

Ministério da Educação
Departamento do Ensino Secundário

CURSO TECNOLÓGICO DE INFORMÁTICA

BASES DE PROGRAMAÇÃO Projecto de Programa

10º ANO e 11º ANO

Autores

Luís Guerra (Coordenador)

Francisco Simões

Manuel C. Pinto

Manuela Silva

Luísa Rodrigues

Maio 2001

A versão agora apresentada refere-se à *Apresentação Geral do Programa* e ao *Desenvolvimento* dos 10º e 11º anos, uma vez que a anteriormente posta à discussão na secção de *Apresentação Geral*, incluía apenas o 10º ano. O *Desenvolvimento* do 12º ano será apresentado em breve.

ÍNDICE

1. INTRODUÇÃO	4
2. APRESENTAÇÃO DO PROGRAMA	5
2.1. FINALIDADES	5
2.2. OBJECTIVOS.....	6
2.3. VISÃO GERAL DOS CONTEÚDOS.....	7
2.4. SUGESTÕES METODOLÓGICAS GERAIS.....	8
2.5. COMPETÊNCIAS A DESENVOLVER.....	9
2.6. AVALIAÇÃO	10
2.7. RECURSOS.	11
3. DESENVOLVIMENTO DO PROGRAMA	11
4. BIBLIOGRAFIA	27

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 como 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.

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 adquiridas.
- 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.

2.2. OBJECTIVOS

São 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.
- Respeitar a propriedade intelectual

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, PHP, ASP, 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 pretendidas.

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 projecto 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

São 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.
- Decidir pela melhor das soluções na realização 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 / PHP / ASP / PERL
- Adaptar-se à evolução das linguagens.

2.6. AVALIAÇÃO

Para cada Unidade Temática, o professor deve ter em atenção os diferentes domínios:

- Cognitivos
 - Testes
 - Fichas de Trabalho
 - Progressão na aprendizagem
 - Demonstração oral de conhecimentos
- Atitudes e valores
 - Comportamento
 - Assiduidade
 - Motivação e interesse
- Aptidões e capacidades
 - Domínio da Língua Portuguesa
 - Destreza com o equipamento
 - Trabalhos de grupo e individuais
 - Organização e método

A avaliação será portanto contínua, registando a evolução do aluno aula a aula e permitindo ultrapassar de imediato qualquer dificuldade.

2.7. RECURSOS.

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

- computadores (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);

3. DESENVOLVIMENTO DO PROGRAMA

3.1. 10º ano

MÓDULO INICIAL

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

Unidade de Ensino/Aprendizagem 1

NOÇÕES BÁSICAS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> - Distinguir as diferentes fases na resolução de um problema. - Distinguir algoritmo de programa. - Distinguir os diferentes tipos de dados. - Reconhecer os diferentes operadores e as regras de prioridade. - Utilizar variáveis e construir expressões. - Construir sequências lógicas de operações em linguagem informal, na resolução de problemas. 	<ul style="list-style-type: none"> • Resolução de problemas • Noções de Algoritmo e Programa • Tipos de dados e operações elementares • Prioridade de operadores • Variáveis e expressões • Construção de algoritmos em linguagem informal 	<p>O professor deverá introduzir a unidade através de exemplos do dia a dia, simples de compreender, onde 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"> - Utilizar fluxogramas na construção de algoritmos. - Utilizar pseudocódigo na construção de algoritmos. - Utilizar as instruções básicas da linguagem Pascal na codificação de algoritmos. - Aplicar estruturas de decisão na elaboração de algoritmos. - Decompor um problema nas suas partes (estruturas) elementares (TOP-DOWN). - Desenvolver programas em Pascal, de forma estruturada e documentada. 	<ul style="list-style-type: none"> • Simbologia <ul style="list-style-type: none"> • Fluxogramas • Pseudocódigo • Noções Básicas de Pascal • Programação Estruturada <ul style="list-style-type: none"> • Estruturas de decisão • Desenvolvimento Estruturado • Elaboração de algoritmos e subalgoritmos e sua tradução para Pascal • Teste de algoritmos/Pascal 	<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 DATA-SHOW.</p> <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 um grande ênfase à explicação da tradução algoritmo/Pascal.</p>	20

