

Delphi 7



<< Iniciante >>

Apresentação

Meu nome é Albert Eije. Já trabalho com informática desde 1993. Primeiramente trabalhei com computação gráfica. Com o tempo “migrei” para a área de desenvolvimento. De início utilizei o DBase/Clipper. Velhos tempos. Depois que apareceu o fascinante Delphi. Já desenvolvi vários sistemas comerciais em Delphi utilizando os mais diversos bancos de dados, tais como Paradox, Access, Interbase, Firebird, MySQL, SQL Server, PostGreSQL e Oracle.

O objetivo desse curso é fazer o alicerce para quem quer aprender a programar. Muitas pessoas acham que basta abrir o Delphi e saber usar seus componentes. Mas isso não basta! Um bom programador precisa ter uma boa base de Lógica de Programação e vamos dedicar dois módulos a este assunto. Outra coisa importante é entender a estrutura do Object Pascal, a linguagem utilizada pelo Delphi. Conheço vários programadores, até bons, que não sabem declarar uma constante. Não sabem trabalhar com vetores e matrizes e quando pergunto o porquê, dizem que nunca precisaram usar esses recursos em seus sistemas. Imagino quantas linhas a mais de programação gastaram por não os terem usado.

Uma outra coisa importante para um programador que deseja fazer sistemas comerciais é a questão dos bancos de dados. Muitos programadores não tem noção de como elaborar um banco de dados, sendo que este é um dos primeiros passos ao se iniciar o desenvolvimento de um sistema. Por isso, vamos aprender quais os tipos de bancos de dados e como selecionar o melhor para o nosso caso. Depois vamos aprender a criar nossas tabelas de forma que quase não exista redundância dos dados.

Finalmente criaremos um programa no Delphi utilizando o banco de dados Paradox. A escolha por esse banco se deu pelo fato da facilidade de trabalhar com o mesmo e porque ele já vem com o Delphi. Vamos também aprender a fazer qualquer tipo de relatório usando o QuickReport. A versão do Delphi utilizada no curso é a 7, mas os conhecimentos adquiridos podem ser aplicados em qualquer versão do Delphi (anterior ou posterior).

Boa parte do material aqui visto foi adquirido na própria internet. Meu trabalho foi compilá-lo em ordem para um melhor aprendizado e acrescentar pontos úteis.

Junto com os módulos irão arquivos e programas necessários para o aprendizado do mesmo.

Bem vindos e mãos a obra!

ÍNDICE

Módulo 01 - Lógica de Programação I

Módulo 02 - Lógica de Programação II

Módulo 03 - Object Pascal I

Módulo 04 - Object Pascal II

Módulo 05 - Delphi - Ambiente (IDE) e Criação de Programas I

Módulo 06 - Delphi - Ambiente (IDE) e Criação de Programas II

Módulo 07 - Bancos de Dados

Módulo 08 - Paradox, BDE e Database Desktop

Módulo 09 - Programa para Controle de Cheques

Módulo 10 - Criação de Relatórios - Quick Report

Apêndice

Módulo 1

Lógica de Programação I

INTRODUÇÃO

A automatização de tarefas é um aspecto marcante da sociedade moderna. O aperfeiçoamento tecnológico alcançado, com respeito a isto, teve como elementos fundamentais a análise e a obtenção de descrições da execução de tarefas em termos de ações simples o suficiente, tal que pudessem ser automatizadas por uma máquina especialmente desenvolvida para este fim, O COMPUTADOR.

Em ciência da computação houve um processo de desenvolvimento simultâneo e interativo de máquinas (hardware) e dos elementos que gerenciam a execução automática (software) de uma dada tarefa. E essa descrição da execução de uma tarefa, como considerada acima, é chamada algoritmo.

O objetivo dessa lição é a Lógica de Programação dando uma base teórica e prática, suficientemente boa, para que o aluno domine os algoritmos e esteja habilitado a aprender uma linguagem de programação. Será mostrado também um grupo de algoritmos clássicos para tarefas cotidianas, tais como: ordenação e pesquisa.

DEFINIÇÃO DE ALGORITMO

O conceito central da programação é o conceito de algoritmos, isto é, programar é basicamente construir algoritmos.

Algoritmo é a descrição, de forma lógica, dos passos a serem executados no cumprimento de determinada tarefa. O algoritmo pode ser usado como uma ferramenta genérica para representar a solução de tarefas independente do desejo de automatizá-las, mas em geral está associado ao processamento eletrônico de dados, onde representa o rascunho para programas (Software). Serve como modelo para programas, pois sua linguagem é intermediária à linguagem humana e às linguagens de programação, sendo então, uma boa ferramenta na validação da lógica de tarefas a serem automatizadas.

Um algoritmo é uma receita para um processo computacional e consiste de uma série de operações primitivas, interconectadas devidamente, sobre um conjunto de objetos. Os objetos manipulados por essas receitas são as variáveis.

É a forma pela qual descrevemos soluções de problemas do nosso mundo, a fim de serem implementadas utilizando os recursos do mundo computacional. Como este possui severas limitações em relação ao nosso mundo, exige que sejam impostas algumas regras básicas na forma de solucionar os problemas, para que possamos utilizar os recursos de hardware e software disponíveis. Pois, os algoritmos, apesar de servirem para representar a solução de qualquer problema, no caso do Processamento de Dados, eles devem seguir as regras básicas de programação para que sejam compatíveis com as linguagens de programação.

LINGUAGEM DE DESCRIÇÃO DE ALGORITMO (LDA)

Para escrevermos algoritmos é preciso uma linguagem clara e que não deixe margem a ambigüidades, para isto devemos definir uma sintaxe e uma semântica, de forma a permitir uma única interpretação das instruções num algoritmo.

Estrutura de um algoritmo

```
Algoritmo Nome_Do_Algoritmo
variáveis
Declaração das variáveis
Procedimentos
Declaração dos procedimentos
Funções
Declaração das funções
Início
Corpo do Algoritmo
Fim
```

Exemplo de um algoritmo cujo objetivo é usar um telefone público.

```
Início
Tirar o fone do gancho;
Ouvir o sinal de linha;
Introduzir o cartão;
Teclar o número desejado;
Se der o sinal de chamar
    Conversar;
    Desligar;
    Retirar o cartão;
Senão
    Repetir;
Fim.
```

Exercícios de Fixação

Um homem precisa atravessar um rio com um barco que possui capacidade de carregar apenas ele mesmo e mais uma de suas três cargas, que são: um leão, uma ovelha e um maço de capim. O que o homem deve fazer para atravessar o rio sem perder suas cargas?

