

# FALLING WORDS



Prof. Nuno Paulino

Prof. Pedro Ferreira Souto

Prof. Pedro Silva

**TURMA 5 - GRUPO 9**

Ricardo Carvalho, up201806791

Tiago Gomes, up201806658

# ÍNDICE

**1. INSTRUÇÕES DE UTILIZAÇÃO**

**2. ESTADO DO PROJETO**

**3. ESTRUTURA/ORGANIZAÇÃO DO CÓDIGO**

**4. DETALHES DE IMPLEMENTAÇÃO**

**5. CONCLUSÕES**

# INSTRUÇÕES DE UTILIZAÇÃO

## 1.JOGO

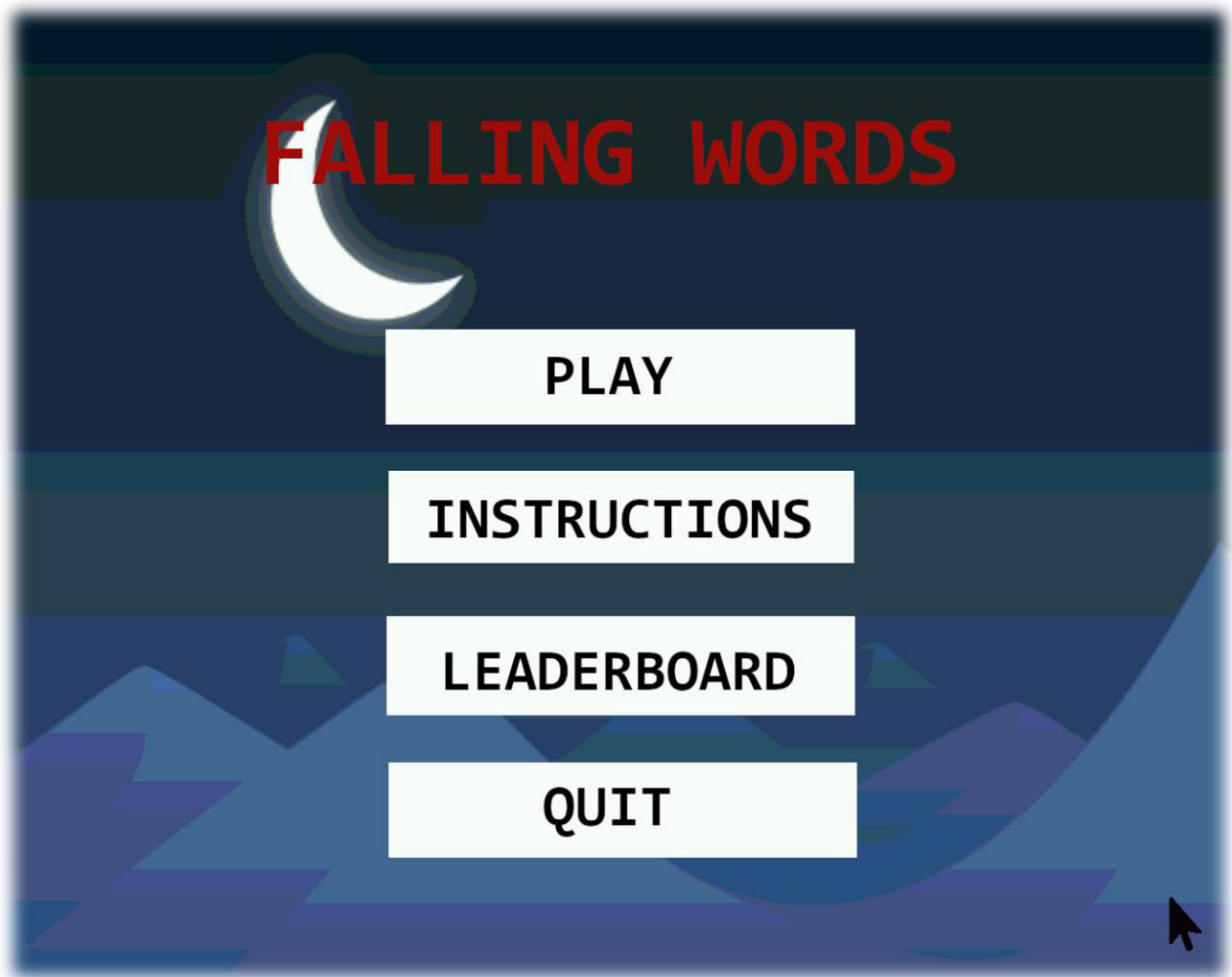
O jogo *Falling Words* é um jogo do estilo *type racer*, em que o objetivo do jogo é escrever palavras que vão caindo ao longo da tela, no menor tempo possível. À medida que o tempo vai passando, a dificuldade aumenta também, fazendo com que a velocidade a que as palavras caem se torne cada vez maior. Cada palavra digitada corretamente aumenta o score, representado pelo troféu no canto superior direito da imagem. O jogo termina quando uma palavra tiver colidido com a margem inferior da tela, aparecendo o *score* do jogador (se for um *score* recorde, será pedido o nome ao utilizador, para posteriormente ser colocado na *leaderboard*).



