



Universidade da Madeira

## ***PizzO - Pizzaria Online***

Componente de avaliação P2 (20%) de Arquitetura de Computadores

Ano letivo: 2017/2018

Data de entrega: 06-05-2018

Data de discussão: 09-05-2018

### **1. Descrição do trabalho**

#### **1.1. Especificação do problema**

Neste projeto os alunos deverão desenvolver um programa que permita o controlo dos pedidos de comida *online* a uma pizzaria que faz entregas. A interface gráfica (ou *display*) deverá simular a página *web* onde os clientes efetuam os seus pedidos *online*. O *display* (periférico de saída) tem dimensões 7x16 (7 linhas de 16 caracteres – *bytes*). O processador a utilizar será o PEPE, conforme ilustrado no Anexo I.

Os clientes deverão estar registados para fazerem o pedido. Deve então ser possível, no caso de um cliente não estar registado, criar o registo antes de efetuar um pedido. Nesse registo é inserido pelo utilizador um *Username* e *Password*, que ficarão armazenados na base de dados da aplicação se o registo for válido. Não deve ser possível fazer registo se algum dos campos não estiver preenchido. A *Password* deve ter pelo menos 3 caracteres. No ecrã deve ser mostrado o código ASCII correspondente ao *Username* (por ex., se o utilizador inserir num dos caracteres o valor hexadecimal 41<sub>16</sub>, no ecrã deverá ser mostrado o carácter "A"). Aquando da inserção da *Password*, no ecrã deve ser mostrado o carácter "\*" como forma de privacidade nos caracteres preenchidos pelo utilizador. No caso do *login* estar incorreto ou do registo não ser válido, o utilizador deverá ser informado.

Depois de validado o *login*, ou após ser efetuado um registo, o utilizador deve poder escolher entre realizar um pedido ou fazer *logout* do sistema.

Aquando da escolha de realização de pedido, deverão estar disponíveis para escolha cinco pizzas diferentes, sendo que cada variedade tem disponível dois tamanhos, pequeno e grande. As pizzas pequenas têm um custo de 5€ e as grandes de 8€.

Por cada 50€ gastos por um cliente, este tem direito a um desconto de 50% na pizza de menor valor na encomenda seguinte (ou atual, caso ela faça ultrapassar um total acumulado de 50€ gastos). Por exemplo, assumo que um cliente tem no seu histórico de encomendas um valor total de 45€, e a encomenda que está a efetuar no momento faz com que o valor total no histórico seja igual ou superior a 50€. Neste caso, esta encomenda terá um desconto de 50% na pizza de menor valor.

Quando o pedido é finalizado, deve ser mostrada ao utilizador a informação do desconto concedido e do valor final do pedido.

### 1.2. Interface com o utilizador

O interface com o utilizador será feito através de um *display* (periférico de saída), de dimensões 7x16 (7 linhas de 16 caracteres – *bytes*). Inicialmente é apresentado no *display* o menu principal, que fornece as opções de *login* ou registo de um novo cliente. A interação do utilizador com o sistema é feita através dos periféricos de entrada.

#### *Periféricos de entrada*

- *Username*
- *Password*
- Botão OK;
- Botão NR\_SEL;

O periférico *Username* e *Password* são utilizados para *login* e registo de clientes. Cada um deles consiste de 8 caracteres.

O botão OK é utilizado para validar a opção numérica escolhida (feita através da introdução do número pretendido no periférico NR\_SEL), nos menus que permitem escolha de várias opções. Assim, em cada menu devem ser listadas as várias opções para depois serem validadas com este botão.

