

Projeto de Arquitetura de Computadores

UAlg ▪ ISE ▪ LESTI

Vitrine Natalícia/Ano Novo

2024/2025

1 – Objetivos

Este projeto pretende exercitar os fundamentos da UC de Arquitetura de Computadores, nomeadamente a programação em linguagem *Assembly* com interação entre CPU e periféricos. O objetivo do projeto consiste em aplicar estes conceitos para criar uma “**Vitrine Natalícia**”, para ser colocada na montra de uma loja (real ou imaginária, à sua escolha). Em alternativa à Vitrine Natalícia, também pode escolher desenvolver uma “**Vitrine de Ano Novo**.” O projeto é obrigatoriamente desenvolvido no **Simulador SIMAC**, na **versão 2.3.0 (Out 2024)**, disponível na Tutoria.

2 – Descrição genérica da vitrine

A vitrine festiva consiste, genericamente, num ecrã multimédia (*MediaCenter*), que irá apresentar elementos decorativos de Natal e/ou Ano Novo, em *Pixel Art*. Alguns dos elementos devem ser animados, podendo ter desde animações simples (e.g., piscar) até outras mais elaboradas, como movimento pelo ecrã. O objetivo da vitrine é comemorar a época natalícia e/ou de fim de ano com elementos gráficos apelativos a quem passa diante de uma loja. Seja criativo!

3 – Detalhes do Projeto

A tabela da página seguinte apresenta a descrição das funcionalidades esperadas para este projeto, através de **versões incrementais** (assim pode começar a desenvolver o seu projeto de uma forma simples, acrescentando complexidade por etapas concretas):

- Considere que cada versão posterior implementa **todas as funcionalidades das versões anteriores**;
- As diferentes versões têm periféricos mínimos que devem obrigatoriamente ser usados;
- As versões posteriores requerem estilos de programação mais elaborados para obter uma avaliação superior, nomeadamente a utilização de processos cooperativos, interrupções, rotinas, comentários adequados e uma boa organização e estruturação do código;
- É sugerida uma nota de avaliação indicativa da componente Material Prático (**MP**; ver secção “Avaliação do Projeto” abaixo) para cada projeto que cumpra os requisitos da versão incremental respetiva; no entanto, as avaliações não são “estanques,” pelo que será sempre valorizado o esforço do aluno em atingir objetivos mais avançados, mesmo quando não se cumpram todos os critérios de funcionalidade. Em suma, apresente sempre o melhor trabalho que conseguir fazer.

V.	Funcionalidade mínima	Periféricos mínimos	Estilo de Programação	~Aval. MP (0–20)
v0	<ul style="list-style-type: none"> – Apresenta <u>elementos decorativos</u> de Natal ou Ano Novo (2025) no ecrã do <i>MediaCenter</i>; – Tem <u>3</u> objetos decorativos diferentes; – Animação <u>simples</u> (e.g., pisca um ou mais objetos). 	<i>MediaCenter</i>	<i>Simples</i>	8
v1	<ul style="list-style-type: none"> – Os elementos decorativos relacionam-se para compor uma <u>cena integrada</u> (tipo “postal” de Natal/Ano Novo); – Usa uma <u>imagem de fundo</u> adequada; – Tem <u>4</u> objetos decorativos diferentes; – Animações <u>simples</u> e <u>complexas</u> (com movimento temporizado); – Permite o uso do <u>teclado</u> para <u>controlo de animações</u>. 	<ul style="list-style-type: none"> – <i>MediaCenter</i> – Teclado (16 teclas, 0–F) 	<i>Simples</i>	12
v2	<ul style="list-style-type: none"> – Alguns <u>efeitos sonoros</u> festivos associados às animações; – Teclado completamente funcional (<u>todas as teclas</u>); – Utilização de <u>pelo menos 6 teclas diferentes</u> para controlo de animações e/ou sons. 	<ul style="list-style-type: none"> – <i>MediaCenter</i> – Teclado (16 teclas, 0–F) 	<i>Avançada</i> <ul style="list-style-type: none"> – Adequada utilização de <u>rotinas</u>; – Código adequadamente comentado. 	14
v3	<ul style="list-style-type: none"> – <u>Duas cenas</u> festivas diferentes, com imagens ou <u>vídeos</u> de fundo apropriados; – Temporizações em <u>tempo-real</u> através de <u>interrupções</u> ao PEPE; – Uso do teclado para controlo de cenas; – Tem <u>6</u> objetos decorativos diferentes. 	<ul style="list-style-type: none"> – <i>MediaCenter</i> – Teclado (16 teclas, 0–F) 	<i>Avançada</i> <ul style="list-style-type: none"> – Adequada utilização de <u>rotinas</u> e <u>interrupções</u>; – Código bem organizado; – Código adequadamente comentado. 	16
v4	<ul style="list-style-type: none"> – <u>Três cenas</u> festivas diferentes, com imagens ou vídeos de fundo apropriados; – Execução baseada em <u>processos cooperativos</u>; – Uso avançado do teclado para controlo geral do programa; – <u>Displays</u> para: (a) apresentar o <u>número da cena atual</u> (0, 1 ou 2) e (b) apresentar a <u>tecla pre-mida</u> no teclado; – Tem <u>10</u> objetos decorativos diferentes; – Pelo menos uma animação complexa com <u>movimento em 2 eixos</u> (horizontal + vertical). 	<ul style="list-style-type: none"> – <i>MediaCenter</i> – Teclado (16 teclas, 0–F) – Displays de 7 segmentos 	<i>Avançada</i> <ul style="list-style-type: none"> – Adequada utilização de <u>rotinas</u>, <u>interrupções</u> e <u>processos</u>; – Código muito bem organizado; – Código muito bem comentado. 	20