# INSTRUÇÕES DE UTILIZAÇÃO

## 2.MENU

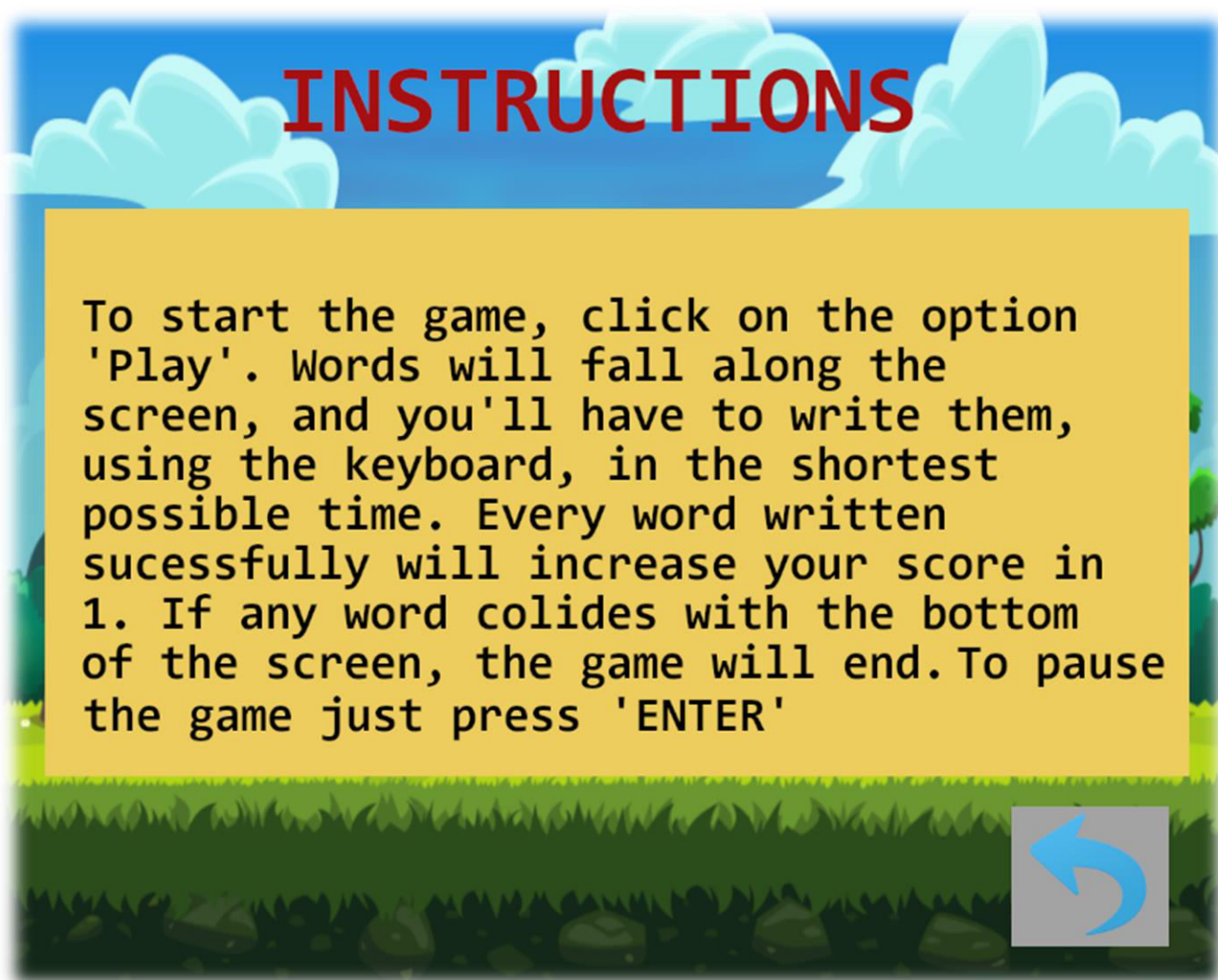
O menu do jogo é constituído por 4 opções: jogar, ver as instruções, ver a *leaderboard* e sair.



