MINISTÉRIO DA EDUCAÇÃO SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA CATARINENSE CÂMPUS VIDEIRA

Curso de Ciência da Computação

Relatório de Estágio Curricular

Taynara C. Dutra

Videira, 17 de Fevereiro de 2016

MINISTÉRIO DA EDUCAÇÃO SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA CATARINENSE CÂMPUS VIDEIRA

Taynara C. Dutra

Relatório de Estágio Curricular

Relatório de Estágio Curricular Obrigatório submetido ao Instituto Federal Catarinense - Câmpus Videira como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Aprovado por:

Diego Ricardo Krohl (Orientador)

Wagner Carlos Mariani

Diego Witti

Videira, 17 de Fevereiro de 2016

SUMÁRIO

| IDEN | NTIFICAÇÃO | 4 |
|-------------|---|----|
| 1. | INTRODUÇÃO | 5 |
| 2. | APRESENTAÇÃO DA EMPRESA | 6 |
| | ATIVIDADES DESENVOLVIDAS | |
| 3.1. | Conversão de grids | 7 |
| | Conversão de relatórios | |
| 3.3. | Ajuste de rotinas | 19 |
| 3.4. | Implementação de rotina | 26 |
| 4. | CONCLUSÃO | 29 |
| 5. | REFERÊNCIAS BIBLIOGRÁFICAS OU REFERÊNCIAS | 30 |

IDENTIFICAÇÃO

| IDENTIFICAÇÃO DA EMPRESA | | | | | |
|--|--|--------|--|--|--|
| DOPPLER SISTEMAS LTDA | | | | | |
| Endereço: Rua Arlindo de Mattos,178 | | | | | |
| Bairro: Santa Tereza | Cidade: Videira | UF: SC | | | |
| CEP: 89560000 | Número Telefone: (49) 3533-2756 | | | | |
| Email: contato@dopplersistemas.com.br | | | | | |

| IDENTIFICAÇÃO DO ESTAGIÁRIO | | | | |
|-----------------------------|------------------------------|--|--|--|
| TAYNARA CERIGUELI DUTRA | | | | |
| Período: - | Turma: 2013.1 | | | |
| Matrícula: 201310099 | Email: tay.dutra@hotmail.com | | | |

| IDENTIFICAÇÃO DO ESTÁGIO | | | | |
|-----------------------------------|----------------------------|--|--|--|
| DESENVOLVIMENTO DE SISTEMAS | | | | |
| Data de Início: 11/06/2015 | | | | |
| Carga Horária Total: 200 horas | Média de Horas Semanal: 10 | | | |
| Supervisor: Adroaldo Pagliari | Email: | | | |

IDENTIFICAÇÃO DE ATIVIDADES 1 Conversão de Grids 2 Conversão de relatórios 3 Ajuste de rotinas 4 Implementação de rotinas

1. INTRODUÇÃO

Em empresas de grande porte é imprescindível que haja um sistema de controle para estoques, contabilidade, produção entre outros setores. Para manter a consistência dos dados armazenados e a integração deles, não é possível utilizar softwares separados ou apenas planilhas do Excel, é necessário um sistema que integre e controle toda a empresa, este sistema é um ERP.

A Doppler Sistemas produz sistemas de informação para sanar a necessidades destas empresas. Atua na cidade de Videira, porém atende toda a região, o software é desenvolvido para empresas grandes de metalúrgica, frutas, produção de calçados e bebidas.

O ERP é composto por vários módulos que consistem nos setores da empresa, e um módulo específico, o módulo de produção. A maioria dos módulos se integram perfeitamente em vários clientes, porém, às vezes há necessidade de customização para adequar-se a maneira como a qual a empresa trabalha.

Para a conclusão do curso de Ciência da Computação é necessário a realização do estágio obrigatório supervisionado. Este é uma etapa importante para o desenvolvimento e aprendizagem de um aluno, pois proporciona a vivência na prática dos conteúdos vistos em sala de aula. Assim, como a troca de conhecimento, ideias e estratégias entre funcionários da empresa. Desta forma, o estudante pode obter um melhor conhecimento sobre a profissão através das experiências reais vivenciadas em uma empresa. (SANTOS, 2014)

As principais atividades realizadas dentro da empresa, estão descritas neste relatório, que consistem principalmente na criação e ajustes de rotinas, implementação de relatórios, procedure e funções para o sistema Doppler, utilizando *Object Pascal* e <u>SQL</u>. Bem como o suporte ao cliente para ajuda-los a realizar tarefas necessárias.

A empresa busca oferecer o melhor para cada cliente, satisfazer as suas particularidades, oferecer agilidade e uma interface amigável para que todos os usuários possam compreender e oferecer o seu melhor.

Todas as tarefas realizadas no dia a dia da Doppler constituem-se em programação. Para realiza-la os conhecimentos obtidos em disciplinas como algoritmos, linguagem de programação I e II e banco de dados foram de profunda importância, apesar da linguagem não ser a mesma aprendida no curso, *Object Pascal* é simples e fácil, com aprendizado rápido com a ajuda dos colegas e supervisor na empresa.