Observação – Se a Ovelha ficar só, devora o capim.

Se o Leão ficar só, devora a ovelha.

Identificadores

São os nomes dados a variáveis, constantes e programas.

Regras Para construção de Identificadores:

- * Não podem ter nomes de palavras reservadas (comandos da linguagem);
- * Devem possuir como 1º caractere uma letra ou Underscore (_);
- * Ter como demais caracteres letras, números ou Underscore;
- * Ter no máximo 127 caracteres;
- * Não possuir espaços em branco;
- * A escolha de letras maiúsculas ou minúsculas é indiferente.

Variáveis

Unidades básicas de armazenamento das informações em nível de linguagens de programação. Os tipos de dados e variáveis utilizados dependem da finalidade dos algoritmos, mas, podemos definir alguns, pelo fato de serem largamente utilizados e implementados na maioria das linguagens, sendo estes:

- * INTEIRO (INTEGER): qualquer número inteiro, negativo, nulo ou positivo.
- * REAL (DOUBLE): qualquer número real, negativo, nulo ou positivo.
- * CHARACTER (STRING): qualquer conjunto de caracteres alfanuméricos.
- * LÓGICO (BOOLEAN): tipo especial de variável que armazena apenas os valores V e F, onde V representa VERDADEIRO e F FALSO.

Declaração de variáveis

Para que os programas manipulem valores, estes devem ser armazenados em variáveis e para isso, devemos declará-las de acordo com a sintaxe:

```
NomeVariável : tipo;
Ex: ValorTotal : Double;
```

Constantes

Constantes são endereços de memória destinados a armazenar informações fixas, inalteráveis durante a execução do programa.

Declaração de constantes

As constantes são eternamente iguais a determinados valores, portanto usamos o sinal de “=”.

Exemplos:

Pi = 3.1416;

Empresa = 'Nivek Informatica'

V = Verdadeiro

Operações Básicas

Na solução da grande maioria dos problemas é necessário que as variáveis tenham seus valores consultados ou alterados e, para isto, devemos definir um conjunto de OPERADORES, sendo eles:

OPERADOR DE ATRIBUIÇÃO

NomeDaVariavel := Valor ou Expressão Atribuída. (":=" é o operador de atribuição utilizado pelo Delphi).

OPERADORES ARITMÉTICOS

+ = Adição	Quociente = Quociente da divisão de inteiros
* = Multiplicação	Resto = Resto da divisão de inteiros
- = Subtração ou inversor do sinal	EXP(a,b) = Exponenciação ab
/ = Divisão	

FUNÇÕES PRIMITIVAS: SEN(x); COS(x); TG(x); ABS(x); INT(x); Raiz(x); PI();

OPERADORES RELACIONAIS

São utilizados para relacionar variáveis ou expressões, resultando num valor lógico (Verdadeiro ou Falso), sendo eles:

= - igual	<> - diferente
< - menor	> - maior
<= - menor ou igual	>= - maior ou igual

OPERADORES LÓGICOS

São utilizados para avaliar expressões lógicas, sendo eles:

* e (and) – e lógico ou conjunção

* ou (or) – ou lógico ou disjunção

* não (not) – negação.

PRIORIDADE DE OPERADORES

Durante a execução de uma expressão que envolve vários operadores, é necessária a existência de prioridades, caso contrário poderemos obter valores que não representam o resultado esperado.

A maioria das linguagens de programação utiliza as seguintes prioridades de operadores:

1º - Efetuar operações embutidas em parênteses "mais internos"

2º - Efetuar funções

3º - Efetuar multiplicação e/ou divisão

4º - Efetuar adição e/ou subtração

5º - Operadores relacionais

6º - Operadores lógicos

OBS: O programador tem plena liberdade para incluir novas variáveis, operadores ou funções para adaptar o algoritmo às suas necessidades, lembrando sempre de que estes devem ser compatíveis com a linguagem de programação a ser utilizada.

COMANDOS DE ENTRADA E SAÍDA

No algoritmo é preciso representar a troca de informações que ocorrerá entre o mundo da máquina e o nosso mundo, para isso devemos utilizar comandos de entrada e saída, sendo que, em nível de algoritmo, esses comandos representam apenas a entrada e a saída da informação, independente do dispositivo utilizado (teclado,

discos, impressora, monitor,...), mas, sabemos que nas linguagens de programação essa independência não existe, ou seja, nas linguagens de programação temos comandos específicos para cada tipo de unidade de Entrada/Saída.

Comando de Entrada de Dados

```
Leia(variável_1, variável_2, ...)
```

Comando de Saída de Dados

```
Imprima(expressão_1, expressão_2, ...)
```

COMANDOS DE CONTROLE DE FLUXO

Para representar a solução de um problema devemos escrever o conjunto de passos a serem seguidos, sendo que a maioria dos problemas exige uma dinâmica na sua solução, impondo assim que os algoritmos executem conjuntos de instruções de acordo com as possíveis situações encontradas no problema original. E de acordo com a Programação Estruturada os mecanismos utilizados para esse controle são: *Seqüência*, *Seleção* e *Repetição*.

SEQÜÊNCIA: usada para executar comandos passo a passo, sabendo que todos eles serão executados na ordem de escrita, sem nenhum desvio. Uma seqüência pode possuir um ou vários comandos, os quais devem ser delimitados pelos identificadores Início e Fim.

```
Inicio
Comando_1
...
Comando_n
Fim
```

CORPO GERAL DE UM PROGRAMA

```
PROGRAMA <<identificador>>;
CONST
    <<identificador>> = <<dado>>
VAR
    <<identificador>> : <<tipo>>;
ÍNICIO
{
COMANDOS DE ENTRADA, PROCESSAMENTO E SAÍDA
<<comando1>>;
<<comandoN>>
}
FIM.
```

; PONTO E VÍRGULA

O sinal de ponto e vírgula “;” indica a existência de um próximo comando (passa para o próximo). Na estrutura ÍNICIO e no comando que antecede a estrutura FIM não se usa “;”.

