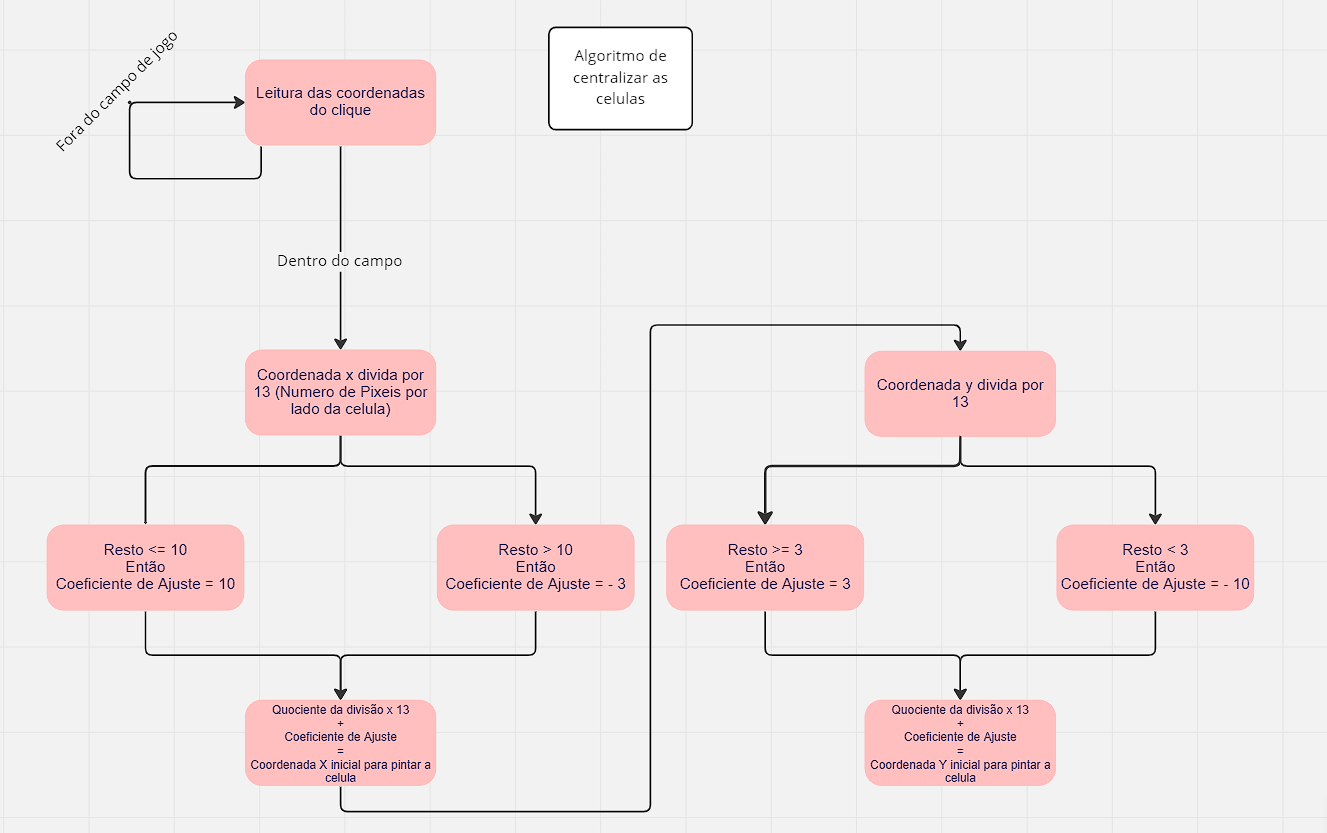
À medida que os jogos vão ocorrendo foi necessário a criação de ficheiros para conseguir guardar as opções existentes no menu. Para isso foi criado um ficheiro Top5.txt que vai guardando e atualizando os status dos cinco melhores jogadores.

Existe também um ficheiro Log.txt que sempre que um jogador sai do jogo é atualizado com as informações do jogador.