2. APRESENTAÇÃO DA EMPRESA

A empresa surgiu em 2011 pelos fundadores Gean Lourenço Fabris e Adroaldo Pagliari. Atua no desenvolvimento de softwares para gerenciamento e automação de processos e recursos corporativos para empresas na cidade de Videira e região. A missão da empresa é oferecer soluções inteligentes e inovadoras, na busca pelo melhor para cada necessidade em tecnologia da informação.

A Doppler Sistemas trabalha com ERPs para empresas de médio a grande porte. O Doppler ERP é um sistema de gerenciamento de processos Administrativos, Fiscais, Contábeis, Comerciais, Financeiros, Suprimentos e de Manufatura e Custos. Permite a integração de empresas dos diversos segmentos. Há três principais módulos de produção, Doppler PPCP, ERP de planejamento, programação e controle de produção. Doppler Meta Mecânico ERP específica para indústria metalúrgica e Doppler Fruticultura, sistema de gestão para indústrias fruticultoras. Para o desenvolvimento do sistema é utilizado a IDE *Delphi*, *Object Pascal* como linguagem de programação e para armazenamento dos dados é utilizado o *Oracle*.

No ano de 2015, equipe é formada por dois programadores, o Adroaldo como gerente de projetos e o Gean que atua no suporte e comercial. Os programadores e o gerente de projetos trabalham em conjunto no desenvolvimento e customização de rotinas para o ERP.

Além do EPR, a Doppler Sistemas trabalha também com a revenda do sistema comercia Fênix, oferecendo a melhor solução em softwares para atender a necessidade de cada empresa, independente da sua área de atuação.

3. ATIVIDADES DESENVOLVIDAS

3.1. Conversão de grids

| Data | | Horas | Setor | Atividade |
|------------|------------|-------|-----------------|-------------------------|
| Início | Término | | | |
| 11/06/2015 | 30/10/2015 | 84:00 | Desenvolvimento | Alteração de componente |

Objetivo: Alterar o componente *DBGrid* para o *CxGrid* nas rotinas do sistema.

Descrição: O sistema Doppler ERP é desenvolvido em linguagem de programação Pascal, a qual é uma linguagem alto nível e estruturada, foi criada por Niklaus Wirth, para ser a primeira linguagem aprendida. Sua estrutura é um pouco mais rígida, ou seja, deve seguir um padrão, para estimular maior organização e disciplina do programador. É fácil de ensinar e de aprender. (CARVALHO, 2006).

Para apresentar dados tabelados o sistema utilizava em suas rotinas o componente *DBGrid* para apresentar dados do banco para o usuário. Segundo Garcia (2013) este componente é muito comum para desenvolvedores *Delphi*, principalmente em soluções comerciais, devido a sua simplicidade e agilidade para apresentar dados de um *DataSet*.

Para otimizar estas rotinas, optou-se por alterar o componente *DBGrid* pelo componente *cxGrid* da *DevExpress*. A suíte *DevExpress* possui um grande diferencial comparado a nativa, devido a sua grande quantidade de recursos, que permite a construção de aplicações de grande porte mais simples até as mais complexas. Dispensa muita codificação, fazendo com que o processo se concentre na configuração da grid. Vale ressaltar, que essa suíte ganhou vários prêmios como melhor suíte de componentes. (MENDES, 200?).

Um exemplo, é rotina CBC2400 – Conciliação Manual, com base nos filtros informados os dados são consultados e adicionados à grid, a Figura 1 mostra a tela antes de sua conversão utilizando o componente *DBGrid*.

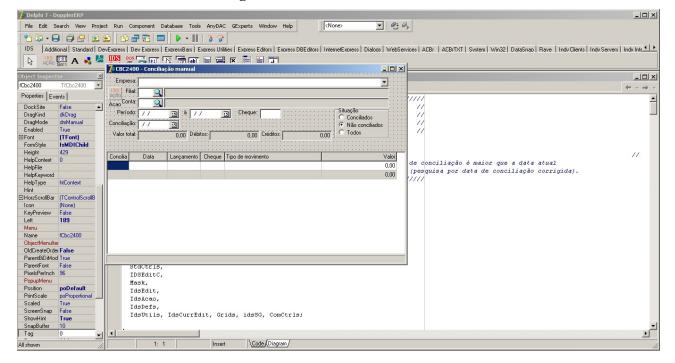


Figura 1: Tela utilizando DBGrid

A conversão é realizada excluindo o componente da tela e todo o código que está relacionado a ela no *script*. Em seguida deve-se adicionar o *DataSource* (DS Principal), *DataSet* e o *CxEstilo* componentes da *cxGrid*. O DataSource, é responsável pela conexão com as tabelas do banco de dados. No *DataSet*, é incluído o nome e o tipo dos campos quer serão carregados. O *CxEstilo*, formata os campos e adiciona um estilo a *grid*, cada linha recebe uma cor e torna visualmente melhor a utilização do usuário.

