

**Ministério da Educação**  
**Departamento do Ensino Secundário**

Programa de  
**BASES DE PROGRAMAÇÃO**  
**10º, 11º e 12º anos**

**Curso Tecnológico de Informática**

Autores:

Luís Guerra (Coordenador)

Francisco Simões

Manuel C. Pinto

Manuel Silva

Luísa Rodrigues

**Homologação**

29/08/2001 (10º ano)

6/12/2002 (11º e 12º anos)

## Índice

<b>1. INTRODUÇÃO.....</b>	<b>3</b>
<b>2. APRESENTAÇÃO DO PROGRAMA.....</b>	<b>4</b>
2.1. FINALIDADES .....	4
2.2. OBJECTIVOS .....	5
2.3. VISÃO GERAL DOS CONTEÚDOS.....	6
2.4.SUGESTÕES METODOLÓGICAS GERAIS.....	7
2.5. COMPETÊNCIAS A DESENVOLVER .....	8
2.6. AVALIAÇÃO.....	9
2.7. RECURSOS.....	10
<b>3. DESENVOLVIMENTO DO PROGRAMA.....</b>	<b>11</b>
3. 1. 10º ano .....	11
3. 2. 11º ano .....	16
3. 3. 12º ano .....	21
<b>4. BIBLIOGRAFIA .....</b>	<b>30</b>

# 1. INTRODUÇÃO

A disciplina *Bases de Programação*, do Curso Tecnológico de Informática, é uma disciplina trienal do tronco comum da componente Científico-Tecnológica, tendo uma carga horária de 3 horas divididas por aulas de 90 minutos ao longo das 33 semanas de cada ano lectivo.

Esta disciplina pretende conduzir o aluno à resolução de problemas de uma forma estruturada e sistematizada, analisando todos os elementos intervenientes.

No 10º ano, através do estudo de algoritmia e programação estruturada em linguagem Pascal, pretende-se criar as bases necessárias para a evolução do aluno na área das linguagens de programação.

No 11º ano, usando o Visual Basic, linguagem de grande divulgação, o aluno irá adquirir não só as bases da programação por objectos, como deverá ser capaz de elaborar pequenas aplicações informáticas utilizando uma ferramenta visual.

No 12º ano e usando uma linguagem de médio nível como o Visual C++, pretende-se a consolidação e alargamento de conhecimentos na área das linguagens de programação. Haverá ainda espaço para o estudo genérico de pelo menos uma linguagem de programação que se considere adequada às exigências informáticas do momento (por exemplo JAVA, PHP, ASP, PERL), com o objectivo de preparar o aluno para o estudo de novas linguagens.

Dada a natureza da disciplina, é aconselhável que não existam mais de dois alunos por posto de trabalho, podendo a turma ser dividida em turnos, ou não, conforme o número de alunos e o número de computadores existentes na sala de aula.

Deverá ser mantida uma estreita colaboração entre todos os professores que leccionem as disciplinas da componente Científico – Tecnológica do Curso Tecnológico de Informática.

## **2. APRESENTAÇÃO DO PROGRAMA**

### **2.1. FINALIDADES**

Contribuindo para a formação de técnicos de informática, a disciplina de Bases de Programação tem como finalidades:

- Desenvolver a capacidade de reflexão, gerando um saber criativo e evolutivo.
- Desenvolver a autoconfiança com base nos saberes e competências adquiridos.
- Fomentar o interesse pela investigação.
- Proporcionar um conjunto de experiências e vivências que permitam aos alunos acompanhar a permanente evolução tecnológica, quer profissionalmente, quer em termos pessoais e humanos.
- Promover o espírito crítico, a capacidade de cooperação, o respeito pelos outros e a responsabilidade, permitindo que os alunos se desenvolvam integralmente enquanto indivíduos e enquanto cidadãos.
- Desenvolver competências de trabalho em equipa.
- Promover a aquisição de competências e versatilidade no desenvolvimento de aplicações informáticas.
- Promover o respeito pela propriedade intelectual.

## 2.2. OBJECTIVOS

- Analisar e resolver problemas.
- Conceber e implementar projectos de natureza interdisciplinar.
- Conhecer diferentes formas de tratamento de dados.
- Conhecer os tipos de representação para o desenvolvimento de algoritmos.
- Conhecer os elementos da linguagem algorítmica.
- Conhecer as noções da programação por objectos.
- Construir algoritmos com base em situações reais.
- Utilizar ferramentas de desenvolvimento.
- Conhecer a sintaxe, o léxico e a semântica das linguagens que integram o programa.
- Aplicar a sintaxe, o léxico e a semântica das linguagens que integram o programa.
- Adaptar-se a novas situações.
- Trabalhar em equipa.
- Conhecer e aplicar normas de segurança de dados.

## 2.3. VISÃO GERAL DOS CONTEÚDOS

### 10º ANO

- NOÇÕES BÁSICAS
  - Análise sistemática de um problema
  - Algoritmos e programas
  - Dados e operações elementares
  - Operadores
  - Variáveis e expressões
- ALGORITMIA E PROGRAMAÇÃO ESTRUTURADA
  - Algoritmos
  - Fluxogramas
  - Pseudocódigo
  - Programação Pascal
- ESTRUTURAS DE DADOS
  - Caracteres
  - Vectores e matrizes
  - Registos
  - Ficheiros

### 11º ANO

- PROGRAMAÇÃO POR OBJECTOS
  - Conceitos e regras
- PROGRAMAÇÃO EM VISUAL BASIC
  - Desenvolvimento de aplicações

### 12º ANO

- PROGRAMAÇÃO EM C++
  - Desenvolvimento de aplicações
- PROGRAMAÇÃO EM VISUAL C++
  - Desenvolvimento de aplicações utilizando um interface gráfico
- PROGRAMAÇÃO EM OUTRAS LINGUAGENS
  - Desenvolvimento de aplicações (JAVA ou PHP ou ASP ou PERL)

## 2.4. SUGESTÕES METODOLÓGICAS GERAIS

Inicialmente, deverá ser apresentado aos alunos o contexto em que irão estar inseridos, em termos de Escola e Curso, bem como o desenvolvimento da disciplina e as respectivas articulações, no sentido de despertar a necessidade da escolha de uma especificação no 12º ano.