Neste projeto espera-se que o aluno seja capaz de desenvolver uma solução personalizada e enquadrada no tema apresentado. **O projeto deverá estar a funcionar no modo *Run do simulador*.**

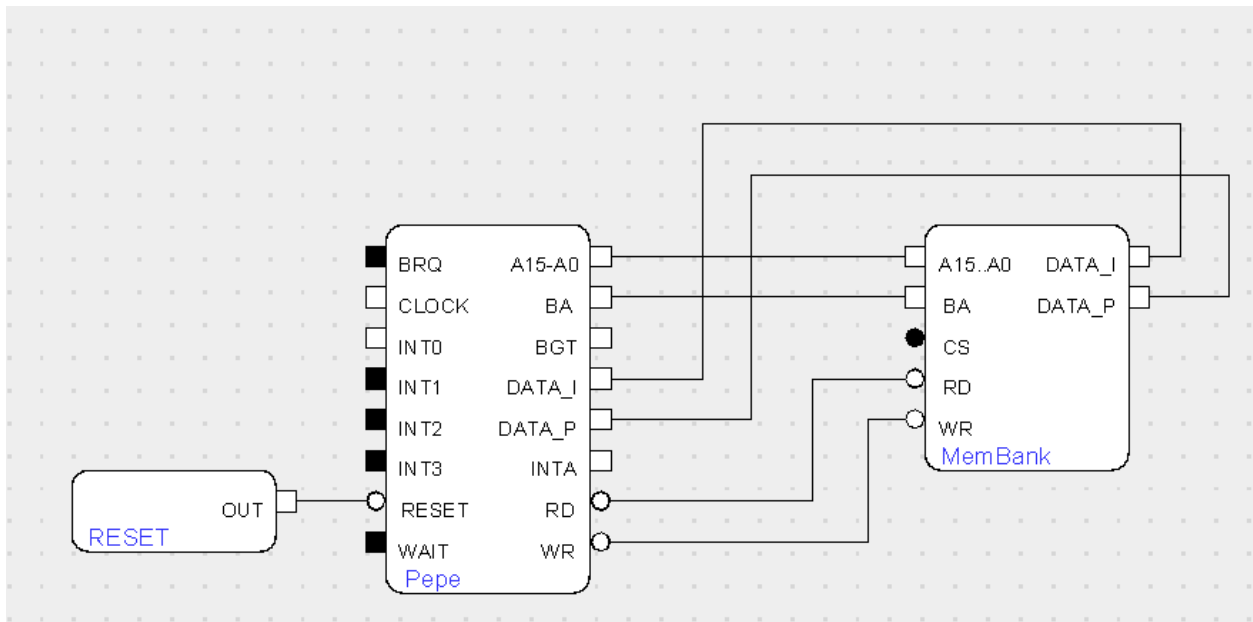
## 2. Plano de trabalhos

No segundo trabalho prático de avaliação da disciplina de Arquitetura de Computadores serão tidos em consideração os seguintes itens:

- Especificação e elaboração dos fluxogramas;
  - Programação em linguagem *assembly* e implementação;
  - Implementação de funções extra;
  - Elaboração do relatório.
- 
- Especificação e fluxogramas
    - Breve estudo de requisitos para a solução de *software*;
    - Desenho dos principais fluxogramas do controlo e das rotinas secundárias.
  - Programação em *assembly* e implementação
    - Programação, em linguagem *assembly* do algoritmo;
    - Verificação experimental do programa.
  - Relatório do trabalho
    - Num máximo de 5 páginas;
    - Capa com a identificação da disciplina, dos docentes e dos alunos;
    - Descrição do projeto e análise de resultados;
    - No anexo A do relatório apresentar os fluxogramas;
    - No anexo B o código em linguagem *assembly*, comentado e organizado.

## 3. Informações importantes

- O segundo trabalho prático de avaliação (relatório e prova oral) tem uma percentagem de 20% na avaliação da disciplina, e a nota mínima é de 8 valores;
- O trabalho prático de avaliação é individual ou em grupo de dois alunos;
- Código que não esteja comentado poderá levar à reprovação do trabalho devido às penalizações;
- O relatório em PDF, o ficheiro com o programa em *assembly*, e o simulador usado deverão ser compactados num único ficheiro ZIP/RAR, que deverá ser enviado para o Gabinete de Apoio ao Estudante (trabalhos@uma.pt) até às 24:00 do dia 06-05-2018. No *e-mail* devem indicar: o vosso nome e número de aluno, o nome da disciplina, a identificação do trabalho e o nome dos docentes;
- No dia da discussão do trabalho, através de prova oral (09-05-2018), cada aluno tem de mostrar o trabalho a funcionar e, por isso, pede-se que levem o seu computador pessoal para analisar a simulação;