4 – Avaliação do Projeto

O projeto é avaliado em **3 componentes**, numa escala de [0, 20] valores:

- Material prático (**MP**; e.g., código-fonte, circuitos, diagramas);
- Apresentação oral (**AO**);
- Relatório técnico (**RT**) simples, até 6 páginas (modelo em anexo).

A nota final (**NF**) do projeto corresponderá à **nota mais baixa (mínima)** entre as obtidas na avaliação das componentes **MP**, **AO** e **RT**:

$$NF = \min(MP, AO, RT).$$

ATENÇÃO

Cópias, plágios ou incapacidade em explicar minimamente o material produzido resultam numa classificação de **NF = 0 (zero) valores**, sem possibilidade de melhoria.

A avaliação do projeto terá em consideração as seguintes linhas orientadoras:

Material prático (MP): o aluno deverá submeter na Tutoria o material prático do projeto, que será avaliado segundo os seguintes critérios:

- **Funcionalidade** (~40%): deve funcionar de acordo com as especificações; inclui a implementação correta dos algoritmos, a execução sem erros e a apresentação adequada dos resultados.
- **Organização** (~30%): deve estar bem estruturado e documentado, com registos, variáveis, rotinas e processos bem nomeados e comentários adequados.
- **Eficiência** (~20%): deve ser eficiente e otimizado, evitando redundâncias e uso desnecessário de recursos.
- **Originalidade e criatividade** (~10%): para além da criatividade já inerente a este projeto, o aluno deve ser capaz de ir além das especificações funcionais e mostrar um pensamento inovador.

Apresentação oral (AO): o aluno deve apresentar o material produzido e explicar os raciocínios de resolução das várias etapas do projeto, de forma cuidada, ordenada, clara e concisa:

- São valorizadas as apresentações organizadas e bem estruturadas, com uma introdução, desenvolvimento e conclusão, e com uma linguagem técnica adequada.
- São penalizadas as apresentações desorganizadas, com uma linguagem demasiado informal, ou com uma apresentação visual desadequada.
- Deverá demonstrar que o material prático funciona corretamente, de preferência com exemplos/testes já preparados para serem executados.