A consciencialização do aluno para as finalidades do curso poderá ser feita recorrendo a filmes onde sejam abordadas as potencialidades da informática no mundo real.

O módulo inicial deverá permitir detectar conhecimentos prévios e interesses dos alunos na área da informática, em geral, e da disciplina, em particular.

A fase inicial desta disciplina é crucial para a compreensão e para o desenvolvimento do raciocínio lógico formal. Os alunos poderão vir a sentir grandes dificuldades no desenrolar da disciplina se não lhes for facultado o tempo e dedicação necessários à construção dos primeiros passos de um algoritmo. Os elementos de dificuldade para a resolução de problemas deverão ser introduzidos gradualmente, dando espaço de criatividade ao aluno para que este desenvolva a sua própria técnica e estilo de programação.

Sempre que possível, devem orientar-se os alunos para experiências por eles propostas, mesmo que tal os coloque numa situação a ser abordada posteriormente, uma vez que esta poderá funcionar como factor de motivação e enquadrador dos conteúdos e competências a adquirir mais tarde pelo próprio indivíduo. É neste âmbito que se poderá, por exemplo, fazer a interdisciplinaridade com as restantes disciplinas do curso, de carácter geral ou científico-tecnológico, propondo projectos de desenvolvimento futuro e transdisciplinares, orientados pelos professores de uma forma articulada com os conteúdos e competências pretendidos.

No que respeita ao relacionamento em grupo, o professor deverá ajudar os alunos a adaptarem-se aos métodos de trabalho dos diferentes indivíduos existentes no grupo, levando-os a reconhecer que a solução dos problemas surgidos não passa por abandonar os projectos ou os colegas, mas por encontrar novas formas de comunicação e de relacionamento entre todos.

É fundamental desenvolver o trabalho em equipa, através da implementação de modalidades de aprendizagem cooperativa.

## 2.5. COMPETÊNCIAS A DESENVOLVER

- Identificar os diversos factores intervenientes num problema.
- Utilizar diferentes estratégias de tratamento dos dados.
- Fasear a resolução de um problema.
- Resolver problemas, encontrando soluções criativas e adequadas.
- Usar vários suportes de informação para resolver dificuldades concretas na utilização das ferramentas de programação.
- Utilizar a “ajuda” para resolver dificuldades concretas na utilização das ferramentas de programação.
- Programar a realização de trabalho de equipa, nomeadamente dividir as tarefas para a realização do trabalho conjunto e definir a estrutura de coordenação.
- Cumprir prazos na realização das tarefas.
- Optar pela melhor decisão na resolução de um problema concreto.
- Aceitar opiniões diferentes e integrar as propostas dos outros na melhoria do trabalho.
- Usar metodologias de análise algorítmica.
- Utilizar diferentes métodos algorítmicos.
- Utilizar estruturas de decisão.
- Criar algoritmos estruturados.
- Traduzir algoritmos em linguagem Pascal.
- Utilizar diferentes estruturas de dados.
- Desenvolver aplicações em Visual Basic.
- Desenvolver aplicações em Visual C++.
- Desenvolver aplicações em JAVA ou PHP ou ASP ou PERL.
- Adaptar-se à evolução das linguagens.
- Utilizar uma linguagem adequada para a criação de relatórios.
- Documentar todo o trabalho desenvolvido.

## 2.6. AVALIAÇÃO

Deverá ser dada ênfase especial à avaliação do desempenho e da manipulação dos programas, assim como a toda a documentação que acompanha os programas, isto é, relatórios de análise e programação e manuais de utilização.

Sugere-se o uso sistemático de grelhas de observação que permitirão avaliar a destreza do aluno com as novas aplicações, o equipamento, a sua capacidade de organização e concentração, a qualidade do trabalho realizado e o relacionamento do aluno em grupo de trabalho.

A avaliação será portanto contínua, registando a evolução do aluno aula a aula, permitindo ultrapassar de imediato qualquer dificuldade. No entanto, existirão momentos em que será necessário proceder, no final de cada unidade, a provas práticas individuais que permitam avaliar os conhecimentos e as competências adquiridos ao longo do processo de ensino/aprendizagem. Por outro lado, a consolidação das aprendizagens e das competências também poderá ser feita através da avaliação do desempenho e de Trabalho de Projecto ou do desenvolvimento de trabalhos de grupo, os quais, sempre que possível, deverão ser apresentados à comunidade escolar, com recurso a meios electrónicos.

## 2.7. RECURSOS

Para a leccionação da disciplina de *Bases de Programação* é necessário um Laboratório de Informática devidamente equipado com:

- computadores (preferencialmente 1 por cada 2 alunos) com sistema operativo actualizado e com pacote de *software* genérico;
- 1 Servidor de rede;
- 1 Retroprojector;
- 1 *Data-Show* ou 1 projector de vídeo;
- 1 Écran de projecção;
- 1 Impressora A4;
- 1 *Scanner*;
- 1 Gravador de CD-ROM para efectuar cópias de segurança da informação, instalado no servidor de rede;
- 1 Rede local de 10/100 Mbs;
- Acesso à Internet;
- Ferramentas de programação: Pascal, Visual Basic, Visual C++ e outras (JAVA, PHP, ASP, PERL);
- Ferramentas de desenho para a criação de relatórios de análise de programação (Microsoft Visio, outros...).