Primeiramente, adiciona-se os componentes para não perder a configuração e assim ser utilizada em todas as rotinas padronizando o sistema. Na Figura 2 é adicionado os componentes para a *cxGrid*.

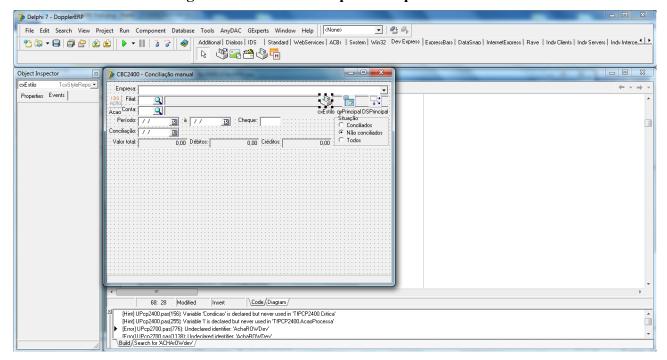


Figura 2: Adicionar componentes para a cxGrid

Em seguida, adiciona-se o componente da cxGrid, como mostrado na Figura 3, e deve-se programar novamente os scripts que foram apagados, porém manipulando agora a *cxGrid* e a maneira com a qual ela trabalha.

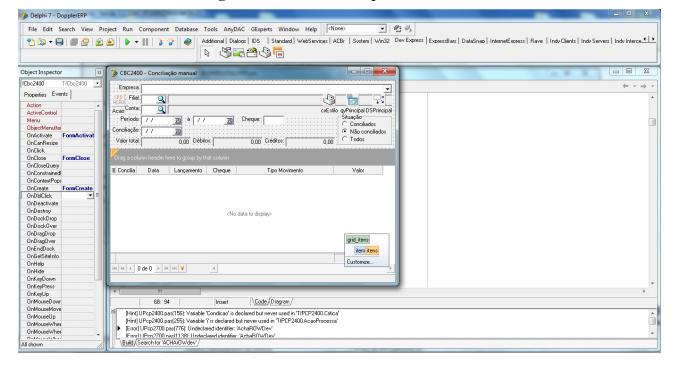


Figura 3: Adicionar componente cxGrid

Por exemplo, para consultar os dados no banco e mostrar ao usuário, utiliza-se funções já préprogramadas no sistema, apenas adicionando a *Unit*. Para realizar a consulta há um *procedure* que segue um padrão de codificação da empresa e esta sincroniza com o componente *Data Set*, preenchendo as colunas da *grid*. Geralmente as *procedures* e funções que precisam ser reprogramadas são as de consulta, duplo *click*, que preenche os campos novamente na tela e a exclusão de linhas na *grid*. A figura 4 mostra a tela com o componente *cxGrid* já compilada.



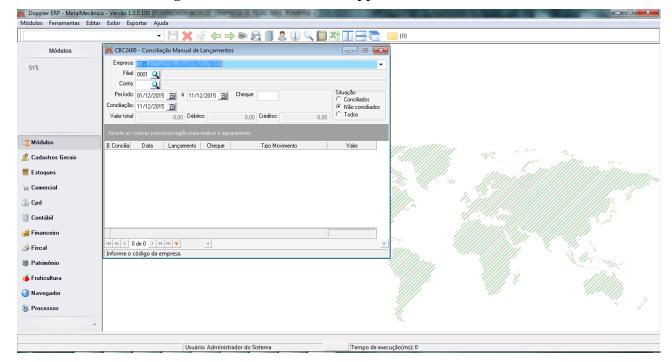


Figura 4: Tela do Sistema Doppler utilizando cxGrid

Dificuldade: Muitas vezes os dados devem ser inseridos na *grid*, porém o componente *cxGrid* não permite fazer isso diretamente.

Encaminhamento realizado: Para que uma informação seja inserida na *grid*, utiliza-se procedures que fazem a consulta previamente no banco inserem em um *work*, tabela auxiliar, então ao inserir uma informação passada na tela, via script insere-se na *work* e a *procedure* de consulta, seleciona os dados da *work*.

3.2. Conversão de relatórios

| Data | | Horas | Setor | Atividade |
|------------|------------|-------|-----------------|--------------------------|
| Início | Término | | | |
| 11/06/2015 | 30/10/2015 | 42:00 | Desenvolvimento | Desenvolver relatório em |
| | | | | outra plata forma |

Objetivo: Desenvolver o relatório utilizando a ferramenta *Fast Reports*.

Descrição: A empresa até pouco tempo utilizava para geração de relatório o *Crystal Reports*. Este software é capaz de gerar relatórios a partir da sincronização com o banco de dados, é uma ferramenta basicamente visual. Para que o relatório seja gerado a partir do ERP, utiliza-se um componente *VCL Crystal Reports* para conexão do *Delphi 7* com o *Crystal Reports*.

