



DELPHI CLIENT SERVER

Curso de programação em linguagens de fácil assimilação e dentro do enfoque geral em conhecer e discutir os fundamentos, os conceitos, as definições e principalmente o aumento de produtividade em desenvolvimento de software, apresentamos o DELPHI.

Por que Delphi?

Para responder esta pergunta, podemos pesquisar inúmeras e diversas fontes, todavia temos que nos convencer de que nada melhor que a própria palavra * de quem usa a tecnologia para falar sobre ela:

5) também não consigo compreender porque muita gente mete o pau no **delphi** sem conhecê-lo de verdade

delphi usa uma das melhores linguagens que é o object pascal... é tão poderosa quanto o C, e tão fácil de compreender quanto o basic... é orientada a objetos, tem tipagem forte, código fácil de ser lido (importante quando se trabalha em EQUIPE), é compilada e não interpretada, além de suportar assembly inline (ótimo para aplicações de baixo nível que necessitam performance absoluta)... quem diz que ela é pior que o basic, realmente nunca estudou object pascal a fundo...

tem gente que reclama dela só por causa do ";" no final de cada linha, mas essas mesmas pessoas reclamam do ponto-final na língua portuguesa??

é chato ter que declarar todas as variáveis com tipos fixos e não poder "misturar" tipos diretamente? sim, com certeza é, mas em linguagens de tipagem fraca, eu tenho dó de quem da manutenção... tanto que o C++ é muito criticado por causa dos problemas de type safety, inclusive um colega citou que a última reunião do comitê que controla o C++ prevê melhorias nessa área...

a IDE é bem funcional e poderosa, e apesar de alguns dizerem o contrário, ela é bem "aberta", pois possui uma API para que possamos desenvolver plugins... eu mesmo utilizo vários plugins gratuitos para facilitar a navegação pelo código, refactoring, wizards para geração de alguns códigos básicos, correção de digitação, declaração de variáveis locais, etc...

é possível utilizar o **delphi** tranquilamente com outras ferramentas de desenvolvimento... eu por exemplo utilizo uma ferramenta uml que já gera código **delphi** baseado no diagrama de classes... essa ferramenta se integra ao **delphi**, e tudo que eu faço em um, aparece no outro... e nada impede de você utilizar outros editores, ou outros compiladores no **delphi**. da pra você compilar seus projetos em **delphi** com o freepascal se quiser...



a IDE também utiliza um conceito que esta sendo cada vez mais utilizado para desenvolvimento de aplicacoes de grande porte, que é a componentizacao... não toh falando só de ficar arrastando TEdit no form, e sim de criacao de componentes funcionais, que abstraem as funcoes dos programas... a **borland** usou o conceito de componentizacao para abstrair o acesso a api do windows, você também pode fazer o mesmo, criar seus componentes para abstrair os acessos aos dados da sua aplicacao, independente de onde eles estejam armazenados...

fora que a biblioteca de componentes feitos por terceiros para o **delphi** é vasta, você acha componente pra fazer de tudo...

de brinde você leva o codigo fonte de todos os componentes e bibliotecas que acompanham o **delphi**... ótima fonte de estudos pra quem quer dominar a ferramenta e a linguagem...

o compilador e debugador são otimos, dizem que o compilador do **delphi** é um dos melhores existentes no mercado... a performance de aplicacoes nativas win32 é muito boa, inclusive eu tenho aplicacoes relativamente pesadas rodando tranquilo em pentium 100, assim como eu tenho aplicacoes de grande porte segurando mais de 300 usuários simultaneos...

o debugador tem varias funcionalidades interessantes, como avaliacao de expressoes "on-the-fly", visualizacao do assembly gerado, poder debugar dentro das bibliotecas, debugger remoto (você roda em uma maquina e debuga na outra), etc...

*

<http://64.233.169.104/search?q=cache:aStpl2aaBFUJ:www.forumpcs.com.br/viewtopic.php%3Fp%3D1544637+ofertas+DELPHI+borland+estat%C3%ADsticas+benef%C3%ADcios+diferenciais+vantagens&hl=pt-BR&ct=clnk&cd=10&gl=br&client=firefox-a>



Objetivo

Desenvolver interface padrão Windows.

Esgotar o uso de controles de acesso a dados, sobre tudo, DBGrid e DBChart.

Uso pleno da linguagem SQL neste Front-End.

Entender e aplicar os métodos únicos de acesso a dados para um SGDB.