### 3. DESENVOLVIMENTO DO PROGRAMA

#### 3. 1. 10º ano

#### MÓDULO INICIAL

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Conhecer termos e conceitos básicos relacionados com as Bases de Programação.</li> <li>- Identificar competências básicas relacionadas com a disciplina e com o curso, nomeadamente no domínio do raciocínio lógico e da algoritmia.</li> <li>- Induzir o espírito de iniciativa e de cooperação.</li> </ul>	<ul style="list-style-type: none"> <li>• O que são as Bases de Programação               <ul style="list-style-type: none"> <li>- Objectivos das Bases de Programação no contexto do Curso Tecnológico de Informática</li> </ul> </li> <li>• Área de desenvolvimento               <ul style="list-style-type: none"> <li>- O papel do analista</li> <li>- O papel do programador</li> </ul> </li> <li>• Organização dos dados               <ul style="list-style-type: none"> <li>- Conceito de dado</li> <li>- Diferentes formas de organização de dados</li> </ul> </li> <li>• Competências envolvidas na programação</li> </ul>	<ul style="list-style-type: none"> <li>- Apresentação pelo professor de exemplos concretos dos conceitos introdutórios a desenvolver nas unidades seguintes, bem como nos anos subsequentes.</li> <li>- Discussão dos alunos em grupo sobre os diferentes conteúdos da unidade.</li> <li>- Realização de tarefas práticas de carácter essencialmente lúdico que permitam aos alunos evidenciar competências relacionadas com a programação, nomeadamente ao nível da algoritmia (p. ex. reorganizar uma história, pela sua sequência lógica).</li> </ul>	4

## Unidade de Ensino/Aprendizagem 1

### NOÇÕES BÁSICAS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"><li>- Distinguir as diferentes fases na resolução de um problema.</li><li>- Distinguir algoritmo de programa.</li><li>- Distinguir os diferentes tipos de dados.</li><li>- Reconhecer os diferentes operadores e as regras de prioridade.</li><li>- Utilizar variáveis e construir expressões.</li><li>- Construir sequências lógicas de operações em linguagem informal, na resolução de problemas.</li></ul>	<ul style="list-style-type: none"><li>• Resolução de problemas</li><li>• Noções de Algoritmo e Programa</li><li>• Tipos de dados e operações elementares</li><li>• Prioridade de operadores</li><li>• Variáveis e expressões</li><li>• Construção de algoritmos em linguagem informal</li></ul>	<p>O professor deverá introduzir a unidade através de exemplos do dia-a-dia, simples de compreender, em que os alunos poderão reconhecer a utilização do raciocínio lógico formal, passando, em seguida, à sistematização através da álgebra, construindo expressões e aplicando os diferentes operadores existentes.</p> <p>O professor deverá introduzir o conceito de algoritmo utilizando problemas de âmbito geral, decompondo uma acção complexa em outras mais simples, como sejam, por exemplo, programar uma viagem de estudo, mudar um pneu num automóvel ou uma lâmpada num candeeiro, etc.</p>	12

**Unidade de Ensino/Aprendizagem 2**  
**ALGORITMIA E PROGRAMAÇÃO ESTRUTURADA**

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Utilizar fluxogramas na construção de algoritmos.</li> <li>- Utilizar pseudocódigo na construção de algoritmos.</li> <li>- Utilizar as instruções básicas da linguagem Pascal na codificação de algoritmos.</li> <li>- Aplicar estruturas de decisão na elaboração de algoritmos.</li> <li>- Decompor um problema nas suas partes (estruturas) elementares (<i>TOP-DOWN</i>).</li> <li>- Desenvolver programas em Pascal, de forma estruturada e documentada.</li> </ul>	<ul style="list-style-type: none"> <li>• Simbologia               <ul style="list-style-type: none"> <li>- Fluxogramas</li> <li>- Pseudocódigo</li> </ul> </li> <li>• Noções Básicas de Pascal</li> <li>• Programação Estruturada               <ul style="list-style-type: none"> <li>- Estruturas de decisão</li> <li>- Desenvolvimento Estruturado</li> </ul> </li> <li>• Elaboração de algoritmos e subalgoritmos e sua tradução para Pascal</li> <li>• Teste de algoritmos/Pascal</li> </ul>	<p>O aluno deverá inserir tipos de dados e operações na construção de algoritmos, utilizando exemplos simples, como seja efectuar um cálculo ou analisar o conteúdo de uma frase.</p> <p>O professor deverá realizar, no quadro, algoritmos simples que clarifiquem a necessidade e potencialidade das estruturas de decisão.</p> <p>O professor deve insistir na diversidade de exemplos, acompanhando de perto a evolução dos alunos. O grau de dificuldade deverá crescer progressivamente.</p> <p>O professor deve deixar espaço para que os alunos debatam entre si os problemas e as suas diferentes vias de resolução.</p> <p>O professor não deverá descurar a estruturação e a boa legibilidade do algoritmo, reforçando que este deve ser claro e eficiente.</p> <p>À medida que o grau de dificuldade dos algoritmos vai aumentando, o professor deve introduzir a necessidade de utilização de funções.</p> <p>O professor irá propor a realização de diversos exercícios para solidificar conhecimentos na criação de algoritmos, levando os alunos a saber utilizar as diferentes estruturas.</p> <p>O professor poderá ter os algoritmos implementados no computador em linguagem Pascal, a fim de os poder executar e mostrar os resultados através do <i>data-show</i>.</p>	20

		<p>Utilizando os algoritmos já elaborados, os alunos poderão transformá-los em pequenos programas Pascal. O professor pode optar por ir traduzindo os algoritmos em linguagem Pascal, à medida que estes vão sendo resolvidos, dando uma grande ênfase à explicação da tradução algoritmo/Pascal.</p>	
--	--	---	--

