CENTRO PAULA SOUZA
GOVERNO DO ESTADO DE SÃO PAULO

**Faculdade de Tecnologia de Americana**
**Curso de Bacharelado em Análise de Sistemas e Tecnologia da Informação**

# BENEFÍCIOS DA UTILIZAÇÃO DE INTERFACES GRÁFICAS MODERNAS EM BASES DE DADOS LEGADAS ORIGINADAS DO COBOL

**AILTON JOSÉ DO AMARAL FILHO**

**Americana, SP**
**2012**

CENTRO PAULA SOUZA
GOVERNO DO ESTADO DE SÃO PAULO

**Faculdade de Tecnologia de Americana**
**Curso de Bacharelado em Análise de Sistemas e Tecnologia da Informação**

# BENEFÍCIOS DA UTILIZAÇÃO DE INTERFACES GRÁFICAS MODERNAS EM BASES DE DADOS LEGADAS ORIGINADAS DO COBOL

**AILTON JOSÉ DO AMARAL FILHO**

**ailton12319@hotmail.com**

**Projeto desenvolvido em cumprimento curricular da disciplina Projeto de Graduação do Curso de Bacharelado em Análise de Sistemas e Tecnologia da Informação da FATEC – Americana, sob orientação do Profº. José Alberto Florentino Rodrigues Filho**

**Área: Banco de Dados**

**Americana, SP**
**2012**

# AGRADECIMENTOS

*Dedico este Trabalho a Deus que me Iluminou para realizá-lo, dedico também a todos os professores que tive na FATEC durante estes quatro anos de Faculdade, pois de certa forma contribuíram para a conclusão deste trabalho*

*Gostaria de Dedicar este Trabalho principalmente ao meu professor Orientador José Alberto Rodriguez que me orientou e me mostrou as diretivas corretas para o desenvolvimento deste trabalho*

*Outras pessoas a quem dedico este trabalho com muito carinho são meus pais, minhas duas irmãs e a todos os funcionários e amigos da FATEC-AM que de maneira indireta também cooperaram para que se fosse realizado este trabalho.*

# RESUMO

Embora seja muito antiga, a linguagem COBOL, ainda é utilizada em larga escala por muitas organizações, principalmente por empresas de grande porte como, por exemplo: IBM, Petrobras e Itaú. As informações que são armazenadas nessas e outras empresas de pequeno porte decorrem de um grande período de tempo. Apesar da linguagem COBOL apresentar diversos benefícios como alto desempenho, segurança e até mesmo documentação (as linhas de comando são muito intuitivas e provem do inglês, facilitando interpretação para os programadores), existe resistência por parte de alguns usuários com relação a sua interface, já que é uma linguagem um tanto quanto antiga, não possui uma interface arrojada e moderna como muitos usuários modernos desejam. Com o objetivo de demonstrar uma alternativa de integração da base de dados COBOL (arquivos isam) com outra linguagem em que a idéia de interface é mais moderna se comparada ao COBOL, além de aproveitar toda a base de dados do COBOL. Propõe ainda a utilização da ferramenta Parkway Connectware como conector de dados ODBC, onde a mesma projeta a base de dados COBOL de forma relacional. Qualquer linguagem que possua acesso ao administrador de dados ODBC conseqüentemente tem acesso ao Parkway Connectware. Neste estudo, além deste conector, serão utilizadas a base de dados COBOL e a ferramenta Delphi.

**PALAVRAS-CHAVES:** COBOL, Delphi, Parkway Connectware  e bases de dados.

# *ABSTRACT*

*Although the COBOL language is a very old, it is still used on a large scale by many organizations, especially large companies such as: IBM, Petrobras and Itaú. The information that is stored in these and other small businesses came through a great period of time. Despite the COBOL language various benefits such as high performance, security and even documentation (command lines are very intuitive and comes from English, facilitating interpretation for the programmers), there is resistance from some users with respect to its interface, as it is a somewhat old language, it does not have a bold and modern interface as many newer computer users desire. Aiming to correct this language "deficiency", this work proposes an integration of COBOL database (isam files) with other language in which the idea of interface is more modern compared to COBOL, and enjoy the whole basis of COBOL data. It also proposes the use of the Parkway Connectware tool as ODBC data connector, where it projects the COBOL database on a relational form. Any language that has administrator access to ODBC data therefore has access to the Parkway Connectware (if properly configured). A study will be done using this data connector previously mentioned, obviously the COBOL database and the Delphi tool. The advantage of this idea is the flexibility of using the same database in both COBOL and Delphi.*

***KEYWORD:*** *COBOL, Delphi, Parkway Connectware and data bases*

# INDÍCE DE TABELAS

Tabela 1 Tabela Demonstrativa das Divisões Cobol.....4

Tabela 2 Escopo Geral do Sistema de Gestão Escolar da ConnectCob.....15

Tabela 3 Dicionário de Dados da Entidade Responsa.....24

Tabela 4 Dicionário de Dados da Entidade Alunos.....25

# INDÍCE DE FIGURAS

Figura 1 : Diagrama genérico envolvendo todos os módulos do Sistema de Gestão Escolar.....15

Figura 2 : Diagrama envolvendo processos do contas à receber.....16

Figura 3 : Diagrama envolvendo processos do contas à pagar.....17

Figura 4 : Diagrama entidade relacionamento dos arquivos usados na aplicação exemplo....19

Figura 5: Arquivo Responsa – *file discription* no COBOL..... 20

Figura 6: Arquivo Responsa – estrutura de chaves no COBOL.....21

Figura 7a: Arquivo Alunos – *file discription* no COBOL..... .22

Figura 7b: Arquivo Alunos – *file discription* no COBOL.....22

Figura 8: Arquivo Alunos – estrutura de chaves no COBOL.....23

Figura 9: Ilustração de Funcionamento da Ferramenta *Parkway Connectware* .....27

Figura 10: Tela de acesso ao *Parkway Connectware*.....28

Figura 11: Acesso a Fonte de Dados WL.....28

Figura 12: Conexão a fonte de dados WL.....29

Figura 13: Criação da Tabela Responsa na Fonte de dados WL.....29

Figura 14: Conexão do *Parkway Connectware* com a tabela RESPONSA da base de dados COBOL.....30

Figura 15: Arquivo RESPONSA projetado no *ParkwayConnectware*.....30

Figura 16: Conexão Delphi com o *ConnectWare*.....31

Figura 17: Conexão com o componente *Query*.....32

Figura 18: Conexão com o componente *DataSource*.....32

Figura 19: Procedure no *Delphi* exemplificando leitura de arquivo(tabela).....33

Figura 20: Script no Delphi exemplificando inserção de registro na tabela RESPONSA.....34

Figura 21: Tela de Cadastro de Responsáveis na aplicação de Gestão Escolar.....35

Figura 22: Tela de Cadastro de Alunos na aplicação de Gestão Escolar.....35

Figura 23: Gráfico do tipo Pizza especificando proporcionalmente
a situação dos alunos.....36

Figura 24: Relatório de Alunos.....36

# SUMÁRIO

**1 INTRODUÇÃO .....1**

**2 REVISÃO BIBLIOGRÁFICA.....2**

2.1 LINGUAGEM DE PROGRAMAÇÃO COBOL.....2

2.1.1 SURGIMENTO DA LINGUAGEM COBOL.....2

2.1.2 CARACTERÍSTICAS DA LINGUAGEM COBOL.....3

2.1.3 ESTRUTURA DE UM PROGRAMA COBOL.....4

2.1.3.1 *IDENTIFICATION DIVISION* (Divisão de Identificação).....5

2.1.3.2 *ENVIRONMENT DIVISION* (Divisão de Ambiente ).....5

2.1.3.3 *DATA DIVISION* (Divisão dos Dados).....5

2.1.3.4 *PROCEDURE DIVISION* (Divisão de Processamento).....5

2.2 A LINGUAGEM COBOL NO CENÁRIO ATUAL .....6

2.3 A FERRAMENTA DE DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES DELPHI.8

2.3.1 A FERRAMENTA RAD *DELPHI*.....8

2.4 CARACTERÍSTICAS DA FERRAMENTA *DELPHI*.....10

2.5 ALTERNATIVAS PARA MODERNIZAÇÃO DE SISTEMAS COBOL.....11

**3 METODOLOGIA , ANÁLISE DE DADOS E PROPOSTA DE INTERFACE.....13**

3.1 CONTEXTO HISTÓRICO DA CONNECTCOB, AVALIACÃO DO SISTEMA E ANALISE PARCIAL DOS DADOS.....13

3.1.1 APRESENTAÇÃO E CARACTERÍSTICAS DA EMPRESA.....13

3.1.2 VISUALIZAÇAO GENÉRICA DO SISTEMA DE GESTÃO ESCOLAR .....15

3.1.3 ARQUIVOS COBOL A SEREM UTILIZADOS NO PROCESSO DE MODERNIZAÇÃO GRÁFICA.....18

3.1.4 DICIONÁRIO DE DADOS DOS ARQUIVOS COBOL.....24

3.1.5 ESPECIFICAÇÕES TÉCNICAS DAS FERRAMENTAS UTILIZADAS NO PROCESSO DE MODERNIZAÇÃO GRÁFICA.....26

3.1.6 PROCESSO DE CONEXÃO DO PARKWAY CONNECTWARE COM O ARQUIVO RESPONSA DA BASE DE DADOS COBOL.....27

3.1.7 CONEXÃO DA BASE DE DADOS WL COM A FERRAMENTA DELPHI E MANIPULAÇÃO DO ARQUIVO COBOL RESPONSA NO DELPHI.....31

**4 CONCLUSÃO.....37**

**REFERÊNCIAS BIBLIOGRÁFICAS.....39**

# INTRODUÇÃO

Este trabalho objetiva evidenciar o surgimento da linguagem COBOL, a importância que a mesma tem no cenário comercial tanto antigamente como nos dias atuais. Será abordado como a COBOL se tornou uma das principais linguagens e por que até hoje existe uma larga escala de empresas com desenvolvimento de sistemas em COBOL.

O conteúdo do trabalho baseia-se em estudos de José E.M. de Carvalho estudioso da linguagem COBOL , assim como nas experiências de William Pereira Alves, estudioso da ferramenta Delphi. Serão evidenciados também determinados empecilhos que o processo de migração da base de dados COBOL para outras bases de dados acarreta, assim como os custos envolvidos em um processo de modernização gráfica de interface COBOL utilizando-se ferramentas em que há a necessidade de aquisição de licenças.

A empresa fictícia CobSolutions será o foco do trabalho, sendo que esta é uma software house que desenvolve sistemas em linguagem COBOL para o setor comercial a mais de 20 anos. **O objetivo geral** do trabalho é avaliar parte da base de dados de um sistema voltado para administração escolar e propor uma adaptação visual desse sistema.

Os **objetivos específicos** são:

- Análise das características e história da linguagem COBOL

- Análise das características e história da ferramenta de desenvolvimento Delphi

- Estudo da ferramenta *Parkway Connectware* e demonstração de sua utilização

- Desenvolvimento de um protótipo de um sistema escolar em Delphi com acesso a uma base de dados COBOL

# 2 REVISÃO BIBLIOGRÁFICA

Nesta seção será abordada características e particularidade da linguagem COBOL bem como da ferramenta RAD Delphi.

## 2.1 LINGUAGEM DE PROGRAMAÇÃO COBOL

Nesta seção, será contextualizada a história e surgimento da linguagem COBOL (*Common Business Oriented Language*), assim como suas particularidades e aplicação no cenário atual.

### 2.1.1 SURGIMENTO DA LINGUAGEM COBOL

A linguagem COBOL foi criada por uma comissão de investigadores de diversas instituições civis no final dos anos 50, mais precisamente na segunda metade do ano de 1959. Em um evento denominado CODASYL (*Conference on Data Systems Language*) patrocinado pelo Pentágono Americano em maio de 1959 diversos fabricantes de computadores e órgãos governamentais americanos se reuniram para sugerir o desenvolvimento de uma linguagem voltada para os negócios.

Depois de diversas congregações foi constituído um novo subcomitê composto por seis pessoas: William Selden e Gertrude Tierney da IBM (multinacional do ramo de Tecnologia da Informação), Howard Bromberg e Howard Discount da RCA(empresa americana pioneira no setor de telecomunicações) e Vernon Reeves e Jean E. Sammet da Sylvania Eletric Products(empresa americana do segmento de produtso elétricos). Esses cidadãos foram os que completaram as especificações da linguagem COBOL.

De acordo com Carvalho (1991) , a principal vantagem que a COBOL trouxe foi uma padronização, ou uniformização entre os fabricantes de computadores. Antes da COBOL a mudança de um computador de determinada marca para um de outra marca impactava na mudança de todos os programas.

A COBOL teve inicialmente suas especificações baseadas na linguagem FLOW-MATIC e COMTRAM originado em 1960 a primeira versão denominada COBOL 60 e sendo posteriormente rapidamente corrigida e substituída pela versão COBOL 61.

A padronização final da COBOL realizada pela *American National Standards Institute* (ANSI) foi feita em 1989.

Para Mccracken e Garbassi (1978), a COBOL foi projetada para sanar ou ao menos atenuar determinados problemas de redação de programas, compatibilidade de computadores, teste de programas e documentação de sistemas.

Conforme Carvalho (1991) a linguagem COBOL foi idealizada para aplicações comerciais, onde se manipulam grandes arquivos de dados.

Carvalho (1993) afirma que existam aproximadamente 125 bilhões de linhas de códigos em todo o mundo, além de que 90% das aplicações comercias foram escritas em COBOL.

## 2.1.2 CARACTERÍSTICAS DA LINGUAGEM COBOL

A linguagem COBOL desde sua concepção possuía características benéficas como uma boa capacidade de auto-documentação, excelente modelagem de dados se comparada ao contexto e época em que foi lançada, principalmente devido ao uso da palavra reservada PICTURE.

Algumas das desvantagens da linguagem COBOL se comparada as linguagens atuais é a não definição de variáveis locais de funções recursivas, além de que nas primeiras versões o número de palavras reservadas que se tinha na linguagem era elevadíssimo, porém com o tempo e com o surgimento de novas versões o COBOL sanou estes problemas.

Há mais de 20 anos críticos dizem que a linguagem COBOL está em extinção, porém nota-se que o mercado cada vez mais necessita de profissionais especializados em linguagem

COBOL, e em decorrência dessa alta procura de especialistas em COBOL e da baixa quantidade de profissionais na área, a remuneração se torna interessante para os conhecedores da linguagem.

De acordo com Mccracken e Garbassi (1978), pelo fato da simplicidade relativa das sentenças COBOL serem similares às da língua inglesa, a COBOL ajuda qualquer pessoa com uma mentalidade lógica a escrever um programa mais ou menos complexo. Ainda segundo Mccracken e Garbassi (1978), o COBOL atenua diversos problemas de verificação e teste de programa por ser escrito em língua inglesa e, portanto, ser algo mais fácil de verificar antes de usar no computador e mais fácil de depurar durante o teste.

## 2.1.3 ESTRUTURA DE UM PROGRAMA COBOL

Tabela 1: **Tabela demonstrativa das divisões COBOL** (AMARAL, 2012)

| **Divisão** | **Função** |
|---|---|
| *IDENTIFICATION DIVISION* | Identificação do programa fonte. |
| *ENVIRONMENT DIVISION* | Descrição tanto do computador como de determinados periféricos utilizados pelo programa. |
| *DATA DIVISION* | De modo geral faz a descrição de arquivos de entrada e saída. |
| *PROCEDURE DIVISION* | Localização do código manipulador dos arquivos. |

#### 2.1.3.1 *IDENTIFICATION DIVISION (Divisão de Identificação)*

Carvalho (1991) afirma que a *Identification Division* é a divisão que identifica o programa em todas as suas etapas de desenvolvimento. Nesta pode ser inclusa a data do programa, a data da compilação e eventuais comentários.

