

# Módulo 2

## Lógica de Programação II

**REPETIÇÃO:** Serve para efetuar um conjunto de ações repetidas vezes. Existem três tipos básicos de repetições, sendo elas.

Enquanto (Expressão Lógica) faça Sequência	O comando Enquanto analisa a Expressão Lógica e enquanto o seu resultado for o valor lógico Verdade a Sequência é executada.
Para variável VAR valor_inicial até valor_final faça Sequência	O comando Para incrementa a variável a partir do valor_inicial de uma unidade até que esta atinja o valor_final. E para cada incremento a Sequência é executada.
Repita Sequência Até (Expressão Lógica)	O comando Repita executa a Sequência até que o valor retornado pela Expressão Lógica seja Verdadeiro.

### ESTRUTURA DE REPETIÇÃO DETERMINADA

Quando uma sequência de comandos deve ser executada repetidas vezes, tem-se uma estrutura de repetição. A estrutura de repetição, assim como a de decisão, envolve sempre a avaliação de uma condição. Na repetição determinada o algoritmo apresenta previamente a quantidade de repetições.

Forma Geral 1:

```
PARA <<VARIABLE DE TIPO INTEIRO>>:=<<VALOR INICIAL>> ATE <<VALOR FINAL>> FAÇA
    <<COMANDO1>>;
```

Forma Geral 2:

```
PARA <<VARIABLE DE TIPO INTEIRO>>:=<<VALOR INICIAL>> ATE <<VALOR FINAL>> FAÇA
    INICIO
        <<COMANDO1>>;
        <<COMANDON>>;
    FIM;
```

A repetição por padrão determina o passo do valor inicial até o valor final como sendo 1. Determinadas linguagens possuem passo -1 ou permitem que o programador defina o passo.

### ALGORITMO SETE

Segue um algoritmo que escreve 10 vezes a frase "VASCO DA GAMA"

```
PROGRAMA REPETICAO;
VAR I:INTEIRO
INICIO
    PARA I:=1 ATE 10 FACA
        ESCREVA ('VASCO DA GAMA')
    FIM.
```

### ALGORITMO OITO

Segue um algoritmo que escreve os 100 primeiros números pares.

```
PROGRAMA PARES;  
VAR I, PAR: INTEGER;  
INICIO  
    PAR:=0;  
    PARA I:=1 ATE 100 FACA INICIO  
        ESCRIVA (PAR);  
        PAR := PAR+2;  
    FIM  
FIM.
```

### ESTRUTURA DE REPETIÇÃO INDETERMINADA COM VALIDAÇÃO INICIAL

É usada para repetir N vezes uma ou mais instruções. Tendo como vantagem o fato de não ser necessário o conhecimento prévio do número de repetições.

Forma Geral 1:

```
ENQUANTO <<CONDIÇÃO>> FACA  
    <<COMANDO1>>;
```

Forma Geral 2:

```
ENQUANTO <<CONDIÇÃO>> FACA ÍNICIO  
    <<COMANDO1>>;  
    <<COMANDON>>;  
FIM;
```

### ALGORITMO NOVE

Segue um algoritmo que calcule a soma dos salários dos funcionários de uma empresa. O programa termina quando o usuário digitar um salário menor que 0.

```
PROGRAMA SOMA_SALARIOS;  
VAR SOMA, SALARIO : REAL;  
INICIO  
    SOMA:=0;  
    SALARIO:=1;  
    ENQUANTO SALARIO>=0 FAÇA INICIO  
        LEIA (SALARIO);  
        SOMA:=SOMA+SALARIO  
    FIM;  
    ESCRIVA (SOMA)  
FIM.
```

OBS: Todas as variáveis que acumulam valores devem receber um valor inicial.

### ESTRUTURA DE REPETIÇÃO INDETERMINADA COM VALIDAÇÃO FINAL

Assim como a estrutura ENQUANTO É usada para repetir N vezes uma ou mais instruções. Sua validação é final fazendo com que a repetição seja executada pelo menos uma vez.

Forma Geral;

```
REPITA  
    <<COMANDO1>>;  
    <<COMANDON>>;  
ATE <<CONDIÇÃO>>;
```

### ALGORITMO DEZ

Segue um algoritmo que calcule a soma dos salários dos funcionários de uma empresa. O programa termina quando o usuário digitar um salário menor que 0.

```
PROGRAMA SOMA_SALARIOS;  
VAR  
SOMA, SALARIO : REAL;
```

```

INICIO
  SOMA:=0;
  REPITA
    LEIA(SALARIO);
    SOMA:=SOMA+SALARIO
  ATE SALARIO<0;
  ESCREVA (SOMA)
FIM.