### Unidade de Ensino/Aprendizagem 3

## ESTRUTURAS DE DADOS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Identificar as estruturas de dados.</li> <li>- Aplicar as estruturas de dados na elaboração de algoritmos e sua tradução para Pascal.</li> <li>- Aplicar os conceitos de registo e de ficheiro em Pascal.</li> </ul>	<ul style="list-style-type: none"> <li>• Cadeia de Caracteres</li> <li>• Vectores e Matrizes</li> <li>• Registos</li> <li>• Ficheiros</li> </ul>	<p>A demonstração da necessidade do uso de vectores pode ser conseguida através de exemplos como o cálculo de médias, de máximos e mínimos, etc., dentro de um conjunto de valores, que poderão ser fornecidos um a um, ou através de uma cadeia de caracteres, que deverá ser validada.</p> <p>O professor poderá demonstrar que, para armazenar dados mais complexos, existem outros tipos de estruturas de dados, como os registos.</p> <p>Deverá ainda utilizar exercícios que permitam distinguir entre a armazenagem de dados em memória e em disco, introduzindo a noção de ficheiro.</p> <p>Os conceitos de registo e ficheiro deverão ser abordados nesta unidade de uma forma simples, sendo posteriormente mais aprofundados nos anos seguintes.</p>	30

### 3. 2. 11º ano

## Unidade de Ensino/Aprendizagem 1 INTRODUÇÃO AO VISUAL BASIC

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Conhecer e compreender os conceitos básicos da POO independentemente da linguagem.</li> <li>- Identificar os elementos básicos constituintes do ambiente de desenvolvimento.</li> <li>- Explicar a diferença entre “Design Time” e “Run Time”.</li> <li>- Conhecer o conceito de “Event-driven Programming”.</li> <li>- Conhecer a finalidade de um “Project File” .</li> <li>- Criar aplicações simples.</li> <li>- Conhecer alguns dos objectos, métodos, propriedades e eventos.</li> <li>- Conhecer algumas propriedades e eventos associados a um formulário.</li> <li>- Utilizar algumas das propriedades das “Text Boxes”, “Command Buttons” e “Labels”.</li> </ul>	<ul style="list-style-type: none"> <li>• Conceitos básicos da POO               <ul style="list-style-type: none"> <li>- Classes</li> <li>- Objectos</li> <li>- Mensagens</li> <li>- Atributos</li> <li>- Métodos</li> <li>- Polimorfismo</li> <li>- Herança</li> </ul> </li> <li>• O ambiente de trabalho do Visual Basic               <ul style="list-style-type: none"> <li>- “Project explorer”</li> <li>- “Properties Windows”</li> <li>- “Toolbox”</li> <li>- “Menu Bar”</li> <li>- etc.</li> </ul> </li> <li>• Objectos Básicos e outras características básicas da linguagem Visual Basic e respectivo ambiente de trabalho</li> </ul>	<p>O professor deverá explicar sucintamente os conceitos envolvidos, comparando a programação procedimental com a programação orientada a objectos.</p> <p>O professor deverá explicar o ambiente mostrando os seus elementos com a ajuda de um data-show ou projector de vídeo.</p> <p>Deverá ainda, com a ajuda de pequenas aplicações, expor os vários conceitos básicos apresentados.</p> <p>Os alunos deverão criar pequenas aplicações semelhantes aos exemplos apresentados.</p>	<p>13</p>

## Unidade de Ensino/Aprendizagem 2

### PROGRAMAÇÃO EM VISUAL BASIC

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Identificar “Form Modules”, “Standard Modules” e “Class Modules”.</li> <li>- Utilizar o “Code Editor” .</li> <li>- Criar “Messages Boxes” .</li> <li>- Usar constantes e argumentos do Visual Basic.</li> <li>- Conhecer a diferença entre declarações “Unload” e “Load” .</li> <li>- Conhecer a diferença entre “Model” e “Modeless” .</li> <li>- Utilizar os métodos “Show” e “Hide”.</li> <li>- Conhecer a diferença entre constantes e variáveis.</li> <li>- Aplicar os vários tipos de dados na declaração de variáveis.</li> <li>- Distinguir entre variáveis Públicas e Privadas.</li> <li>- Conhecer os operadores lógicos e de comparação.</li> <li>- Usar as estruturas de controlo (“If .. EndIf”, “Select Case”, “Do .. Loop”, “While .. Wend” e “For .. Next” ).</li> <li>- Conhecer a diferença entre procedimentos e funções.</li> <li>- Criar funções com argumentos e retorno de valores.</li> </ul>	<ul style="list-style-type: none"> <li>• “Code Editor”</li> <li>• “Message Boxes”</li> <li>• Os diferentes tipos de módulos em VB</li> <li>• Introdução a alguns dos métodos, constantes, declarações e argumentos</li> <li>• Declarações para controlo da execução de um programa</li> <li>• Declaração de variáveis, constantes, procedimentos e funções</li> <li>• “Menu Editor”</li> <li>• “Toolbar Control”</li> </ul>	<p>O professor deverá exibir pequenos exemplos demonstrativos, acompanhando-os de uma breve descrição dos vários conceitos envolvidos.</p> <p>Os alunos deverão criar pequenas aplicações semelhantes aos exemplos apresentados.</p> <p>Os alunos deverão criar aplicações que envolvam diversos formulários permitindo-lhes compreender o conceito de variável pública e privada.</p> <p>Os alunos deverão testar os diferentes tipos de dados, criando, para isso, variáveis para cada um, e verificando o seu comportamento em situações distintas.</p> <p>Deverão ainda ser criadas aplicações que envolvam funções com e sem argumentos, permitindo conhecer as suas diferenças.</p> <p>O aluno poderá melhorar o aspecto gráfico das aplicações criadas anteriormente, desenvolvendo um ambiente mais amigável para o utilizador.</p>	22