# INSTRUÇÕES DE UTILIZAÇÃO

## 3.INSTRUÇÕES

Nas instruções, é apresentada informação de como se joga.



# INSTRUÇÕES DE UTILIZAÇÃO

## 4. LEADERBOARD

Na *leaderboard*, é apresentado um quadro com 3 lugares, correspondentes ao primeiro, segundo e terceiro lugar. Cada lugar tem a sua medalha correspondente, além do nome e do score associado.



LEADERBOARD		
POS.	NAME	SCORE
	<b>TIAGO</b>	<b>14</b>
	<b>RICARDO</b>	<b>13</b>
	<b>LCOM</b> 	<b>12</b>



# ESTADO DO PROJETO

Dispositivos	Funcionalidade	Interrupções
Teclado	Digitar as palavras e o nome, caso haja pontuação recorde	Sim
Rato	Escolher as opções no menu e para clicar no botão voltar	Sim
Timer	Usado para esperar um certo tempo para desenhar as imagens no ecrã	Sim
Placa Gráfica	Desenho de imagens	Não
Real Time Clock(RTC)	Usado para meter o fundo de dia ou de noite, consoante a hora atual e para medir o tempo de jogo	Não

## 1.TECLADO

O teclado é usado somente no jogo, para digitar as palavras que vão caindo pelo ecrã. Pode ser usado também para pôr o jogo em pausa, tendo para isso o utilizador que clicar na tecla *Enter*. É usado também para escrever o nome, caso a pontuação seja uma pontuação recorde.

Usamos funções conhecidas do 'Lab 3' que tratam das interrupções, nomeadamente *keyboard\_subscribe\_int()*, *keyboard\_unsubscribe\_int()* e *kbc\_ih()*.

Para pôr o jogo em pausa, usamos a função *pauseGameHandlerKeyboard()*, que verifica se o scancode é equivalente à de uma tecla 'Enter'.

Para escrever o nome na secção 'Novo Recorde', usamos a função *askNameIfRecordHandlerKeyboard()*, que converte os scancodes recebidos do teclado para caracteres.

A função *gameWordsHandlerKeyboard()*, é a responsável por atualizar a palavra de acordo com o scancode recebido.

## 2.RATO

O rato é usado para navegar pelas opções do menu, tendo de se clicar no botão esquerdo do rato na caixa da opção pretendida. Pode ser usado também para clicar no botão com a seta azul, para voltar ao menu inicial, e no botão com o visto verde, para submeter o nome na secção de 'Novo Recorde'.

Para além das funções que tratam das interrupções, entre elas, *mouse\_\_subscribe\_int()*, *mouse\_unsubscribe\_int()* e *mouse\_ih()*, temos



também as funções que tratam do *data reporting*, *our\_mouse\_enable\_data\_reporting()* e *mouse\_reset\_state()*, e uma função responsável pela leitura dos pacotes do rato, *packet\_parsing()*.

A função *mainMenuHandlerMouse()* associa-se ao menu principal e verifica se o utilizador clicou no botão esquerdo do rato e averigua se esse clique foi dado dentro dos limites de cada opção.

A função *leaderboardHandlerMouse()* é usada para saber se, estando o utilizador na aba 'Leaderboard' o utilizador clicou com o botão esquerdo do rato dentro dos limites do botão voltar (representado por uma seta azul).

A função *askNameHandlerMouse()* é usada para saber se o utilizador, ao estar na secção 'Novo Recorde' clicou no botão confirmar, para submeter o seu nome para a *Leaderboard*.