Relatório Técnico (RT): o aluno deverá submeter na Tutoria o relatório técnico do projeto, que será avaliado, na sua globalidade, em linha com as seguintes ponderações:

Critério	%
Clareza e Qualidade do Manual de Utilizador	~25%
Abrangência e Profundidade dos Comentários	~35%
Análise Crítica (pontos não alcançados, méritos e melhorias)	~20%
Organização e Estrutura do Relatório	~10%
Qualidade da Escrita e Formatação	~5%
Originalidade e Criatividade	~5%

- **Clareza e Qualidade do Manual de Utilizador:** deve ser fácil de entender e usar, com instruções claras sobre o funcionamento do programa.
- **Abrangência e Profundidade dos Comentários:** os comentários devem ser relevantes, refletir uma compreensão profunda do projeto e abordar todas as áreas solicitadas.
- **Análise Crítica:** avalia o quão bem o aluno identificou as limitações do seu trabalho, as forças e as possíveis melhorias.
- **Organização e Estrutura do Relatório:** o relatório deve seguir a estrutura proposta, de forma lógica e coerente.
- **Qualidade da Escrita e Formatação:** a escrita deve ser clara, sem erros ortográficos ou gramaticais, e o relatório deve seguir as normas de formatação do modelo fornecido.
- **Originalidade e Criatividade:** o aluno apresentou soluções inovadoras ou perspectivas únicas no seu trabalho.

5 – Periféricos do PEPE

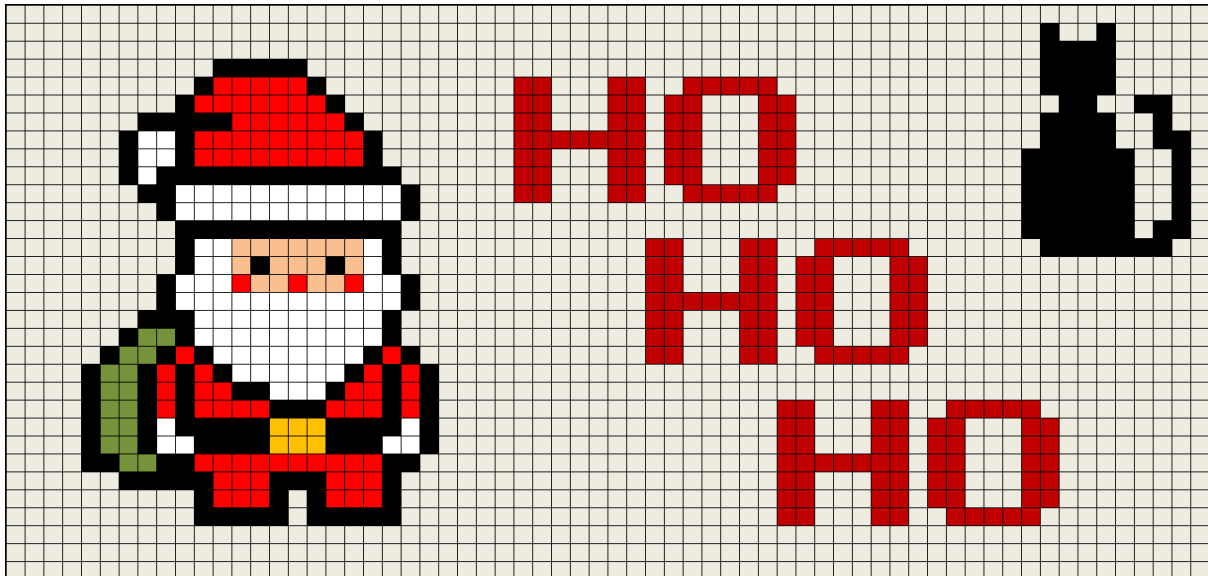
O projeto deverá usar, pelo menos, os seguintes periféricos (de acordo com a versão):

- Ecrã multimédia (*MediaCenter*) de 32×64 pixels (ARGB, de 4 bits/canal);
- Teclado (de 16 teclas) para controlo;
- Conjunto de displays de 7 segmentos.

5.1 – *MediaCenter*: ecrã, cenários, sons e objetos

O módulo *MediaCenter* do simulador possui variadas capacidades multimédia, permitindo definir imagens de fundo, reproduzir sons e vídeos, vários planos sobrepostos de imagens (construídas pixel a pixel, em *Pixel Art*), um cenário frontal para texto, etc. O **Guião Prático 4** fornece mais detalhes sobre este módulo.

- A imagem seguinte ilustra um possível ecrã do *MediaCenter* (que poderá ser posteriormente sobreposto a uma imagem de fundo).