ALGORITMO UM

Segue um Algoritmo que lê o nome e as 4 notas bimestrais de um aluno. Em seguida o Algoritmo calcula e escreve a média obtida.

```
PROGRAMA MEDIA_FINAL;
VAR
NOTA1, NOTA2, NOTA3, NOTA4, MEDIA: REAL;
NOME : CARACTERE [35]
INICIO
    LEIA (NOME);
    LEIA (NOTA1, NOTA2, NOTA3, NOTA4);
    MEDIA := (NOTA1 + NOTA2 + NOTA3 + NOTA4) / 4;
    ESCREVA (NOME, MEDIA);
FIM.
```

ALGORITMO DOIS

Segue um Algoritmo que lê o raio de uma circunferência e calcula sua área.

```
PROGRAMA AREA_CIRCUNFERENCIA;  
CONST PI = 3.1416;  
VAR RAI0, AREA : REAL;  
INICIO  
  LER (RAIO); {PROCESSAMENTO}  
  AREA := PI * SQR(RAIO); {ENTRADA}  
  ESCREVA ('AREA = ', AREA); {SAÍDA}  
FIM.
```

{LINHAS DE COMENTÁRIO}

Podemos inserir comentários em um Algoritmo para aumentar a compreensão do mesmo, para isso basta que o texto fique entre Chaves “{}”.

Exemplo:

```
LER (RAIO); {ENTRADA}
```

'ASPAS SIMPLES'

Quando queremos exibir uma mensagem para a tela ou impressora ela deve estar contida entre aspas simples, caso contrário, o computador irá identificar a mensagem como Variável Indefinida.

Exemplo:

```
ESCREVER ('AREA OBTIDA = ', AREA) {COMANDO DE SAÍDA}  
AREA OBTIDA = X.XX {RESULTADO GERADO NA TELA}
```

SELEÇÃO: usada para tomar decisões, ou seja, desviar a execução do algoritmo de acordo com uma condição, podendo ser simples ou composta.

Simple	Composta
Se (Expressão Lógica) Então Seqüência_1	Se (Expressão Lógica) Então Seqüência_1 Senão Seqüência_2

ESTRUTURAS DE DECISÃO

Executa uma seqüência de comandos de acordo com o resultado de um teste. A estrutura de decisão pode ser Simple ou Composta, baseada em um resultado lógico.

Simple:	Composta 1:
SE <<CONDIÇÃO>> ENTÃO <<COMANDO1>>	SE <<CONDIÇÃO>> ENTÃO <<COMANDO1>> SENÃO <<COMANDO1>>

Composta 2:

```
SE <<CONDIÇÃO>> ENTÃO INICIO  
  <<COMANDO1>>;  
  <<COMANDON>>;  
FIM  
SENÃO INICIO  
  <<COMANDO1>>;  
  <<COMANDON>>;  
FIM;
```

ALGORITMO TRÊS

Segue um Algoritmo que lê 2 números e escreve o maior.

```

PROGRAMA ACHA_MAIOR;
VAR A, B : INTEIRO;
INICIO
  LEIA (A, B);
  SE A>B ENTÃO
    ESCRIVA (A)
  SENÃO
    ESCRIVA (B)
FIM.

```

ALGORITMO QUATRO

Segue um Algoritmo que lê o nome e as 4 notas bimestrais de um aluno. Em seguida o Algoritmo calcula e escreve a média obtida pelo aluno escrevendo também se o aluno foi aprovado ou reprovado.

Média para aprovação = 6.

```

PROGRAMA MEDIA_FINAL;
VAR
  NOTA1, NOTA2, NOTA3, NOTA4, MEDIA: REAL;
  NOME : CARACTERE [35]
INICIO
  LEIA (NOME);
  LEIA (NOTA1, NOTA2, NOTA3, NOTA4);
  MEDIA := (NOTA1 + NOTA2 + NOTA3 + NOTA4) / 4;
  SE MEDIA >= 6 ENTÃO
    ESCRIVA ('APROVADO')
  SENÃO
    ESCRIVA ('REPROVADO')
  ESCRIVA (NOME, MEDIA)
FIM.

```

NINHOS DE SE

Usados para tomadas de decisões para mais de 2 opções.

Forma Geral:

```

SE <<CONDIÇÃO>> ENTÃO
  <<COMANDO1>>
SENÃO SE <<CONDIÇÃO>> ENTÃO
  <<COMANDO1>>
SENÃO
  <<COMANDO1>>

```

ALGORITMO CINCO

Segue um Algoritmo que lê 3 números e escreve o maior.

```

PROGRAMA ACHA_MAIOR;
VAR A, B, C : INTEIRO;
INICIO
  LEIA (A, B, C);
  SE (A>B) E (A>C) ENTÃO
    ESCRIVA (A)
  SENÃO SE (B>A) E (B>C) ENTÃO
    ESCRIVA (B)
  SENÃO
    ESCRIVA (C)
FIM.

```

ESTRUTURAS DE CONDIÇÃO

A estrutura de condição equivale a um ninho de SE's.

Forma Geral:

```
FACA CASO
  CASO <<CONDIÇÃO1>>
    <<COMANDO1>>;
  CASO <<CONDIÇÃO2>>
    <<COMANDO1>>;
  OUTROS CASOS
    <<COMANDO1>>;
FIM DE CASO
```

ALGORITMO SEIS

Segue um Algoritmo que lê 3 números e escreve o maior.

```
PROGRAMA ACHA_MAIOR;
VAR A, B, C : INTEIRO;
INICIO
  LEIA (A, B, C);
  FACA CASO
    CASO (A>B) E (A>C)
      ESCREVA (A);
    CASO (B>A) E (B>C)
      ESCREVA (B);
    OUTROS CASOS
      ESCREVA (C);
  FIM DE CASO
FIM.
```