<ul style="list-style-type: none"><li>- Usar funções do Visual Basic para manipulação de "Strings".</li><li>- Criar menus e submenus utilizando o "Menu Editor" .</li><li>- Conhecer as propriedades que podem ser usadas no "Menu Editor" .</li><li>- Criar uma "ToolBar" usando o "ToolBar Control" .</li></ul>			
---	--	--	--

**Unidade de Ensino/Aprendizagem 3**  
**VALIDAÇÃO DE DADOS E TRATAMENTO DE ERROS**

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Criar aplicações de validação de dados.</li> <li>- Criar aplicações que usem o “Mask Edit Control”.</li> <li>- Utilizar “Break Points”.</li> <li>- Visualizar o valor das variáveis na “Watch Window”.</li> <li>- Distinguir entre os modos “Design”, “Run” e “Debug” .</li> <li>- Executar passo a passo programas utilizando “Call Stack”.</li> </ul>	<ul style="list-style-type: none"> <li>- Validação de dados</li> <li>- Função “InputBox”</li> <li>- Funções de Conversão de Dados</li> <li>- Ferramentas de controlo e detecção de erros</li> </ul>	<p>O professor deverá referir a importância de uma boa validação de dados em qualquer aplicação.</p> <p>Os alunos deverão construir aplicações que validem diferentes tipos de dados.</p> <p>O professor deverá ter especial cuidado na exposição dos vários conceitos de “debug”, dando relevo à importância desta fase.</p> <p>Os alunos poderão utilizar todo o material desenvolvido anteriormente para experimentar as várias etapas na detecção e tratamento de erros.</p>	14

**Unidade de Ensino/Aprendizagem 4**  
**O “ADO DATA CONTROL” EM VISUAL BASIC**

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Utilizar o “ADO Data Control” para visualizar registos numa base de dados.</li> <li>- Utilizar o “ADO Data Control” para procurar, apagar, adicionar e modificar registos.</li> <li>- Utilizar o “Data Environment Designer”.</li> </ul>	<ul style="list-style-type: none"> <li>• “ADO Data Control”</li> <li>• “Data Environment Designer”</li> </ul>	<p>O professor deverá exemplificar as diferentes ferramentas existentes, através do <i>data-show</i> ou vídeo projector.</p> <p>Os alunos deverão criar pequenas aplicações que utilizem o “ADO Data Control” com bases de dados em Access.</p> <p>Os alunos poderão trabalhar em articulação com a disciplina de Aplicações Informáticas.</p>	17

### 3. 3. 12º ano

## Unidade de Ensino/Aprendizagem 1 CONCEITOS BÁSICOS E FERRAMENTAS DE VISUAL C++

### SUBUNIDADE 1.1. – CONHECIMENTOS BÁSICOS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"><li>- Distinguir tipos de dados e palavras-chave do C++.</li><li>- Implementar classes, objectos e funções no C++.</li><li>- Utilizar construtores e destrutores em C++.</li><li>- Reconhecer os conceitos básicos da POO no C++.</li><li>- Distinguir classes derivadas de classes-base e enquadrá-las numa hierarquia de classes.</li><li>- Utilizar funções amigáveis em C++.</li><li>- Realizar gestão de memória interna com a função <i>malloc</i> e os operadores <i>new</i> e <i>delete</i>.</li></ul>	<ul style="list-style-type: none"><li>• Tipos e Referências ou Apontadores Qualificadores</li><li>• Classes, Objectos e Funções do C++</li><li>• Classes Derivadas, Hierarquia de Classes e Herança</li><li>• Polimorfismo e funções de membro virtuais</li><li>• Sobrecarga de funções e sobrecarga de operador</li><li>• Variáveis de membro estáticas</li><li>• Classes amigáveis e funções amigáveis</li><li>• Operadores <i>new</i> e <i>delete</i></li></ul>	<p>O professor, no início desta unidade deverá realizar uma revisão dos conceitos básicos da Programação Orientada a Objectos ( POO ).</p> <p>Todos os elementos da linguagem C++ devem ser introduzidos gradualmente, recorrendo sempre a exemplos ilustrativos da sua utilização.</p> <p>Os exemplos devem ser apresentados, na medida do possível em Visual C++, sem que o professor entre em grande detalhe, pois a linguagem irá ser estudada de forma mais aprofundada.</p>	4

## SUBUNIDADE 1.2. – INTRODUÇÃO AO VISUAL C++

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Conhecer o ambiente de desenvolvimento do Visual C++.</li> <li>- Utilizar o <i>VisualWorkbench</i> e os seus utilitários:               <ul style="list-style-type: none"> <li>• O Editor;</li> <li>• O Gestor de Recursos <i>AppStudio</i>;</li> <li>• Os Assistentes <i>AppWizard</i> e <i>ClassWizard</i>;</li> <li>• O <i>SourceBrowser</i>, o Depurador, o Compilador e o <i>Linker</i>.</li> </ul> </li> <li>- Reconhecer a importância da biblioteca MFC na escrita de programas para Windows.</li> <li>- Criar programas simples em Visual C++.</li> </ul>	<ul style="list-style-type: none"> <li>• O Ambiente de Programação do Visual C++</li> <li>• O Editor de Visual C++</li> <li>• Assistentes do Visual C++ e do <i>App Studio</i></li> <li>• Programação com o Visual C++</li> <li>• O Processo de construção do Visual C++</li> <li>• A biblioteca MFC</li> <li>• Escrita de um programa em Visual C++</li> </ul>	<p>O professor deverá realizar programas simples que clarifiquem a necessidade e a potencialidade da utilização da biblioteca MFC na construção de programas em ambiente Windows.</p> <p>Deverá ainda insistir na diversidade de exemplos, acompanhando de perto a evolução dos alunos na compreensão dos exemplos indicados. O grau de dificuldade dos exemplos deverá crescer progressivamente.</p> <p>O professor deverá também levar os alunos a construir pequenas aplicações para Windows utilizando o <i>AppWizard</i>.</p> <p>Poderá, ainda, apresentar exercícios de modificação de interface de programas já construídos com a ajuda, por exemplo, do editor de recursos.</p>	<p>3</p>

