



AGRUPAMENTO DE ESCOLAS
DE OLIVEIRA DE FRADES

ESCOLA BÁSICA E SECUNDÁRIA DE OLIVEIRA DE FRADES

Planificação Anual – 2024/2025



REPÚBLICA
PORTUGUESA
EDUCAÇÃO, CIÊNCIA E INOVAÇÃO

Disciplina: **Programação**

Ano: 10^o

Turmas: TPI

Anual

TEMAS/UNIDADES	CONTEÚDOS/APRENDIZAGENS ESSENCIAIS	N.º de AULAS PREVISTAS	
<p>MÓDULO: 0804 – Algoritmos (Professor responsável – José Viegas)</p>	<ol style="list-style-type: none"> 1. Linguagem estruturada <ol style="list-style-type: none"> 1.1. Estruturas lógicas 1.2. Desenho das estruturas diagramáticas 1.3. Sintaxe da linguagem 2. Construção de um algoritmo <ol style="list-style-type: none"> 2.1. Noções de ação e estado da ação 2.2. Ações e a sua sintaxe 2.3. Verbos 2.4. Sintaxe 2.5. Alinhamento das frases 3. Estruturas lógicas básicas e o seu controlo <ol style="list-style-type: none"> 3.1. Estrutura sequencial, alternativa e repetitiva 3.2. Condições e regras de inicialização e alteração 3.3. Estruturas diagramáticas como representação algorítmica 4. Tipos de dados em algoritmia <ol style="list-style-type: none"> 4.1. Constantes e Variáveis 5. Descrição da entrada e saída de dados <ol style="list-style-type: none"> 5.1. Elementos de linguagem 6. Técnicas de construção de algoritmos <ol style="list-style-type: none"> 6.1. Contadores 6.2. Totalizadores 6.3. Expressões aritméticas 7. Validação de dados Construção de procedimentos <ol style="list-style-type: none"> 7.1. Procedimentos de entrada e saída e chamada de procedimentos 7.2. Funções predefinidas 	30	1º Período
<p>MÓDULO: 0806 - Princípios metodológicos de programação (Professor responsável – José Viegas)</p>	<ol style="list-style-type: none"> 1. Metodologias em programação local <ol style="list-style-type: none"> 1.1. Evolução 1.2. Vantagens e desvantagens do seu uso 1.3. Tipos de metodologias 1.4. Desenho estruturado 1.5. Orientadas por objeto 1.6. Orientadas aos dados 2. Princípios do método <ol style="list-style-type: none"> 2.1. Teoria dos conjuntos aplicada aos dados 2.2. Representações gráficas dos conjuntos 2.3. Operações sobre conjuntos 2.4. Dados agrupados em conjuntos 2.5. Dados de entrada 2.6. Dados de saída 2.7. Frequência dos subconjuntos 	30	1º e 2º Período

	<ol style="list-style-type: none"> 3. Estrutura de um programa <ol style="list-style-type: none"> 3.1. Regras de dedução para a identificação dos procedimentos 3.2. Sequência das instruções nos procedimentos 3.3. Condições o seu emprego no controlo das estruturas lógicas 4. Corpo do programa <ol style="list-style-type: none"> 4.1. Desenvolvimento da sequência 4.2. Tratamento da estrutura de dados 4.3. Regras de gestão 5. Desenho de formatos para entrada e saída dos dados em ecrã <ol style="list-style-type: none"> 5.1. Normas de desenho 		
<p>MÓDULO: 0809 - Programação em C/C++ fundamentos (Professor responsável – José Viegas)</p>	<ol style="list-style-type: none"> 1. Estrutura de um programa em C/C++ <ol style="list-style-type: none"> 1.1. Função main () 1.2. Estrutura de um programa 2. Dados em C <ol style="list-style-type: none"> 2.1. Variáveis e constantes 2.2. Tipos de dados 2.3. Variáveis de ponto flutuante 3. Estudo e emprego da função printf() <ol style="list-style-type: none"> 3.1. Constantes e variáveis 3.2. Formatação 3.3. Sequências de escape 3.4. Carateres gráficos 3.5. Valores de virgula-flutuante 3.6. Vários argumentos em printf() 4. Cadeia de carateres e entrada e saída de dados formatados <ol style="list-style-type: none"> 4.1. Uma cadeia de carateres 4.2. Função scanf() 4.3. Códigos de formatação e modificação para a função scanf() 4.4. Operador de endereço & 4.5. Função strlen() 4.6. Directiva #define 4.7. Funções getche(), getch(), getchar() e putchar() 5. Operadores e expressões em C/C++ <ol style="list-style-type: none"> 5.1. Operadores e expressões aritméticas 5.2. Operadores de atribuição +=, -=, *=, /= e %= 5.3. Operadores cast e relacionais 6. Estruturas repetitivas em C/C++ <ol style="list-style-type: none"> 6.1. Implementação com a instrução <i>while</i>, <i>for</i> e <i>do while</i> 7. Estruturas repetitivas complexas <ol style="list-style-type: none"> 7.1. Estruturas matriciais 7.2. Condições de controlo 8. Estruturas alternativas em C/C++ 	60	2º e 3º Período