#### 2.1.3.2 *ENVIRONMENT DIVISION (Divisão de Ambiente )*

Carvalho (1991) argumenta que a *Environment Division* especifica os métodos de expressão dos aspectos do processamento de dados que dependem das características físicas do computador.

#### 2.1.3.3 *DATA DIVISION (Divisão dos dados)*

De acordo com Carvalho (1991) esta divisão armazena todos os dados a serem manipulados e / ou processados pelo programa durante seu processamento. Na COBOL assim como em qualquer outra linguagem há o processamento e armazenamento tanto de dados internos como dados externos, sendo que os dados externos são aqueles presentes em arquivos. Em contrapartida os dados internos consistem em dados adicionais em áreas de armazenamento de memória.

#### 2.1.3.4 *PROCEDURE DIVISION* (Divisão de Processamento)

Responsável pelo processamento de todo o algoritmo desenvolvido e conseqüentemente o local onde há todo o processamento e manipulação dos dados contidos no *data-division*. Para Carvalho (1991) a procedure *division* é o coração dos programas em COBOL, sendo a divisão que através de comandos específicos visa atingir o objetivo final.

## 2.2 A LINGUAGEM COBOL NO CENÁRIO ATUAL

A linguagem COBOL ainda é utilizada de maneira maciça em algumas empresas nacionais como multinacionais. A grande questão é que são poucos que possuem conhecimento nessa linguagem, e devido a este fato há uma supervalorização dos profissionais com conhecimento nessa área.

De acordo com Carvalho (1993) 50% das novas aplicações em desenvolvimento serão escritas e atualizadas em COBOL. Desde muito antes a linguagem COBOL é de certa forma discriminada e dada como morta, porém determinadas evidências do contexto atual comprovam a grande utilização desta linguagem.

Pedro Britto (2012), um gerente de vendas da IBM faz uma alerta ao pessoal da nova geração da computação: a necessidade do aprendizado da linguagem COBOL. Isso porque, segundo Britto a procura de profissionais que ainda conheçam COBOL é constante por parte da organização.

Ainda para Britto (2012) existe um preconceito no mercado com relação a aprendizagem de COBOL, e em detrimento disso a IBM inclusive vem desenvolvendo soluções que facilitem a aproximação com a linguagem, como por exemplo o ambiente Eclipse que viabiliza tanto o desenvolvimento para COBOL como para JAVA.

De acordo com Weissmann (2006) foi realizado um estudo em 1999 e foi constatado que naquele dado momento cerca de 80 % das aplicações corporativas em execução no mundo eram escritas em COBOL .

Um exemplo de uso maciço de uso da linguagem COBOL é na bolsa de valores de Nova Iorque, sendo que boa parte de seus sistemas para operações financeiras possuem programas desenvolvidos em linguagem COBOL, sendo que estes programas executam em Mainframes da IBM.

Segundo Rodrigues (2007) a longevidade do COBOL se deve principalmente ao grande número de aplicações escritas nas mais diferentes versões da linguagem, representando uma porção significativa do legado existente nas grandes empresas.

Celso Cardoso (diretor da C&C Microinformática), citado por Rodrigues (2007) sintetiza em três as razões da permanência da linguagem no mercado: a plataforma é extremamente segura, proporciona enorme legibilidade e se documenta sozinha.

Alexssandro Tolomei – gerente de produtos da empresa DTS Latin America Technology, ainda citado por Rodrigues (2007) afirma que 70 % da receita da DTS provêm de sistemas em COBOL.

Ingo Feslau – diretor da empresa T-Systems , citado também por Rodrigues (2007) afirma que em Blumenau(SC) foi inaugurado um centro objetivando-se manter pessoas treinadas em COBOL, sendo que essa foi uma atitude impulsionada devido a necessidade de mercado.

Em contrapartida um estudo da TIOBE Software (2012), evidencia que a linguagem COBOL já não está tão no topo se comparada à alguns anos atrás. Segundo esse estudo da TIOBE que foi publicado em setembro de 2012, o qual objetiva identificar a popularidade das linguagens de programação, a linguagem C é a mais popular e o COBOL se encontra na posição de número 25. É importante ressaltar que esse é um estudo genérico, já que o uso maciço da linguagem COBOL se dá no setor comercial.

É importante ressaltar que muitas ferramentas COBOL já foram modernizadas de modo a suportarem sistemas distribuídos. A Micro Focus também desenvolveu o Visual COBOL que possibilita a expansão das aplicações COBOL a novas plataformas e em qualquer ambiente de desenvolvimento, porém essa solução apresenta um custo relativamente alto.

## 2.3 A FERRAMENTA DE DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES DELPHI

Nesta seção será abordada a ferramenta de desenvolvimento Delphi, bem como suas características, história e contexto tecnológico em que foi lançada.

### 2.3.1 A FERRAMENTA RAD DELPHI

Certamente pode-se dizer que o Delphi foi um produto inovador, principalmente se comparada ao contexto tecnológico em que foi lançado. Delphi teve seu surgimento na década de 90, seu lançamento certamente foi extremamente esperado, pois era um ambiente de desenvolvimento com recursos avançados se comparados ao da época. A interface que o Delphi proporciona é extremamente agradável do ponto de vista do usuário.

De acordo com Norton (1997) o *Delphi* surgiu como uma tentativa de levantar a *Borland* como empresa (objetivo alcançado), além de tratar-se de um derivado do Pascal para os anos 90. É importante salientar que o *Pascal* é uma linguagem extremamente antiga e utilizada muito como ferramenta didática.

Assim como a linguagem JAVA é baseada em C, a ferramenta Delphi é toda baseada em Pascal. De acordo com Norton (1997) o Pascal é derivado da PL/1 (Programming Language 1) e do ALGOL (ALGOrithmic Language). A linguagem Pascal recebeu esse nome em homenagem a Blaise Pascal – um matemático francês que, em 1690 com apenas 18 anos criou a primeira máquina de calcular.

O Prof. Niklaus Wirth da Universidade Técnica de Zurique foi quem inventou a linguagem Pascal aproximadamente no inicio da década de 70. Com o surgimento dos microcomputadores aproximadamente no final da década de 70 a linguagem Pascal foi sendo cada vez mais inserida no contexto computacional.

Norton (1997) argumenta que o compilador necessário para usar o Pascal exigia muito pouca memória. Porém com o passar do tempo a linguagem Pascal acabou sendo um pouco esquecida e de certa forma deixou de ser usada em escolas técnicas e universidades como instrumento de aprendizado.

Ainda segundo Norton (1994), a comunidade de programação como um todo ficou atenta e entusiasmada com o surgimento do Turbo Pascal em 1984, já que está versão era realmente muito fácil de se utilizar, além de proporcionar um compilador de código nativo e facilitar o acesso aos periféricos comuns como o teclado e a tela.

O Turbo Pascal passou por variadas atualizações, e o fato de ser muito rápido conquistou diversos fãs. Além de proporcionar diversos recursos de Programação Orientada a Objeto e boa funcionalidade.

Segundo Pereira (2003) em 1991 ainda foi lançada outra versão do Turbo Pascal for Windows, algo que já dava sinais de uma investida da Borland em alguma ferramenta baseada em Pascal voltada para o sistema operacional Windows, algo que se concretizou no primeiro semestre de 1995.

Pereira (2003) afirma que o Delphi é uma variação do Borland Pascal 7.0, denominada Object Pascal e que baseia-se num conceito denominado RAD (Rapid Application Development – Desenvolvimento Rápido de Aplicação), um conceito que auxilia principalmente a etapa relacionada à criação da interface com o usuário.

## 2.4 CARACTERÍSTICAS DA FERRAMENTA DELPHI

A ferramenta de desenvolvimento Delphi possui algumas características particulares que facilitam no uso da linguagem. Existem diversos recursos de componentes que acessam

dados, recursos que possibilitam o desenvolvimento de interfaces ainda mais agradáveis do ponto de vista do usuário, além de ser uma linguagem extremamente intuitiva e com uma gama de profissionais que a conhece.

Diferentemente de COBOL onde a mão de obra é escassa, o DELPHI é conhecido por grande parte dos programadores em todo mundo. O benefício desse aspecto é que em diversos fóruns na internet e comunidades de programadores é possível se encontrar conteúdos com uma relativa facilidade se comparada ao COBOL, em contrapartida o mercado está de determinada forma saturado com relação ao Delphi, não remunerando tão bem os profissionais que realmente conhecem esta ferramenta.

De acordo com Pereira (2003) embora o Delphi seja de alto nível o mesma possui determinados recursos que possibilitam um aprofundamento maior nas camadas internas do Windows, como possibilidade de chama às funções da API do Windows e de criação de rotinas em Assembly.

Outra vantagem significativa do Delphi é a reutilização de código fonte. É possível no Delphi a criação de um método ou uma rotina e a transformação desta em um componente que pode ser agregado a paleta do Delphi, podendo deste modo inserir o componente em outra eventual aplicação.

Ainda segundo Pereira (2003) o bom desempenho de programas em Delphi está diretamente relacionado à maneira como foi escrita sua biblioteca de componentes visuais (VCL). Houve uma otimização dos algoritmos desses componentes impactando nos desempenho dos programas que utilizam os mesmos.

Outros aspecto a se ressaltar são os componentes de geração de relatórios que o DELPHI proporciona como o QuickReports e o RaveReports, além de possibilitar desenvolvimento para Web, apesar de ser uma linguagem não muito indicada para este tipo de aplicação.

## 2.5 ALTERNATIVAS PARA MODERNIZAÇÃO DE SISTEMAS COBOL

Existem alternativas que não sejam a utilização da ferramenta Parkway Connectware para a modernização de aplicações COBOL, porém na maioria dos casos a inviabilidade financeira dessas soluções acaba influenciando na tomada de decisão dos stakeholders que tentam de alguma se adaptar ao mercado.

Existe uma ferramenta da Brivi (uma empresa brasileira voltada ao mercado COBOL) que objetiva modernizar aplicações COBOL. A ferramenta é denominada isCOBOL, a mesma possui diversos componentes que de certa forma possibilita relativa portabilidade a aplicações COBOL.

Dentre os componentes que ela oferece está o isCOBOL Compiler que consisti em um compilador COBOL compatível com os padrões ANSI, talvez a principal vantagem desse compilador é a tradução do código fonte para uma classe JAVA, característica que possibilita maior flexibilidade as aplicações COBOL.

Outra ferramenta é a isCOBOL IDE que viabiliza ao desenvolvedor um ambiente baseado em Eclipse, auxiliando em todas a fases da criação de um aplicativo desde o desenho, codificação, até mesmo a execução.

O isCOBOL ESQL Generator também é uma ferramenta interessantíssima pois possibilita que toda a codificação de acesso a dados COBOL seja convertido para sintaxes SQL, possibilitando o acesso a qualquer fonte de dados sem a necessidade de alteração de código fonte.

Outra Ferramenta também relevante é a MicroFocus Visual COBOL, que também auxilia na modernização de aplicativos COBOL, oferecendo diversas opções para desenvolvimento COBOL como .NET, UNIX, LINUX, Java Virtual Machine e Cloud Computing.

Através da utilização de uma IDE o Visual COBOL permite a escrita do código COBOL em uma ambiente mais moderno, podendo se desenvolver através do Microsoft Visual Studio, Eclipse e JVM.

Outra ferramenta também da Micro Focus é o NET Express que por sua vez é a última versão do compilador COBOL para desenvolvimento em ambiente Windows, este compilador

possui ambiente integrado para facilitar o desenvolvimento de todos os tipos de aplicações tanto com telas padrão Windows como WEB(HTML).

O Net Express possibilita também o aproveitamento dos fontes COBOL de modo que ao recompilá-lo é possível compatibilizar o mesmo para ambiente Windows, esta solução da Micro Focus é utilizada em larga escala, um exemplo clássico de utilização do Net Express é o bankline do Itaú.

Dentre outras as citadas acima são as principais ferramentas para modernização de aplicações COBOL, Estas ferramentas são de fato interessantíssima, porém os custos envolvidos nesses processos são relativamente altos se atribuídos a empresas de desenvolvimento de pequeno e médio porte, principalmente porque geralmente são pagas licenças por máquinas, estações de trabalho e a maioria dos clientes não se disponibilizam a pagar valores estratosféricos para se obter uma modernização gráfica, apesar de cobrarem essa adaptação a todo o momento dos desenvolvedores.

Em decorrência desses aspectos o trabalho propõr a utilização da ferramenta Parkway Connectware juntamente com a ferramenta Delphi para sanar estes problemas de adaptação a novos recursos gráficos utilizando-se dados COBOL, já que o custo não é tão exorbitante se comparada às ferramentas descritas acima.

# 3 METODOLOGIA, ANÁLISE DE DADOS E PROPOSTA DE INTERFACE

Nesta seção será abordada o contexto histórico da ConnectCob, o estudo e proposta de uma moderna interface e análise parcial dos dados.

## 3.1 CONTEXTO HISTÓRICO DA CONNECTCOB

Nesta seção será abordado alguns aspectos relevantes no que diz respeito a empresa ConnectCob desde a sua fundação até os dias de hoje, avaliando de forma objetiva as necessidades que se existia por parte de seus clientes com relação ao processamento de dados naquela época e as particularidades que existem hoje que acabam prejudicando a empresa pelo fato de o seu sistema ser considerado ultrapassado.

Será também realizada um avaliação parcial do sistema de administração escolar e análise parcial dos dados, com explicações de funcionamento dos arquivos COBOL, ou seja, a estruta do arquivo propriamente dita como os tipos utilizados em determinados campos e sua organização de chaves. Será feito também um dicionário de dados dos arquivos utilizados para a melhor compreensão da existência de determinados campos.

### 3.1.1 APRESENTAÇÃO E CARACTERÍSTICAS DA EMPRESA

A empresa ConnectCob atua no setor de desenvolvimento de software há mais de 20 anos, onde um dos propietários antes mesmo de fundar a empresa trabalhava no CPD(Central de Processamento de Dados) de uma empresa do ramo têxtil. Esse sujeito começou a trabalhar com computação em seu surgimento. Sistemas em COBOL, naquele contexto tecno-mercadológico era o que se tinha de mais interessante no momento. O mesmo realizou toda a informatização dessa empresa têxtil, ou seja, todas as rotinas administrativas que eram realizadas de maneira manual como Contas a Receber, Contas a Pagar, Faturamento, Controle de Estoque entre outros foram adaptados a passaram a ser realizados via Computador.

Os proprietários daquela empresa se entusiasmaram com toda aquela informatização e o profissional que realizou todo aquele processo adaptativo ganhou ainda mais respeito. Posteriormente o mesmo decidiu abrir uma empresa(ConnectCob) de desenvolvimento e deixou de trabalhar no CPD dessa grande empresa. Algum tempo depois adquiriu novos clientes desse mesmo grupo de investidores, dentre eles uma organização que atuava no setor educacional, um colégio.

Com o passar dos anos os filhos desses investidores assumiram estas empresas e foi nesse momento que a ConnectCob se viu extremamente pressionada. Com um conceito novo

em tecnologia esses herdeiros não compactuavam muito com a interface que os sistemas ConnectCob proporcionava e começaram a pressionar no sentido de obter um aspecto visual mais moderno e mais compatível com as tendências atuais. Nesse momento que a linguagem COBOL apresentou sua maior desvantagem: a sua interface arcaica se comparada as atuais.