EXERCÍCIOS

- 01) Crie um Algoritmo não computacional, que troque um pneu de carro.
- 02) Implemente um algoritmo capaz de encontrar o maior dentre 3 números inteiros quaisquer. Suponha todos serem distintos.
- 03) Implemente um algoritmo que leia 3 números quaisquer e os imprima em ordem crescente.
- 04) Escreva um subprograma capaz de calcular a média aritmética de três parâmetros passados.
- 05) Crie um algoritmo que pegue o nome de três pessoas, pegue a data de nascimento delas e identifique o sexo das mesmas. Depois mostre os nomes dessas pessoas seguido pelo sexo e idade e diga no final do algoritmo qual dos três é o mais velho.
- 06) NÃO $2 ** 3 < 4 ** 2$ OU $ABS(INT(-15 / 2)) < 10$
- 07) $3 * (C / 4 + 5) < -8 * 3 + (15 \text{ MOD } 8 - 3)$ OU $5 ** 2 > INT(C * 0.7)$ onde $C = 20$
- 08) $A ** 3 / B + 5 - C * D > C * D + A - B$ OU $A // 2 / D < 18 - A$ onde $A = 9$, $B = 3$, $C = 4$ e $D = 2$
- 09) $5 + A * B ^ 3 - 16 // 4 - D$ E $6 / A * C / (A - B) = 234$ OU $A / 4 - (7 + 5 * C) < A ** 2 - 3 * B$
- 10) Faça um algoritmo para calcular o peso ideal de uma pessoa sabendo sua altura.
DADO: para homens ($72,7 * \text{altura}$) – 58;
para mulheres ($62,1 * \text{altura}$) – 44,7.
- 11) Faça um algoritmo para ler nove número inteiros quaisquer. Tirar a média aritmética dos três primeiros, depois a média dos outros três e por fim a média dos três últimos. Escreva as três médias e a média das três médias.
- 12) Faça um algoritmo para ler os catetos de um triângulo retângulo e calcular e imprimir a sua hipotenusa.
- 13) Faça um algoritmo para ler duas variáveis inteiras e trocar o conteúdo lido de uma pela outra.
- 14) Faça um algoritmo para ler dois números e imprimir o maior, o menor ou então dizer se são iguais.
- 15) Faça um algoritmo para ler três números e imprimir se estes podem ou não formar um triângulo.
Observação – Para formar os lados de um triângulo cada um dos valores tem que ser menor que a soma dos outros dois.
- 16) Faça um algoritmo que leia as três notas, as faltas e o nome de um aluno e imprima sua situação. (“APROVADO”, “REPROVADO POR FALTA” ou “REPROVADO POR MÉDIA”)
Observação – A média para aprovação é 5.0 e o limite de faltas é 17.
- 17) Uma pessoa precisa comprar 3 remédios. Porém tem somente R\$ 100,00. Faça um algoritmo que leia o nome e o preço de cada medicamento e escreva quais os medicamentos que ela pode comprar, se é que pode. (Para facilitar o algoritmo faça a compra por ordem de leitura.)
- 18) Faça um algoritmo que leia 9 números e escreva quantos números pares foram lidos.
- 19) Leia o nome e a idade de três pessoas e escreva seus nomes em ordem crescente de idade.
- 20) Leia três palavras e escreva se das palavras lidas, as três são diferentes, as três são iguais ou pelo menos duas são iguais.

Veja o Conteúdo Exclusivo do Curso Delphi Avançado



1 CD - 130 Minutos

- * Compreender o modelo Entidade-Relacionamento e Lógico-Relacional;
- * Saber definir e identificar: Entidades, Relacionamentos, Atributos e tudo mais que tenha a ver com o Modelo Entidade-Relacionamento;
- * Conhecer os tipos de relacionamentos: Condicionais e Incondicionais;
- * Saber os graus de relacionamento: Um-Para-Um, Um-Para-Muitos, Muitos-Para-Muitos;
- * Compreender uma realidade e a partir dessa compreensão criar um modelo de dados e logo depois o banco de dados em si;
- * Dominar o conceito de chaves, índices e integridade referencial;
- * Saber normalizar um modelo através das Formas Normais (1FN a 5FN);
- * Usar programas para a criação do modelo de dados;
- * Utilização do DBDesigner.



1 CD - 159 Minutos

- * Definição e História da Linguagem SQL;
- * Entendendo as partes da Linguagem: DDL, DML, DCL e Transactions Control;
- * Instrução Select (alias, order by, like, asc, desc, group by, having, where, not, in, exists, sum, max, min, avg, count, between, distinct, etc);
- * Instrução Insert Into;
- * Instrução Update;
- * Instrução Delete;
- * Consultas Encadeadas (SubQueries);
- * Junções (inner join, left join, right join e full outer join);
- * Criação, alteração e deleção de tabelas e índices;
- * Controle de Transações;
- * Diversos exemplos feitos no programa SQL Explorer;
- * Criação de um sistema no Delphi implementando os comandos SQL juntamente com o componente Query. Cadastro de Clientes, Cadastro de

Usuarios e Formulário Mestre Detalhe com duas tabelas (CVenda e DVenda).



2 CDs - 340 Minutos

- * Introdução: Definição, História, Características, Licenças e Versões;
- * Instalação da versão 3.23 no Windows 98, tipos de arquivos, estrutura de diretórios do MySQL, uso do Console (prompt), etc;
- * Utilização da ferramenta gráfica MySQLFront 2.3, criação de bancos de dados, tabelas, explicação dos tipos de campos, importação de dados do paradox, atributos especiais de campos do MySQL, uso do Manual de Referência, uso das funções do MySQL;
- * Instalação da Versão 4 do MySQL no Windows XP, uso do MySQLFront 3.2, importação de arquivos do Excel, Exportação de dados para o formato HTML, etc;
- * Instalação do MySQL no Linux Slacware, uso do console no Linux, uso da ferramenta gráfica PHPMyAdmin;
- * Instalação e configuração da versão 5 no Windows XP;
- * Uso da ferramenta gráfica DBManager, tipos de tabelas, foreign keys, constraints, views, backup e restauração de dados, controle de transações;

- * Uso da ferramenta gráfica MySQL Administrator, controle de usuários, backup e restauração de dados, replicação;
- * Uso da ferramenta gráfica MySQL Query Browser;
- * Uso da ferramenta gráfica MySQL Migration Toolkit, migração de um banco de dados do MS-Access;
- * DBDesigner - Sincronização de dados e engenharia reversa;
- * Apresentação dos sites oficiais do MySQL - Internacional e Nacional;
- * Stored Procedures e Triggers.

Curso Delphi Avançado - 15 Módulos (19 CDs)



2 CDs - 254 Minutos

- * Introdução: Definição, História, Características, Licenças e Especificações Firebird;
- * Comparativo das arquiteturas Classic x SuperServer;
- * Descrição dos tipos de dados suportados pelo Firebird;
- * Dialetos;
- * Tipos de Transações;
- * Instalação passo a passo da versão 1.5 do Firebird no Windows;
- * Explicação da estrutura de diretórios;
- * Serviços executados no Windows e opções no Painel de Controle;
- * Utilização das seguintes ferramentas gráficas: IBExpert, IBEasy+, FlameRobin, IBAccess, SQLHammer, DBak Win, IB First AID, Interbase Grant Manager;
- * Criação de tabelas, triggers, generatos, domains, views, queries;
- * Controle de acesso para usuários;
- * Backup do BD;

- * Constraints: Check, PK, FK, Unique;
- * Corrigindo BDs corrompidos;
- * Criação de um projeto de BD;
- * Utilização do ISQL (console);
- * Criação de triggers para controle de estoque;
- * Exceptions;
- * Criação de Stored Procedures;
- * UDFs - Conceito; Criação de uma UDF no Delphi e uso no Firebird;
- * Corrupção de Dados: Conhecendo, evitando, corrigindo;
- * Utilização do Gfix (console);
- * Comparativo: MySQL x Firebird;
- * Apresentação dos sites importantes internacionais e nacionais;
- * Instalação do Firebird Cliente numa máquina virtual Windows 98;
- * Conexão com o banco de dados através da rede;
- * Apresentação da documentação.