A função *gameOverHandlerMouse()* é usada para saber se o utilizador, ao estar na secção de *GameOver*, clicou com o botão esquerdo numa de 3 opções: jogar outra vez, voltar ao menu principal ou sair do jogo.

### 3.TIMER

Nós usamos o *timer* para desenhar todos os sprites, passado um certo número de interrupções do *timer*. É responsável pela maioria das coisas

presentes a nível visual no jogo, tais como a pontuação, nível de dificuldade, *leaderboard*...

Temos 3 funções responsáveis pelas interrupções: *timer\_subscribe\_int()*, *timer\_unsubscribe\_int()* e *timer\_int\_handler()*.

## 4. PLACA GRÁFICA

A placa gráfica é responsável por desenhar tudo o que o utilizador vê no ecrã. Usamos a função *vg\_init()* que inicializa o modo gráfico com as descrições apresentadas a seguir: o modo gráfico usado foi o 11A, com uma resolução de 1280x1024, com a cor representada no modo direto. Este modo possui 65536 cores diferentes.

Usamos *double buffering* no nosso projeto, o que tornou o jogo significativamente mais rápido e fluido.

Temos uma função que desenha as palavras com a cor desejada *displayWordWithColor()*.

As letras que caem ao longo do ecrã, à medida que vão sendo digitadas as letras correspondentes, a palavra atualiza-se: por exemplo, se a palavra for 'carro' e escrevermos 'c' no teclado, a palavra ficará 'arro' e assim por diante, até desaparecer por completo e aparecer uma nova.

Usamos a nossa variável *y0\_word*, que representa a coordenada y da palavra que está a cair, para verificar se esta colidiu com o fundo do ecrã.

## 5.RTC

O RTC(Real Time Clock) é usado para ler a hora no momento em que um utilizador está a jogar, e para contar o tempo de jogo. Se a hora corresponder a um período noturno, o jogo aparece com um fundo alusivo à noite. Se for diurno, o fundo corresponde ao dia. A função responsável por isto chama-se *read\_time()*.

# ORGANIZAÇÃO DO CÓDIGO

## 1.game

No *game* estão presentes todas as funções ligadas diretamente ao jogo, como por exemplo, funções que determinam o nível de dificuldade, funções que pedem o nome ao utilizador caso haja pontuação recorde ou funções responsáveis por pôr o jogo em pausa.

**Percentagem: (30%/70%) (Ricardo/Tiago)**

**Percentagem no trabalho: 18%**

## 2.RTC

No RTC encontram-se funções responsáveis por ler a hora e os segundos atuais, e desse modo perceber se é de dia ou de noite.

**Percentagem: (30%/70%) (Ricardo/Tiago)**

**Percentagem no trabalho: 2%**

## 3.mouse

No *mouse* encontram-se funções relacionadas com o rato: funções que ativam e desativam as interrupções, que leem pacotes vindos do rato e que ativam ou desativam o *data reporting*.

**Percentagem: (50%/50%) (Ricardo/Tiago)**

**Percentagem no trabalho: 4%**

## 4.keyboard

No *keyboard* está toda a informação relacionada com o teclado, o periférico mais importante no nosso jogo. Lá, estão presentes funções

responsáveis por ativar e desativar interrupções vindas do teclado, bem como funções que lidam com essas mesmas interrupções (*kbc\_ih()*).

**Percentagem: (50%/50%) (Ricardo/Tiago)**

**Percentagem no trabalho: 2%**

## **5. Leaderboard**

A *leaderboard* contém tudo o que está relacionada com parte dos recordes e *high scores*. A função *read\_leaderboard\_file()* tem como objetivo ler o ficheiro de texto “Leaderboard.txt” e pôr os nomes que lá estão no quadro presente na secção da *leaderboard*. Quando fechamos o jogo, ele atualiza o ficheiro de texto, escrevendo os nomes presentes no quadro da *leaderboard* no ficheiro de texto, por meio da função *write\_leaderboard\_file()*.