### SUBUNIDADE 1.3. – FERRAMENTAS DO VISUAL C++

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Criar projectos em Visual C++.</li> <li>- Utilizar o <i>AppStudio</i> para gerir recursos, incluindo mapas de bits e menus.</li> <li>- Construir aplicações em Visual C++.</li> <li>- Criar gestores de mensagens utilizando o <i>ClassWizard</i>.</li> <li>- Utilizar o depurador de Visual C++ para eliminar erros nas aplicações desenvolvidas.</li> </ul>	<ul style="list-style-type: none"> <li>• Projectos em Visual C++</li> <li>• Adição de um Mapa de bits com o <i>App Studio</i></li> <li>• Construção de uma aplicação em Visual C++</li> <li>• Edição de Menus com o <i>App Studio</i></li> <li>• Criação de gestores de mensagens com o <i>Class Wizard</i></li> <li>• Escrita de código para gestores de mensagens</li> <li>• O depurador do Visual C++</li> </ul>	<p>O professor deverá levar os alunos a construir pequenas aplicações para Windows utilizando o <i>AppWizard</i>.</p> <p>Poderá, ainda, apresentar exercícios de modificação de interface de programas já construídos, com a ajuda, por exemplo, do editor de recursos.</p> <p>O professor deverá iniciar o aluno na utilização de projectos como a unidade fundamental de desenvolvimento de aplicações no Visual C++.</p> <p>Deverá também, nesta unidade, ser realizado um estudo mais detalhado de algumas classes importantes da biblioteca MFC, especialmente as incluídas nos projectos gerados com o <i>AppWizard</i>.</p> <p>Para o efeito do exposto no parágrafo anterior, estas classes deverão ser incluídas em projectos apresentados aos alunos e o papel das funções pertencentes aos objectos instanciados das classes deverá ser explicado em detalhe.</p> <p>Exemplos de criação de gestores de mensagens com o <i>ClassWizard</i> devem também ser fornecidos e explicados aos alunos.</p> <p>O funcionamento do depurador, como elemento importante na eliminação de erros dos programas, deve ser alvo de alguma atenção. O executar, passo a passo, de programas previamente concebidos e a eliminação de erros previamente colocados em programas podem ser boas alternativas de verificação do funcionamento deste importante utilitário.</p>	3

**Unidade de Ensino/Aprendizagem 2**  
**PROGRAMAÇÃO DE “INTERFACES” EM VISUAL C++**

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Reconhecer a versatilidade da hierarquia de classes MFC.</li> <li>- Criar classes personalizadas para realização de tarefas específicas.</li> <li>- Criar e implementar recursos para menus.</li> <li>- Compreender o sistema de mensagens do Windows.</li> <li>- Distinguir as diversas mensagens relacionadas com a utilização do rato e do teclado.</li> <li>- Criar e personalizar barras de ferramentas e de estado.</li> <li>- Criar e aplicar caixas de diálogo.</li> <li>- Desenhar e utilizar controlos em caixas de diálogo.</li> </ul>	<ul style="list-style-type: none"> <li>• Programação com classes MFC</li> <li>• Os menus</li> <li>• Rato, Teclado e Mensagens</li> <li>• Barras de Estado e Barras de Ferramentas</li> <li>• Controlos</li> <li>• Caixas de Diálogo</li> </ul>	<p>Nesta unidade, o professor deverá começar por indicar um quadro geral de classes de aplicação geral da biblioteca MFC.</p> <p>Deverão ser apresentados exercícios, de grau crescente de dificuldade, da personalização de classes em situações concretas de desenvolvimento de aplicações.</p> <p>Os alunos deverão ser encaminhados na criação e utilização de menus em aplicações.</p> <p>Deve ser realizada uma descrição tão lata quanto possível do sistema de mensagens típico da API do Windows.</p> <p>Deve-se distinguir entre a armazenagem de dados em memória e em disco, introduzindo a noção de ficheiro.</p> <p>Os conceitos de registo e ficheiro deverão ser abordados nesta unidade de uma forma simples.</p>	<p>15</p>

### Unidade de Ensino/Aprendizagem 3

## IMPRESSÃO, GRÁFICOS, SOM E VÍDEO EM VISUAL C++

### SUBUNIDADE 3.1. – IMPRESSÃO

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Criar programas com impressão, usando para o efeito a API do Windows.</li> <li>- Utilizar o assistente <i>AppWizard</i> na criação de programas de pré-visualização de impressão.</li> <li>- Reconhecer a importância da Classe <i>View</i> e suas funções na impressão em Windows.</li> <li>- Usar a Classe <i>CprintInfo</i> e respectivas funções na obtenção de informações sobre o trabalho de impressão em curso.</li> <li>- Utilizar o objecto <i>Cpreview DC</i> na pré-visualização de impressões em aplicações Visual C++.</li> <li>- Utilizar objectos da Classe <i>CprintDialog</i> para configurar e parametrizar impressões.</li> </ul>	<ul style="list-style-type: none"> <li>• Suporte de impressão na API do Windows</li> <li>• A impressão com MFC</li> <li>• A Classe <i>CprintInfo</i></li> <li>• A arquitectura de pré-visualização</li> <li>• A Classe <i>CprintDialog</i></li> </ul>	<p>Nesta unidade, é importante dar uma ênfase especial à criação de um Projecto que inclua a pré-visualização e impressão em Visual C++.</p> <p>A mesma impressão, a título exemplificativo, deverá ser considerada usando a API do Windows e alternativamente com o uso das classes MFC, alertando para as diferenças na programação da saída na impressora, entre as duas formas.</p> <p>Todas as funções definidas nas classes MFC importantes para a impressão e pré-visualização da impressão deverão ser analisadas e utilizadas em exemplos.</p>	3