1 CD - 116 Minutos

- * Definição e estrutura das empresas;
- * Níveis de decisão das empresas;
- * O Analista de Sistemas;
- * Engenharia de Software;
- * Ciclo de vida de um sistema;
- * Coleta de Informações: Entrevistas e pesquisas;
- * Entrevistas: planejamento, preparação, comportamento, linguagem, fatos x opiniões, desejos dos usuários, observações gerais;
- * Contratos e lei: análise das principais cláusulas de um contrato (identificação das partes, documentação, prazo de entrega, códigos fontes, confiabilidade);
- * Lei do Software - comentários sobre os principais artigos;
- * O que é melhor: autônomo ou empresa;
- * Estudo de caso: Sistema para Hotel - apresentação de um estudo de caso mostrando desde o início um problema enfrentado por um hotel e a

solução adotada por um analista de sistemas. Neste estudo são mostrados e comentados a proposta e o contrato utilizado pelo analista;

- * Apresentação de recibos: simples e RPA;
- * Vantagens do aluguel de software;
- * A carreira do consultor independente;
- * Comentários acerca do Guia do Empreendedor e do manual A Pequena Empresa e o Novo Código Civil;
- * Prosperidade - Mapas Mentais - explanação a apresentação do programa freemind.

Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 208 Minutos

- * Compreensão da realidade do SisCom;
- * Criação das tabelas (Produto - Cliente - Fornecedor - Funcionario - Banco - Departamento - Cartao - Unidade - Plano_Conta - Tipo_Pgto - CFOP - Pagamento - Recebimento - C_NFE - D_NFE - C_Venda - D_Venda - C_Requisicao - D_Requisicao - C_Cotacao - D_Cotacao - C_Pedido - D_Pedido - Configuracao - Nivel_Acesso);
- * Comentário detalhado sobre cada tabela e seus campos;
- * Apresentação do DER do SisCom e comentários sobre cada relacionamento;
- * Criação das tabelas e dos relacionamentos no DBDesigner;
- * Sincronização do DBDesigner com o MySQL;
- * Migrando o BD do MySQL para o Firebird;
- * Conectando o DBDesigner com o Firebird através de ODBC;
- * Conectando o DBDesigner com o Firebird diretamente;
- * Fazendo Engenharia Reversa de tabelas do Firebird no DBDesigner;

- * Visualizando a estrutura de arquivos do DBDesigner para fazê-lo conectar-se ao Firebird;
- * Visualizando o Código Fonte do DBDesigner dentro do Delphi;
- * Criação passo a passo do BD no Firebird utilizando o IBExpert;
- * Padronização de campos e tabelas;
- * Erros encontrados por falta de padronização.



2 CDs - 416 Minutos

- * Apresentação da Suite de Componentes ZeosDBO;
- * Vantagens de usar o ZeosDBO;
- * Instalação do ZeosDBO;
- * Explicação sobre cada componente do ZeosDBO;
- * Comparação dos componentes do ZeosDBO com os seus equivalentes no BDE;
- * Diferença entre sistemas MDI e SDI e explicação sobre qual o melhor;
- * Início do desenvolvimento do sistema SisCom;
- * Criação do Formulário Principal: Menu, Barra de Ferramentas, Barra de Status. Todos os hints do sistema serão enviados para a barra de status, possibilitando uma ajuda online através de dicas rápidas para os usuários;
- * Carregando um logotipo para a tela principal do sistema;
- * Uso de Application.MessageBox;
- * Componentes de acesso ao banco em DataModule;
- * Criação dos cadastros utilizando ZTable (equivalente a TTable). Para

quem vem do paradox ficará muito fácil desenvolver dessa forma;

- * Criação dos cadastros utilizando ZQuery (Equivalente a TQuery). Utilização de Edits Simples no lugar dos DBEdits, com procedimentos específicos para o controle de edição e gravação dos campos;
- * Utilização de comandos dos tipos TTable (Append, Edit, Post, Delete) e cadastros usando apenas SQL puro (Insert Into, Update Set, Delete From);
- * O sistema roda AUTOMATICAMENTE em Firebird e MySQL. O usuário escolhe qual banco usar. Não existe alteração nenhuma no código;
- * Utilização dos Generators (firebird) para autonumerar campos no sistema;
- * Implementação de uma rotina onde o usuário procura por qualquer informação em qualquer campo do sistema, bastando para isso clicar no título do DBGrid;
- * Criação do nível de acesso para controle de acessos dos usuários aos módulos do sistema;
- * Verificação e análise de várias mensagens de erro;
- * Implementação de um controle de auditoria, onde o sistema armazena qual o usuário que inseriu ou alterou registros, juntamente com a data e hora.

Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 236 Minutos

- * Introdução a tecnologia dbExpress;
- * Vantagens e desvantagens do dbExpress;
- * Comparativo dbExpress x BDE;
- * Apresentação de todos os componentes da paleta dbExpress;
- * Utilização do SQLQuery + DataSetProvider + ClientDataSet para criar o Contas a Pagar;
- * Desenvolvimento do Contas a Pagar passo a passo;
- * Acessando Firebird e MySQL ao mesmo tempo através do SQLConnection;
- * Utilizando um método de procura ao clicar no título do DBGrid, tendo por base um período passado por 2 MaskEdits;
- * Pegando o código AutoIncrement no MySQL para o ClientDataSet;
- * Pegando o código do Generator no Firebird para o ClientDataSet (Stored Procedures);
- * Explicação do uso de ActionLists para otimizar o trabalho;

* O SisCom funciona automaticamente com MySQL e Firebird. Os módulos de cadastro estão utilizando a tecnologia Zeos. O módulo de Contas a Pagar está usando a tecnologia dbExpress + ClientDataSet.