Para tornar a geração do relatório mais rápida, os relatórios seriam todos desenvolvidos novamente utilizando outro gerador, o *Fast Report*. Segundo Cantu (200?) Este atualmente pode ser considerado uma das melhores ferramentas para geração de relatórios no *Delphi/BCB*, rápido, leve e pode ser integrado dentro do executável; facilitando a implantação do ERP nos clientes.

Com base nas informações apresentadas no relatório em Crystal, cria-se um novo relatório em *Fast Report*, utilizando as mesmas consultas realizada no banco de dados e os parâmetros recebidos da tela, para ser utilizado como filtro. Para isso, no sistema na tela de cadastro de rotinas do sistema, informa-se qual é rotina do relatório, e então escolhe a opção para criar o relatório, como mostrado na Figura 5.

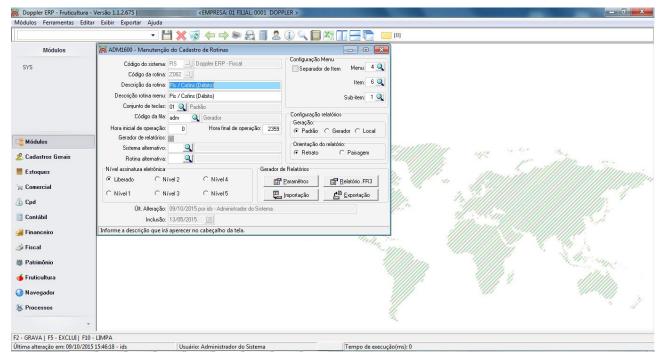


Figura 5: Tela de Cadastro de Rotina - Sistema Doppler

O sistema possui muitos relatórios, que auxiliam os clientes a consultar informações sobre custos, estoques, notas fiscais, etc. O relatório consiste em uma tela, na qual, o usuário preenche os campos para filtrar os dados que desejada e clica em Gerar. Então os filtros são passados ao relatório, que busca as informações no banco pertinentes e expõe ao cliente. Na figura 6 pode-se visualizar o relatório em branco na ferramenta *Fast Reports*.

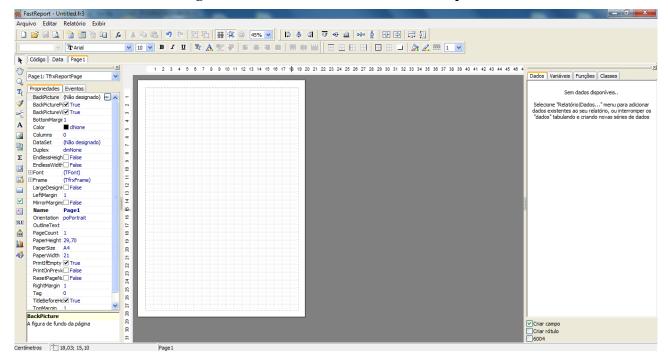


Figura 6: Relatório em branco - Fast Reports

Primeiramente deve-se adicionar o componente *ADQuery* e dentro dele inserir a consulta em *SQL* para que os dados sejam utilizados na apresentação das informações, já informando quais serão os filtros, como exemplificado na Figura 7.

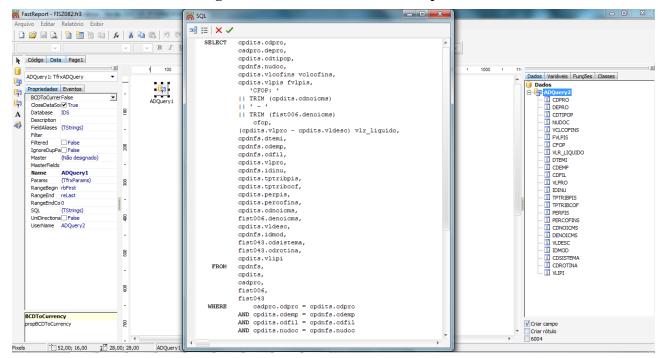


Figura 7: Consulta SQL - Fast Reports

Na Figura 8, pode ser visualizado a criação dos parâmetros, estes recebem os dados informados na tela.

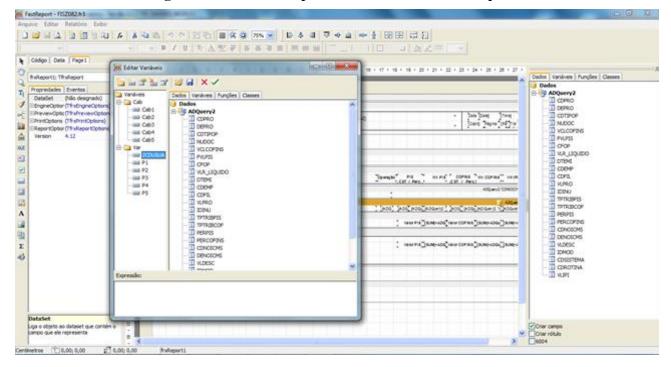


Figura 8: Parâmetros para filtrar dados - Fast Reports

O Fast Report possui inúmeros componentes, os mais utilizados são agrupamentos, caixa de texto, cross-tab, somatórios (count, min, max, media), subrelatórios, etc. A parte visual, é feita através do "clica e arrasta", os campos para serem listados que retornaram da consulta ao banco de dados, se encontram ao lado esquerdo e são adicionados ao MasterData, que listará tudo. Ao adicionar grupos, o MasterData se limitará a listar os dados pertencentes a cada grupo.