```

## ALGORITMO ONZE

Segue um algoritmo que escreve os 100 primeiros números pares.

```

PROGRAMA PARES_2;
VAR
  I, PAR, CONTADOR : INTEIRO;
INICIO
  CONTADOR:=0;
  PAR:=0;
  REPITA
    ESCREVA(PAR);
    PAR:=PAR+2;
    CONTADOR:=CONTADOR+1;
  ATE CONTADOR=100
FIM.

```

## VARIÁVEIS COMPOSTAS UNIDIMENSIONAIS (VETORES)

Até agora variáveis podiam comportar apenas uma informação por vez. Veremos agora que uma variável pode comportar um conjunto de informações primitivas, desde que devidamente ORGANIZADAS, DIVIDIDAS.

Para entender as variáveis compostas unidimensionais, imaginemos uma locomotiva com um número finito de vagões, representando uma estrutura de dados, e seus vagões, partições desta estrutura.

### DECLARAÇÃO

tipo LISTA = vetor( CI .... CF ) de < tipo >;  
LISTA = lista de variáveis;

### ONDE

CI (Coluna Inicial) – representa o limite inicial do vetor;  
CF (Coluna Final) – representa o limite final do vetor;  
< tipo > - representa qualquer um dos tipos primitivos de variáveis;

### EXEMPLO

tipo CLASSE = vetor( 1 ... 40 ) de reais;  
CLASSE : NÚMEROS;

### NÚMEROS

8.5	7.4	5.8	2.9	9.2	.....	7.3	5.4
1	2	3	4	5		39	40

Podemos notar que CI e CF são sempre números inteiros e CF sempre será maior que CI.

O número de elementos do vetor será dado por CF – CI + 1. Isto significa que as posições do vetor são identificadas a partir de CI, com incrementos unitários, até CF.

CI	CI + 1	CI + 2	CI + 3	CI + 4	.....	CF – 1	CF
----	--------	--------	--------	--------	-------	--------	----

## OUTROS EXEMPLOS

tipo CLASSE = vetor( 1 ... 50 ) de inteiros;  
CLASSE : VET1, VET2, VET3;

VET1							
9	8	7	6	5	.....	4	2
1	2	3	4	5		39	40
VET2							
8	7	5	2	9	.....	7	5
1	2	3	4	5		39	40
VET3							
2	2	2	2	2	.....	2	2
1	2	3	4	5		39	40

## EXERCÍCIOS DE FIXAÇÃO

01- Ler e armazenar um conjunto de 30 números inteiros.

```

programa trinta;
variáveis
tipo NUMEROS = vetor ( 1 ... 30 ) de inteiros;
NUMEROS : CONJUNTO;
NUM, CONT: inteiro;
início
.   CONT := 1 ;
.   faça enquanto CONT <= 30 ;
.       leia( NUM ) ;
.       CONJUNTO ( CONT ) := NUM;
.       CONT := CONT + 1;
.   fimfaça;
fim.

```

02- Construa um vetor de nome ALUNOS com 39 posições onde cada uma delas será preenchida pelo nome dos alunos desta classe.

```

programa alunos;
variáveis
tipo NMALUNO = vetor ( 1 ... 39 ) de caracteres;
NMALUNO : ALUNOS;
NOMES: caracter;
CONT: inteiro;
.   CONT := 1 ;
.   faça enquanto CONT <= 39 ;
.       leia( NOMES ) ;
.       ALUNOS ( CONT ) := NOMES;
.       CONT := CONT + 1;
.   fimfaça;
fim.

```

## VARIÁVEIS COMPOSTAS BIDIMENSIONAIS (MATRIZES)

Os vetores têm como principal característica a necessidade de apenas um índice para endereçamento – são estruturas unidimensionais.

Uma estrutura que precisa de mais de um índice são denominadas estruturas multidimensionais.

Suponha que, além do acesso pelo elevador até um determinado andar, tenhamos também a divisão desse andar em apartamentos. Para chegar a algum deles não basta só o número do andar, precisamos também do número do apartamento. Neste caso precisamos de dois números para chegarmos onde queremos, ou seja, dois índices. Chamamos de estrutura bidimensional (MATRIZ) aquela que precisa de dois índices para chegar ao seu conteúdo.