## SUBUNIDADE 3.2. - GRÁFICOS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Compreender Mapas de bits do Windows, dependentes e independentes de dispositivo.</li> <li>- Carregar, guardar, copiar e apresentar Mapas de Bits dependentes do dispositivo.</li> <li>- Apresentar e mover caracteres sobre o fundo.</li> <li>- Criar sequências de animação complexas, utilizando animação passo a passo e <i>sprites</i>.</li> <li>- Compreender a arquitectura de Mapas de Bits independentes do dispositivo e a forma de os criar e utilizar.</li> <li>- Evitar cintilações e cortes em operações de cópia de mapas de bits</li> <li>- Utilizar gráficos vectoriais.</li> </ul>	<ul style="list-style-type: none"> <li>• Mapas de Bits</li> <li>• Mapas de Bits dependentes do dispositivo ( DDB)</li> <li>• Animação com <i>Sprites</i></li> <li>• Utilização de molduras de <i>buffers</i> em programas de animação</li> <li>• Rectângulos delimitadores de sequências de animação</li> <li>• Mapas de Bits independentes de dispositivos (DIB)</li> <li>• Paletas de cor</li> <li>• Criação e utilização de DIB´s</li> </ul>	<p>O professor deverá começar por abordar a interface gráfica do Windows GDI como parte fundamental da geração de gráficos por programação.</p> <p>Deverão ser apresentados exercícios com utilização de Mapas de Bits em Visual C++.</p> <p>Deverá também ser explicada a diferença entre Mapas de Bits dependentes e independentes de dispositivo.</p> <p>É importante que exemplos de animação com texto e animações complexas com <i>sprites</i> sejam explicados aos alunos, assim como exemplos de utilização de gráficos vectoriais.</p>	<p style="text-align: center;">6</p>

### SUBUNIDADE 3.3. - SOM E VÍDEO

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Reproduzir ficheiros de som e vídeo utilizando a API do Windows.</li> <li>- Utilizar funções da classe <i>Cwave</i> para reproduzir som.</li> <li>- Utilizar funções da classe <i>Cmidi</i> para reproduzir som MIDI.</li> <li>- Utilizar a Interface de Controlo Multimédia disponível na livraria <i>CCAudio</i>.</li> </ul>	<ul style="list-style-type: none"> <li>• A classe <i>Cwave</i></li> <li>• Execução de um ficheiro de som usando a API do Windows</li> <li>• A classe <i>Cmidi</i></li> <li>• A livraria de Classes <i>CCAudio</i></li> <li>• A interface de Controlos Multimédia</li> </ul>	<p>Os conteúdos desta unidade devem ser abordados de um modo superficial, utilizando as API do Windows e a MFC.</p> <p>Os exemplos devem estar previamente construídos, esclarecendo os alunos acerca das funções utilizadas.</p>	4

## Unidade de Ensino/Aprendizagem 4

### PROGRAMAÇÃO DE BASES DE DADOS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Compreender o conceito de ODBC.</li> <li>- Aplicar e configurar o ODBC em aplicações Visual C++ com bases de dados.</li> <li>- Utilizar Classes <i>Cdatabase</i>, <i>Recordset</i> e outras no acesso a Bases de Dados em Visual C++.</li> <li>- Realizar operações sobre Bases de Dados como: Filtrar, Ordenar, Salvar e Eliminar registos assim como deslocar-se na Base de Dados.</li> <li>- Realizar Consultas parametrizadas.</li> <li>- Utilizar objectos DAO para aceder a Bases de Dados Microsoft Access.</li> </ul>	<ul style="list-style-type: none"> <li>• ODBC               <ul style="list-style-type: none"> <li>○ Configuração do ODBC</li> <li>○ A Classe <i>COBDCDynamic</i>;</li> <li>○ A Utilização do ODBC para consulta dinâmica de dados</li> </ul> </li> <li>• Classes MFC de bases de dados               <ul style="list-style-type: none"> <li>○ As Classes <i>Cdatabase</i> e <i>Recordset</i></li> <li>○ Utilização das Classes MFC para utilização de Bases de Dados</li> <li>○ Operações sobre registos em Visual C++</li> </ul> </li> <li>• Programação de Bases de dados DAO               <ul style="list-style-type: none"> <li>○ A hierarquia DAO</li> <li>○ Os objectos DAO</li> <li>○ As Classes <i>CDAODatabase</i>, <i>CDWorkspace</i> e <i>CDAORecordset</i></li> </ul> </li> <li>• Criação de um pequeno Projecto de gestão de bases de dados</li> </ul>	<p>Exercícios simples de acesso a Bases de Dados em geral através do ODBC e em particular com classes MFC com objectos DAO no caso das Bases de Dados em MS-ACCESS.</p> <p>Um Projecto de demonstração de acesso a uma Base de Dados deverá ser apresentado aos alunos, sintetizando os conteúdos desta unidade.</p>	8