**Anexo I - Processador PEPE e ligações à memória de dados**

**Anexo II – Código ASCII**

Bin	Oct	Dec	Hex	Sinal	Bin	Oct	Dec	Hex	Sinal	Bin	Oct	Dec	Hex	Sinal
0010 0000	040	32	20	(espaço)	0100 0000	100	64	40	@	0110 0000	140	96	60	`
0010 0001	041	33	21	!	0100 0001	101	65	41	A	0110 0001	141	97	61	a
0010 0010	042	34	22	"	0100 0010	102	66	42	B	0110 0010	142	98	62	b
0010 0011	043	35	23	#	0100 0011	103	67	43	C	0110 0011	143	99	63	c
0010 0100	044	36	24	\$	0100 0100	104	68	44	D	0110 0100	144	100	64	d
0010 0101	045	37	25	%	0100 0101	105	69	45	E	0110 0101	145	101	65	e
0010 0110	046	38	26	&	0100 0110	106	70	46	F	0110 0110	146	102	66	f
0010 0111	047	39	27	'	0100 0111	107	71	47	G	0110 0111	147	103	67	g
0010 1000	050	40	28	(	0100 1000	110	72	48	H	0110 1000	150	104	68	h
0010 1001	051	41	29	)	0100 1001	111	73	49	I	0110 1001	151	105	69	i
0010 1010	052	42	2A	*	0100 1010	112	74	4A	J	0110 1010	152	106	6A	j
0010 1011	053	43	2B	+	0100 1011	113	75	4B	K	0110 1011	153	107	6B	k
0010 1100	054	44	2C	,	0100 1100	114	76	4C	L	0110 1100	154	108	6C	l
0010 1101	055	45	2D	-	0100 1101	115	77	4D	M	0110 1101	155	109	6D	m
0010 1110	056	46	2E	.	0100 1110	116	78	4E	N	0110 1110	156	110	6E	n
0010 1111	057	47	2F	/	0100 1111	117	79	4F	O	0110 1111	157	111	6F	o
0011 0000	060	48	30	0	0101 0000	120	80	50	P	0111 0000	160	112	70	p
0011 0001	061	49	31	1	0101 0001	121	81	51	Q	0111 0001	161	113	71	q
0011 0010	062	50	32	2	0101 0010	122	82	52	R	0111 0010	162	114	72	r
0011 0011	063	51	33	3	0101 0011	123	83	53	S	0111 0011	163	115	73	s
0011 0100	064	52	34	4	0101 0100	124	84	54	T	0111 0100	164	116	74	t
0011 0101	065	53	35	5	0101 0101	125	85	55	U	0111 0101	165	117	75	u
0011 0110	066	54	36	6	0101 0110	126	86	56	V	0111 0110	166	118	76	v
0011 0111	067	55	37	7	0101 0111	127	87	57	W	0111 0111	167	119	77	w
0011 1000	070	56	38	8	0101 1000	130	88	58	X	0111 1000	170	120	78	x
0011 1001	071	57	39	9	0101 1001	131	89	59	Y	0111 1001	171	121	79	y
0011 1010	072	58	3A	:	0101 1010	132	90	5A	Z	0111 1010	172	122	7A	z
0011 1011	073	59	3B	;	0101 1011	133	91	5B	[	0111 1011	173	123	7B	{
0011 1100	074	60	3C	<	0101 1100	134	92	5C	\	0111 1100	174	124	7C	
0011 1101	075	61	3D	=	0101 1101	135	93	5D	]	0111 1101	175	125	7D	}
0011 1110	076	62	3E	>	0101 1110	136	94	5E	^	0111 1110	176	126	7E	~
0011 1111	077	63	3F	?	0101 1111	137	95	5F	_					