1 CD - 140 Minutos

- * Desenvolvimento do Contas a Receber do SisCom utilizando a tecnologia dbExpress + ClientDataSet;
- * Implementação do conceito de reconciliação de erros no SisCom;
- * Tradução do Formulário de Reconciliação de Erros para o Português;
- * Utilização apenas do SQLConnection para realizar consultas SQL;
- * Explicação detalhada do uso do SQLDataSet (Table, Query, StoredProc);
- * Utilização de parâmetros para realizar consultas SQL;
- * Manipulação dos dados do ClientDataSet antes de serem enviados para o Banco de Dados - ainda na memória do micro cliente (utilização da propriedade Delta do ClientDataSet);
- * Criação de consultas Mestre/Detalhe. (Dados mostrados numa mesma DBGrid e posteriormente em duas DBGrids);
- * Criação de bancos de dados locais com o ClientDataSet (utilização de arquivos XML);

- * Criação de Campos Agregados;
- * Utilização do componente SimpleDataSet;
- * Utilização do componente SQLMonitor para mapear o que está ocorrendo entre a aplicação e o banco de dados (tudo que está ocorrendo em baixo nível é mostrado num Memo);
- * Implementando o Controle de Transações com o dbExpress;
- * Explicação sobre o nível de isolamento (TransIsolationLevel) das transações do dbExpress.

Após assistir às vídeo aulas do curso você estará apto para desenvolver qualquer sistema comercial. Este conteúdo você não encontra em nenhum curso presencial ou online. Conteúdo único e exclusivo.

Curso Delphi Avançado - 15 Módulos (19 CDs)



2 CDs - 447 Minutos

- * Apresentação de cada componente da paleta ADO - principais propriedades;
- * Criando uma conexão no SisCom para Firebird e MySQL (alternando entre um e outro de acordo com a escolha do usuário);
- * Instalação dos Drivers ODBC para MySQL e Firebird e criação de DSN para acesso ao banco de dados do SisCom;
- * Criação do Módulo Requisição de Compras;
- * Utilização do ADOConnection, ADOQuery, ADOTable e ADOSToredProc;
- * Análise de erros ao construir o módulo;
- * Construção do formulário para importar itens para a requisição;
- * Trabalhando com passagem de parâmetros;
- * Localizando e eliminando erros do projeto do BD;
- * Criação do Módulo Cotação de Compras;
- * Utilização do repositório de formulários do Delphi;
- * Utilização de um componente permite abrir um menu em um botão;

- * Construção do formulário para importar os itens das requisições que vão compor a cotação;
- * Construção do Módulo Confirmação da Cotação (neste formulário o usuário insere os preços e condições enviadas pelos fornecedores);
- * Construção do Módulo Pedido de Compras (neste formulário o usuário visualiza as cotações e pode fazer pedidos aos fornecedores baseado nas informações prestadas pelos mesmos. É permitido realizar vários pedidos através de uma cotação);
- * Construção do formulário para importar os itens das cotações.



1 CD - 165 Minutos

- * Construção do Sistema de Vendas utilizando os componentes da paleta ADO (ADOConnection, ADOQuery, ADOTable, ADOSToredProc);
- * Chamadas a Stored Procedures armazenadas no BD com passagem e retorno de parâmetros;
- * Importação dos dados da tabela oficial CFOP de um BD diferente;
- * Análise de alguns erros que ocorrem com ADO;
- * Usando ADO com ClientDataSet;
- * Acessando o BD sem o uso do componente ADOConnection;
- * Construindo consultas Mestre-Detalhe com ADO (3 tabelas no exemplo);
- * ADO e Controle de Transações;
- * Enviando informações para o BD sem Result Set com o componente ADOCommand.



1 CD - 188 Minutos

- * Descrição detalhada dos 13 Componentes da paleta IBX (Interbase Express);
- * Comentários sobre suas propriedades;
- * Construção do Formulário de Entrada de Notas Fiscais (IBDataBase, IBTransaction, IBDataSet, IBQuery, IBUpdateSQL);
- * Passagem de parâmetros;
- * Atualização do Estoque via código feito no Delphi;
- * Atualização do Estoque via Triggers criadas no Firebird (AfterInsert, AfterUpdate, AfterDelete);
- * Construção do Formulário para Ajuste de Preços;
- * Código para ajustar preços na tabela de produtos utilizando uma taxa (aumentando e diminuindo valores).

Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 198 Minutos

- * Instalação do Quick Report no Delphi 7;
- * Descrição detalhada dos componentes da Suite Quick Report;
- * Desenvolvimento dos relatórios de cadastro do SisCom utilizando o Quick Report;
- * Criação de relatórios Mestre-Detalhe com o Quick Report;
- * Exportando relatórios para TXT e HTML através do Quick Report;
- * Descrição detalhada dos componentes da Suite Rave Reports;
- * Desenvolvimento dos relatórios de Contas a Pagar e a Receber utilizando o Rave Reports;
- * Criação de relatórios Mestre-Detalhe com o Rave Reports;
- * Utilização de imagens e códigos de barras usando o Rave Reports;
- * Exportando relatórios para PDF através do Rave Reports;
- * Apresentação do Componente PrintFast (componente que envia o relatório diretamente para a impressora - excelente para impressoras matriciais e Clippeiros de plantão);
- * Instalação do PrintFast no Delphi 7;

* Desenvolvimento dos seguintes relatórios usando o PrintFast: Requisição de Compras, Mapa Comparativo de Preços e Pedido de Compras;

* Apresentação dos Geradores de Relatórios: FreeReport e Fortes Reports.



1 CD - 168 Minutos

- * Análise de 10 Manuais em formato PDF;
- * Criação do Manual do SisCom em formato PDF;
- * Criação de um arquivo HLP (ajuda) utilizando ferramentas gratuitas;
- * Chamando a ajuda de dentro do SisCom ao pressionar a tecla F1 (ajuda sensível ao contexto);
- * Desenvolvimento de um manual HTML para o SisCom;
- * Convertendo arquivos HTML para CHM (HTML Help);
- * Convertendo arquivos CHM para 42 outros formatos (PDF, HLP, RTF, TXT, XLS, DOC e muitos outros);
- * Utilização de uma ferramenta que cria o Help e já salva em diversos formatos diferentes (HLP, CHM, PDF e outros). Ferramenta muito fácil de utilizar;
- * Chamando a ajuda CHM de dentro de uma aplicação Delphi ao pressionar F1 (ajuda sensível ao contexto);
- * Utilização de uma ferramenta que cria o arquivo de ajuda baseado nas

imagens capturadas das telas do SisCom. O usuário clica em qualquer parte das janelas capturadas e a ajuda referente àquela parte da janela é apresentada.



