



Escola Secundária

Dr. Joaquim de Carvalho

Figueira da Foz

Nº Projeto: _____

Nº Curso: _____ Nº Ação: _____

Cursos Profissionais



Planificação Anual

2016/2017

Curso Profissional de Técnico de Gestão de Equipamentos Informáticos

SISTEMAS DIGITAIS E ARQUITETURA DE COMPUTADORES

11º ANO

MÓDULO 5 - Introdução à Lógica Programável		24 aulas de 45' = 18h
Data Início/Final: 19 setembro 2016 a 13 outubro 2016	Datas avaliação – 06 out 2016 e 13 out 2016 (TP)	

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

MÓDULO 6 - Fundamentos de Programação		45 aulas de 45' = 34h
Data Início/Final: 17 outubro 2016 a 05 junho 2017	Datas avaliação – FA - 13 dezembro 2 FA + 1 Projeto 1 FA + 1 Projeto	

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

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