**Percentagem: (30%/70%) (Ricardo/Tiago)**

**Percentagem no trabalho: 10%**

## **6.proj**

O *proj* é o módulo mais importante do projeto, pois nele está a base de todo o jogo. Contém um ciclo que mantém o jogo a correr, acedendo a opções escolhidas pelo utilizador. O ciclo termina, bem como o jogo,

assim que o utilizador pressione a tecla 'ESC' ou clique na opção 'QUIT' do menu principal.

**Percentagem: (35%/65%) (Ricardo/Tiago)**

**Percentagem no trabalho: 10%**

## **7.timer**

No *timer* encontram-se funções responsáveis pela ativação e desativação das interrupções provenientes do timer, bem como outras funções relacionadas diretamente com este dispositivo.

**Percentagem: (50%/50%) (Ricardo/Tiago)**

**Percentagem no trabalho: 1%**

## **8.scancodes**

No módulo *scancodes* estão presentes funções relacionadas com *scancodes* provenientes do teclado, entre elas, funções que convertem *scancodes* em *chars* e que verificam se um *scancode* é válido para assumir como uma letra para uma palavra em jogo.

**Percentagem: (50%/50%) (Ricardo/Tiago)**

**Percentagem no trabalho: 1%**

## **9.sprite**

Neste módulo estão presentes as funções relacionadas com a criação de *sprites*, o desenho destes e ainda funções responsáveis por dar cor aos *sprites*.

**Percentagem: (30%/70%) (Ricardo/Tiago)**

**Percentagem no trabalho: 3%**

## **10.words**

Como o nome indica, é neste módulo que estão presentes todas as funções relacionadas com as palavras que caem durante o jogo. Dentre elas, funções que tratam do aparecimento de uma palavra no ecrã, funções que tratam de atualizar a palavra caso o utilizador insira a letra correta e funções que geram uma palavra aleatória proveniente do ficheiro “words.txt”.

**Percentagem: (30%/70%) (Ricardo/Tiago)**

**Percentagem no trabalho: 30%**

## **11.video\_card**

No *video\_card* estão presentes funções que iniciam o modo gráfico e funções que desenham várias coisas presentes no jogo, como o fundo e o cursor do rato.

**Percentagem: (10%/90%) (Ricardo/Tiago)**

**Percentagem no trabalho: 18%**

## **12.utils**

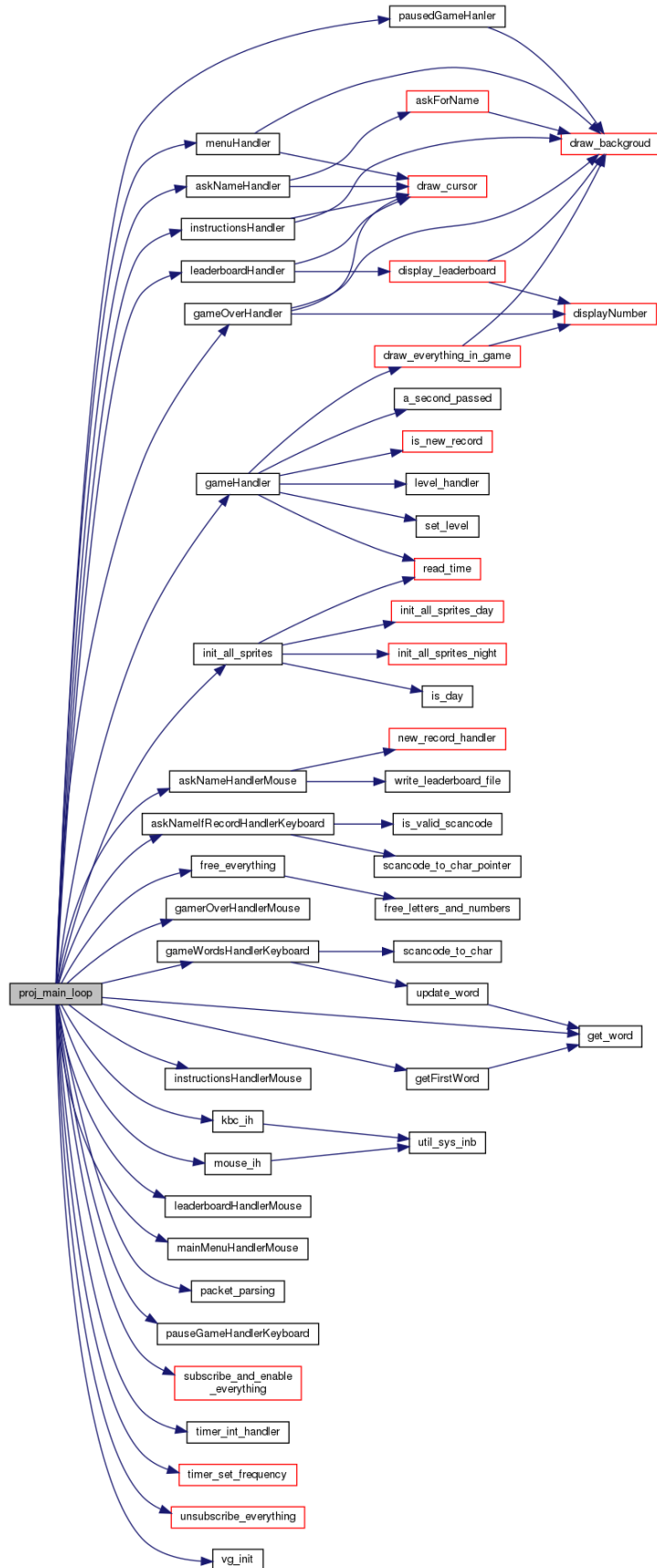
No módulo *utils*, proveniente do lab2, estão funções mais gerais, mas bastante importantes, como `util_sys_inb()`.