Apesar de a ConnectCob ainda permanecer forte com esse grupo de investidores (única e exclusivamente porque desenvolve soluções sob-medida), a pressão por uma interface moderna se acresce proporcionalmente ao passar dos anos e a empresa precisa encontrar uma solução.

Como citado anteriormente as ferramentas Parkway Connectware e Delphi serão utilizadas com o objetivo de solucionar este problema, e para isso alguns aspectos deverão ser analisados, como demostrado nos itens seguintes.

### 3.1.2 VISUALIZAÇÃO GENÉRICA DO SISTEMA DE GESTÃO ESCOLAR

Na figura 1 é demosntrado um diagrama genérico envolvendo todos os módulos que contemplam o sistema de gestão escolar da ConnectCob

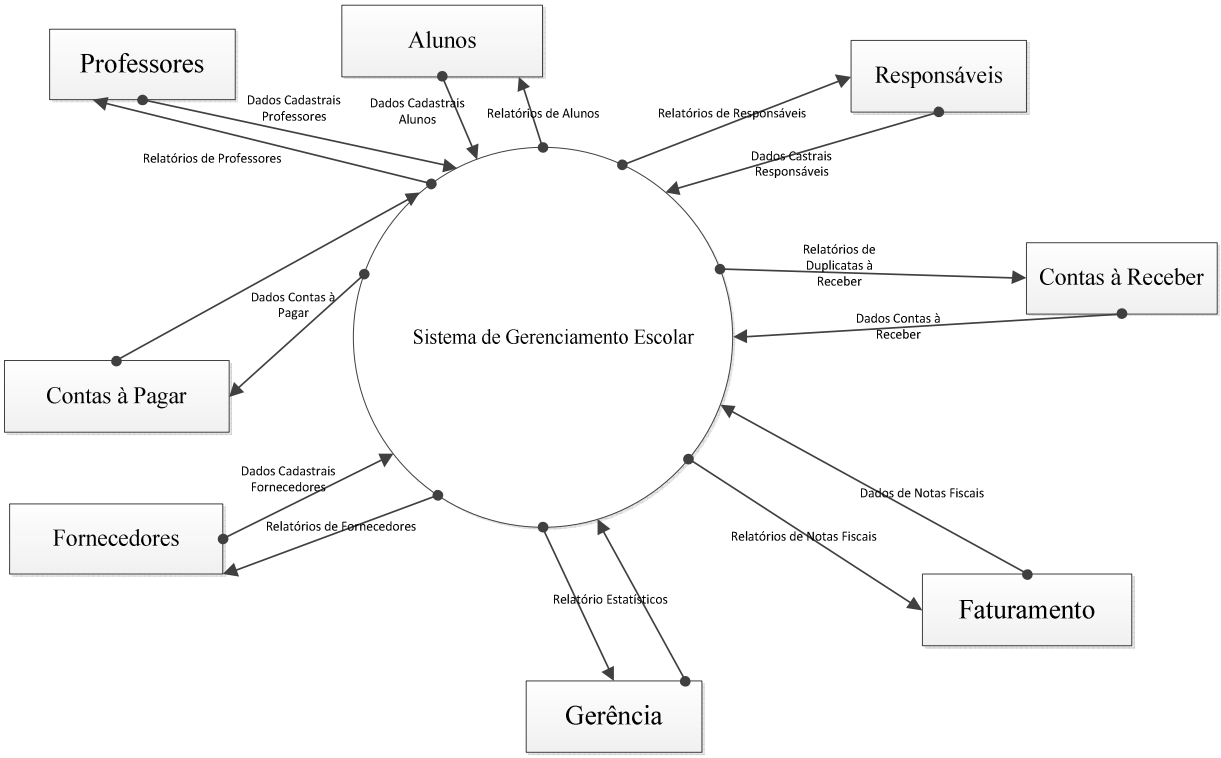



Figura 1**: Diagrama genérico envolvendo todos os módulos do Sistema de Gestão Escolar** (AMARAL, 2012)

Tabela 2: **Escopo geral do sistema de Gestão Escolar da ConnectCob** (AMARAL, 2012)

| **Objetivo Geral do sistema** |
|---|
| O foco do Sistema de Gestão Escolar é gerenciar todos os aspectos relacionados a uma escola, ou seja, realizar todo um controle de alunos e responsáveis por alunos, prover um cadastro de professores contratados, bem como dados cadastrais de fornecedores, de modo a controlar o financeiro da empresa (contas à pagar e contas à receber) e o faturamento. Todas essas informações servem de base para emissão de relatórios gerenciais. |

Os cadastros de alunos, responsáveis, professores, fornecedores e parâmetros são essenciais para o sistema, pois é a partir destes que será alimentado o contas a receber e o contas à pagar. Na figura 2 e 3 respectivamente são representados de maneira sintética os vínculo desses cadastros com o contas à receber e o contas à pagar da empresa.

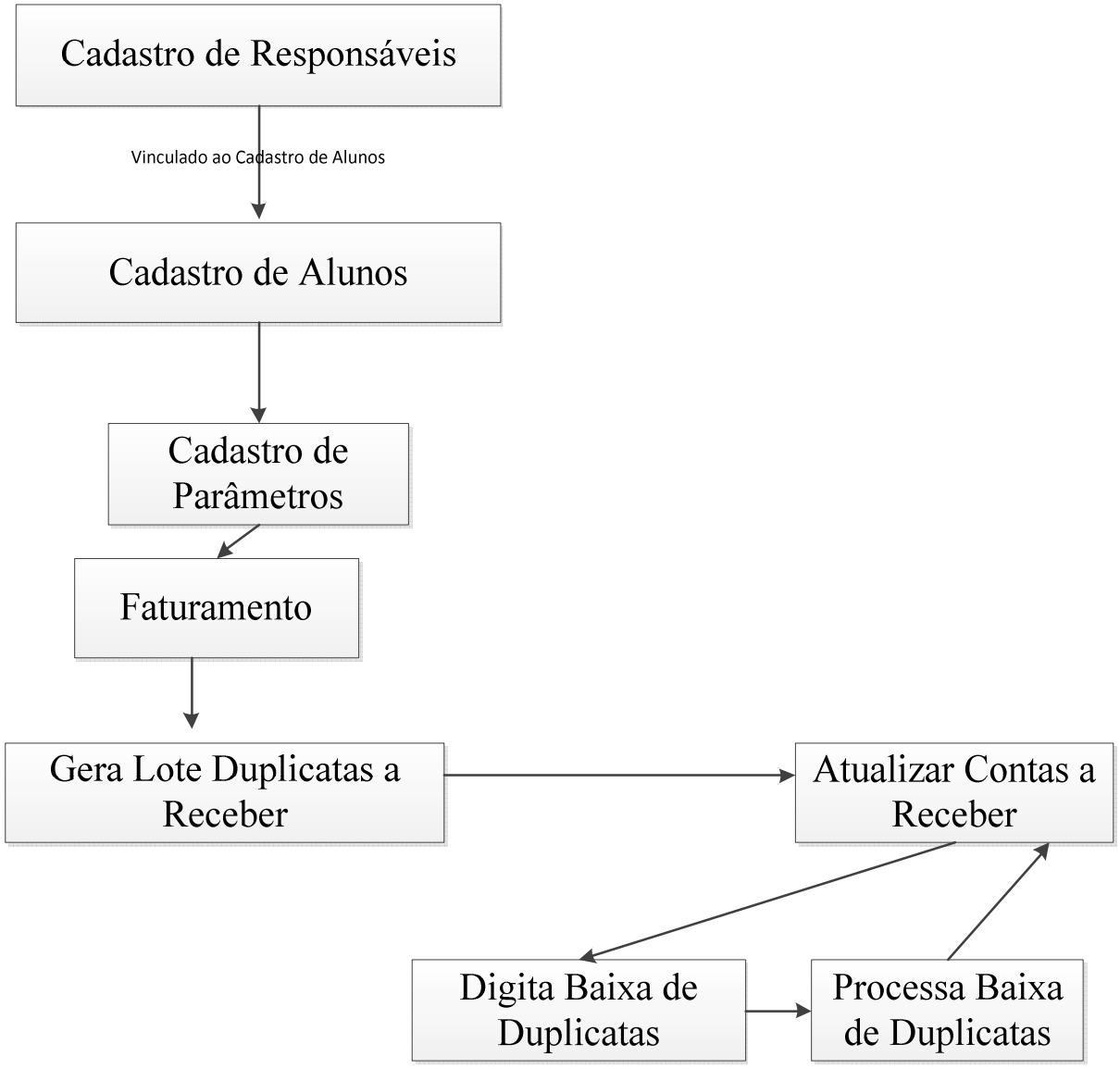



Figura 2**: Diagrama envolvendo processos do contas à receber** (AMARAL, 2012)

No Contas à Receber pode-se perceber que o Cadastro de Alunos é o cadastro de clientes. Cada aluno possui um responsável, e cada aluno está vinculado no cadastro de parâmetros. O faturamento gera um lote de duplicatas com base nas informações dos parâmetros juntamente com as informações dos alunos, atualizando o contas à receber. Naturalmente quando a conta foi recebida é necessário se realizar a digitação de baixa do contas à receber e processar as mesmas

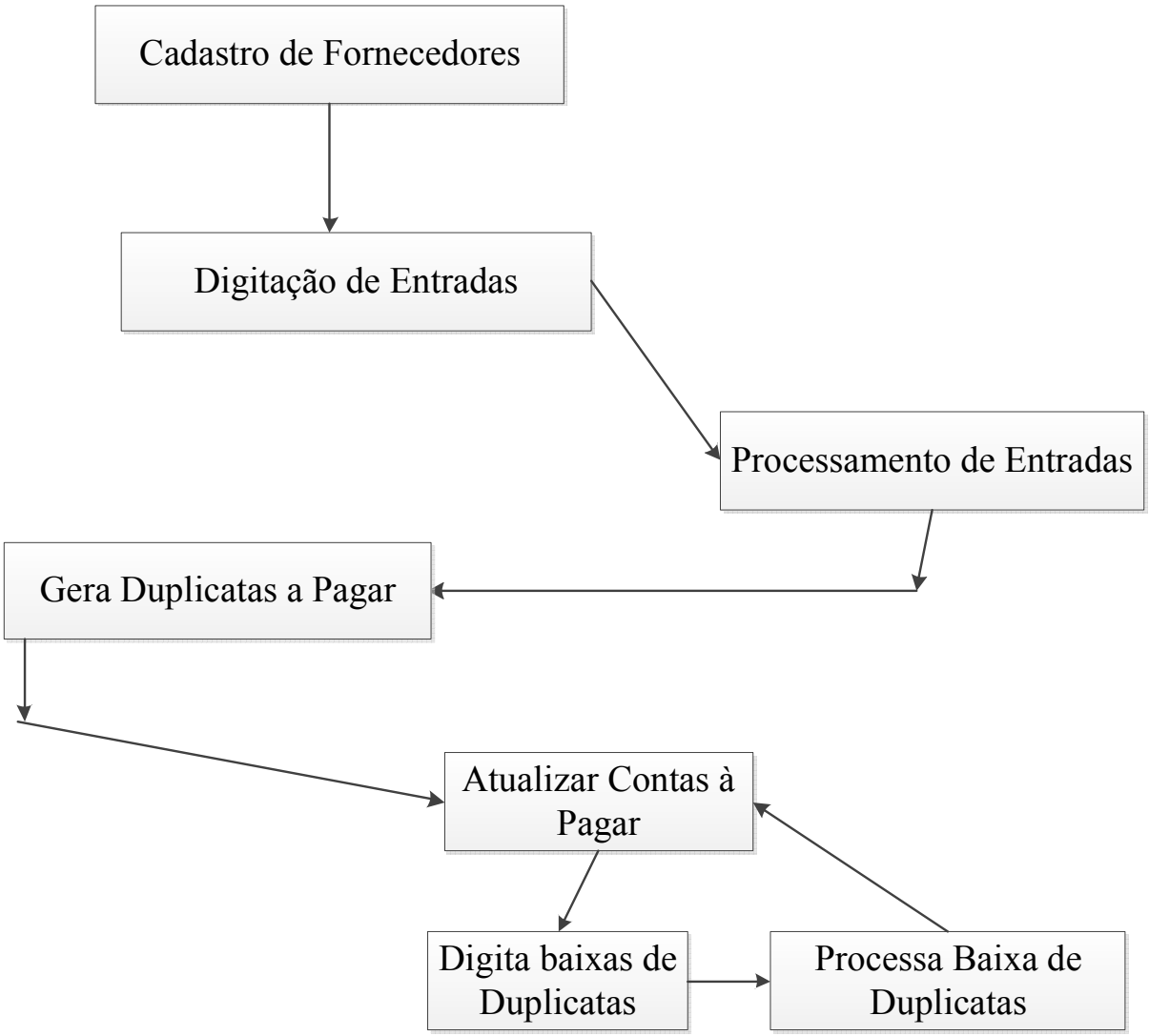



Figura 3**: Diagrama envolvendo processos do contas à pagar** (AMARAL, 2012)

No Contas à Pagar, assim como em toda empresa, é necessário um cadastro de fornecedores. No momento de Digitação de entradas é solicitado o fornecedor e outras informações específicas como valores, data de emissão e valor total de IPI e ICMS. Uma vez digitada, essas entradas são processadas gerando duplicatas à pagar e atualizando o módulo de contas à pagar. Uma vez realizado o pagamento da conta, há uma digitação de baixas de duplicatas e processamento das mesmas.

### 3.1.3 ARQUIVOS COBOL A SEREM UTILIZADOS NO PROCESSO DE MODERNIZAÇÃO GRÁFICA

A aplicação exemplo que será realizada para demonstrar o processo de modernização gráfica utilizará apenas dois arquivos COBOL. O arquivo de alunos e o arquivo de responsáveis serão utilizados. Esses dois arquivos são os principais, pois em muitos outros cadastros essas informações serão requisitadas.

Algumas técnicas do modelo conceitual são extremamente úteis para o entendimento das relações entre as tabelas, em virtude disso será apresentado o DER (Diagrama Entidade Relacionamento) de cada arquivo COBOL, bem como o esquema relacional desses arquivos.