O relatório já finalizado trazendo as informações com os campos adicionados, agrupados e com seus totalizadores, é mostrado na Figura 9.

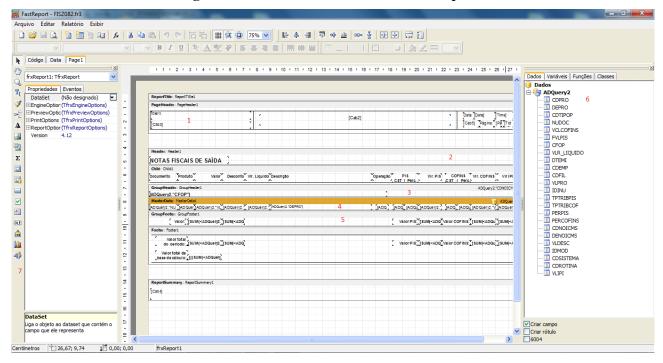


Figura 9: Relatório Finalizado - Fast Reports

- Item 1: Cabeçalho padrão dos relatórios, com nome da empresa, título do relatório, data de geração, etc.
 - Item 2: Cabeçalho do relatório, com títulos dos campos gerados no Master Data.
 - Item 3: Agrupamento.
 - Item 4: Listagem das informações consultadas no banco de dados.
 - Item 5: Somatórios dos campos agrupados.
 - Item 6: Campos consultados no banco de dados.
 - Item 7: Ferramentas disponibilizadas pelo Fast Report.

Por fim, na Figura 10 e 11 é possível visualizar o relatório gerado pelo sistema em sua versão em *Crystal* e *Fast Report* respectivamente.



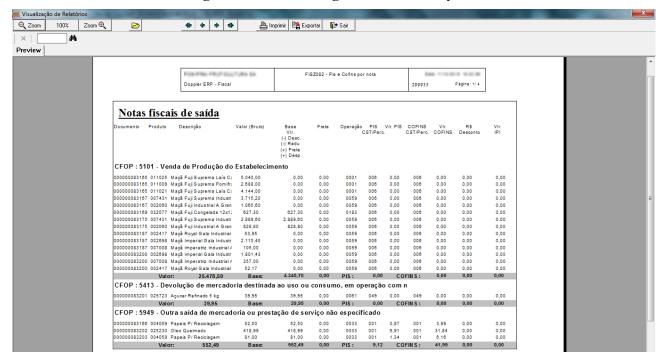
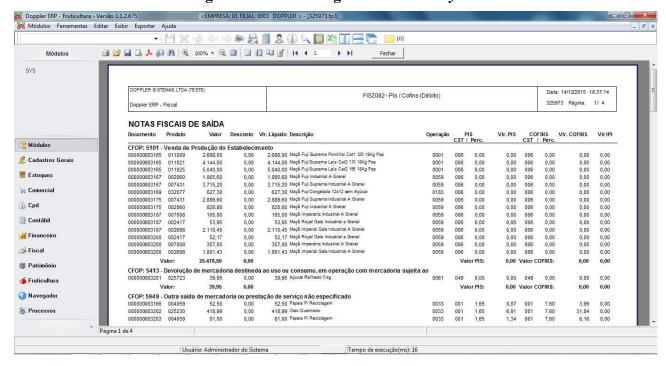


Figura 10: Relatório gerado versão Crystal

Figura 11: Relatório gerado versão Crystal



Fonte: O autor



Dificuldade: Em alguns relatórios na sua versão em *Crystal* os totalizadores estavam na parte de cima do agrupamento, *Header*; e somente em seguida os dados eram listados. No *Fast Reports*, isso não é possível devido a ele calcular os totais com base nos dados já listados.

Encaminhamento realizado: No caso em que os somatórios precisam ser listados junto com os grupos, antes dos dados serem listado, cria-se *procedures* e *Works*. Então os dados são consultados nas procedures e inseridos na *work*, esta é a tabela auxiliar, então os totalizadores são calculados e inseridos em colunas. No relatório apenas consulta os dados da *work*.

3.3. Ajuste de rotinas

| Data | | Horas | Setor | Atividade |
|------------|------------|-------|-----------------|--------------------------------|
| Início | Término | | | |
| 11/06/2015 | 30/10/2015 | 54:00 | Desenvolvimento | Alterações de telas do sistema |

Objetivo: Alterar telas do sistema conforme necessidade.

