

# Sistema de Gerenciamento para Indústria Gráfica

Lucas Bahri<sup>1</sup>, William Diogo Pasternak<sup>2</sup>, Elena Mariele Bini<sup>3</sup>

<sup>1,2,3</sup> Tecnologia em Análise e Desenvolvimento de Sistemas

Faculdade Guairacá – Guarapuava – Paraná

lucas\_bahri@hotmail.com, wiliampasternak@gmail.com,  
elenamarielebini@gmail.com

**Abstract.** *In this article, we present the steps taken to develop a system of printing industry, which aims to overcome with the use of software developed lack of automated management processes. To develop this system we used Java technology, along with OpenSwing framework, Hibernate and MySQL database Using the system has brought benefits, enabled the company to take greater control of all their records and assisting in the administration of business.*

**Resumo.** *Neste artigo, apresentam-se as etapas realizadas para o desenvolvimento de um sistema para gestão de indústria gráfica, que tem por objetivo suprir a falta de gerenciamento automatizado dos processos da empresa. Para o desenvolvimento deste sistema empregou-se a tecnologia Java, juntamente com os frameworks OpenSwing e Hibernate e o banco de dados MySQL. A utilização do sistema trouxe benefícios, pois possibilitou à empresa ter um maior controle dos seus cadastros, auxiliando assim na administração dos negócios.*

## Introdução

Indústrias gráficas fazem parte da área de informação e comunicação atuam em diversos segmentos, destacando-se o editorial e gráfico. Os processos mais comuns são serviços de pré-impressão, impressão dos produtos e serviços de acabamento. Grande parte dos trabalhos gráficos utiliza papel, sendo-os principais produtos finais: embalagens, editorial, formulários, promocional, artigos de papelaria, pré-impressão, impressos comerciais.

Partindo da necessidade de uma empresa gráfica de possuir um sistema gestor, desenvolveu-se o sistema BORG - Sistema de Gerenciamento de Indústria Gráfica. O *software* foi desenvolvido em linguagem de programação Java, utilizando o *framework* OpenSwing. Para tal desenvolvimento também foram empregadas às ferramentas Astah para a elaboração de diagramas propostos pela UML (*Language Unified Modeling*), o ambiente de desenvolvimento Netbeans 6.8 para programação. Para persistência e recuperação de dados foi utilizado o banco de dados MySQL, juntamente com o *Data Base Management System* (SGBD) MySQL Workbench e o *framework* Hibernate.

O objetivo deste trabalho é apresentar as etapas de desenvolvimento de um *software* para gerenciamento de uma empresa gráfica capaz de oferecer a empresa um maior controle de todos seus produtos, vendas, pedidos e cadastros auxiliando na administração dos negócios.

## **Fundamentação Teórica**

Para Laudon e Laudon (2007), as empresas estão a cada momento sempre tentando melhorar a eficiência de suas operações para conseguir maior lucratividade. Ainda segundo Laudon e Laudon (2007), dentre as ferramentas que as empresas dispõem, as tecnologias e os sistemas de informação estão entre as mais importantes para atingir altos níveis de eficiência e produtividade, também se torna a principal ferramenta em que as empresas apostam para desenvolver novos produtos e serviços.

Já O'Brien (2004) afirma que sistemas e tecnologias de informação tornaram-se componentes vitais para atingir o sucesso de empresas e organizações, é tão importante possuir uma compreensão de sistemas de informação como entender qualquer outra área nos negócios.

Os sistemas de informação baseados em computador usam a informática a como recurso para melhorar sua eficiência, as tecnologias empregadas melhoram a capacidade e a velocidade das funções de coleta, armazenamento, processamento e distribuição da informação (Audy, Andrade e Cidral, 2005).

Uma das tecnologias empregadas para este trabalho foi o paradigma de programação orientado a objetos, segundo Mendes (2009) é a forma de pensar e escrever o código existem diversos paradigmas com conceitos e abstrações, entre os paradigmas de programação o de orientação a objetos é mais usado, segundo Horstmann (2010) a programação orientada a objetos é o paradigma dominante nos dias atuais, tendo substituído as técnicas da programação estruturada. Para Mendes (2009), na orientação a objetos o mundo real é visto como sendo constituído de objetos autônomos, que interagem entre si e cada objeto tem seu próprio estado, atributos e comportamento. Claro e Sobral (2008) afirmam que a orientação a objetos torna o desenvolvimento mais rápido, pois ele ocorre em blocos de códigos, com isso pode-se alcançar uma maior qualidade.

Uma linguagem orientada a objetos de grande representatividade é o Java, para Mendes (2009), o paradigma da orientação a objetos só ganhou credibilidade com a linguagem Java, de acordo com Horstmann (2010) a linguagem surgiu, quando um grupo de engenheiros, resolveu criar uma linguagem de computador que pudesse ser utilizada para dispositivos de consumidor.