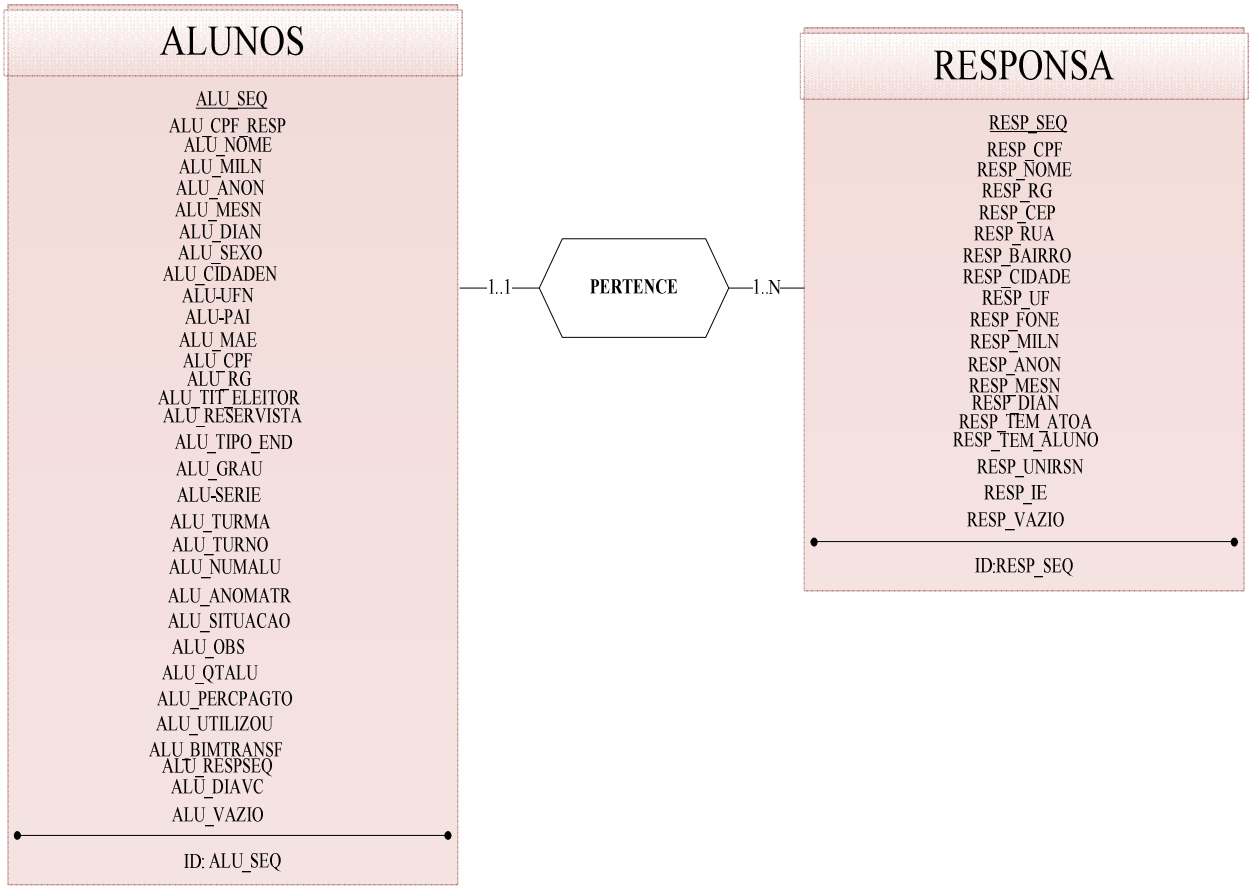



Figura 4**: Diagrama entidade relacionamento dos arquivos usados na aplicação exemplo** (AMARAL, 2012)

### 3.1.3.1 ESQUEMA RELACIONAL

RESPONSA(RESP_SEQ, RESP_CPF, RESP_NOME, RESP_RG, RESP_CEP, RESP_RUA, RESP_BAIRRO, RESP_CIDADE, RESP_UF, RESP_FONE, RESP_MILN, RESP_ANON, RESP_MESN, RESP_DIAN, RESP_TEM_ATOA, RESP_TEM_ALUNO, RESP_UNIRSN, RESP_IE, RESP_VAZIO)

ALUNOS(ALU_SEQ, ALU_CPF_RESP, ALU_NOME, ALU_MILN, ALU_ANON, ALU_MESN, ALU_DIAN, ALU_SEXO, ALU_CIDADEN, ALU_UFN, ALU_PAI, ALU_MAE, ALU_CPF, ALU_RG, ALU_TIT_ELEITOR, ALU_RESERVISTA, ALU_TIPO_END, ALU_GRAU, ALU_SERIE, ALU_TURMA, ALU_TURNO, ALU_NUMALU, ALU_ANOMATR, ALU_SITUACAO, ALU_OBS, ALU_QTALU, ALU_PERCPAGTO, ALU_UTILIZOU, ALU_BIMTRANSF, ALU_RESPSEQ, ALU_DIAVC, ALU_VAZIO )

ALU_RESPSEQ REFERENCIA RESPONSA

### 3.1.3.2 ESTRUTURA DOS ARQUIVOS NO COBOL

Segue abaixo a descrição dos arquivos que serão utilizados na aplicação exemplo, onde existe a especificação dos aspectos físicos dos dados, tamanho e nome e assim por diante no próprio COBOL.

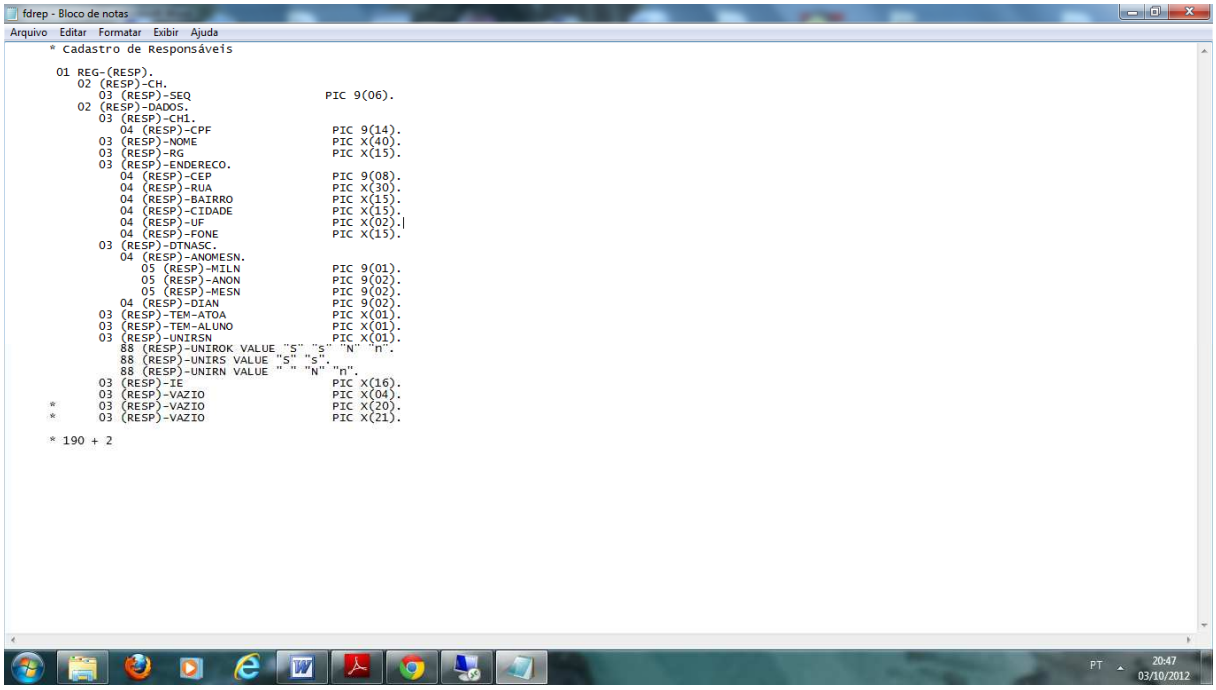



Figura 5**: Arquivo Responsa – file discription no COBOL** (AMARAL, 2012)

Abaixo segue a maneira como se desenvolve a chave principal e alternativa do arquivo Responsa no COBOL.

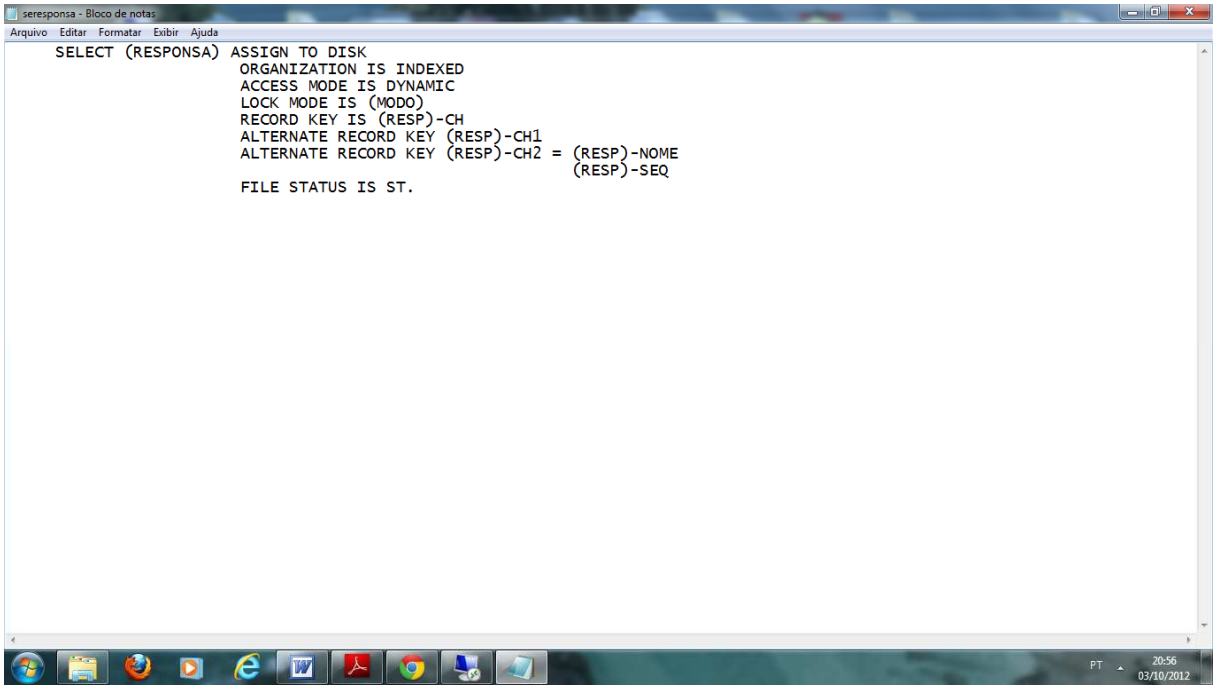
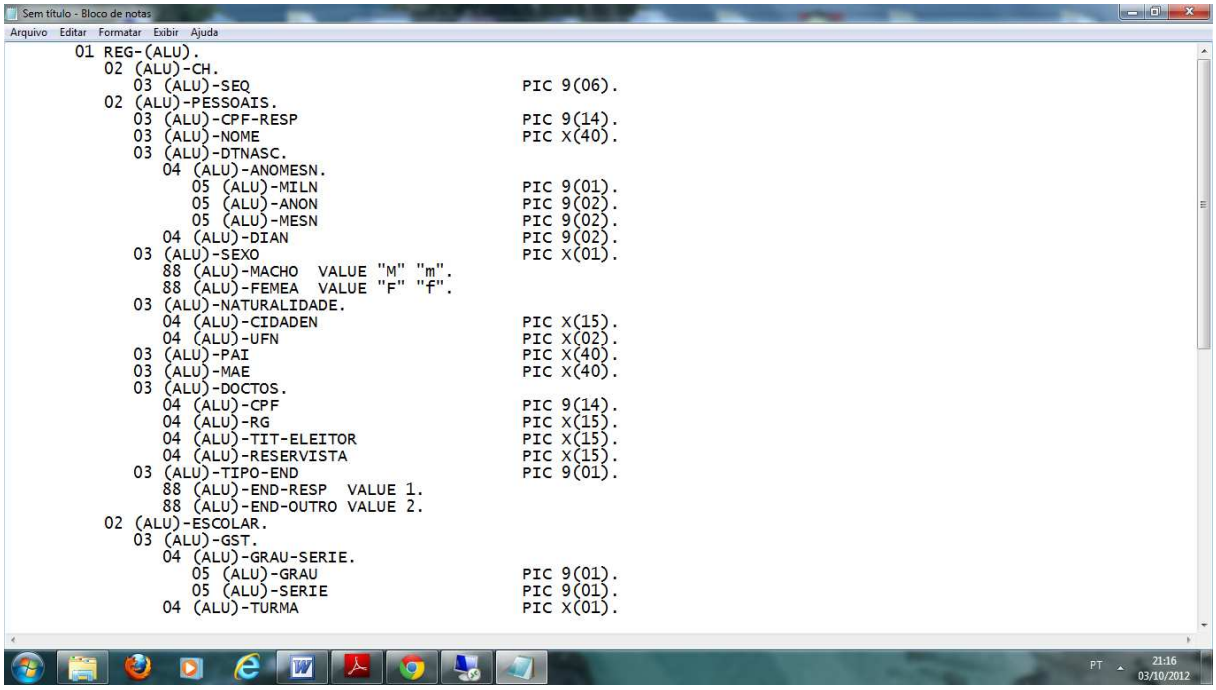


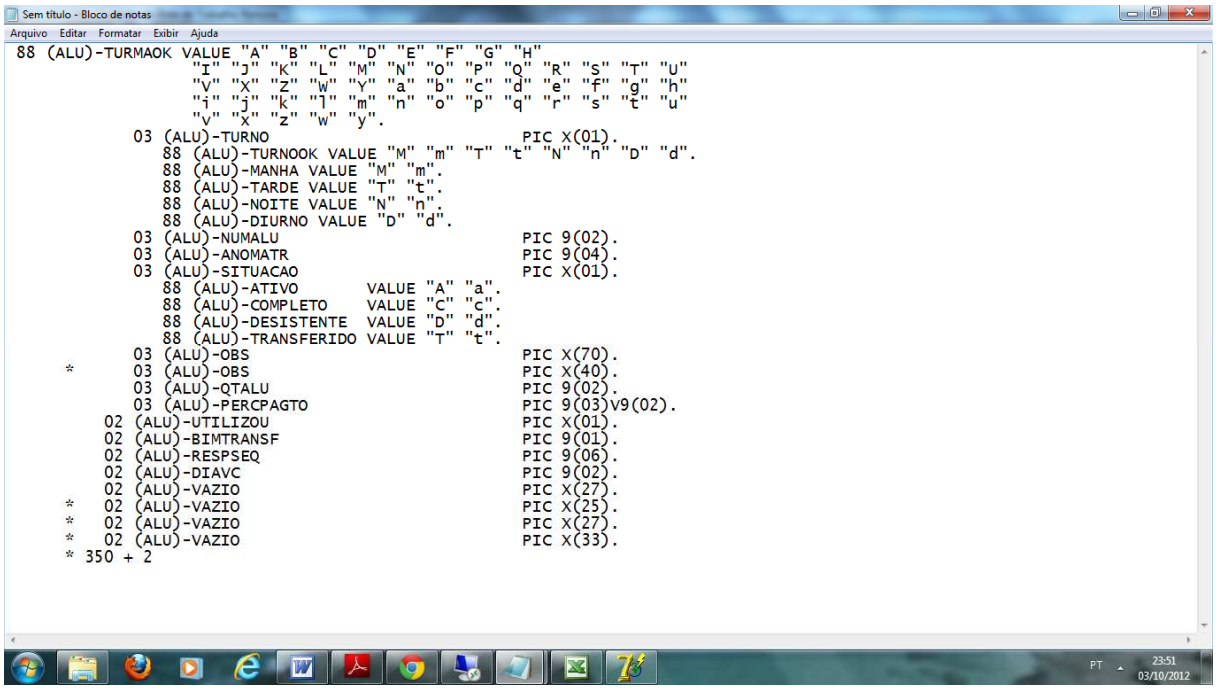

Figura 6**: Arquivo Responsa – estrutura de chaves no COBOL** (AMARAL, 2012)