- O ecrã da vitrine deverá ter, pelo menos, 32×64 pixels (linhas \times colunas). Esta é a resolução padrão do *MediaCenter*, mas poderá opcionalmente utilizar uma resolução superior.
- Cada pixel pode ter uma cor diferente, com 4 componentes (*Alpha*, *Red*, *Green* e *Blue*, ou **ARGB**), todas com 4 bits (valores sem sinal, de **0** a **F**). A primeira componente define a opacidade (**0** é totalmente transparente, **F** totalmente opaco). O pixel pode estar **ligado** (com a cor definida pelas 4 componentes) ou **desligado** (com tudo a **zero**, caso em que não se vê, pois fica transparente).
- Todos os elementos festivos (excepto as imagens/videos de fundo) são desenhados pixel a pixel.
- Atrás deste ecrã de pixels está a imagem (ou vídeo) de fundo, que tipicamente tem uma resolução bastante superior (embora deva ter um fator de forma – *aspect ratio* – semelhante, retangular 2×1 , para que não apareça de forma distorcida).
- É possível alterar a imagem/vídeo de fundo através do programa do PEPE. Por isso, espera-se que use uma diferente para cada cena criada (especialmente a partir da **v3** do projeto).
- Não é fornecida nenhuma imagem nem vídeo para cenários, mas existem inúmeras imagens adequadas que pode obter de forma gratuita na Web para uso pessoal. O design multimédia fica ao seu gosto.
- Também devem existir efeitos sonoros, que mais uma vez podem ser obtidos na Web. Ficheiros de som pequenos são o ideal. O módulo *MediaCenter* permite, no entanto, restringir os tempos de início e fim de um som (ou de um vídeo), caso haja necessidade.

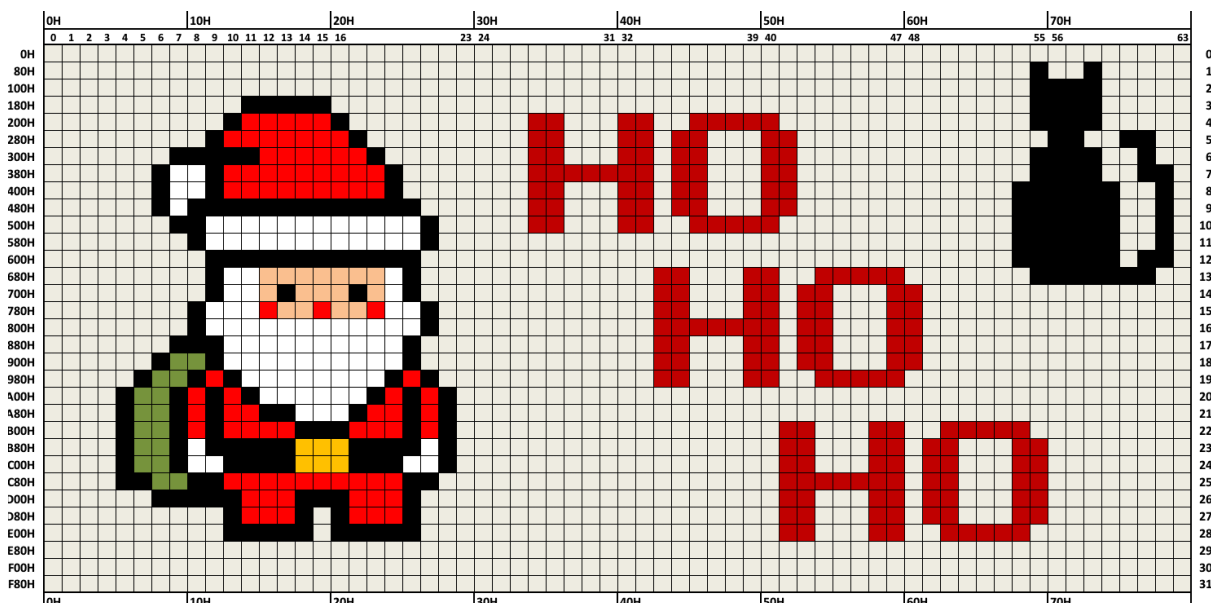
FUNDAMENTAL

Depois de definir os ficheiros de imagem, vídeo e som que quiser acrescentar ao módulo *MediaCenter*, deve salvá-los na mesma pasta onde tem o circuito (ficheiro **.cir**), ou numa sub-pasta. Esta é a única forma destes ficheiros serem incluídos no ficheiro de descrição do circuito de forma portátil. Caso contrário, é guardado o *path* absoluto do ficheiro, e depois o projeto não funciona noutro computador (nomeadamente, no do docente que vai avaliar o projeto!).

