

Tenologias e Arquiteturas de Computadores

2021 / 2022

Trabalho Prático

Ricardo Almeida de Aguiar Tavares - 2021144652 – LEI

João Choupina Ferreira da Mota - 2020151878 – LEI

Coimbra, 12 de junho de 2022

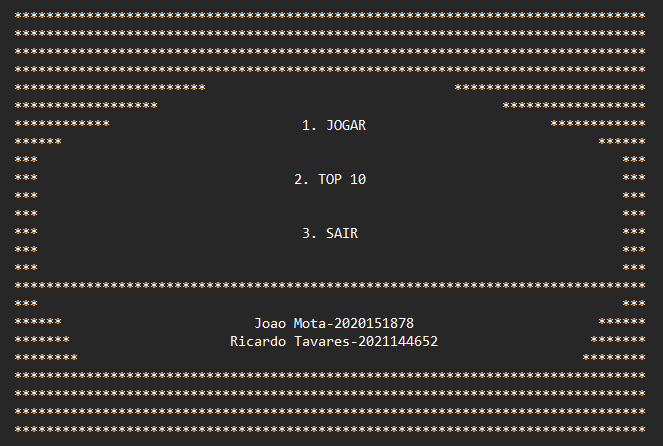
# Introdução

Projeto criado no âmbito da unidade curricular de Tecnologias e Arquiteturas de Computadores com o intuito de criar um jogo em linguagem *Assembly 8086.*

O jogo recriado é bastante popular tendo o nome de Sopa de Letras. Este tem como objetivo encontrar as palavras indicadas no lado direito do tabuleiro.

# Funcionamento

## Menu



## Jogo

Escolhendo a 1ª opção visível no Menu (1. Jogar) , o utilizador é automaticamente direcionado para um segundo menu, onde o utilizador poderá escolher o nível de jogo que pretende jogar (1. BASICO) (2. AVANCADO).

Uma imagem com texto

Descrição gerada automaticamente

Este utilizará as teclas ( ← , ↑ , → , ↓ ) para se mover dentro do tabuleiro e a tecla ENTER para selecionar cada letra da palavra encontrada. O mesmo acontece no jogo avançado, onde apenas é alterado o número de palavras a encontrar. O tempo será limitado e assim que o tempo chegue ao fim, é apresentada a interface “Winner”, apresentada na figura abaixo, e a sua pontuação irá ser adicionada à lista dos TOP10 jogadores. Caso desista a meio do jogo irá ser redirecionado para o menu “Game Over “.





Uma imagem com texto

Descrição gerada automaticamente

## Top 10

Usando a opção TOP10 disponível no menu será requerido ao jogador o seu nome de utilizador que ficará visível na lista. A permissão para o uso desta opção apenas será concedida caso a pontuação obtida num único jogo exceda a do último colocado já presente na lista. Quase a pontuação não seja suficiente para ingressar no TOP10, o programa irá avisar o utilizador e regressar ao menu principal.

Uma imagem com texto

Descrição gerada automaticamente

# Funções Utilizadas

## ( Mais Relevantes )

**Main** - permite a execução do jogo;

**IMP\_FICH** - imprime o que está num ficheiro de texto;

**APAGA\_ECRAN** - limpa o ecrã;

**LER\_TEMPO** - guarda as horas / minutos / segundos em variáveis;

**PRINTDIGIT** - imprime um número de 16bits para a posição do cursor através de uma série de divisões por 10 e conversões de número para carácter. Coloca um espaço no final do número;

**PRINTDIGITPLUS** - imprime o número para uma string com o mesmo algoritmo da PRINTDIGIT;

**NPLAYER** - lê e grava um nome de jogador fornecido

**LE\_TECLA** - interpreta os inputs do teclado do computador para a movimentação do avatar e construção do labirinto, inicia também o timer, data atual e horas do computador;

**GAME\_OVER** - display do ecrã de derrota;

**ADICIONAR\_TOP10** - verifica se a pontuação obtida pelo jogador é suficiente para entrar no top 10, se não, avisa o jogador que a pontuação foi suficiente, se sim, substitui o primeiro valor menor que encontra;

**END\_GAME** - termina o programa.

# Funcionalidades Implementadas

|  |  |
| --- | --- |
| Módulo 1 | Implementado |
| Módulo 2 | Não implementado |
| Módulo 3 | Não totalmente implementado, a navegação na grelha encontra-se a funcionar, porém não está a ser feita a leitura das palavras |
| Módulo 4 | Segundos despendidos desde o início de jogo – Implementado  Número de palavras encontradas – Não implementado  Número de tentativas falhadas (letras selecionadas que não pertencem a nenhuma das palavras a identificar) – Não implementado  Top10 – Não totalmente implementado  Leitura do nome do jogador - Não totalmente implementado |
| Dois níveis de dificuldade | Implementado |
| Gestão do repositório de palavras | Não implementado |
| Identificação de palavra de forma diferente | Não implementado |