Como se pode perceber está é a definição de chaves de um arquivo indexado, onde a chave principal é o item de grupo resp-ch que possui como subitem resp-seq. O arquivo ainda possui duas chaves alternativas. A chave 1 – CH1 que é definido no arquivo como o CPF do responsável e a chave 2 que consiste no nome do responsável e seqüência respectivamente.



```
Sem título - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
       01 REG-(ALU).
          02 (ALU)-CH.
             03 (ALU)-SEQ                          PIC 9(06).
          02 (ALU)-PESSOAIS.
             03 (ALU)-CPF-RESP                     PIC 9(14).
             03 (ALU)-NOME                         PIC X(40).
             03 (ALU)-DTNASC.
                04 (ALU)-ANOMESN.
                   05 (ALU)-MILN                   PIC 9(01).
                   05 (ALU)-ANON                   PIC 9(02).
                   05 (ALU)-MESN                   PIC 9(02).
                04 (ALU)-DIAN                      PIC 9(02).
             03 (ALU)-SEXO                         PIC X(01).
                88 (ALU)-MACHO  VALUE "M" "m".
                88 (ALU)-FEMEA  VALUE "F" "f".
             03 (ALU)-NATURALIDADE.
                04 (ALU)-CIDADEN                   PIC X(15).
                04 (ALU)-UFN                       PIC X(02).
             03 (ALU)-PAI                          PIC X(40).
             03 (ALU)-MAE                          PIC X(40).
             03 (ALU)-DOCTOS.
                04 (ALU)-CPF                       PIC 9(14).
                04 (ALU)-RG                        PIC X(15).
                04 (ALU)-TIT-ELEITOR               PIC X(15).
                04 (ALU)-RESERVISTA                PIC X(15).
             03 (ALU)-TIPO-END                     PIC 9(01).
                88 (ALU)-END-RESP  VALUE 1.
                88 (ALU)-END-OUTRO VALUE 2.
          02 (ALU)-ESCOLAR.
             03 (ALU)-GST.
                04 (ALU)-GRAU-SERIE.
                   05 (ALU)-GRAU                   PIC 9(01).
                   05 (ALU)-SERIE                  PIC 9(01).
                04 (ALU)-TURMA                     PIC X(01).
```

Figura 7a**: Arquivo Alunos – file description no COBOL** (AMARAL, 2012)



```
Sem título - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
 88 (ALU)-TURMAOK VALUE "A" "B" "C" "D" "E" "F" "G" "H"
                  "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T" "U"
                  "V" "X" "Z" "W" "Y" "a" "b" "c" "d" "e" "f" "g" "h"
                  "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u"
                  "v" "x" "z" "w" "y".
             03 (ALU)-TURNO                        PIC X(01).
                88 (ALU)-TURNOOK VALUE "M" "m" "T" "t" "N" "n" "D" "d".
                88 (ALU)-MANHA VALUE "M" "m".
                88 (ALU)-TARDE VALUE "T" "t".
                88 (ALU)-NOITE VALUE "N" "n".
                88 (ALU)-DIURNO VALUE "D" "d".
             03 (ALU)-NUMALU                       PIC 9(02).
             03 (ALU)-ANOMATR                      PIC 9(04).
             03 (ALU)-SITUACAO                     PIC X(01).
                88 (ALU)-ATIVO       VALUE "A" "a".
                88 (ALU)-COMPLETO    VALUE "C" "c".
                88 (ALU)-DESISTENTE  VALUE "D" "d".
                88 (ALU)-TRANSFERIDO VALUE "T" "t".
             03 (ALU)-OBS                          PIC X(70).
      *      03 (ALU)-OBS                          PIC X(40).
             03 (ALU)-QTALU                        PIC 9(02).
             03 (ALU)-PERCPAGTO                    PIC 9(03)V9(02).
          02 (ALU)-UTILIZOU                        PIC X(01).
          02 (ALU)-BIMTRANSF                       PIC 9(01).
          02 (ALU)-RESPSEQ                         PIC 9(06).
          02 (ALU)-DIAVC                           PIC 9(02).
          02 (ALU)-VAZIO                           PIC X(27).
      *   02 (ALU)-VAZIO                          PIC X(25).
      *   02 (ALU)-VAZIO                          PIC X(27).
      *   02 (ALU)-VAZIO                          PIC X(33).
      * 350 + 2
```

Figura 7b**: Arquivo Alunos – file discription no COBOL** (AMARAL, 2012)

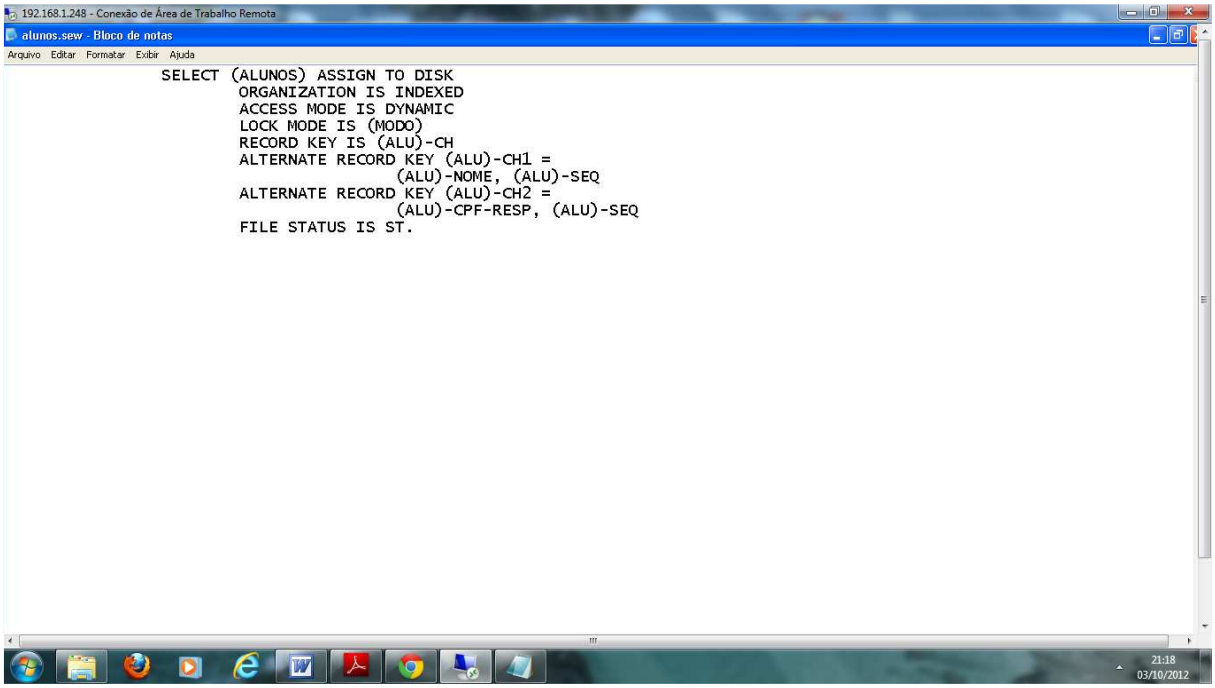



Figura 8**: Arquivo Alunos – estrutura de chaves no COBOL** (AMARAL, 2012)

Neste arquivo a chave principal é o código seqüencial do aluno alu_seq, e possui duas chaves alternativas sendo que a primeira é o nome do aluno e código seqüencial e a segunda é o CPF do responsável e o código seqüencial do aluno respectivamente.

### 3.1.4 DICIONÁRIO DE DADOS DOS ARQUIVOS COBOL

Tabela 3: **Dicionário de Dados da Entidade Responsa** (AMARAL, 2012)

| Atributo | Classe | Domínio | Tamanho | Descrição |
|---|---|---|---|---|
| RESP SEQ | Determinante | Numérico | 6 | Código Sequencial - Chave Principal |
| RESP_CPF | Determinante | Numérico | 14 | CPF/CNPJ do Responsável - Faz parte Chave Alternativa |
| RESP_NOME | Determinante | Texto | 40 | Nome Responsável - Faz parte Chave Alternativa |
| RESP_RG | Simples | Texto | 15 | RG do responsável |
| RESP_CEP | Simples | Numérico | 8 | CEP Responsável |
| RESP_RUA | Simples | Texto | 30 | Rua do Responsável |
| RESP_BAIRRO | Simples | Texto | 15 | Bairro do Responsável |
| RESP_CIDADE | Simples | Texto | 15 | Cidade do Responsável |
| RESP_UF | Simples | Texto | 2 | UF do Responsável |
| RESP_FONE | Simples | Texto | 15 | Fone do Responsável |
| RESP_MILN | Simples | Numérico | 1 | Milênio da data de Nascimento do Responsável |
| RESP_ANON | Simples | Numérico | 2 | Ano da data de Nascimento do Responsável |
| RESP_MESN | Simples | Numérico | 2 | Mês da data de Nascimento do Responsável |
| RESP_DIAN | Simples | Numérico | 2 | Dia da data de Nascimento do Responsável |
| RESP_TEM_ATOA | Simples | Texto | 1 | Campo de sobra no arquivo - Não usado |
| RESP_TEM_ALUNO | Simples | Texto | 1 | Campo de sobra no arquivo - Não usado |
| RESP_UNIRSN | Simples | Texto | 1 | Boletos Unidos S/N |
| RESP_IE | Simples | Texto | 16 | Inscrição Estadual da Empresa (Responsável pode ser uma empresa) |
| RESP_VAZIO | Simples | Texto | 4 | Campo de sobra no arquivo - Não usado |

Tabela 4: **Dicionário de Dados da Entidade Alunos** (AMARAL, 2012)

| Atributo | Classe | Domínio | Tamanho | Descrição |
|---|---|---|---|---|
| ALU SEQ | Determinante | Numérico | 6 | Código Sequencial - Chave Principal |
| ALU_CPF_RESP | Determinante | Numérico | 14 | CPF/CNPJ do Responsável - Faz parte Chave Alternativa |
| ALU_NOME | Determinante | Texto | 40 | Nome Aluno - Faz parte Chave Alternativa |
| ALU_MILN | Simples | Numérico | 1 | Milênio da data de Nascimento do Aluno |
| ALU_ANON | Simples | Numérico | 2 | Ano da data de Nascimento do Aluno |
| ALU_MESN | Simples | Numérico | 2 | Mês da data de Nascimento do Aluno |
| ALU_DIAN | Simples | Numérico | 2 | Dia da data de Nascimento do Aluno |
| ALU_SEXO | Simples | Texto | 1 | Sexo do Aluno |
| ALU_CIDADEN | Simples | Texto | 15 | Cidade Aluno |
| ALU_UFN | Simples | Texto | 2 | UF Aluno |
| ALU_PAI | Simples | Texto | 40 | Pai Aluno |
| ALU_MAE | Simples | Texto | 40 | Mãe do Aluno |
| ALU_CPF | Simples | Numérico | 14 | CPF do Aluno |
| ALU_RG | Simples | Texto | 15 | RG Aluno |
| ALU_TIT_ELEITOR | Simples | Texto | 15 | Titulo de Eleitor do Aluno |
| ALU_RESERVISTA | Simples | Texto | 15 | Reservista do Aluno |
| ALU_TIPO_END | Simples | Numérico | 1 | Tipo de Endereço do Aluno - Mesmo do Responsável ou diferente |
| ALU_GRAU | Simples | Numérico | 1 | Grau do Aluno - 1º, 2º |
| ALU_SERIE | Simples | Numérico | 1 | Serie do Aluno |
| ALU_TURMA | Simples | Texto | 1 | Turma do Aluno - 8ª A, 8ª B |
| ALU_TURNO | Simples | Numérico | 1 | Turno do Aluno - Noturno, Matutino ... |
| ALU_NUMALU | Simples | Numérico | 2 | Campo de Sobra - Não Utilizado |
| ALU_ANOMATR | Simples | Numérico | 4 | Ano de Matricula do Aluno |
| ALU_SITUACAO | Simples | Texto | 1 | Situação do Aluno - Ativo, Inativo |
| ALU_OBS | Simples | Texto | 40 | Campo de Observação |
| ALU_QTALU | Simples | Numérico | 2 | Campo de Sobra - Não Utilizado |
| ALU_PERCPAGTO | Simples | Numérico | 5 | Porcentagem de desconto do aluno sobre a mensalidade integral - Determinados Alunos são Bolsistas |
| ALU_UTILIZOU | Simples | Texto | 1 | Campo de Sobra - Não Utilizado |
| ALU_BIMTRANSF | Simples | Numérico | 1 | Bimestre no qual o aluno ingressou na escola |
| ALU_RESPSEQ | Simples | Numérico | 6 | Código do Responsável - "Chave estrangeira", elemento para se acessar arquivo de responsáveis |
| ALU_DIAVC | Simples | Numérico | 2 | Dia do Vencimento do Pagamento do Aluno |
| ALU_VAZIO | Simples | Texto | 27 | Campo de Sobra - Não Utilizado |

### 3.1.5 ESPECIFICAÇÕES TÉCNICAS DAS FERRAMENTAS UTILIZADAS NO PROCESSO DE MODERNIZAÇÃO GRÁFICA

Para a concretização da Aplicação exemplo foi utilizado o Micro Focus COBOL versão 3.4.23. No COBOL foram desenvolvido os cadastros utilizando-se arquivos ISAM.

A ferramenta RAD Delphi versão 7 também foi utilizada no processo, por ser uma ferramenta com componentes visuais eficientes, por possuir excelentes componentes de conexão a bases de dados e por possuir diversos recursos que auxiliam no processo de desenvolvimento.

A ferramenta Parkway Connectware também foi utilizada por flexibilizar o acesso a dados COBOL, permitindo que quaisquer dados ISAM não relacionais possam ser utilizados como uma verdadeira base de dados SQL.

As versões existentes do Parkway Connectware são:

- ConnectWare stand-alone for Micro Focus ISAM (U$ 980.00)
- Connectware Server Unix/Linux (U$ 2,350.00)
- Connectware Server Windows (U$ 2,350.00)
- Additional Client Windows (U$ 215.00)
- Additional Client Linux (U$ 215.00)

A versão utilizada no projeto foi a versão stand-alone, já que no mesmo não se obtinham usuários concorrentes na base de dados. Esta versão dispensa a utilização de um servidor para base de dados (hardware e software).

Existe a versão demonstração porém esta limita a utilização em um determinado número de dias. Percebe-se que a versão para servidor tanto Linux quanto Windows custa 2,350.00 dólares. A versão cliente (utilização por usuário concorrente) também é comum tanto para Linux quanto para Windows custando 215.00 dólares.

### 3.1.6 PROCESSO DE CONEXÃO DO PARKWAY CONNECTWARE COM O ARQUIVO RESPONSA DA BASE DE DADOS COBOL

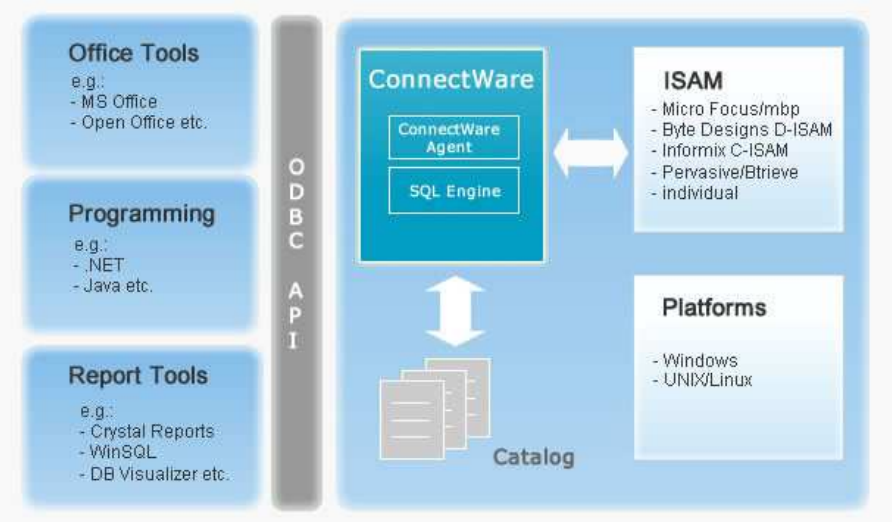



Figura 9**: Ilustração de Funcionamento da Ferramenta Parkway Connectware** (AMARAL,2012)

Na figura 10 é demonstrado como o Parkway é acessado depois de devidamente instalado.