Descrição: Para ajustar o sistema conforme as necessidades dos clientes, alterações devem ser realizadas em telas já prontas. Muitas vezes é inclusão de campos, filtros para pesquisa, correção de algum erro, alterar campos para *comboedits*, etc.

Um exemplo de ajuste de rotina, ocorreu na Ctbz018, o qual é um relatório e está retornando dados duplicados, como mostrado na imagem abaixo.

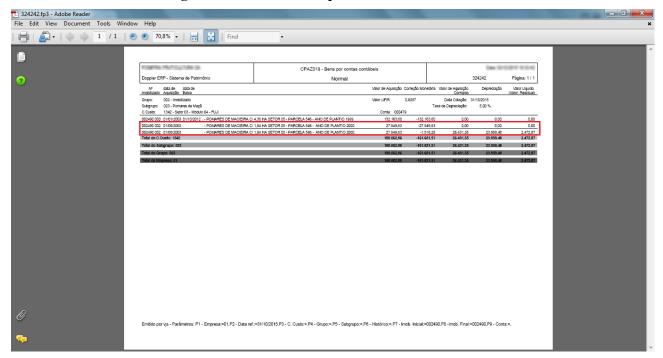


Figura 12: Relatório imprimindo dados errôneos

Para solucionar este problema era necessário buscar onde o erro estava acontecendo, que poderia ser na consulta do relatório em sua tabela auxiliar ou diretamente na *procedure*, a qual consultava os dados do banco. E foi na *procedure* que estava faltando um filtro, o qual era passado na tela; gerando duplicidade às informações. O relatório correto foi gerado e pode ser visualizado na Figura 13.



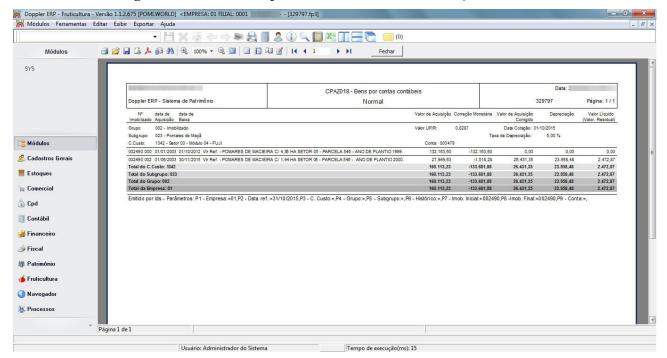


Figura 13: Relatório ajustado retornando informações corretas

Outro exemplo, é a CTB3300, a qual ao comparar com outra tela, informando os mesmos parâmetros o valor total de movimentações não estava correto. A Figura 14, mostra a *grid* com o valor do totalizador errado. Bem como, deveria adicionar uma nova coluna de Centro de Resultado na *grid* para permitir realizar o agrupamento.

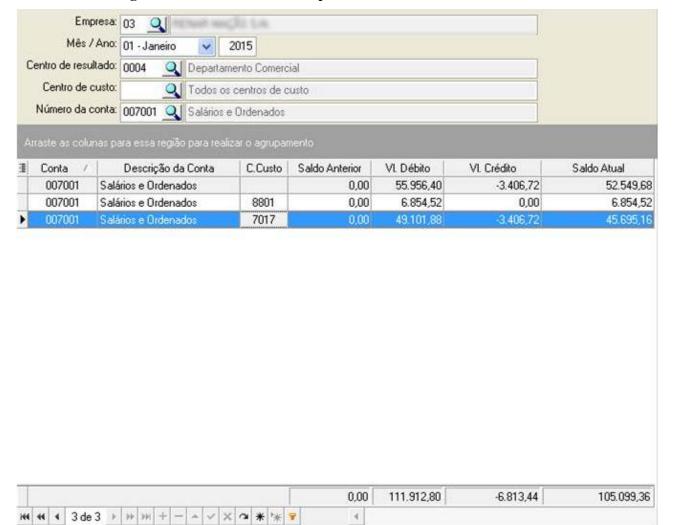


Figura 14: Rotina CTB3300 apresentando totalizadores inválidos

Para acertar o valor, adicionou-se mais um parâmetro Tipo de Estrutura, a tela para fazer a filtragem correta e retirou-se a primeira linha que retornava da consulta, esta se tratava de um totalizador. Porém, o problema estava na falta de *union* na consulta com as chaves estrangeiras. A imagem abaixo mostra a tela com os valores certos e a coluna C. Resultado incluída.

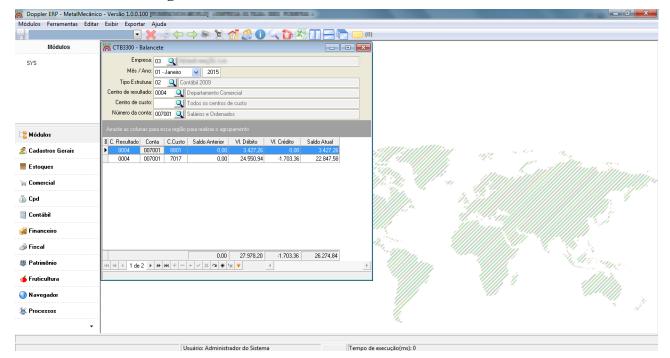


Figura 15: CTB3300 com valores certos e nova coluna

A tela FIS1109 do sistema é a responsável pelo Cadastro de código de serviços visualizada na Figura 16, a qual era necessária incluir um campo de Valor Limite de IRRF tanto na tela quanto no banco de dados.