### DECLARAÇÃO

tipo IDENTIFICADOR = matriz( LI...LF, CI...CF ) de < tipo >;  
IDENTIFICADOR: lista de variáveis;

### ONDE

LI...LF, CI...CF – são os limites dos intervalos de variação dos índices da variável, onde cada par de limites está associado a um índice;

LI...LF à linhas da matriz;

CI...CF à colunas da matriz.

< tipo > - representa qualquer um dos tipos primitivos de variáveis;

### EXEMPLO

tipo SALA = matriz( 1 ... 4, 1...4 ) de inteiros;

SALA: MSALA;

	1	2	3	4
1				
2				
3				
4				

Matriz MSALA

O intervalo em destaque é a célula (APARTAMENTO) MSALA(2, 3)

Para utilizar o vetor, nós o inserimos em um único laço de repetição, fazendo com que haja variação do seu índice. Como numa estrutura multidimensional possuímos mais de um índice, faz-se necessária a utilização de mais laços de repetição. As matrizes mais utilizadas são as bidimensionais que devem possuir dois laços de repetição.

### EXERCÍCIOS DE FIXAÇÃO

01- Ler e armazenar um conjunto de 36 números inteiros em uma matriz.

Programa trinta e seis;

Variáveis

tipo NUMEROS = matriz ( 1...6, 1...6 ) de inteiros;

NUMEROS : NUM1;

NUM, CONTL, CONTC: inteiro;

início

CONTL := 1 ;

faça enquanto CONTL <= 6 ;

. CONTC := 1 ;

. faça enquanto CONTC <= 6 ;

. . leia( NUM ) ;

. . NUM1 ( CONTL, CONTC ) = NUM;

. . CONTC := CONTC + 1;

. fimfaça;

. CONTL := CONTL + 1 ;

fimfaça;

fim.

02- Construa uma matriz de nome IDADE com 18 posições onde cada uma delas será preenchida pela idade de um conjunto de pessoas.

```
Programa idade;
Variáveis
tipo ALUNOS = matriz ( 1...6, 1...3 ) de inteiros;
ALUNOS : IDADE;
ID, CONTL, CONTC: inteiro;
início
CONTL := 1 ;
faça enquanto CONTL <= 6 ;
.   CONTC := 1 ;
.   faça enquanto CONTC <= 3 ;
.       ID := -1 ;
.       faça enquanto ID <= 0 ;
.           leia( ID )
.       fimfaça ;
.       IDADE ( CONTL, CONTC ) = ID;
.       CONTC := CONTC + 1;
.   fimfaça;
.   CONTL := CONTL + 1 ;
fimfaça;
fim.
```

## EXERCÍCIOS

- 01) Faça um algoritmo que imprima a tabuada do número 7.
- 02) Faça agora o mesmo algoritmo para calcular a tabuada de qualquer número.
- 03) Faça um algoritmo que leia 50 números reais e escreva o número maior e o menor deles.
- 04) Construa um algoritmo que leia um conjunto de dados contendo altura e sexo (MAS ou FEM) de N pessoas e calcule e escreva:
- a maior e a menor altura do grupo e de quem é esta altura. ( Se HOMEM ou MULHER.)
  - a média de altura entre as mulheres.
  - O número de homens e a diferença percentual entre estes e as mulheres.
- 05) Elabore um algoritmo que efetue a soma de todos os números inteiros ímpares que são múltiplos de três e que se encontram no conjunto dos números de 1 até 500.
- 06) Preciso fazer um levantamento do meu estoque de mercadorias que tenho na minha loja de artigos de construção. Para isso preciso saber qual o material que vendo e a quantidade de cada material. Me ajude, fazendo a entrada e a saída de mercadorias da minha loja no último mês de venda. O algoritmo termina quando encontrar a palavra "ACABOU". Minhas mercadorias com o estoque de um mês atrás são:
- |                  |                       |                        |
|------------------|-----------------------|------------------------|
| CIMENTO 40 sacos | CAL 50 alqueires      | TIJOLO 15.000 unidades |
| AREIA 200 m3     | TELHA 25.650 unidades |                        |
- 07) Durante uma corrida de automóveis com N voltas de duração foram anotados para um piloto os tempos, em minutos, registrados em cada volta. Faça um algoritmo que calcule:
- seu melhor tempo.
  - A volta em que o melhor tempo ocorreu.
  - O tempo médio das N voltas.
- 08) Uma locadora de CDs e DVDs está precisando fazer um sistema de computação para organizar seu negócio. As informações mais importantes para o dono é:
- quantos filmes estão alugados.
  - quantos clientes estão com seus filmes.
  - Quantos clientes pagam a vista.
  - Quantos clientes só pagam quando devolvem o filme.
  - Quanto será o apurado da loja na devolução de todos os filmes.
- Faça um algoritmo que escreva estas informações para a locadora.
- Observação à O algoritmo termina quando em uma variável de nome SAIR estiver o conteúdo "SIM".
- 09) Faça um algoritmo para ler e montar um vetor de números inteiros com 50 posições. Depois de montado, o vetor ler 10 números inteiros quaisquer e para cada número lido imprimir se este existe ou não no vetor.
- 10) Faça um algoritmo que leia e monte dois vetores 1 e 2 de N posições com números reais positivos. Depois de montados procure por cada elemento do vetor 1 no vetor 2 e armazene as suas respectivas posições ( somente do primeiro número achado ) em um vetor 3, sendo que os elementos do vetor 1 que não forem encontrados no vetor 2 deverão ter como valor o número zero no vetor 3.
- 11) Ler um vetor de nomes e imprimir os nomes das posições PARES e em seguida os nomes das posições IMPARES.