Figura 10**: Tela de acesso ao Parkway Connectware** (AMARAL, 2012)

A figura 11 demonstra o acesso a fonte de dados WL (base de dados do Sistema), que deve ser devidamente adicionada logo após a configuração Parkway Connectware.

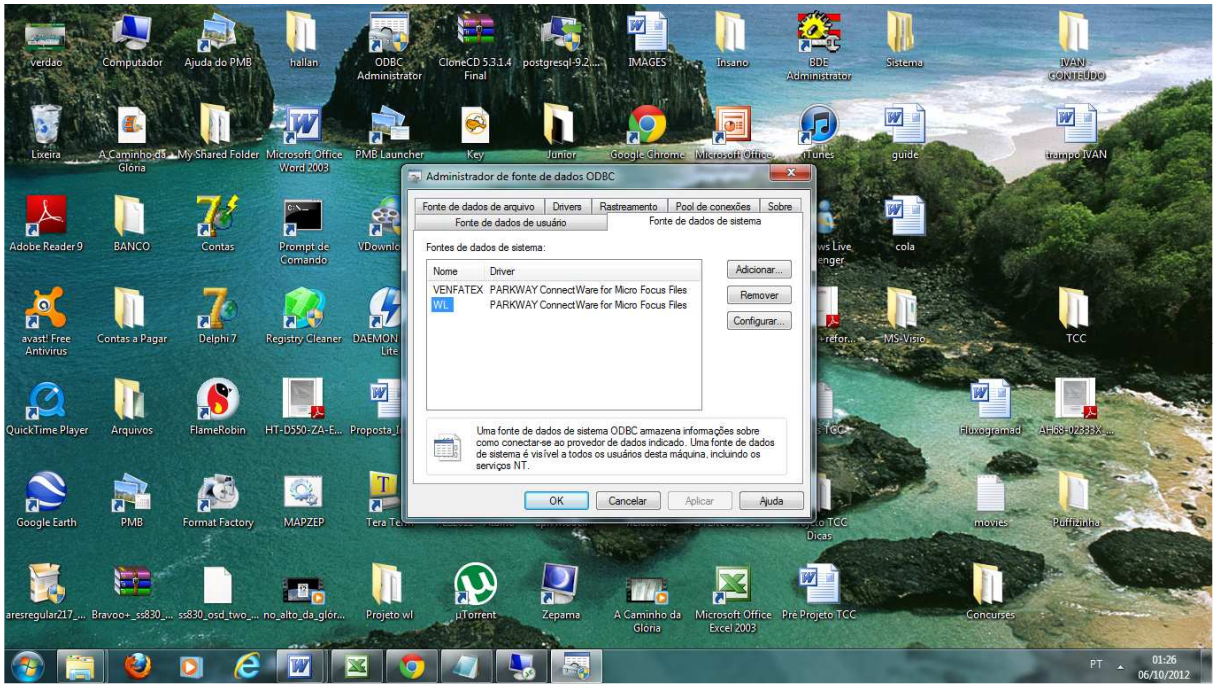



Figura 11**: Acesso a fonte de dados WL** (AMARAL, 2012)

Na figura 12 demonstra-se como se faz à conexão a base de dados WL

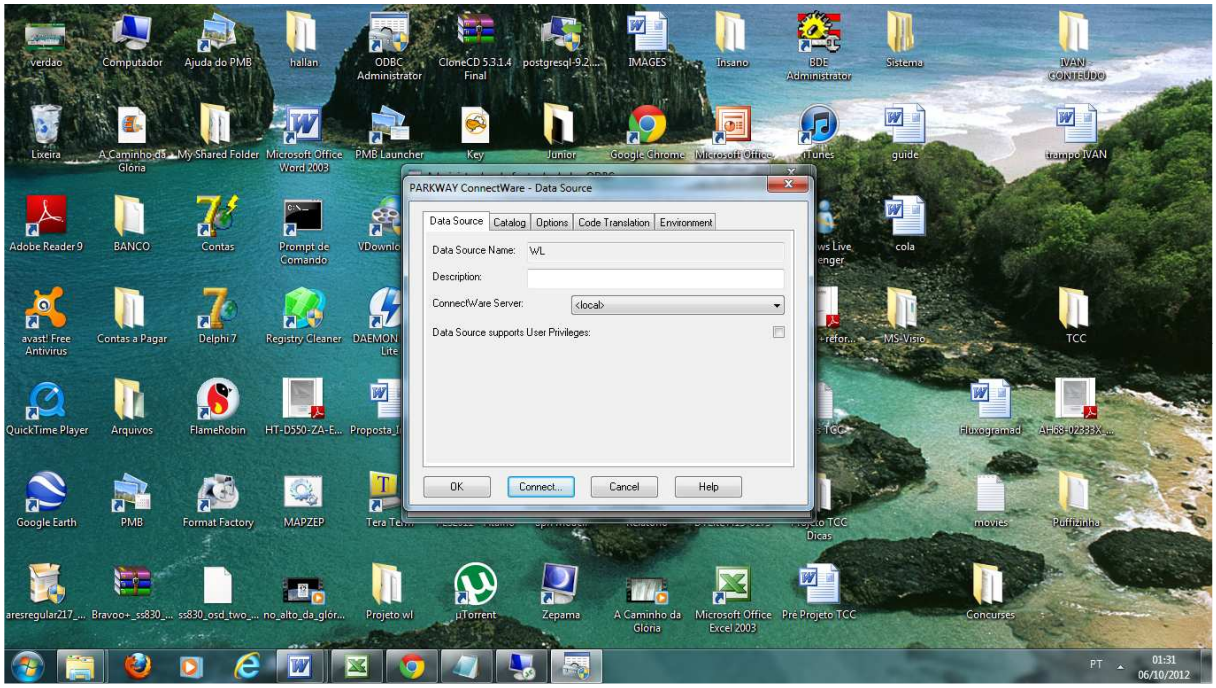



Figura 12**: Conexão a fonte de dados WL** (AMARAL, 2012)

A figura 13 evidencia a criação da tabela RESPONSA no ConnectWare

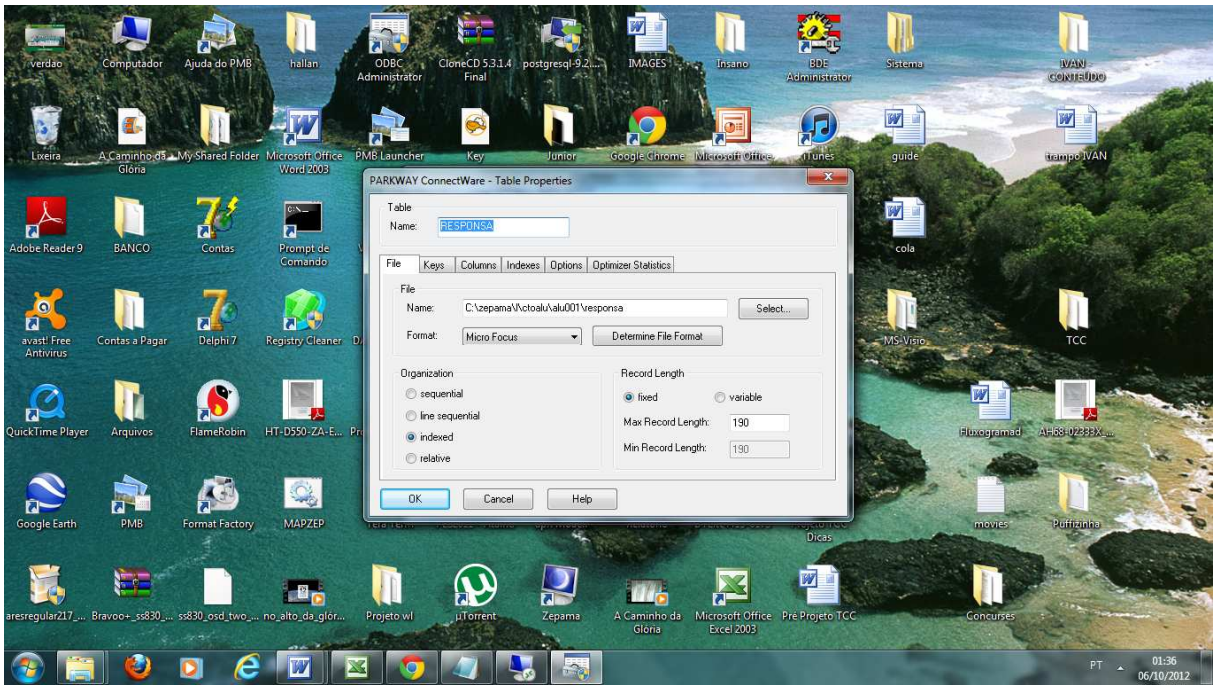



Figura 13**: Criação da Tabela Responsa na Fonte de dados WL** (AMARAL, 2012)

A figura 14 demonstra a conexão do Parkway ConnectWare propriamente dita com os arquivos COBOL

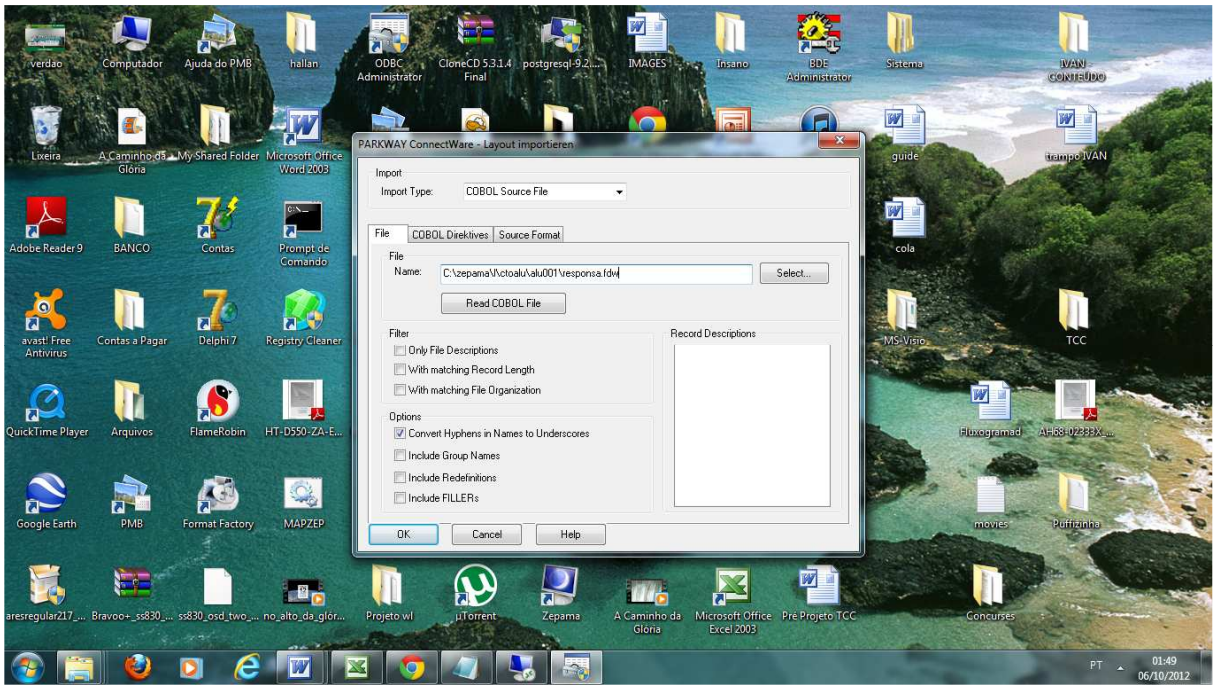



Figura 14**: Conexão do Parkway Connectware com a tabela RESPONSA da base de dados COBOL** (AMARAL, 2012)

Ao clicar em Read COBOL File é feita a conexão e os campos do arquivo COBOL aparecerão na ferramenta em forma de “colunas”, como é demonstrado na figura 15.

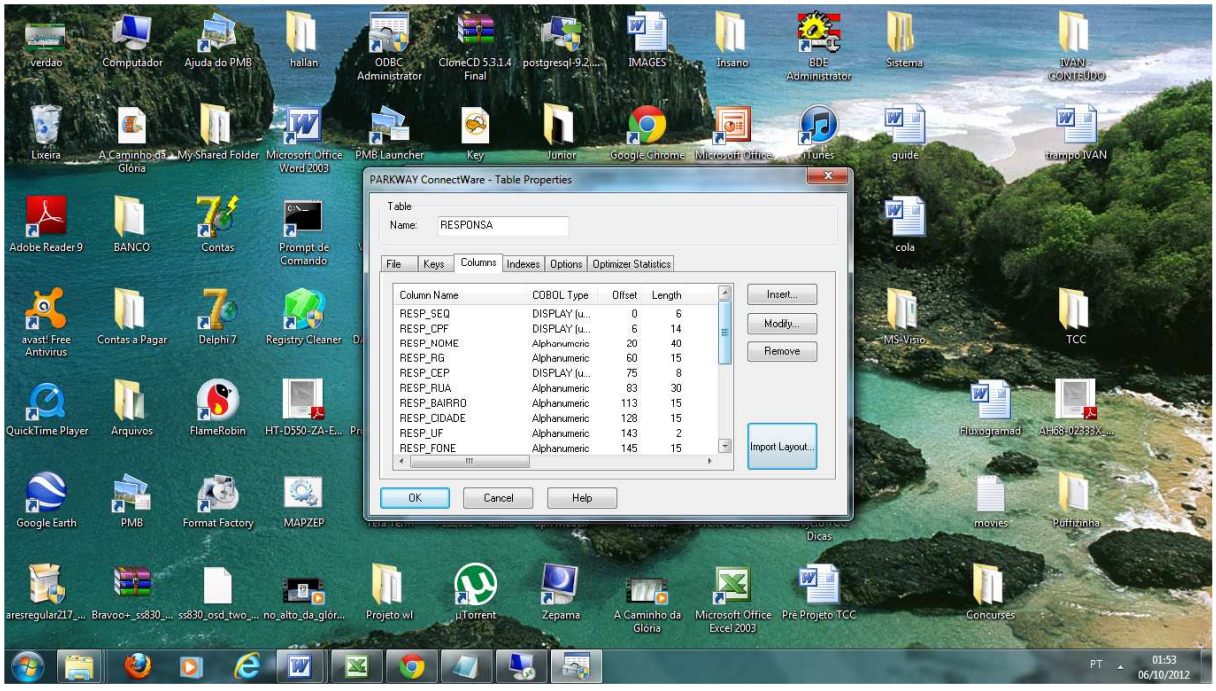



Figura 15**: Arquivo RESPONSA projetado no ParkwayConnectware** (AMARAL, 2012)

### 3.1.7 CONEXÃO DA BASE DE DADOS WL COM A FERRAMENTA DELPHI E MANIPULAÇÃO DO ARQUIVO COBOL RESPONSA NO DELPHI

Para que se faça a conexão da base de dados COBOL com o Delphi inicialmente precisa-se iniciar a ferramenta Delphi e clicar em New → DataModule. Depois de criado o DataModule é necessário que se inclua o componente DataBase da aba BDE no DataModule, conforme é demonstrado na figura 15.