**Unidade de Ensino/Aprendizagem 5**  
**LINGUAGEM DE PROGRAMAÇÃO PARA A WEB**

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> <li>- Entender os mecanismos de comunicação de uma linguagem de programação da Web com uma Base de Dados.</li> <li>- Aplicar uma linguagem de <i>scripting</i> ou outra para construção de HTML dinâmico na Web.</li> <li>- Construir um pequeno Projecto de uma aplicação de Base de dados na Internet com uma linguagem apropriada.</li> </ul>	<ul style="list-style-type: none"> <li>• Estudo de uma linguagem de programação a ser utilizada na geração de HTML dinâmico</li> <li>• Aplicação num pequeno Projecto de HTML dinâmico com suporte de uma plataforma de Base de Dados</li> </ul>	<p>Dá-se total liberdade de escolha nesta unidade ao professor, no sentido de abordar uma linguagem das mais utilizadas ou mais conhecidas de geração de HTML dinâmico, tal como páginas ASP, linguagem de <i>scripting</i> PHP, PERL e outras como JAVA e VISUAL C++.</p> <p>Deverá ser proposto aos alunos, na parte final da unidade, um pequeno projecto de exploração de geração de HTML dinâmico, suportado por uma Base de Dados.</p> <p>Os temas deste Projecto podem estar relacionados com eventuais necessidades de gestão da informação das escolas neste domínio.</p>	20

## 4. BIBLIOGRAFIA

### 10º Ano

- Aguiar, L. (1988). *Fundamentos de programación. Algoritmos y estructura de datos*. Madrid: McGraw-Hill.
- Agustín, G. (1987). *Programación estructurada. Diseño, Validación y Transformación de Programas*. Córdoba-Argentina: SEPA.
- Alonso, M. & Rumeu, S. (1992). *Metodología de la Programación. Programación Estructurada*. Madrid: Editorial Paraninfo.
- Farrer, H. *et al.* (1985). *Algoritmos Estructurados*. Rio de Janeiro: Editora Guanabara.
- Guerreiro, P. (1992). *Pascal Técnicas de Programação*. Lisboa: FCA.
- Guimarães & Lages (1988). *Algoritmos e Estrutura de Dados*. Rio de Janeiro: Livros Técnicos e Científicos Editora –AS.
- Hoffman, M. *et al.* (1988). *Algorítmica*. Barcelona: Marcombo.
- Leestma, S. & Nyhoff, L. (1988). *Programación en Pascal*. Madrid: Prentice Hall Iberia.
- Lima, J. (1993). *Programação de Computadores*. Porto: Porto Editora.
- Llorca F. *et al.* (1992). *Programación com el Lenguage Pascal*. Madrid: Paraninfo.
- Lopes, J. (1989). *Técnicas da Programação*. Lisboa: Editorial O Livro.
- Lovelle, J. *et al.* (1994). *Introducción a la Programación Estructurada y Orientada a Objectos com Pascal*. Oviedo: Autor.
- O'Brien, S. & Nameroff, S. (1996). *Turbo Pascal 7: Manual de Referência*. Madrid: McGraw-Hill.
- Pinto, W. (1990). *Introdução ao Desenvolvimento de Algoritmos e Estruturas de Dados*. São Paulo: ÉRICA Editora.
- Schemitz, E. *et al.* (1985). *Pascal e Técnicas de Programação*. Rio de Janeiro: Livros Técnicos e Científicos Editora –SA.
- Tremblay, J. & Bunt, R. *Ciência dos computadores - uma abordagem algorítmica*. Madrid: McGraw-Hill.

## 11º Ano

- Campos, L., Vilar, S. & Lúcio, L. (1999). *Programação em Visual Basic 6*. Lisboa: FCA.
- Connell, J. (1998). *Beginning Visual Basic 6 Database Programming*. Wrox Press.
- Curland, M. & Storage, B. (1999). *Advanced Visual Basic 6*. Addison Wesley.
- Ferreira, J. (2001). *Técnicas Avançadas em Visual Basic 6 - Curso Completo*. Lisboa: FCA.
- Fox, D. (1999). *Pure Visual Basic 6: QUE*.
- Nina, N. (1999). *Visual Basic 6 Curso Completo*. Lisboa: FCA.
- Schneider, D. (1999). *An Introduction to Programming with Visual Basic 6.0*. US Imports & PHIPES.

## 12º Ano

- Andrews, M. (1998). *Aprenda Visual C++ Já* . Lisboa: McGraw-Hill, Microsoft Press.
- Bates, J. & Tompkins, T. (1999). *Microsoft Visual C++ 6*. Madrid: Prentice Hall.
- D'Andrea, E. (1999). *Visual C++ 6. Curso de Iniciación*. Barcelona: INFORBOOK'S.
- Damas, L. (1999). *Linguagem C*. Lisboa: FCA.
- Guerreiro, P. (2000). *Programação com Classes em C++*. Lisboa: FCA.
- Kruglinski, D., Shepherd & Wingo, S. (1999). *Programación avanzada con Microsoft Visual C++ 6.0*. Madrid: McGraw-Hill, Microsoft Press.
- Leinecker, R. & Archer, T. (1999). *La biblia de Microsoft Visual C++ 6*. Madrid : ANAYA Multimédia.
- Pappas, C. & Murray, W. (1999). *Visual C++ 6.0. Manual de Referência*. Madrid: Osborne McGraw Hill.
- Rodrigues, P., Pereira, P. & Sousa, M. (1998). *Programação em C++. Conceitos Básicos e Algoritmos*. Lisboa: FCA.
- Rodrigues, P., Pereira, P. & Sousa, M. (1998). *Programação em C++. Algoritmos e Estruturas de Dados*. Lisboa: FCA.
- Sphar, C. (1999). *Aprenda Microsoft Visual C++ 6.0 Ya* . Madrid: McGraw-Hill, Microsoft Press.
- Zaration, B. (1999). *Microsoft Visual C++ 6.0. Manual do Programador*. Madrid: McGraw Hill, Microsoft Press.