



Escola Secundária

**Dr. Joaquim de Carvalho**

Figueira da Foz

401470

Nº Projeto: \_\_\_\_\_

Nº Curso: \_\_\_\_\_ Nº Ação: \_\_\_\_\_

**Cursos Profissionais**



**Planificação Anual**

**2017/2018**

**Curso Profissional de Técnico de Gestão de Equipamentos Informáticos**

**SISTEMAS DIGITAIS E ARQUITETURA DE COMPUTADORES**

**11º ANO**

<b>MÓDULO 5 - Introdução à Lógica Programável</b>		<b>24 aulas de 45' = 18h</b>
<b>Data Início/Final:</b> 13 setembro 2016 a 12 outubro 2017	<b>Datas avaliação</b> – 12 out 2017	

<b>Objetivos Gerais</b>	<b>Conteúdos</b>	<b>Situações de aprendizagem/avaliação</b>	<b>Estratégias</b>	<b>Aplicação dos critérios de avaliação</b>
<ul style="list-style-type: none"> <li>▪ Saber justificar as vantagens de se utilizarem circuitos de lógica programável em substituição dos circuitos tradicionais.</li> <li>▪ Compreender a classificação dos circuitos de lógica programável.</li> <li>▪ Conhecer a estrutura interna de uma PAL e de uma GAL.</li> <li>▪ Conhecer a estrutura de um programa em CUPDL (ou PALASM).</li> <li>▪ Conseguir implementar circuitos combinatorios utilizando GALs.</li> <li>▪ Opcionalmente conseguir implementar circuitos sequenciais simples utilizando GALs.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Vantagem da utilização lógica programável sobre a lógica tradicional.</li> <li>▪ Classificação e estrutura interna dos principais tipos de circuitos de lógica programável.</li> <li>▪ Diferentes tipos de programar um circuito deste tipo: Através de um editor de esquemáticos ou através de linguagem de programação: ABEL, VHDL, CPUDL ou PALASM.</li> <li>▪ Estrutura de um programa em CPUDL (ou PALASM).</li> <li>▪ Exemplos de programação de circuitos combinatoriais simples em CPUDL, utilizando GALs.</li> <li>▪ Programação e teste de um circuito combinatorial utilizando uma GAL.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Trabalhos práticos.</li> <li>▪ Observação direta do trabalho desenvolvido e registo.</li> <li>▪ Ficha de avaliação.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Realização de trabalhos práticos.</li> <li>▪ Utilização de recursos do laboratório (dispositivos eletrónicos, nomeadamente portas lógicas e <i>breadboard</i>).</li> <li>▪ Utilização de <i>software</i> adequado à elaboração e ensaio de circuitos virtuais.</li> <li>▪ Utilização da <i>Internet</i>.</li> <li>▪ Utilização de simulador.</li> </ul>	<p><b>Dentro dos 60%</b>            Testes sumativos 30%            Trabalho prático 30%</p> <p><b>Dentro dos 30%</b>            Realização de todos os exercícios propostos.</p> <p><b>Desempenho 10%</b></p>

<b>MÓDULO 6 - Fundamentos de Programação</b>	<b>184 aulas de 45' = 160h</b>
<b>Data Início/Final:</b> 12 outubro 2017 a junho 2017	<b>Datas avaliação</b> – FA - 12 dezembro   2 FA + 1 Projeto   1 FA + 1 Projeto

<b>Objetivos Gerais</b>	<b>Conteúdos</b>	<b>Situações de aprendizagem/avaliação</b>	<b>Estratégias</b>	<b>Aplicação dos critérios de avaliação</b>
<ul style="list-style-type: none"> <li>▪ Iniciar o desenvolvimento de raciocínios algorítmicos e aquisição de métodos de desenvolvimento de programas.</li> <li>▪ Abordar linguagens: sintaxe e semântica.</li> <li>▪ Conhecer os conceitos de instrução, dados e programa.</li> <li>▪ Utilizar uma linguagem natural e uma linguagem gráfica, no desenho e teste de Algoritmos.</li> <li>▪ Utilizar um ambiente integrado de desenvolvimento de programas para edição, compilação e teste</li> <li>▪ Estudar tipos de dados simples.</li> <li>▪ Conhecer e utilizar instruções de decisão.</li> <li>▪ Conhecer e utilizar instruções de controlo de fluxo.</li> <li>▪ Iniciar o desenvolvimento de raciocínios algorítmicos e aquisição de métodos de desenvolvimento de programas.</li> <li>▪ Abordar linguagens: sintaxe e semântica.</li> <li>▪ Conhecer os conceitos de instrução, dados e programa.</li> <li>▪ Utilizar uma linguagem natural e uma linguagem gráfica, no desenho e teste de Algoritmos.</li> </ul>	<ul style="list-style-type: none"> <li>▪ <b>Definição de Linguagem.</b></li> <li>▪ Conceitos de Sintaxe, Semântica, Gramática e Expressão.</li> <li>▪ Exemplo com um subconjunto da Linguagem Natural.</li> <li>▪ Erros de Sintaxe e de Semântica nas frases (expressões) de uma Linguagem (gramática).</li> <li>▪ <b>Definição de Algoritmo como processo descritivo de uma Linguagem.</b></li> <li>▪ Exemplificação (meramente conceptual) de algoritmos simples.</li> <li>▪ Desenvolvimento conceptual de tipos de informação e respetivo armazenamento (conceito intuitivo de variável num algoritmo).</li> <li>▪ Valores Numéricos, Alfanuméricos e Lógicos.</li> <li>▪ Desenvolvimento conceptual da possibilidade de o algoritmo dispor da capacidade de recolher informação do utilizador e de enviar informação para o utilizador.</li> <li>▪ Exemplos em linguagem natural envolvendo mecanismos intuitivos de Decisão Binária e Decisão Múltipla.</li> <li>▪ Exemplos em linguagem natural envolvendo mecanismos de repetição condicionada por uma expressão lógica.</li> <li>▪ Desenvolvimento de algoritmos, fazendo uso de uma linguagem gráfica com o objetivo de analisar o seu fluxo de execução sequencial.</li> <li>▪ <b>Estudo e utilização de um ambiente integrado de desenvolvimento de programas para edição, compilação e teste de programas:</b> <ul style="list-style-type: none"> <li>– Estrutura de um programa.</li> <li>– Tipos de variáveis. Tipos simples.</li> <li>– Instruções: Afetação, Input e Output de informação.</li> <li>– Mecanismos de controlo de programa (seleção simples, seleção múltipla, repetição condicional, repetição incondicional).</li> </ul> </li> <li>▪ Funções Simples.</li> <li>▪ Implementação de Algoritmos de complexidade crescente</li> <li>▪ Utilização das ferramentas de <i>Debugging</i> disponíveis:</li> </ul>	<ul style="list-style-type: none"> <li>▪ Trabalhos teórico e/ou práticos.</li> <li>▪ Observação direta do trabalho desenvolvido e registo.</li> <li>▪ Relatórios e/ou fichas de avaliação.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Realização de exercícios práticos.</li> <li>▪ Realização de projetos.</li> <li>▪ Utilização de recursos do laboratório (linguagens de programação, Pascal e Visual C#).</li> <li>▪ Utilização da <i>Internet</i>.</li> </ul>	<p><b>Dentro dos 60%</b> Testes sumativos 40% Trabalho prático 20%</p> <p><b>Dentro dos 30%</b> Realização de todos os exercícios propostos.</p> <p><b>Desempenho 10%</b></p>