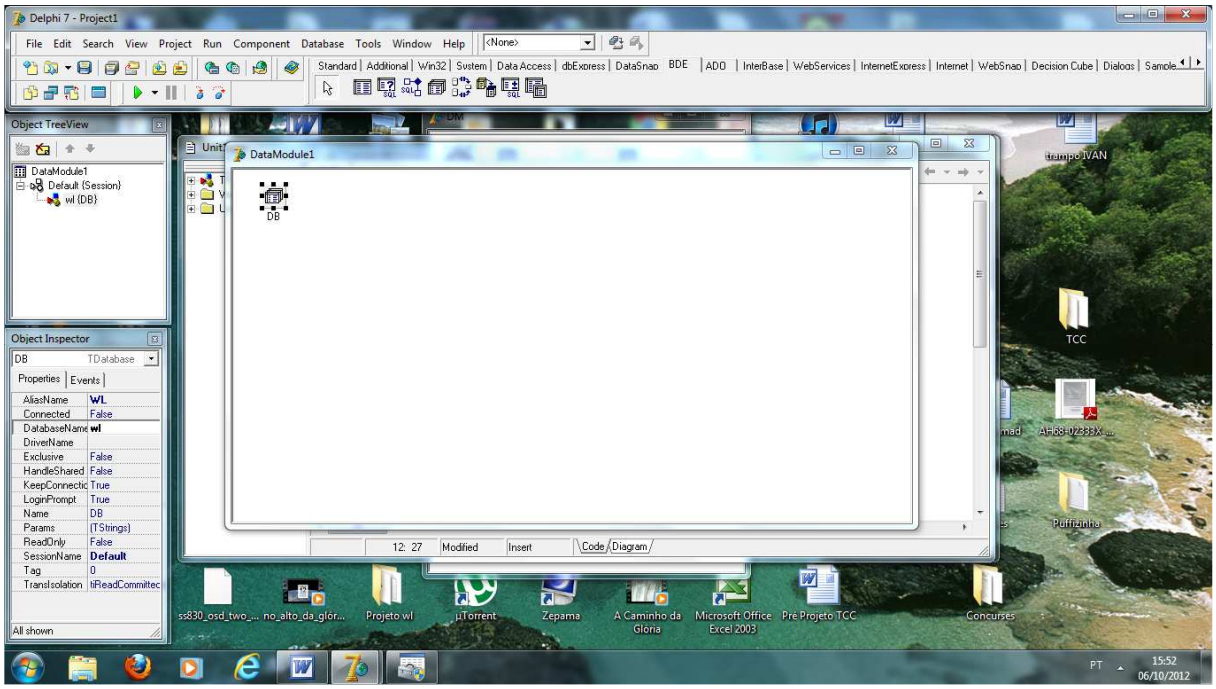



Figura 16**: Conexão Delphi com o ConnectWare** (AMARAL, 2012)

Conforme evidenciado na figura acima para se conectar a base de dados é necessário selecionar a fonte de dados WL previamente definida no Parkway Connectware na propriedade “Alias Name” do Delphi, definir um nome para a base de dados na propriedade “Data Base Name” e definir um nome para o componente na propriedade “Name”.

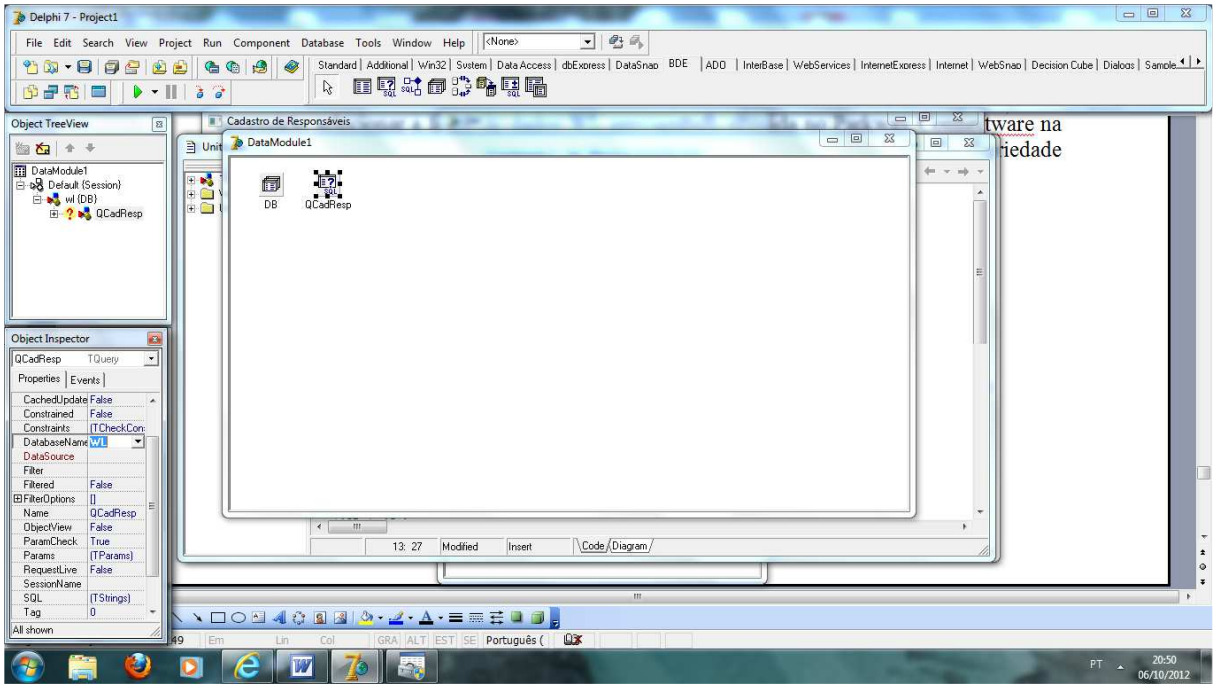



Figura 17**: Conexão com o componente Query** (AMARAL, 2012)

A figura acima demonstra a inclusão do componente Query, é necessário que na propriedade “DataBase Name” se coloque WL(nome da fonte de dados Connectware) e se de um nome ao componente, no exemplo acima o componente foi nomeado de QCadResp.

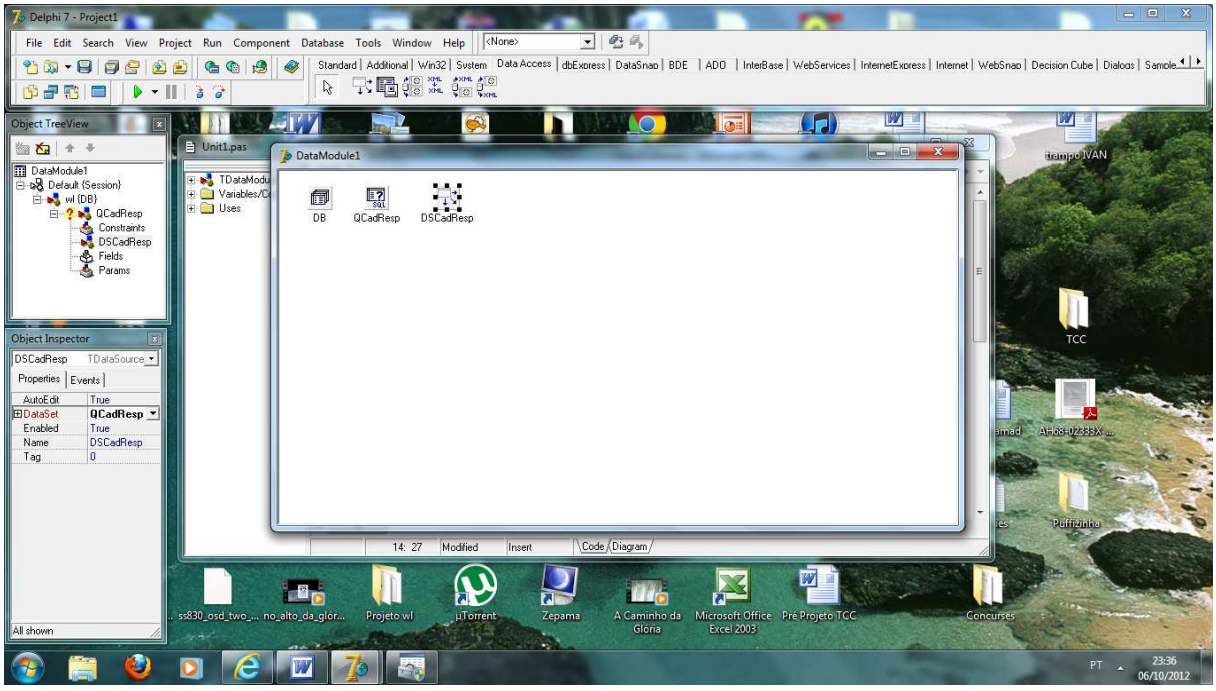



Figura 18**: Conexão com o componente DataSource**

A figura 18 exemplifica a inserção do componente DataSource no DataModule, após isso é necessário selecionar QCadResp (Componente Query Tabela Responsa) na propriedade “DataSet” do componente DataSource. Já está preparada a conexão com a tabela RESPONSA. Para toda tabela é necessário se realizar esse procedimento, ou seja, para toda a tabela que se deseja conectar com Delphi é necessário um componente Query e um componente DataSource.

**Manipulação dos dados da tabela Responsa no Delphi**

É possível manipular os dados da tabela responsa no Delphi através de comandos SQL ANSI, como por exemplo, foi desenvolvida uma procedure no Delphi denominada lerreg que faz a leitura dos dados no arquivo COBOL, formata-os e projeta nos campos (Edit) do formulário. Na figura abaixo é demonstrado esse processo

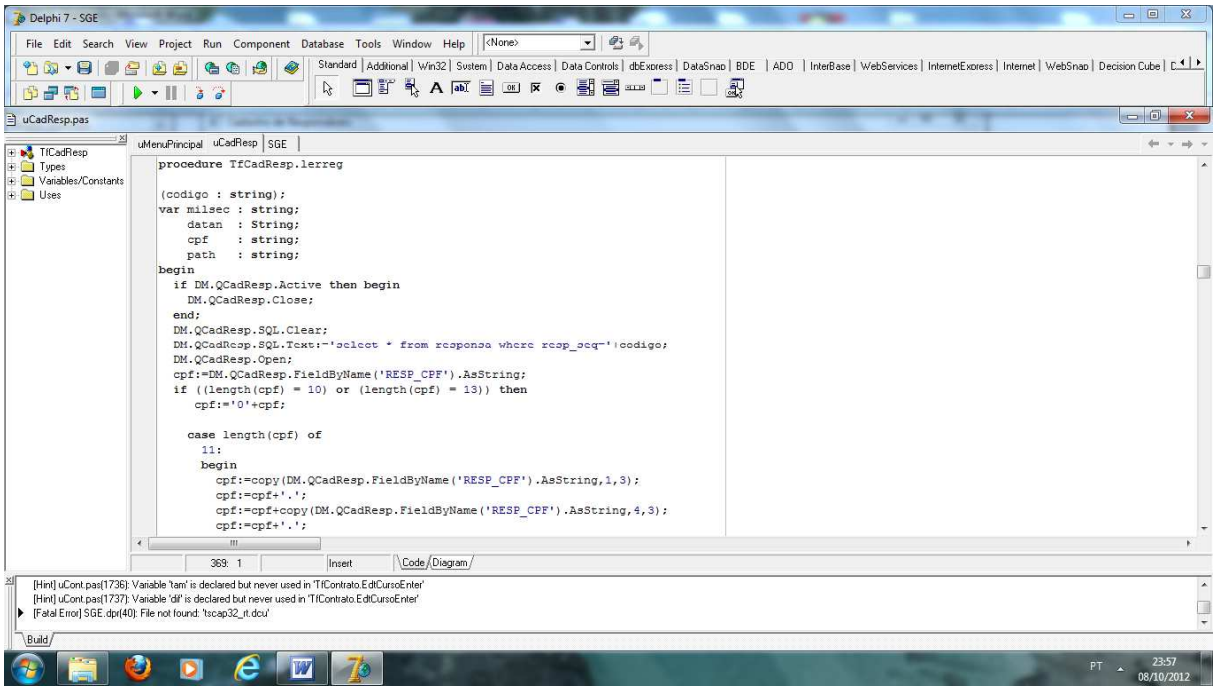



Figura 19**: Procedure no Delphi exemplificando leitura de arquivo(tabela)** (AMARAL, 2012)

É importante ressaltar que nem todos os comandos SQL ANSI são aceitos pela ferramenta parkway Connectware, como por exemplo o comando limit, porém os comandos básicos de manipulação como insert, delete, update e select além de alguns outros como inner join, group by e order by entre outros são aceitos.

A ilustração abaixo demonstra como é realizada a inserção dos dados no arquivo RESPONSA.

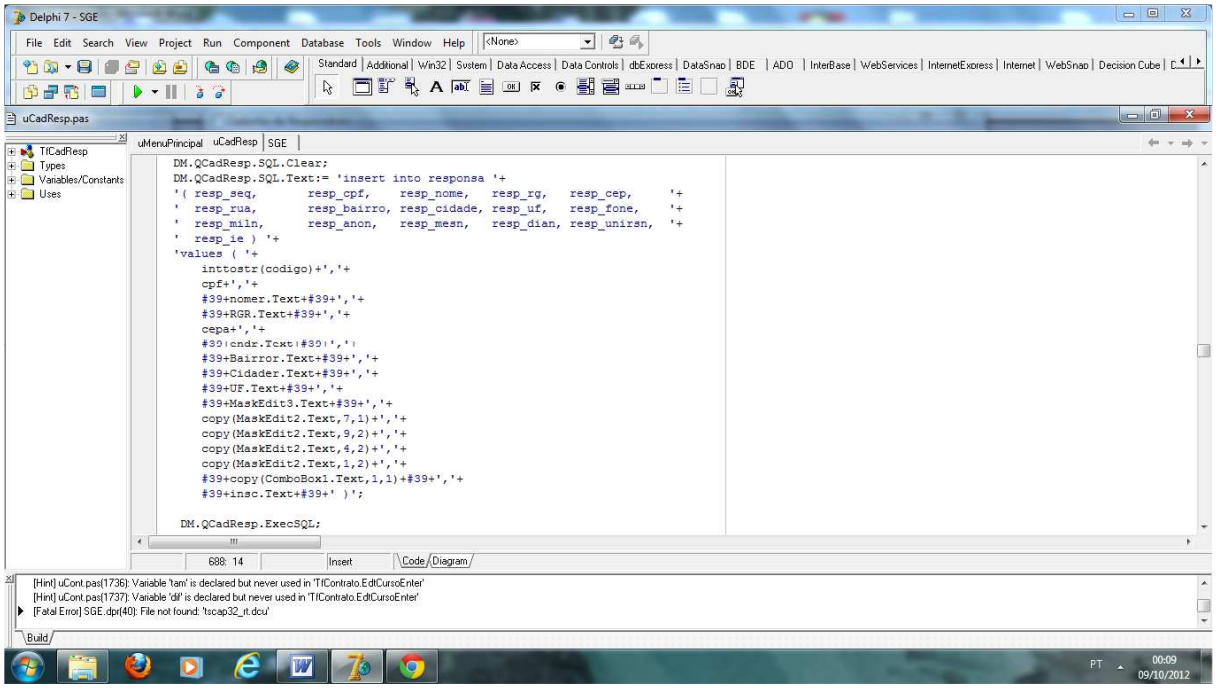



Figura 20: **Script no Delphi exemplificando inserção de registro na tabela RESPONSA** (AMARAL, 2012)

Através destes comandos SQL ANSI é possível manipular os dados. No Delphi a manipulação é realizada através destes comandos e no COBOL a manipulação é realizada através de comandos nativos do COBOL como Write, Rewrite, Read e Delete. A grande vantagem dessa idéia é a utilização dos mesmos dados tanto em COBOL como em qualquer outra ferramenta (na aplicação exemplo o Delphi).