Cria-se também um ficheiro com o nome “xxxxdd\_mm\_aaaa\_hh\_mm” e extensão “.gam” quando é utilizada a opção Guardar. De seguida temos a opção de carregar o ficheiro que guardámos.

# 4. Detalhe dos módulos do Sistema

Dentro do jogo, irá haver a possibilidade de colocar células, contudo não é aleatório a colocação pelo utilizador, estas deverão ser centralizadas. Então como solução inventou-se um algoritmo para que se pudesse resolver a centralização.



A necessidade do ajuste é devido ao inicio do campo do jogo (coordenadas que não são multiplas do número de pixeis de lado por célula).

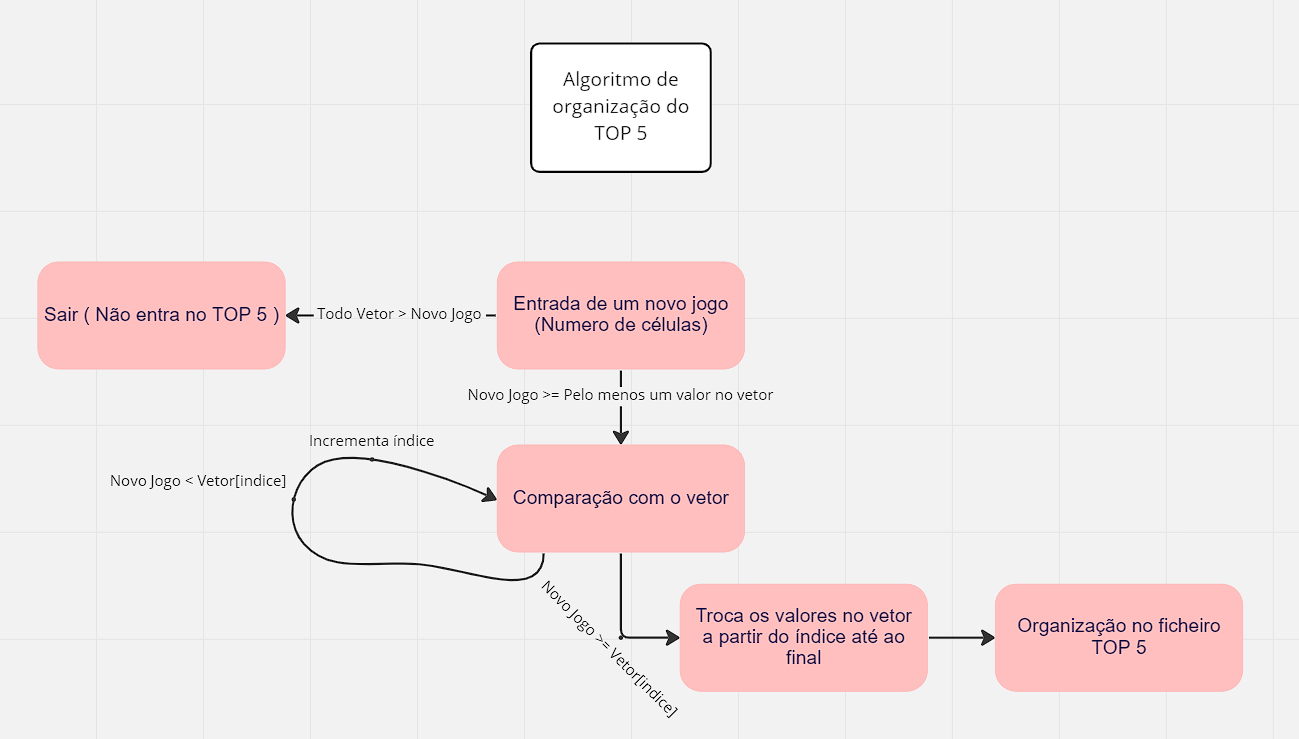
Após isso quando se clica iniciar, irá verificar o ecrã à procura de células vivas e colocar num vetor o valor 1 caso ela esteja viva e 0 caso contrário. Escusado é dizer que ele coloca no vetor conforme a posição da célula.