Unidade de Ensino/Aprendizagem 3

ESTRUTURAS DE DADOS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none">- Identificar as estruturas de dados.- Aplicar as estruturas de dados na elaboração de algoritmos e sua tradução para pascal.- Aplicar os conceitos de registo e de ficheiro em Pascal.	<ul style="list-style-type: none">• Cadeia de Caracteres• Vectores e Matrizes• Registos• Ficheiros	<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 PROGRAMAÇÃO ORIENTADA A OBJECTOS (POO)

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
- Conhecer e compreender os conceitos básicos da POO independentemente da linguagem	<ul style="list-style-type: none">• Conceitos básicos<ul style="list-style-type: none">- Classes- Objectos- Mensagens- Atributos- Métodos- Polimorfismo- Herança	O professor deverá explicar sucintamente os conceitos envolvidos, comparando a programação procedural com a programação orientada a objectos	2

Unidade de Ensino/Aprendizagem 2

INTRODUÇÃO AO AMBIENTE DE DESENVOLVIMENTO

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> - Identificar os elementos básicos constituintes do ambiente de desenvolvimento - Explicar a diferença entre “Design Time” e “Run Time” - Conhecer o conceito de “Event-driven Programming” - Conhecer qual a finalidade de um “Project File” 	<ul style="list-style-type: none"> • O ambiente de trabalho do Visual Basic <ul style="list-style-type: none"> - “Project explorer” - “Properties Windows” - “Toolbox” - “Menu Bar” - etc. 	<p>O professor deverá explicar o ambiente mostrando os seus elementos com a ajuda de um data-show</p> <p>Deverá ainda, com a ajuda de pequenas aplicações, expor os vários conceitos básicos apresentados.</p>	2

Unidade de Ensino/Aprendizagem 3
CONCEITOS FUNDAMENTAIS DE VISUAL BASIC

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> - Criar aplicações simples - Conhecer alguns dos objectos, métodos, propriedades e eventos - Conhecer algumas propriedades e eventos associados a um formulário - Utilizar algumas das propriedades das "Text Boxes", "Command Buttons" e "Labels" 	<ul style="list-style-type: none"> • Objectos Básicos e outras características básicas da linguagem Visual Basic e respectivo ambiente de trabalho 	<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>	6

Unidade de Ensino/Aprendizagem 4
EDIÇÃO DE CÓDIGO E FORMULÁRIOS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> - Identificar “Form Modules”, “Standard Modules” e “Class Modules” - Utilizar o “Code Editor” - Criar “Messages Boxes” - Usar constantes e argumentos do Visual Basic - Conhecer a diferença entre declarações “Unload” e “Load” - Conhecer a diferença entre “Model” e “Modeless” - Utilizar os métodos “Show e “Hide - Aplicar a declaração “End” 	<ul style="list-style-type: none"> • “Code Editor” • “Message Boxes” • Os diferentes tipos de módulos em VB • Introdução a alguns dos métodos, constantes, declarações e argumentos • Conceitos básicos 	<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>	8

Unidade de Ensino/Aprendizagem 5

VARIÁVEIS E PROCEDIMENTOS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> - Aplicar os vários tipos de dados na declaração de variáveis - Distinguir entre variáveis Públicas e Privadas - Conhecer a diferença entre constantes e variáveis - Conhecer a diferença entre procedimentos e funções - Criar funções com argumentos e retorno de valores - Usar funções do Visual Basic para manipulação de "Strings" 	<ul style="list-style-type: none"> • Declaração de variáveis, constantes, procedimentos e funções 	<p>O professor deverá exibir pequenos exemplos demonstrativos acompanhando-os de uma breve descrição dos vários conceitos envolvidos</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>	10