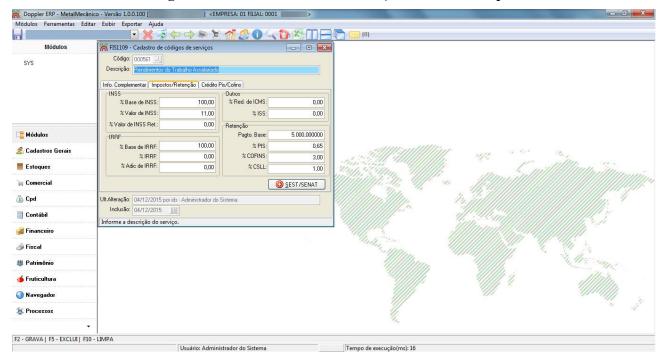


Figura 16: FIS1109 antes da inserção do novo campo

O campo foi adicionado juntamente com os *scripts* de validação e inserção na tela. No banco de dados, apenas foi necessário digitar o comando: *Alter table fist009 add vllimirrf number (17,2)*, "fist009" é a tabela e "vllimirrf" o nome do campo adicionado. A tela com o novo campo já adicionado pode ser vista na imagem abaixo.



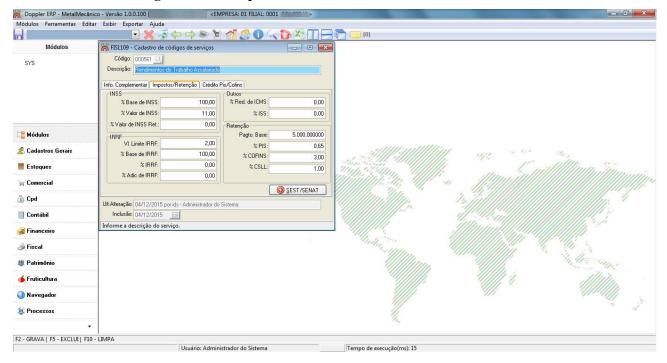


Figura 17: Campo VIr Limite IRRF adiconado à FIS1109

Dificuldade: Não há muita dificuldade nas tarefas citadas e executadas acima. O que acontece é que muitas vezes na busca por um erro não se sabe de qual lugar ele está partindo, se é na procedure ou no relatório por exemplo.

Encaminhamento realizado: A solução para tornar o processo mais fácil é iniciar a busca pelo ponto inicial, no caso acima na procedure pois a consulta é gerada na mesma, no caso da rotina CTBZ018 estava faltando um filtro.



3.4. Implementação de rotina

| Data | | Horas | Setor | Atividade |
|------------|------------|-------|-----------------|----------------------------------|
| Início | Término | | | |
| 11/06/2015 | 30/10/2015 | 20:00 | Desenvolvimento | Desenvolvimento de novas rotinas |

Objetivo: Desenvolver novas rotinas para usuários ou controle interno do sistema Doppler.

Descrição: Muitas vezes o usuário do sistema precisa de funcionalidades que o sistema ainda não possui, seja uma nova tela de cadastro ou de consulta. A tela é feita utilizando a ferramenta de desenvolvimento *Delphi*, assim como todo o sistema. Um exemplo, é a rotina Adm7110, que é uma tela de controle interno, Consulta de *Lock*. O objetivo é que através do componente cxGrid listasse os dados consultados do banco.

Após a tela ser implementada, a rotina deve ser cadastrada no sistema através da Adm1600, como mostra a figura abaixo.

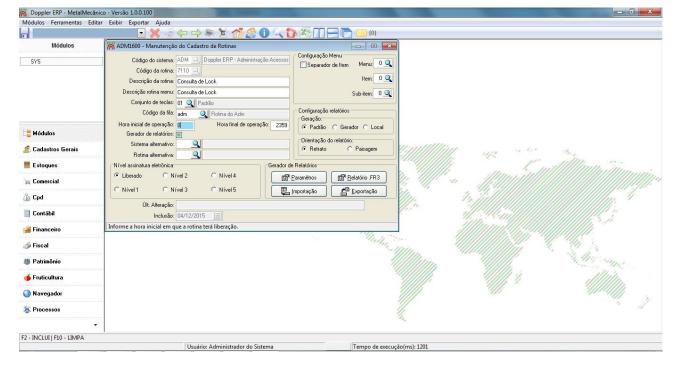


Figura 18: Cadastro da rotina Adm7110

Fonte: O autor

Após, o cadastro deve ser feito a liberação da tela neste caso como se trabalha nas bases de



homologação, ou seja, base de testes é utilizado o usuário ids, usuário administrador. A liberação da tela por usuário é feita através da rotina Adm2100, na qual deve-se informar qual o módulo e qual o usuário. Todas as telas serão listadas e as que não tiverem liberadas, ficarão com N nas colunas, assim como mostra a Figura 19 para efetuar a liberação deve-se dar dois cliques para trocar para S e salvar.