De seguida irá se fazer comparações dentro do vetor de modo a obter a proxima geração. Então usamos um outro vetor para puder se guardar os resultados da comparação (que no fundo é a nova geração).

Na comparação, é necessário comparar com as 8 células à volta da célula em causa, mas nos cantos nunca irá haver 8 células pelas quais são necessárias para a comparação. Então, como solução, usou-se um campo com “células mortas à volta”, (como se fosse uma camada à volta do campo) de modo que assim, em qualquer lado do campo, irá sempre ter 8 células para comparar. No final das comparações irá colocar no ecrã as novas células e apagar as antigas conforme o segundo vetor (nova geração).

Concluindo, é possível fazer geração a seguir de geração enquanto o jogador quiser ou apenas sair.

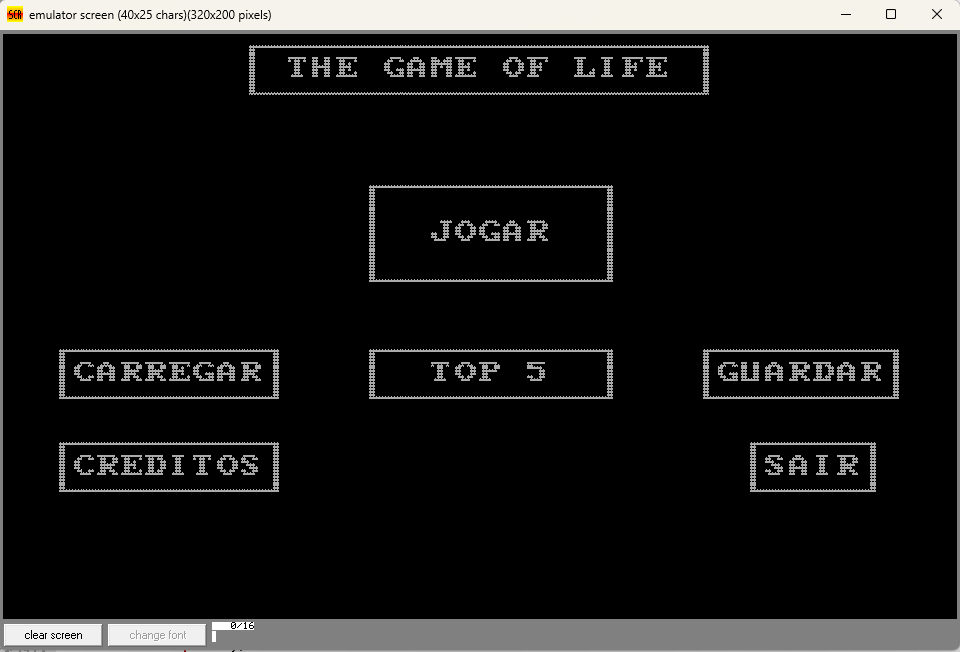
Se o utilizador sair, o Top5 será atualizado, cujo valores deste estão num vetor de modo a puder ordenar o ficheiro. Isto é, se o o número de células for menor que todos os que estão no vetor, ele não irá fazer nada, como está em baixo.



O guardar é basicamente um ‘dump’ do vetor inicial mais as estatisticas do jogo pretendido ( geração, células e nome) e o carregar é a leitura desse ficheiro e colocar no ecrã a geração e as estatísticas.

O ficheiro Log tem os mesmos principios que o TOP 5, contudo não é necessário ordem nem nenhum algoritmo, apenas guarda num ficheiro a informação dos jogadores.

# 5. Testes e Resultados

* No início do jogo clicamos numa tecla qualquer e apresenta o menu do jogo:
* Depois clicamos “JOGAR” e o programa pede o nome do jogador:

Graphical user interface

Description automatically generated

* Agora temos a opção de colocar as celulas e iniciar o jogo:

Text

Description automatically generated

* Colocamos as células:

A picture containing graphical user interface

Description automatically generated

* Clicamos “INICIAR” e o jogo começa e realiza a proxima geração. Quando o programa acaba de fazer a geração temos a opção de fazer a próxima geração ou então sair para o menu do jogo:

A screenshot of a computer

Description automatically generated with medium confidence

* No menu temos vários botões como, por exemplo, o guardar. Nesta opção guardamos num ficheiro a geração, as células e as suas posições.