- Desenhar um objeto no ecrã (à frente do cenário de fundo) é desenhar os seus pixels, com **cor** definida ou **desligados** (transparentes), em posições (linha e coluna) adjacentes do ecrã, de acordo com a forma definida para esse objeto.
- Toma-se um dos pixels do objeto (canto superior esquerdo, por exemplo) como indicador da posição no ecrã desse objeto e todos os restantes pixels desse objeto são desenhados em posições relativas a esse pixel de referência.
- Para desenho neste estilo gráfico – **Pixel Art** – sugere-se que utilize um editor apropriado (e.g., <https://www.pixilart.com/>) ou um editor de **Icons** (e.g., <https://redketchup.io/icon-editor>, podendo em seguida converter uma imagem existente, ou “Add a New Image Entry”, no formato de 32×32 BMP). Em alternativa, também pode usar um editor normal de imagens (e.g., o *Paint* do Windows, com o zoom no máximo, a 1000%).
- Para facilitar os passos seguintes, na pasta do projeto encontra-se o ficheiro **img2argb.py**, em Python, que permite converter os pixels de uma imagem (à escolha) numa tabela de WORDS do PEPE. Para utilizá-lo, execute, na linha de comandos: **python img2argb.py <imagem>**. Por exemplo, **python img2argb.py santa.png**, que criará o ficheiro **santa-PEPE.txt** na pasta em que se encontrava a imagem original, como representado na figura seguinte.



- O ficheiro Excel **ecrã-32x64-vazio.xlsx**, que é fornecido, simula as quadrículas do ecrã e mostra que o ecrã é na realidade uma memória (note os endereços de cada linha no lado esquerdo), em que cada pixel é uma palavra (2 bytes) dessa memória, com a cor do pixel. Há 64 pixels, ou 128 bytes, por cada linha do ecrã (figura seguinte).

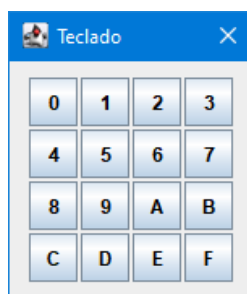


- Pode usar este ficheiro para planear o tamanho e forma dos seus objetos (mas recomenda-se que use um editor de *Pixel Art* configurado para uma imagem de dimensão 32×64), pois não têm de ter as formas e tamanhos mostrados na figura anterior, nem as mesmas cores (e cada objeto pode ter pixels de cores diferentes). A figura é apenas uma sugestão ilustrativa. Use a sua criatividade – aqui pode inventar!
- Cada pixel pode ser desenhado escrevendo diretamente na memória do ecrã ou por meio de um comando. Escolha o seu método (ver **Guião Prático 4**).
- Mover um objeto é: (a) apagá-lo da sua posição atual, (b) mudar a respetiva posição e (c) desenhá-lo na nova posição. O efeito visual é o objeto parecer mover-se.
- O módulo *MediaCenter* possibilita que cada objeto seja desenhado num ecrã de pixels diferente (todos os ecrãs de pixels são sobrepostos). Isto permite que os objetos que se movimentem na mesma direção se sobreponham de forma graciosa (como se fossem janelas diferentes num sistema operativo), em vez de um destruir o desenho do anterior.
- Para os objetos que nunca se sobrepõem, pode usar o mesmo ecrã de pixels. Terá de definir o número de ecrãs de pixels que quer usar no *MediaCenter*, em modo *Design* no seu painel de configuração.

O **Guião Prático 4** ensina a trabalhar com o módulo *MediaCenter*, em termos de pixels (e a desenhá-los), ecrãs de pixels, cenários e sons.