Adotar controle de tratamento de erro de forma global e centralizada.

Construir aplicação para distribuição do software desenvolvido, bem como na camada de acesso a dados(BDE).

Resumir todo o aprendizado em torno de um "projeto de aula", com características de uma aplicação comercial real.

Público alvo

Desenvolvedores, programadores, analista de Sistemas, estudantes de informática.

Benefícios

Será capaz de aplicar o conhecimento conceitual de uma ferramenta de quarta geração. Com exercícios introdutórios, aspectos da modelagem da interface serão alcançados. Consolidando o treinamento, o participante terá contato com o que há de mais avançado para acesso e anipulação de dados, culminando com o desenvolvimento no modelo client server (c/s).

Metodologia de ensino

Exposição interativa com apresentação de estudo de casos e exercícios práticos, propiciando um suporte para elucidação de dúvidas durante e após o término imediato do curso.

Pré requisitos

Não existem pré requisitos mandatários para este treinamento; no entanto, experiência de trabalho em programação, conhecimento de lógica formal e algoritmos são recomendados.

Conteúdo Programático

1. Introdução ao Ambiente Delphi

- Conceito de programação visual.
- Conceito de programação OOP.
- O potencial de ambiente Delphi.
- Entendendo o objeto (propriedades, métodos e eventos).
- Implementando formulários com controles básicos (menus, status-bar e painéis).
- Compilando a aplicação.

2. Fundamentos

- Conhecendo a Estrutura da Unit.
- Controle de variáveis.
- Depurando a aplicação.
- Usando funções de Conversão.

3. Acesso Clássico aos Dados

- MiddleWare necessário BDE dbExpress
- O Software Borland Database Engine
- Usando o gerenciador SQL Explorer.
- Criando conectividade.
- Implementando formulário para acesso e manutenção de dados.
- A estrutura básica pelo uso dos controles TDataBase, TTable e TDataSource combinados com os diversos Data Controls.
- Conhecendo e aplicando os diversos métodos de conectividade e manipulação de dados.
- Usando todo o potencial das propriedades de TDataField, tanto para aspectos visuais quanto funcional.
- Tratando as regras de validação pelo uso das propriedades de TdataField.

4. Acesso a Dados para o Modelo C/S

- Diferença entre Banco de Dados de processamento local e Banco de Dados Cliente Servidor.
- Parametrização (propriedades) dos controles de dados para acesso a um SGDB.
- Controlando Transações.
- Usando TQuery para acesso.
- Usando TSQLUpdate combinando com TQuery para manutenção de dados.
- Usando TSQLStoredProcedure para acesso a Stored Procedure.
- Tratando Exception de Servidor.
- Aplicando e entendendo ""Cached Update" (bufeização de dados).
- Uso pleno da linguagem SQL.
- Desenvolvimento de um módulo do projeto de aula que implementa todos os conceitos indicados aqui.



5. Desenvolvimento de Relatórios

- Os componentes do TQReport.
- Gerando relatório básico (lista).
- Gerando relatório completo (grupo).
- Gerando relatório com quebra e sub-totalização.
- Controlando a aparência dos relatórios.

6. Outros Módulos do Projeto de Aula

- Rotina para tratamento de erros, com exibição em português.
- Desenvolvimento de um módulo para exibição de dados em formato gráfico.
- Usando o repositório do Delphi para armazenar modelos reutilizáveis.

7. Módulos Especiais (conceito)

- Evoluindo do modelo cliente Servidor para o modelo de Objetos Distribuídos.
- Criando servidor COM+.
- Implementando um Cliente COM.
- Introdução ao Desenvolvimento Web.

Carga Horária: 40 horas

Mini Currículo:

Rômulo Francisco Maia.

Diretor Executivo da Trevor Tecnologia e Consultoria Financeira LTDA.

Mestre pela UFMG em Ciência da Computação com ênfase em Usabilidade.

Especialização em Análise de Sistemas.

Especialização em Ensino Superior.

Mais de 15 anos como Analista programador, sendo que há oito anos vem usando a linguagem DELPHI.

Possui patente de Software junto ao INPI de um sistema de prontuário eletrônico de pacientes, com a UNIFESP – Escola Paulista de Medicina, software desenvolvido em DELPHI.

Também possui experiência de mais de 15 anos como professor universitário e coordenador de curso superior.

Currículo completo disponível no site do CNPq Lattes.

Membro do CodeGear GROUP.