Graphical user interface

Description automatically generated

* Graphical user interface

  Description automatically generatedDepois de guardar os ficheiros dos jogadores podemos carregá-los clicando “CARREGAR” e escrevendo o nome do jogador:

A screenshot of a computer

Description automatically generated with medium confidence

* Quando o jogo carrega temos a opção de colocar mais células, iniciar ou sair (neste teste coloquei mais células):

Graphical user interface

Description automatically generated

A picture containing graphical user interface

Description automatically generated

* O jogo depois de carregado faz a próxima geração:

A picture containing graphical user interface

Description automatically generated

* Clicamos sair e podemos voltar a guardar o jogo para ficar atualizado nos ficheiros:

Graphical user interface

Description automatically generated

* De seguida clicamos na opção “TOP 5”:
* Depois clicamos na opção “CREDITOS”:

Text

Description automatically generated

* Text

  Description automatically generatedPor fim, clicamos “SAIR”:

# 6. Conclusões

Neste trabalho, no âmbito da cadeira de Microprocessadores, concretizou-se todos os objetivos propostos, desde o próprio jogo a funcionar até ao Top5. Na nossa opinião, poderiamos ter organizado melhor o código, aperfeiçoado algumas funções e fazer algumas extras. Estas são umas ideias que pretendíamos concretizar :

* A possibilidade do jogador puder escolher o tamanho das células;
* A existência de uma dificuldade de jogo (isto é, aumentar ou diminuir o número possivel de células colocadas);
* Um Menu mais colorido;
* E até um ‘easter egg’.

Um dos inúmeros problemas foi a comparação a partir de uma geração, no qual usámos um vetor para pudermos concretizar, contudo é necessário 8 células para dizer algo sobre a célula, mas isso não acontece nas pontas do ecrã. Encontramo-nos num impasse, tinhamos duas soluções, ou fazer os casos especificos das pontas (que iria reduzir a eficiência do programa), ou faríamos uma “camada extra de células mortas”, pudendo assim aceder ao vetor e comparar sem qualquer problema.

Outro problema, foi o algortimo da centralização, que inicialmente não havia problema, pois não havia uma área limitada do jogo, mas devido à demora do jogo com muitas células, vimo-nos obrigados a reduzir o tamanho das células e o tamanho da área de jogo. Com uma nova área de jogo, vieram problemas, tais como escrever por cima das barreiras, e não era possivel colocar uma célula corretamente na ponta da área de jogo.

No final apercebemo-nos que havia uma solução, através do resto da divisão podiamos adicionar ou retirar ao valor final ‘semi-centralizado’ (significa que fazia parte da centralização e não era completamente funcional) um valor para corrigir esse problema. (Ver Imagem 1.)

Pontos fortes no nosso trabalho são:

* A existência de uma barra de progresso, em que vai enchendo à medida que vai realizando a nova geração;
* A eficiencia e eficacia dos algoritmos de comparacão e de colocar as células;
* O sair;
* A organização no ficheiro TOP 5 e como é realizada.

# 7. Referências Bibliograficas

As referências devem ser identificadas no texto com o seguinte formato [Apelido do 1º Autor, ano] (e.g. [Taub, 2013]) e aqui deve ser colocada a referência inteira que permita ao leitor chegar ao documento, no seguinte formato:

Taub, 1982 Herbert Taub, “Digital Circuits and Microprocessors”, McGraw-Hill Book Co., 1982, ISBN: 0070629455)

Nota: as referências devem ser ordenadas alfabéticamente e depois por ano de publicação. Caso tenham várias publicações do mesmo autor no mesmo ano, devem juntar ao final do número do ano uma letra sequencial a partir de “a” (e.g. [Taub, 1982a])>