A aplicação exemplo no Delphi ficou com um visual moderno e agradável, além de ficar extremamente funcional. Abaixo algumas figuras demonstram o visual das telas desenvolvidas.

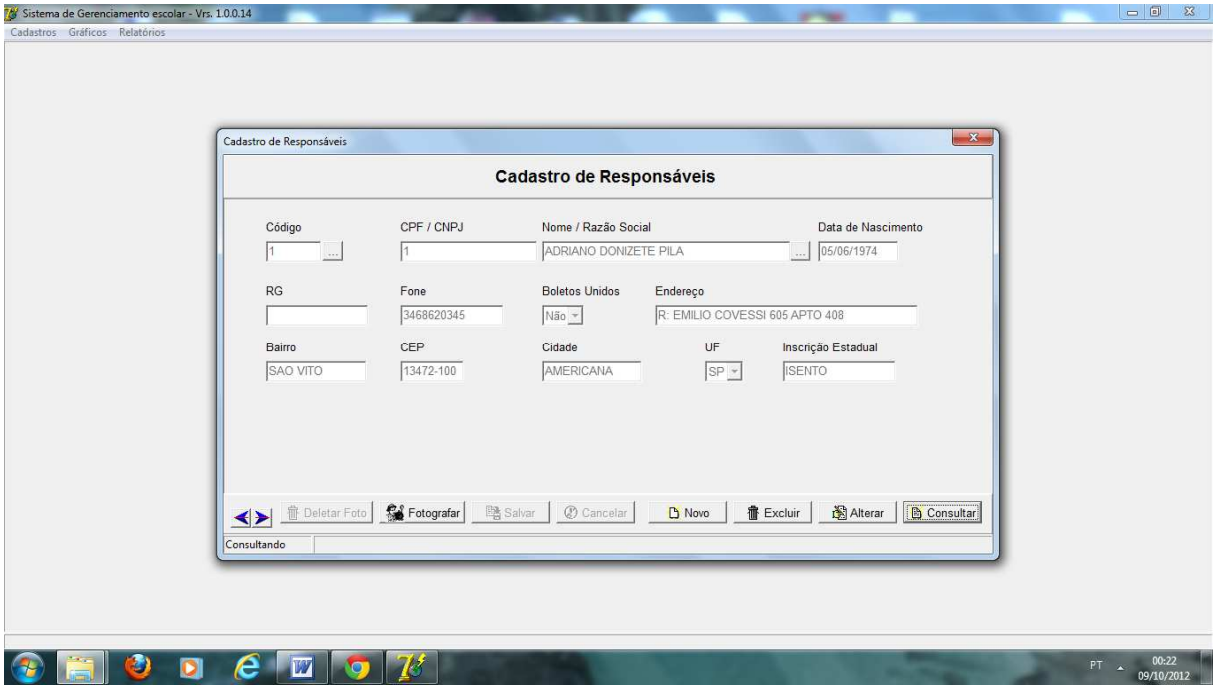



Figura 21: **Tela de Cadastro de Responsáveis na aplicação de Gestão Escolar** (AMARAL, 2012)

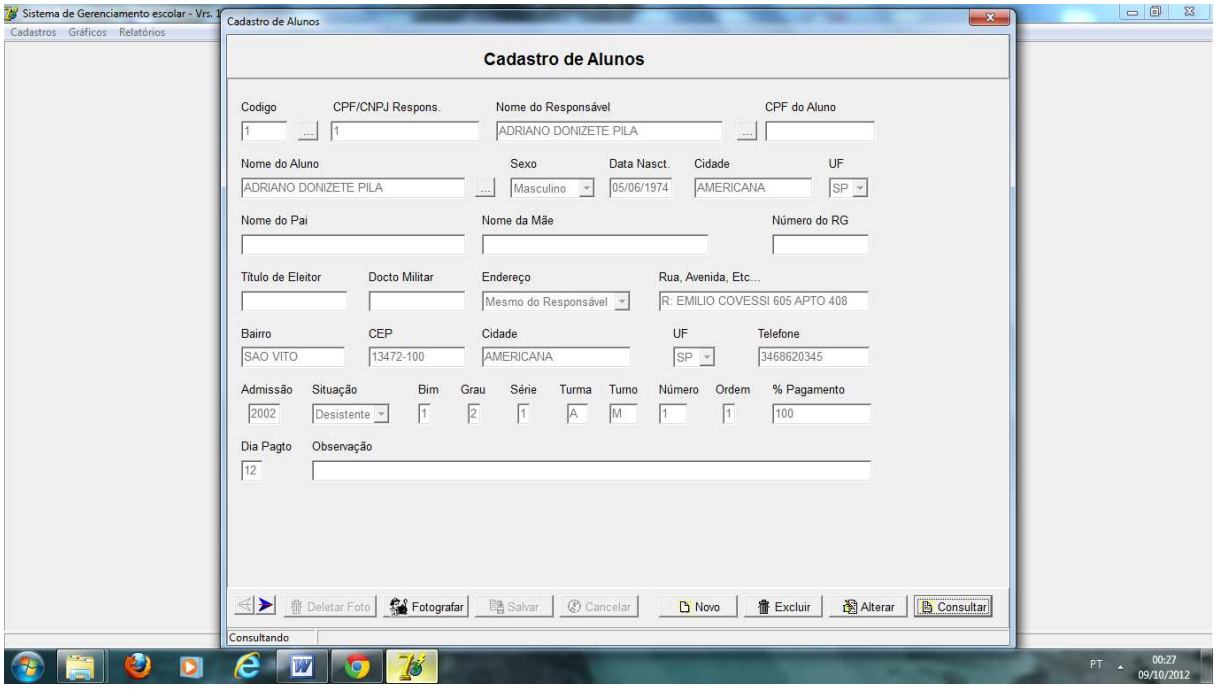



Figura 22: **Tela de Cadastro de Alunos na aplicação de Gestão Escolar** (AMARAL, 2012)

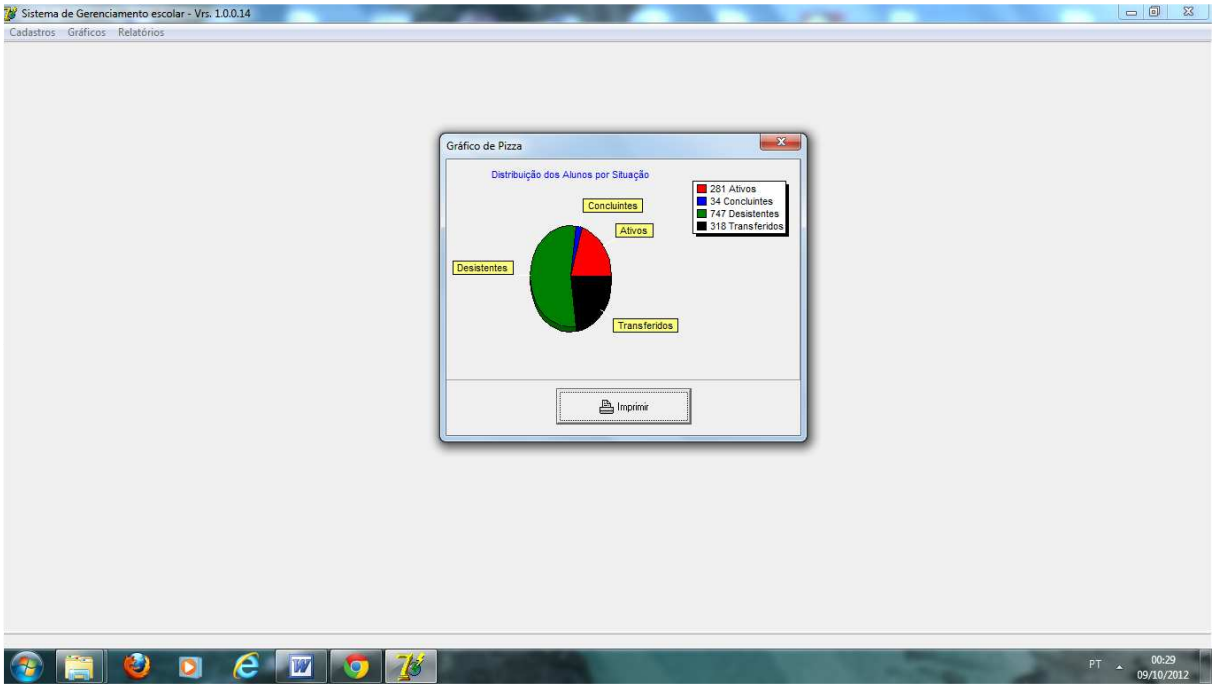



Figura 23: **Gráfico do tipo Pizza especificando proporcionalmente a situação dos alunos** (AMARAL, 2012)

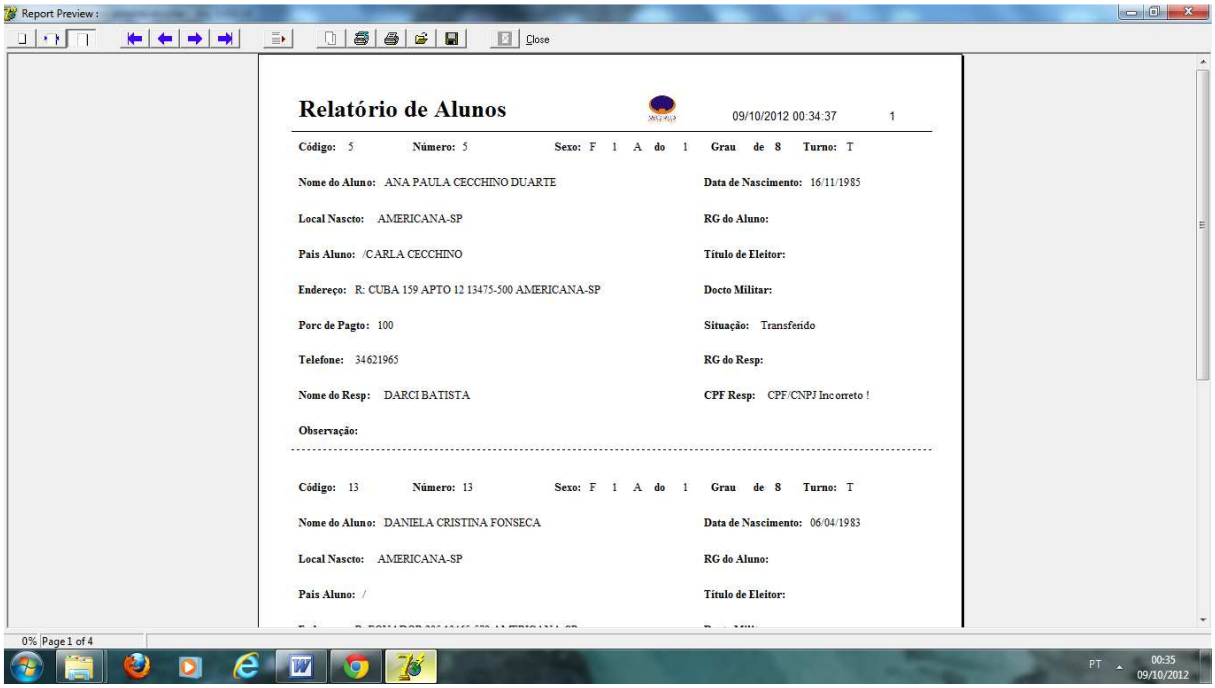



Figura 24: **Relatório de Alunos** (AMARAL, 2012)

# 4 CONCLUSÃO

Neste trabalho, é demonstrada as funcionalidades de algumas ferramentas para desenvolvimento de software como o COBOL e o Delphi, bem como a usabilidade, características e história.

Como pode-se constatar a linguagem COBOL proporciona um alto nível de segurança, além de possuir uma sintaxe relativamente simples e auto documentável. Apesar da linguagem apresentar todos esses benefícios, a interface que a mesma proporciona (um tanto que arcaica) não é bem vista por parte dos usuários.

O trabalho aborda a existência de algumas ferramentas para as organizações que ainda possuem sistemas funcionando em linguagem COBOL, porém desejam modernizar suas aplicações, utilizando a mesma base de dados e obtendo um custo relativamente baixo.

Com o objetivo de solucionar esta deficiência de interface, o trabalho propõe um processo de modernização de interface aproveitando toda a base de dados COBOL. Para isso foi utilizada a ferramenta alemã Connectware (responsável por disponibilizar os arquivos COBOL em forma de tabelas) e a ferramenta de desenvolvimento Delphi, já que a mesma possui uma sintaxe relativamente simples e intuitiva, além de proporcionar uma interface agradável e moderna do ponto de vista dos usuários.

Os arquivos COBOL são projetados em forma de tabelas de banco de dados, por meio da ferramenta Connectware. A ferramenta RAD Delphi conecta-se a base de dados COBOL via Connectware, onde é possível se realizar toda a manipulação da base dados COBOL via comandos SQL ANSI.

Para realizar a conexão da base de dados COBOL com a ferramenta RAD Delphi foi estudada a ferramenta Connectware onde é possível compartilhar por meio de um conector ODBC a base de dados COBOL com qualquer outra ferramenta de desenvolvimento. O resultado final da conexão se mostrou extremamente satisfatório.

Conclui-se que este trabalho contribui e oferece alternativas para empresas que possuem sistemas baseados em COBOL para uma nova linguagem mantendo-se a mesma base de dados, utilizando-se a ferramenta Parkway Connectware.

No trabalho foram desenvolvidas apenas algumas telas, pois foi elaborada uma aplicação exemplo, porém como trabalho futuro pode-se estudar e desenvolver uma aplicação completa, com o objetivo de se obter um conhecimento mais abrangente sobre esse processo de modernização.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, William Pereira. **Delphi 7 Aplicações Avançadas de Banco de Dados.** Ed, Érica Ltda São Paulo. 2002, p. 16 – 17.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Citação**: NBR-10520/ago - 2002. Rio de Janeiro: ABNT, 2002.

BRITTO, Pedro. **Estudantes, dediquem-se mais ao COBOL — palavras de um gerente da IBM**, 14 mai. 2007. Disponível em: < http://tecnoblog.net/104147/ibm-cobol-faculdade/>.Acesso em 06 jun. 2012.

CARVALHO, José E. M. de. **Microsoft COBOL 4.0.** 1Ed, Makron Books: São Paulo. 1991, p. 27 – 29.

MCCRACKEN, D. D.; GARBASSI, U. et al. **Programação Cobol**. São Paulo: ATLAS, 1978.

NORTON, Peter. **Guia para o Delphi 2.** Ed, Makron Books São Paulo. 1997.

RODRIGUES, N. **Duro de Matar**, 14 mai. 2007. Disponível em: <**http://pcworld.uol.com.br/reportagens/2007/05/idgnoticia.2007-05-14.708755063/** >.
Acesso em: 09 set. 2012.

_____. **Referências**: NBR-6023/ago. 2002. Rio de Janeiro: ABNT, 2002.

RODRIGUES, N. **Duro de Matar**, 21 mai. 2007. Disponível em: < **http://www.projetosespeciais.com.br/remuneracao.htm**>.
Acesso em: 15 set. 2012.

SOARES, Edileuza. **A História do COBOL**. 2012. Disponível em < http://www.cadcobol.com/histor_1.htm> Acesso em 07 set.2012.

SEVERINO, Antônio J. Metodologia do Trabalho científico. 23ª Ed. São Paulo: Cortez, 2004. P.122-123.

WEISSMAN LOBO, Henrique, 25 out. 2008. Disponível em: < **http://www.itexto.net/devkico/?p=135**>.
Acesso em: 10 set. 2012.