5.2 – Teclado e controlo da vitrine

A vitrine é controlada pelo utilizador por meio de um teclado, tal como o da figura seguinte, atuado por clique do rato nas “teclas”:



A atribuição de teclas a comandos é livre e ao seu gosto. Um exemplo de atribuição é o seguinte:

Tecla	Efeito
0, 1 e 2	Trocar entre cenas natalícias (0 ou 1) e de Ano Novo (2)
4 e 5	Mover o pai natal para a esquerda ou para a direita
6	Os pares de letras “HO HO HO” aparecem em sequência
7	Apaga do ecrã “HO HO HO”
B	Liga/desliga os efeitos sonoros da vitrine
C	Liga ou reinicia a vitrine
D	Suspende/continua as animações da vitrine
E	Desliga a vitrine (todos os pixels a preto)

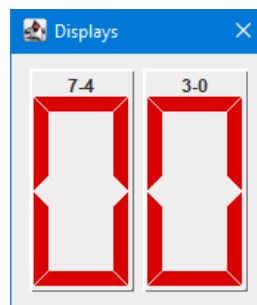
IMPORTANTE

Cada tecla deve executar apenas um comando. Para um novo comando, mesmo que igual, tem de se largar a tecla e carregar de novo.

O **Guião Prático 3** ensina a trabalhar com o teclado.

5.3 – Displays e cenas festivas

Existem dois displays de 7 segmentos, que servem para apresentar informação de controlo aos empregados da loja. Desta forma, tanto o teclado como os displays não precisam de estar colocados na montra da loja e podem estar colocados num ponto de acesso mais fácil, como ao pé de uma caixa registadora:



Estes poderão implementar a seguinte funcionalidade:

- O **primeiro display** (à esquerda), pode mostrar o número da cena festiva em exibição no ecrã do *MediaCenter* (e.g., 0, 1 ou 2).
- O **segundo display** (à direita), pode mostrar durante 2 segundos a tecla que foi premida no teclado, de forma que haja feedback das interações com o teclado.
 - O display da direita pode também apresentar outra informação contextualizada, por exemplo, se for implementada a funcionalidade de desligar a vitrine por teclado (e.g., tecla **E**), este display poderá apresentar a letra **E** em contínuo, enquanto estiver desligada.

O objetivo é usar os displays para facilitar o controlo da vitrine, apresentando informação relevante sobre o estado da mesma, pelo que fica ao seu critério pensar em funcionalidades adicionais que poderiam ser mostradas neste periférico.

6 – Entrega do projeto

O projeto terá de ser submetido na Tutoria da disciplina até às **06h00** do dia **13 de janeiro de 2024** (**segunda-feira**), através de um ficheiro (**grupo_A1_A2_A3.zip**, em que **A** é o número de aluno dos elementos do grupo, **por ordem crescente**), contendo os seguintes ficheiros:

- 1 – Um ficheiro **grupo_A1_A2_A3.pdf**, correspondendo ao relatório técnico simples de formato semilivre, mas com a seguinte informação (juntamente com este enunciado, é fornecido um modelo de relatório):
 - Identificação dos elementos do grupo, pelos números de aluno e nomes (1 a 3 pessoas/grupo);
 - Definições relevantes, se tiverem feito algo diferente do que o enunciado pede ou indica (teclas diferentes, funcionalidade a mais, etc.);
 - Indicação concreta das funcionalidades pedidas que o código enviado NÃO satisfaz;
 - Eventuais outros comentários ou conclusões.
- 2 – Um ficheiro **grupo_A1_A2_A3.asm** com o código, pronto para ser carregado no simulador e executado (deve ter o número do grupo, números de aluno e nomes). Deve criar uma cópia da versão mais recente do código, limpando eventual “lixo” e coisas temporárias, de modo a compilar e executar a funcionalidade pedida. A organização do código e comentários serão avaliados, como descrito anteriormente;

IMPORTANTE

Não se esqueça de identificar no ficheiro de código (**grupo_A1_A2_A3.asm**) em comentários, logo no início do programa, o número e nome dos alunos que participaram no seu desenvolvimento.

- 3 – Todos os ficheiros de imagem, vídeo e som usados no módulo *MediaCenter*;
- 4 – Um ficheiro **grupo_A1_A2_A3.cir** com o circuito do projeto, mas guardado depois de definir no módulo *MediaCenter* todos os ficheiros de imagem e som usados. Não se esqueça que estes ficheiros devem estar guardados no mesmo diretório do circuito, ou num subdiretório deste.

IMPORTANTE

O circuito do projeto encontra-se em **projeto-vitrine-natalicia.cir**. É semelhante ao circuito do **Guião Prático 5**. No entanto, se pretender implementar interrupções (**Guião Prático 6**) e processos (**Guião Prático 7**), utilize como base os circuitos dos guiões correspondentes.