	<ul style="list-style-type: none"> 8.1. Implementação com a instrução <i>if, if else e switch</i> 8.2. Condições e os operadores lógicos 8.3. Alternativas múltiplas 8.4. Instruções <i>break, continue e go to</i> 9. Funções <ul style="list-style-type: none"> 9.1. Estrutura e argumentos de uma função 9.2. Variáveis locais 9.3. Funções recursivas 9.4. Classes de armazenamento 10. Matrizes <ul style="list-style-type: none"> 10.1. Declaração e inicialização de matrizes 10.2. Armazenamento e leitura de dados de uma matriz 10.3. Matrizes unidimensionais/multidimensionais 10.4. Matrizes como argumento de funções 11. Cadeias de caracteres (string) <ul style="list-style-type: none"> 11.1. Constantes, Variáveis e Matriz de strings 11.2. Leitura e escrita de strings 		
<p>MÓDULO: 0810 - Programação em C/C++ - avançada (Professor responsável – José Viegas)</p>	<ul style="list-style-type: none"> 1. Apontadores <ul style="list-style-type: none"> 1.1. Definição e declaração de apontadores 1.2. Apontadores para passar dados para uma função 1.3. Operador indireto (*) 1.4. Atribuição de valores e variáveis apontadores 1.5. Operações com apontadores 1.6. Apontadores para matrizes e para strings 1.7. Apontadores para apontadores 1.8. Apontadores para funções apontadores como argumento de uma função 2. Estruturas <ul style="list-style-type: none"> 2.1. Definição e declaração de estruturas 2.2. Inicialização de estruturas 2.3. Leitura e escrita de valores nos elementos de uma estrutura 2.4. Matrizes de estruturas 2.5. Apontadores para estruturas 2.6. Estruturas em lista ligada 2.7. Declaração e operações de e sobre listas ligadas 3. Uniões <ul style="list-style-type: none"> 3.1. Definição, declaração iniciação de uniões 3.2. Uniões de estruturas 3.3. Problemas com uniões 4. Impressão de caracteres por acesso direto à memória <ul style="list-style-type: none"> 4.1. Sistema binário, hexadecimal e octal 4.2. Operadores sobre os bits 4.3. Conversão entre sistemas 5. Campos bit <ul style="list-style-type: none"> 5.1. Pré-processor C/C++ e as diretivas 5.2. Diretiva #define 5.3. Macros e funções - Comparação 	60	3º Período

	<p>5.4. Directivas #include #undef, #if, #ifdef, #ifndef, #else, #endif e #error</p> <p>6. Livraria do C/C++</p> <p>6.1. Acesso à livraria #include</p> <p>6.2. Livraria Math-#include</p> <p>7. Alocação e desalocação de memória</p> <p>7.1. Funções malloc(), calloc() e free()</p> <p>7.2. Ficheiros em C/C++</p> <p>7.3. Níveis de leitura e escrita em ficheiros</p> <p>8. Abertura e fecho de ficheiros</p> <p>8.1. Leitura e escrita em ficheiros</p> <p>8.2. Condições de erro</p> <p>8.3. Buffers</p> <p>8.4. Outras funções para manipular ficheiros</p>		
Atividades de avaliação	<ul style="list-style-type: none"> • Diagnóstica • Formativa contínua • Comportamento e postura dentro da sala de aula • Participação adequada • Observação direta • Assiduidade e pontualidade • Autonomia • Cooperação e relacionamentos interpessoais • Trabalhos práticos 		
Total de Aulas Previstas (50 minutos)			180

Nota: Os conteúdos a abordar serão lecionados de forma integrada entre as diversas disciplinas da componente técnica e professores responsáveis pela leção de cada módulo, contribuindo de forma ativa para o desenvolvimento integral dos alunos e constituindo-se como atividades facilitadoras do desenvolvimento da Prova de Aptidão Profissional e da Formação em Contexto de Trabalho.

O Grupo Disciplinar