- 12) Ler um conjunto de nomes e notas dos alunos de uma turma e imprimir os nomes dos alunos que tiveram nota acima da média das notas da turma.
- 13) Faça um algoritmo que construa uma matriz  $M$  por  $N$  de números reais e se esta matriz for quadrada ( $M = N$ ) imprimir sua diagonal principal.
- 14) Faça um algoritmo que construa uma matriz  $50 \times 30$  e depois de construída inverter suas linhas. Exemplo: a última linha será a primeira e a primeira a última. A antepenúltima será a segunda e a segunda será antepenúltima e assim por diante.
- 15) Faça um algoritmo para ler 3 notas de um conjunto de 60 alunos de uma turma armazenando numa matriz, onde os números das linhas representam os números dos alunos e cada coluna uma de suas três notas. Depois de construída a matriz imprimir o número do aluno e sua respectiva média e no final a média geral da turma.

## Veja o Conteúdo Exclusivo do Curso Delphi Avançado



### 1 CD - 130 Minutos

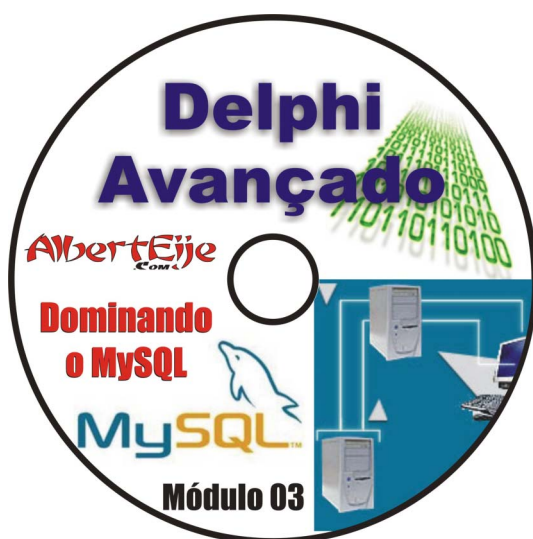
- \* Compreender o modelo Entidade-Relacionamento e Lógico-Relacional;
- \* Saber definir e identificar: Entidades, Relacionamentos, Atributos e tudo mais que tenha a ver com o Modelo Entidade-Relacionamento;
- \* Conhecer os tipos de relacionamentos: Condicionais e Incondicionais;
- \* Saber os graus 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.

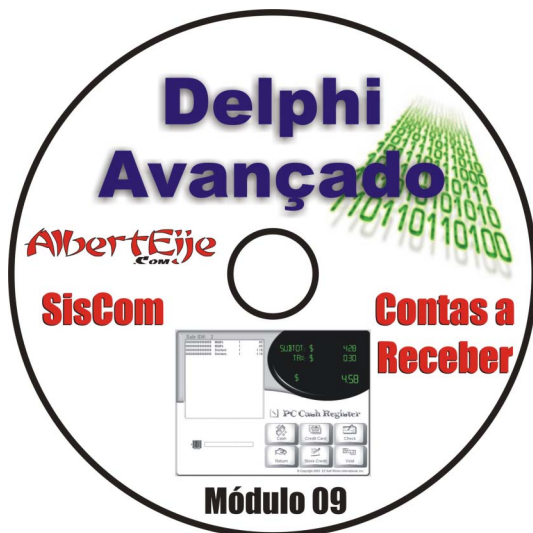
## Curso Delphi Avançado - 15 Módulos (19 CDs)



### 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;
- \* Confeção 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 (inicio, 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;