7 – Apresentação do projeto

Todas as apresentações do projeto serão **presenciais e individuais**, e decorrerão, o mais tardar, nos dias úteis imediatamente seguintes à data limite para submissão do projeto, de **13 a 17 de janeiro de 2025**. Os horários disponíveis para apresentação serão comunicados atempadamente e funcionarão por marcação através de link próprio. Se não tiver disponibilidade para apresentação presencial durante este período, entregue o seu projeto mais cedo e solicite a apresentação do mesmo ao docente durante o período de uma Aula PL, ou por marcação regular para apresentação de TP. Não serão realizadas apresentações remotas (por teleconferência).

8 – Estratégia de implementação

Os Guiões Práticos estão alinhados com as funcionalidades parciais a atingir, em termos de desenvolvimento do projeto. Tente cumpri-los, de forma a garantir que o projeto estará concluído na data de entrega. No entanto, recomenda-se vivamente que sejam usadas rotinas/processos para suportar as diversas ações da vitrine, tornando o **Guião Prático 5** uma referência essencial.

Como ordem geral de implementação do projeto, recomenda-se a seguinte estratégia (pode naturalmente adotar outra):

1. Rotinas de ecrã, para desenhar/apagar:
 - um pixel numa dada linha e coluna (de 0 a 31 e 0 a 63, respetivamente);
 - um objeto/elemento gráfico genérico, descrito por uma tabela que inclua a sua largura, altura e a cor ARGB de cada um dos seus pixels (para facilitar este processo, ver o ficheiro de conversão **img2argb.py**, descrito acima). Use um desses pixels – por exemplo o canto superior esquerdo – como referência da posição do objeto (linha e coluna) e desenhe os restantes pixels relativamente às coordenadas desse pixel de referência;
2. Teclado (**v1–v2**);
3. Interrupções (**v3**) para temporizações em tempo real dos elementos multimédia;
4. Processos cooperativos e displays (**v4**).

IMPORTANTE:

- As rotinas de interrupção param o programa principal enquanto estiverem a executar. Por isso, devem apenas assinalar aos processos quando ocorrem, por meio de variáveis (LOCKS, no caso de usar processos cooperativos). O processamento do jogo deve ser feito pelos processos e não pelas rotinas de interrupção;
- Se usar valores negativos (por exemplo, –1 para somar à coluna de um objeto para ele se deslocar para a esquerda), as variáveis correspondentes devem ser de 16 bits (declaradas com WORD, e não BYTE).

Tenha ainda em consideração as seguintes recomendações:

- Faça PUSH e POP de todos os registos que use numa rotina e não constituam valores de saída (mas não sistematicamente de todos os registos, de R0 a R11!). É muito fácil de não reparar que um dado registo é alterado durante um CALL, causando erros que podem ser difíceis de depurar. Atenção ao emparelhamento dos PUSHs e POPs, bem como à ordem relativa;
- Vá testando todas as rotinas que fizer e quando as alterar. É muito mais difícil descobrir um bug num programa já complexo e ainda não testado;
- Estruture bem o programa, com zona de dados no início, quer de constantes, quer de variáveis, e rotinas auxiliares de implementação de cada processo junto a ele;
- Produza comentários abundantes, não se esquecendo de cabeçalhos para as rotinas com descrição, registos de entrada e de saída (veja exemplos nos guiões de laboratório);