**Percentagem: (50%/50%) (Ricardo/Tiago)**

**Percentagem no trabalho: 1%**

**A seguir, mostramos o `call_graph` da função `proj_main_loop`:**





## DETALHES DA IMPLEMENTAÇÃO

Tudo o que foi planeado fazer por nós no início do projeto, foi feito. Apesar de ainda termos pensado em implementar a *serialPort*, decidimos que o nosso jogo não era o ideal para isso.

Sem dúvida que o módulo mais difícil para nós de implementar foi o *words*, visto que tivemos de pegar nos sprites de cada letra, ver o comprimento de cada uma e solucionar uma fórmula que pegasse nesse comprimento e fizesse o display dessa mesma palavra no ecrã. Outro módulo que nos causou dificuldades foi o do *video\_card*, porque é um dispositivo com o qual tivemos de trabalhar bastante neste projeto, e por isso tínhamos de saber trabalhar bem com ele.

No início tudo começou por ser difícil, pois não sabíamos por onde começar, mas com muita insistência e persistência da nossa parte, muito tempo perdido a encontrar *bugs* e erros que iam aparecendo, fomos melhorando o projeto a cada dia.

A ferramenta usada para a edição de imagens foi o GIMP, e foi com esta mesma ferramenta que convertíamos as imagens para o formato “.xpm”.

A implementação do *double buffering* trouxe melhorias significativas ao nosso projeto, tornando-o mais rápido, e com menos *lag*.

# CONCLUSÕES

Concluindo, achamos que a unidade curricular ‘Laboratórios de computadores’ ajudou-nos e ensinou-nos imenso, quer seja a programar em C, uma linguagem desconhecida por nós, apesar de termos conhecimentos de C++, mas ajudou-nos sobretudo a trabalhar com periféricos e a pesquisar informação.

Como aspetos negativos, temos a apontar a falta de organização da página da unidade curricular. Os tópicos estão extremamente desorganizados, há assuntos que estão nos slides, mas há uns que estão nos guiões, e perdíamos muito tempo para procurar alguma informação que quiséssemos.

A relação entre os créditos da unidade curricular e o esforço que ela exige é também, a nosso ver, algo que de todo não está correto. Esta unidade curricular, comparada com a unidade curricular ‘Algoritmos e Estrutura de Dados’, que tem 7,5 créditos, ao passo que ‘Laboratórios de Computadores’ tem 6, é algo injusto, visto que o esforço requerido nesta unidade curricular é muito superior ao da unidade curricular de ‘Algoritmos e Estrutura de Dados’.

Como aspetos positivos, temos a salientar a disponibilidade de toda a equipa docente no esclarecimento de dúvidas, por via e-mail, e, apesar da desorganização da informação, tanto nos guiões como nos slides, a informação lá presente era muito útil e ia sempre ao encontro do que pretendíamos.