<ul style="list-style-type: none"> <li>▪ Utilizar um ambiente integrado de desenvolvimento de programas para edição, compilação e teste</li> <li>▪ Estudar tipos de dados simples.</li> <li>▪ Conhecer e utilizar instruções de decisão.</li> <li>▪ Conhecer e utilizar instruções de controlo de fluxo.</li> <li>▪ Saber fazer <i>DEBUGGING</i> e o visionamento passo-a-passo da execução de algoritmos.</li> <li>▪ Conhecer e manipular estruturas de dados estáticas (vetores e matrizes).</li> <li>▪ Saber decompor um programa em sub-programas modularização).</li> <li>▪ Utilizar parâmetros na construção de sub-programas.</li> <li>▪ Conhecer os níveis de visibilidade ("Scope") das variáveis de um programa.</li> <li>▪ Estudar tipos de dados compostos.</li> <li>▪ Saber analisar as necessidades de estruturas de informação utilizando as estruturas de dados apropriadas.</li> <li>▪ Estudar as formas de armazenamento de informação em memória secundária como forma de manter.</li> </ul>	<ul style="list-style-type: none"> <li>– Observação do valor de variáveis.</li> <li>– Execução de algoritmos em modo "STEP by STEP."</li> <li>– Definição de "Breakpoints".</li> <li>– Execução de algoritmos por Troços.</li> <li>▪ <b>Estruturas de dados estáticas (unidimensionais):</b> <ul style="list-style-type: none"> <li>– Declaração e Manipulação.</li> </ul> </li> <li>▪ <b>Estudo de algoritmos de manipulação de Arrays:</b> <ul style="list-style-type: none"> <li>– Algoritmos de iniciação.</li> <li>– Algoritmos de pesquisa sequencial.</li> <li>– Algoritmos de inserção e remoção de elementos: No Início (à Cabeça - FIFO); no Fim (à Cauda - LIFO).</li> <li>– Algoritmos de ordenação.</li> </ul> </li> <li>▪ <b>Estruturas de dados estáticas (multidimensionais)</b></li> <li>▪ Análise <i>top-down</i>, versus <i>bottom-up</i>: <ul style="list-style-type: none"> <li>– Diferenças.</li> <li>– Declaração</li> <li>– Utilização</li> </ul> </li> <li>▪ Regras de "Scope" para a utilização de variáveis.</li> <li>▪ <b>Variáveis Globais e Variáveis Locais:</b> <ul style="list-style-type: none"> <li>– Período de existência das variáveis.</li> <li>– Regras de "Scope" para a utilização de variáveis.</li> </ul> </li> <li>▪ <b>Passagem de parâmetros a sub-programas:</b> <ul style="list-style-type: none"> <li>– Passagem por Valor.</li> <li>– Passagem por Referência de Endereço.</li> </ul> </li> <li>▪ Tipos de dados compostos: <ul style="list-style-type: none"> <li>– Sintaxe.</li> <li>– Manipulação</li> </ul> </li> <li>▪ <b>Estruturas de dados compostos.</b> <ul style="list-style-type: none"> <li>– Desenho de aplicações que envolvam estruturas de dados de baixa complexidade.</li> <li>– Ficheiros como variáveis suportadas em disco.</li> <li>– Declaração de variáveis do tipo ficheiro da mesma forma que são declaradas variáveis em RAM.</li> <li>– Associação do nome físico do ficheiro (ao nível do Sistema Operativo) ao nome lógico do ficheiro (ao nível do programa). Abertura e fecho de ficheiros.</li> <li>– Acesso a ficheiros.</li> <li>– Manipulação de ficheiros.</li> <li>– Exemplos de ficheiros de texto pré definidos: COM (Porta Série), LPT1 ou PRN</li> <li>– (Porta Paralela. Exercícios envolvendo estes ficheiros (por exemplo, imprimir um ficheiro de texto).</li> </ul> </li> </ul>			
---	--	--	--	--