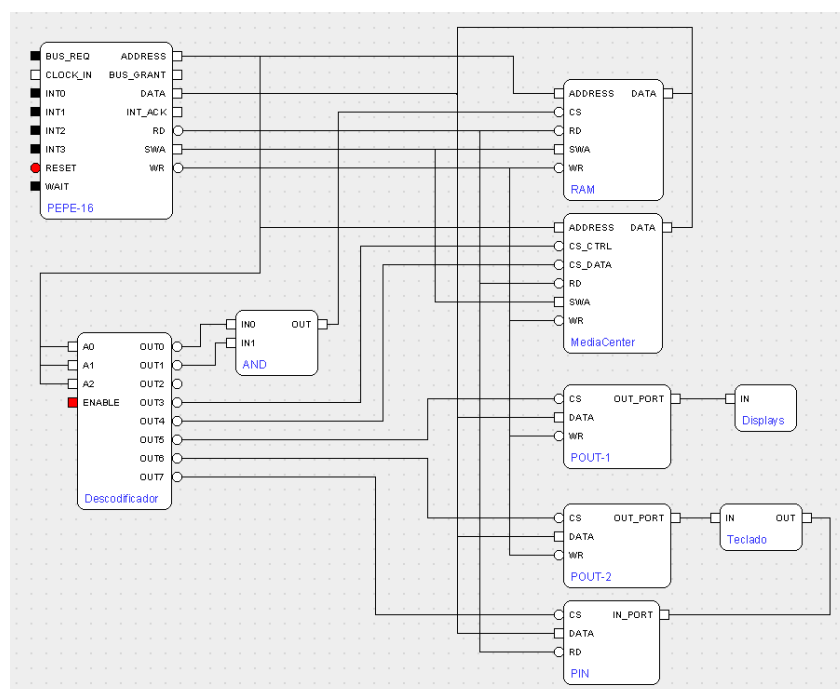
- Não coloque constantes numéricas (com algumas exceções, como 0 ou 1) pelo meio do código. Defina constantes simbólicas no início e use-as depois no programa;
- Como boa prática, as variáveis em memória devem ser de 16 bits (WORD), para suportarem valores negativos sem problemas. O PEPE só sabe fazer aritmética em complemento para 2 com 16 bits;
- Os periféricos de 8 bits e as tabelas com BYTE devem ser acedidos com a instrução MOVB. As variáveis definidas com WORD (que são de 16 bits) e periféricos de 16 bits devem ser acedidos com MOV;
- Não duplique código (com *copy-paste*). Use uma rotina com parâmetros para contemplar os diversos casos em que o comportamento correspondente é usado.

9 – Dispositivos e circuito do projeto

O mapa de endereços (em que os dispositivos podem ser acedidos pelo PEPE) é o seguinte:

Dispositivo	Endereços
RAM	0000H a 3FFFH
MediaCenter (acesso aos comandos)	6000H a 6069H (ver Guião 4)
MediaCenter (acesso à memória)	8000H a 8FFFH
POUT-1 (periférico de saída de 8 bits)	0A000H
POUT-2 (periférico de saída de 8 bits)	0C000H
PIN (periférico de entrada de 8 bits)	0E000H

A figura seguinte mostra o circuito a usar (fornecido no ficheiro **projeto-vitrine-natalicia.cir**), semelhante ao do **Guião Prático 5**. Use este circuito para as versões **v0** a **v2** do projeto. Para a versão **v3** utilize o circuito do **Guião Prático 6**, e para a **v4** utilize o circuito do **Guião Prático 7**.



Os módulos seguintes têm um painel de controlo em execução (no modo “Simulation”):

- **MediaCenter** – módulo multimédia que inclui um ecrã de 32×64 pixels. Este ecrã é accedido por comandos ou como se fosse uma memória de 2048 pixels (ou 4096 bytes: 128 bytes em cada linha, 32 linhas). Este periférico tem 2 *chip selects*, um para acesso pela memória e outro para acesso pelos comandos. Pode ver no ficheiro de excel **ecrã-32x64-vazio.xlsx** os endereços de cada byte (relativos ao endereço de base do ecrã). O **Guião Prático 4** fornece mais detalhes;
- Dois **displays de 7 segmentos**, ligados aos bits 7–4 e 3–0 do periférico POUT-1 (8 bits);
- **Teclado**, de 4×4 teclas, com 4 bits ligados ao periférico POUT-2 e 4 bits ligados ao periférico PIN (bits 3-0). A deteção de qual tecla foi carregada é feita por varrimento. Atenção, que estes periféricos são de 8 bits e devem ser acedidos com MOV_B (e não MOV). Note também que só os 4 bits de menor peso (3 a 0) são significativos. Os restantes (7 a 4) estão no ar e leem valores aleatórios. Por isso, deve usar uma máscara para os eliminar ao tentar detetar teclas premidas;
- **Memória** (RAM), que tem 16 bits de dados e 14 bits de endereço, com capacidade de endereçamento de byte, tal como o PEPE e o MediaCenter;
- **PEPE-16** (processador de 16 bits).

Bom trabalho!