1 CD - 175 Minutos

- * Modificando o código do SisCom para que o mesmo funcione em rede;
- * Testando o funcionamento do SisCom em rede (máquina virtual);
- * Identificação das bibliotecas necessárias para a instalação numa máquina cliente;
- * Trabalhando com arquivos Ini (configuração do acesso aos dados através desses arquivos);
- * Desenvolvimento de um instalador personalizado no Delphi (o instalador se encarrega de copiar todos os arquivos necessários para o funcionamento do SisCom e instala o MySQL e o Firebird);
- * Utilização de ferramentas para criação de instaladores: Setup2Go, CreateInstall, InnoSetup);
- * Compactação de executáveis;
- * Criando uma biblioteca de funções e integrando a mesma ao SisCom;
- * Aperfeiçoamento da janela de Login;
- * Desenvolvimento de uma Splash Screen para o SisCom utilizando uma

imagem feita no CorelDraw e exportada para o SisCom. Utilização de uma barra de progresso.

Curso Delphi Avançado - Módulos Plus



2 CDs - 224 Minutos

- * Introdução a Orientação a Objetos
- * Histórico do Paradigma de Orientação a Objetos
- * Explicações sobre: Abstração; Objeto; Encapsulamento; Mensagem; Classe; Herança; Polimorfismo; Classificação; Associação; Agregação; Generalização; Especialização
- * Introdução a UML - Unified Modeling Language
- * Histórico da notação UML
- * Fases do desenvolvimento de um sistema em UML: Análise de requisitos; Análise; Design (projeto); Programação; Testes
- * Visões: Visão de Componentes; Visão Lógica; Visão de Use-Case; Visão de Organização; Visão de Concorrência
- * Modelos de Elementos: Classes; Objetos; Estado; Pacote; Componente; Relacionamentos (associação, generalização, dependência e refinamento)
- * Diagramas: Use-Case; Classes; Objetos; Estado; Seqüência;

Colaboração; Atividade; Componente; Execução

- * Estudos de caso: conta corrente e aplicações financeiras; locadora; sistema de matrícula; posto de gasolina; biblioteca.
- * Cada estudo de caso acima é feito em uma ferramenta diferente.

Obs: Este Módulo é pré-requisito para o Módulo Delphi OOP



1 CD - 220 Minutos

- * Introdução à Linguagem Object Pascal (Delphi Language);
- * Variáveis - declaração e exemplos; Variáveis Locais e Globais;
- * Constantes - declaração e exemplos;
- * Tipos de Dados - Inteiros, Reais, Texto, Ordinais;
- * Rotinas de conversão de tipos; TypeCasting;
- * Operadores - unários, multiplicativos, direção de bits, aditivos, relacionais;
- * Estruturas de Repetição - While, For, Repeat; Quebras de Laço (continue, break, exit, halt);
- * Tipos Definidos pelo Usuários - SubRange, Enumerações, Ponteiros, Records, Arrays, Sets;
- * Procedimentos, Funções e Métodos;
- * Passagem de parâmetros - explicações sobre passagem por valor e por referência; Uso do With;
- * Desenvolvimento de diversas rotinas para praticar os conceitos acima;
- * Explicações sobre a estrutura da Unit: Áreas: Unit, Interface,

Implementation, Var, Const, Uses, Referência Circular;

- * Sobrecarga de métodos: Overload - Exemplo de vários métodos de mesmo nome para mostrar o funcionamento;
- * Blocos protegidos - explicações sobre exceções (try, except, finally, raise);
- * Explicações sobre parâmetros formais e parâmetros reais;
- * Classes e Objetos; Desenvolvimento de diversas classes no Delphi;
- * Métodos construtores e destrutores; Métodos Getters e Setters;
- * Instanciação das classes - Desenvolvimento de uma aplicação Console para analisar o funcionamento;
- * Encapsulamento - Operadores de visibilidade (private, public, protected, published) - explicações e prática de cada um deles;
- * Herança entre classes; * Utilização de Property (propriedades);
- * Sobreposição de Métodos - Virtual/Override - Sobrescrever métodos em classes descendentes - Conceito de Polimorfismo;
- * Desenvolvimento do SisCom OO - explicações sobre o objetivo do SisCom (Sistema Comercial);
- * Exibição do DER e da UML do SisCom. Diferenças entre os dois diagramas;
- * Apresentação do SisCom desenvolvido durante o Curso Delphi Avançado;
- * Desenvolvimento passo a passo do Formulário Padrão de Cadastro. Através desse formulário, todos os outros forms de cadastro serão criados. Utilização do conceito de Herança Visual do Delphi;
- * Integração Delphi / ModelMaker; * Engenharia reversa do SisCom para o ModelMaker;
- * Alterações em modelo no ModelMaker e atualização em Tempo Real na aplicação no Delphi;
- * Explicações para desenvolvimento dos diagramas UML do SisCom no ModelMaker;

Obs: O Módulo Plus Entendendo UML é pré-requisito para este módulo

Curso Delphi Avançado - Módulos Plus



2 CDs - 339 Minutos

- * Explicações sobre ECF (Emissor de Cupom Fiscal) e TEF (Transferência Eletrônica de Fundos);
- * Instalação e utilização de Emulador de Impressora Fiscal para utilização junto com o sistema de Frente de Caixa;
- * Análise dos manuais das Impressoras Bematech, Yanco, Daruma e Zanthus;
- * Criação passo a passo da Interface do Sistema de Frente de Caixa;
- * Implementação dos comandos para utilização com a impressora fiscal: Leitura X, Redução Z, Relatórios de Memória Fiscal; Abertura e Fechamento de Cupons; Envio de Formas de Pagamento; Envio e Cancelamento de Itens; Etc;
- * Criação de um formulário de configurações onde o próprio usuário define as formas de pagamentos que vai utilizar, em quais ocasiões vai solicitar senhas, qual impressora está utilizando e o caminho onde as fotos dos produtos estão armazenadas (as fotos dos produtos são armazenadas

fora do banco de dados);

- * Criação de um formulário para localização de produtos, caso o sistema não consiga ler o código ou o produto não tenha código de barras;
- * Criação de um formulário para armazenar os dados da venda;
- * Estudo e implementação da Solução TEF;