Unidade de Ensino/Aprendizagem 6
CONTROLO DA EXECUÇÃO DE UM PROGRAMA

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> - Conhecer os operadores lógicos e de comparação - Usar a declaração "If ... Else" - Usar a declaração "Select Case" - Usar a declaração "Do ... Loop" ou "While ... Wend" - Usar a declaração "For ... Next" 	<ul style="list-style-type: none"> • Declarações para controlo da execução de um programa 	<p>O professor deverá mostrar pequenos exemplos demonstrativos acompanhando-os de um breve descrição dos vários conceitos envolvidos</p> <p>Os alunos deverão criar pequenas aplicações semelhantes aos exemplos apresentados</p>	6

Unidade de Ensino/Aprendizagem 7

CONTROLO, DETECCÃO E TRATAMENTO DE ERROS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none">- Utilizar "Break Points"- Visualizar o valor das variáveis na "Watch Window"- Distinguir entre os modos "Design", "Run" e "Debug"- Executar passos a passo programas utilizando "Call Stack"	<ul style="list-style-type: none">• Ferramentas de controlo e detecção de erros	<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>	4

Unidade de Ensino/Aprendizagem 8

CONTROLOS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
- Utilizar os Standard controlos - Adicionar controlos ActiveX a um projecto e utilizá-los num programa .	• Standard controlos e ActiveX controlos	Os alunos deverão criar uma aplicação que envolva a utilização de novos controlos	4

Unidade de Ensino/Aprendizagem 9

VALIDAR A INTRODUÇÃO DE DADOS

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none">- Criar aplicações de validação de dados- Criar aplicações que usem o "Mask Edit Control"	<ul style="list-style-type: none">• Validação de dados• Função "InputDialog"	<p>O professor deverá transmitir 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>	6

Unidade de Ensino/Aprendizagem 10
INTERFACE COM O UTILIZADOR

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none"> - Criar menus e submenus utilizando o "Menu Editor" - Conhecer as propriedades que podem ser usadas no "Menu Editor" - Criar um "ToolBar" usando o "ToolBar Control" - Compreender os eventos associados ao dispositivo apontador em acções de "Drag" e "Drop" 	<ul style="list-style-type: none"> • "Menu Editor" • "ToolBar Control" • Implementação do "Drag" e "Drop" numa aplicação 	<p>O professor deverá exemplificar as diferentes ferramentas existentes, através do data show.</p> <p>O aluno poderá melhorar o aspecto gráfico das aplicações criadas anteriormente, desenvolvendo um ambiente mais amigável para o utilizador.</p>	6

Unidade de Ensino/Aprendizagem 11

ACESSO A DADOS USANDO O “ADO DATA CONTROL”

Objectivos	Conteúdos	Sugestões Metodológicas	Nº aulas (90 min.)
<ul style="list-style-type: none">- Utilizar o “ADO Data Control” para visualizar registos numa base de dados- Utilizar o “ADO Data Control” para procurar, apagar, adicionar e modificar registos- Utilizar o “Data Environment Designer	<ul style="list-style-type: none">• “ADO Data Control”• “Data Environment Designer”	<p>O professor deverá exemplificar as diferentes ferramentas existentes, através do data show.</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>	12

4. BIBLIOGRAFIA

Aguilar, 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.

Pascal Técnicas de Programação, Lisboa, Lidel.

Farrer, H. et al (1985). *Algoritmos Estructurados*. Rio de Janeiro, Editora Guanabara.

Guimarães, & Lages (1988). *Algoritmos e Estructura de Dados*, Rio de Janeiro, Livros Técnicos e Científicos Editora –AS.

Hoffman, M. et al (1988). *Algorítmica*, Barcelona, Marcombo.

Leestma, S. & Nyhoff, L. (1988). *Programacion 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 Programción Estructurada y Orientada a Objectos com Pascal.*, Oviedo, Edição dos autores

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:* McGraw-Hill.