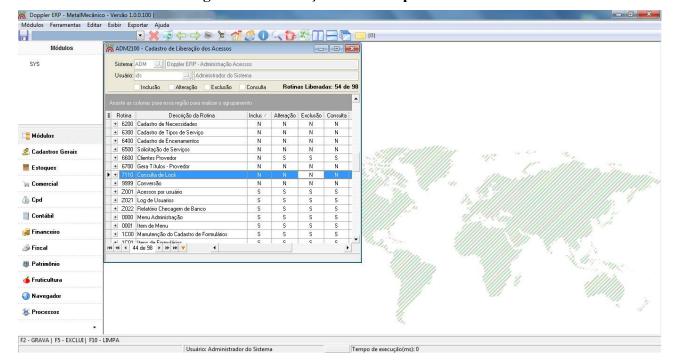


Figura 19: Liberação de rotina para usuário

Fonte: O autor

E por fim, basta chamar a rotina desenvolvida, a qual é apresentada na Figura 20.



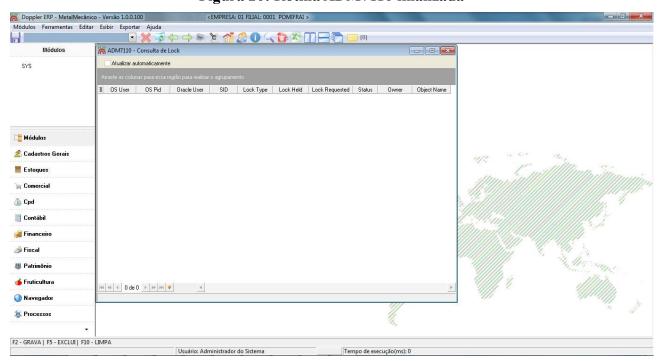


Figura 20: Rotina ADM7110 finalizada

4. CONCLUSÃO

Para poder atuar na empresa, foi necessário aprender uma nova linguagem de programação, novas ferramentas e *PL/SQL*. Assim como, aprender a programar seguindo os padrões já adotados, para resultar em um código limpo e compreensível a todos os membros da equipe.

Uma das partes interessantes do estágio é poder ver como o usuário final interage com o sistema. Muitas vezes os programadores não se dão conta das necessidades dos usuários e como precisam compreender o que fazer e como fazer no sistema desenvolvido para ele. Foi visto na prática, como a usabilidade interfere no rendimento de uma pessoa e consequentemente na empresa, a qual utiliza o ERP.

Além disso, o estágio foi de suma importância para conhecer o mercado de Tecnologia da Informação e as necessidades da região nesta área; assim como, os softwares existentes e suas aplicações. Da mesma forma, conhecer várias pessoas e suas funções, as divisões de tarefas, unir saberes em prol de um objetivo final da equipe, fortaleceu o lado profissional preparando o estagiário para o mercado de trabalho. Por fim, foi uma experiência de grande valor, a qual agregou um grande aprendizado na área de desenvolvimento de software e banco de dados.

5. REFERÊNCIAS BIBLIOGRÁFICAS OU REFERÊNCIAS

CANTU, C. H. **Qual gerador de relatórios utilizar com Delphi.** 200?. Disponível em: http://www.edudelphipage.com.br/opinioes/qual-gerador-de-relatorios-utilizar-com-delphi/carlos-cantu. Acesso em: 28 nov 2015.

CARVALHO, F. P. **Apostila de linguagem de programação I.** 2006. Faculdade de Informática de Taquara. Disponível em: < https://fit.faccat.br/~fpereira/apostilas/apostila_prog1_ago2006.pdf acesso em: 17 fev de 2016.

MENDES, S. C. **Componente cxGrid da DevExpress.** 200?. Disponível em: < http://www.devmedia.com.br/componente-cxgrid-da-devexpress-revista-clube-delphi-magazine-147/26881 acesso em: 29 nov de 2015.

SANTOS, D. S. Aimportância do estágio para a vida acadêmica e profissional do aluno. 2014. Disponível em: < http://www.portaleducacao.com.br/iniciacao-profissional/artigos/58044/a-importancia-do-estagio-para-a-vida-academica-e-profissional-do-aluno acesso em: 07/02/2016.

GARCIA, D. **Dicas sobre o componente DBGrid do Delphi.** 2013. Disponível em: < https://drgarcia1986.wordpress.com/2013/01/17/dicas-sobre-o-componente-dbgrid-do-delphi/ acesso em: 07 fev de 2016.