2 CDs - 275 Minutos

- * SQL Server - Apresentação - Características;
- * Instalação passo a passo do SQL Server e Ferramentas Avançadas - Instalação do .NET Framework;
- * Apresentação da ferramenta de configuração (SQL Server Configuration Manager);
- * Apresentação e configuração do SQL Browser - ferramenta para aceitar conexões remotas;
- * Explicação e configuração dos tipos de conexões (Shared Memory, Named Pipes, TCP/IP, VIA);
- * Criação de aliases para o Banco de Dados;
- * Apresentação da ferramenta de configuração (SQL Server Surface Area Configuration);
- * Configuração e uso da ferramenta para gerenciamento de bancos de dados feitos em SQL Server (Microsoft SQL Server Management Studio Express);

- * Explicação sobre Collations; Detalhamento sobre os tipos de dados;
- * Criação e configuração de um banco de dados;
- * Criação das tabelas, índices, relacionamentos, constraints, etc do banco de dados utilizando ferramenta para modelagem;
- * Criação do campo Auto-Increment; Criação de Views atualizáveis;
- * Apresentação de ferramenta gráfica adicional para gerenciamento do banco de dados (SQL Manager 2005 for SQL Server);
- * Criação de Checks Constraints; Criação de Usuários/Roles para acesso ao banco;
- * Confecção de relatórios utilizando a ferramenta;
- * Exportando dados (Excel, Access, Word, RTF, HTML, PDF, TXT, CSV, DBF, XML);
- * Importando dados (Excel, Access, DBF, XML, TXT, CSV);
- * Explicação sobre Stored Procedures e Triggers;
- * Operações com o Banco (extração de dados, backup e restore);
- * Apresentação de ferramenta gráfica adicional para gerenciamento do banco de dados (SQL Lite);
- * Conhecendo as funções de sistema (agregação, matemáticas, data e hora, string, diversas...);
- * Desenvolvimento de uma aplicação do Delphi para acesso ao banco de dados (cadastros, vendas, etc). Utilização das tecnologias de acesso ADO e dbExpress;
- * Formulário mestre-detelhe e campos de lookup; Explicação e criação de Trigger para atualização do estoque;
- * Utilização de transações (início, cancelamento e confirmação);
- * Funcionamento da aplicação em rede através de máquina virtual W98;

Curso Delphi Avançado - Módulos Plus

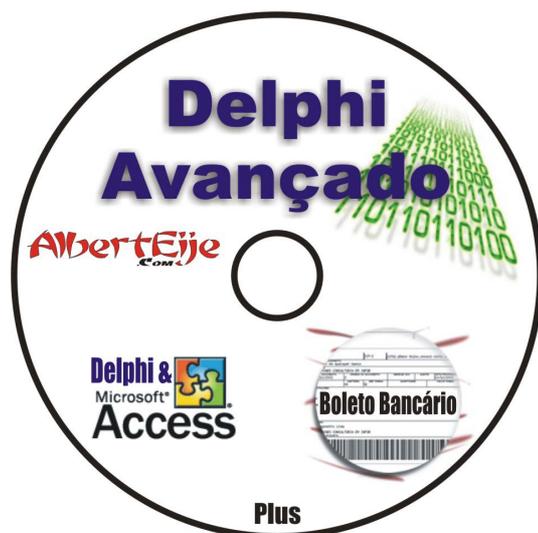


1 CD - 129 Minutos

- * Introdução a Criptologia (Criptografia e Criptoanálise);
- * Utilizando na prática a Esteganografia (esconder arquivos em imagens);
- * Análise de vários componentes que implementam a Criptografia no Delphi;
- * Apresentação de vários métodos para Proteção de Programas (Nag-Screen, Período, Limitação de Registros, HardLocks, Identidade Única);
- * Implementando na prática várias técnicas para proteção de programas (Por Arquivo, Por Diretório, Por Registro, etc);
- * Desenvolvimento de um sistema que bloqueia o uso do sistema no usuário. O sistema grava as informações no registro do Windows. Pega informações da máquina do usuário (HD, BIOS) e combina tais informações com Constantes implementadas pelo programador (Phi e Pi). O desbloqueio do sistema se dá ao informar a Contra-Chave (Demonstrado passo a passo);
- * Apresentação de vários componentes para Delphi que implementam a

proteção de programas;

- * Dicas importantes para colocar em prática no momento da criação de uma proteção;
- * Apresentação de Engenharia Reversa (Descompiladores);
- * Utilização na prática de alguns descompiladores (DeDe, Source Rescuer, ResHacker);
- * Demonstração de técnica para evitar a Engenharia Reversa de programas feitos em Delphi;
- * Demonstração de como Crackear um programa feito em Delphi utilizando um Editor HexaDecimal (passo a passo);



1 CD - 229 Minutos

- * MSAccess - Apresentação;
- * Criação de Tabelas (explicação dos tipos de dados);
- * Especificações do Access - limitações do banco de dados;
- * Definindo relacionamentos e criando integridade referencial. Explicações sobre cardinalidade;
- * Criação de consultas (Visual e SQL);
- * Criação de formulários. Formulários prontos para impressão. Formulários vinculados a código em VBA. Subformulários;
- * Criação de uma consulta vinculada a dados de um formulário. Visualização dos dados de uma consulta em um subformulário;
- * Desenvolvimento de relatórios;
- * Desenvolvimento de uma página WEB que acessa e altera os dados da tabela;
- * Introdução a VBA (Visual Basic for Applications);
- * Acessando o banco de dados através do Delphi (tecnologia DAO);
- * Criação de um pequeno aplicativo no Delphi acessando o BD Access.

Utilização de Acion Lists;

- * Demonstração do Assistente de tabelas do Access que ajuda a normalizar o Banco de Dados;
- * Demonstração do Analisador de Desempenho do Access;
- * Definindo e alterando senhas no Access;
- * Acessando os relatórios desenvolvidos no Access através do Delphi: visualização em tela e envio para impressão sem que o usuário perceba que o relatório foi feito no Access;
- * Criação de índices primários e secundários;
- * Introdução ao conceito de boletos bancários;
- * Apresentação do Manual da Federação Brasileira dos Bancos;
- * Criação passo a passo de um registro de 240 posições definido no manual da Febraban;
- * Apresentação de uma solução pronta (componente open source) para a impressão de boletos bancários em Delphi, emissão do arquivo de remessa que é enviado pelas empresas aos bancos e leitura do arquivo de retorno que é enviado pelos bancos para as empresas;
- * Instalação e utilização do referido componente: impressão de boletos bancários em QuickReport (visualização personalizada e RaveReports, exportação do boleto para PDF, geração do arquivo de remessa);
- * Demonstrando a impressão de boletos bancários com dados vindos de uma tabela do banco de dados Access;
- * Orientações para migrar o componente para uma versão posterior do Delphi;