Segundo Laudon e Laudon (2007) para cada ambiente computacional em que a linguagem Java é usada, existe uma máquina virtual Java, que interpreta o código, assim não está vinculado a um processador ou sistema operacional específico, ainda de acordo com Laudon e Laudon (2007) a plataforma Java migrou para celulares, automóveis,

players de música, videogames e sistemas de televisão a cabo. Afirma Horstmann (2010) que existem diversas linguagens de programação disponíveis e poucas nem mesmo se aproximam do poder do Java, a biblioteca que elas possuem praticamente não é uma biblioteca, forçando-o a criar uma própria, já o Java possui uma coleção enorme contendo uma grande quantidade de códigos reutilizáveis.

Para desenvolvimento de *softwares* em Java diferentes ferramentas podem ser empregadas, entre elas o *Netbeans Integrated Development Environment* (IDE) de acordo o site oficial do *Netbeans* (2014), ela é uma ferramenta de desenvolvimento gratuita, com código-fonte aberto e possui uma grande comunidade de usuários e desenvolvedores em todo o mundo, permite o desenvolvimento rápido e fácil de aplicações *desktop* Java, móveis e *Web*, além de uma ampla variedade de *plug-ins* que estão disponíveis para todos os tipos de desenvolvimento.

O emprego do *framework OpenSwing* é uma alternativa para o desenvolvimento em Java. De acordo o Eije (2010) *OpenSwing* é um *framework* de código aberto que pode ser utilizado para desenvolver aplicações Java baseadas em *Swing*. Segundo Arnold, Gosling e Holmes (2007) os componentes *Swing* são um poderoso conjunto de controles de interfaces gráficas, eles são escritos para parecer e se comportar para o usuário de forma idêntica em todos os sistemas. Segundo o Eije (2010) com a suíte de componentes gráficos avançados, o desenvolvedor não precisará gastar tempo e esforço desenvolvendo novos componentes, os controles gráficos do *framework OpenSwing* podem ser comparados aos utilizados em *Delphi* e *Visual Studio*. Ainda segundo Eije (2010) os componentes do *OpenSwing* são totalmente desacoplados e baseados no padrão *Model View Controller* (MVC), o *framework* segue as especificações Java Beans, deste modo pode ser utilizado em diversas IDEs como *Eclipse* e *NetBeans*.

Outra alternativa para o desenvolvimento em Java é a aplicação do *framework Hibernate*. Segundo King et al (2013) a meta do *Hibernate* é aliviar o desenvolvedor em 95% de dados comuns de persistência relacionados às tarefas de programação. Para Elliott e O'brien (2009) é uma maneira fácil de trabalhar e eficientemente com informações de um banco de dados relacional na forma de objetos naturais Java, realiza suas tarefas tão facilmente que você quase esquece que está trabalhando com banco de dados.

De acordo com Oliveira (2002), um Sistema de Informação é desenvolvido esperando a entrada de dados que serão processamentos e gerar saída das informações. Com o tempo, surgiu a necessidade de armazenar os Dados, então o armazenamento e a recuperações das Informações passaram a representar um papel fundamental na tecnologia. Ainda segundo Oliveira (2002) em 1970, membros da IBM, publicaram um trabalho chamado “Um Modelo Relacional De Dados para Grandes Bancos De Dados Compartilhados” com esse trabalho se estabeleceu princípios sobre gerência de Banco de Dados, essa foi à base utilizada na criação de uma linguagem padrão a linguagem *Structured Query Language* (SQL).

Segundo Abreu (2002) SQL se tornou um padrão para manipulação de dados em banco de dados. As entidades *American National Standard Institute* (ANSI) e a *International Standards Organization* (ISSO 9) vem padronizando a linguagem.

De acordo com Heuser (1998) a SQL é oferecida em praticamente todos os sistemas de gerência de Banco de Dados, ainda Heuser (1998) define SGBD (*Data Base Management System*) como *softwares* que surgiram com o objetivo de facilitar a programação de Banco de Dados, tornar mais simples a manutenção e aumentar a produtividade.

O *MySQL* é um sistema de gerenciamento de banco de dados (SGBD) que utiliza a linguagem SQL, de acordo com *Mysql* Ab (2008) *MySQL* é o mais popular sistema de gerenciamento de banco de dados SQL *Open Source*, sendo um servidor robusto, muito rápido, fácil de usar, multitarefa e multiusuário. Ainda, segundo o *Mysql* Ab (2008) o Servidor *MySQL* pode ser usado em sistemas de produção com alta carga e missão crítica bem como pode ser embutido em programa de uso em massa.

No desenvolvimento de *software*, a *Language Unified Modeling* (UML) pode ser aplicada buscando uma visão ampla e detalhada do sistema. De acordo com Lima (2011) a modelagem existe porque não é complicado entender os sistemas completamente. Ainda segundo Lima (2011) cita o exemplo da construção civil onde plantas e maquetes são desenvolvidas para representar edifícios, quanto maior a complexidade do produto, maior a necessidade de técnicas de modelagem, com um conjunto de visões suficientes para garantir sua compreensão. Para Bezerra (2007) a UML é uma linguagem visual para modelar sistemas orientados a objetos, que define elementos gráficos para serem utilizados na modelagem de sistemas. Com esses elementos gráficos se torna mais fácil construir diagramas que representam diversas perspectivas de um sistema.

O diagrama de caso de uso é uma das representações existentes no conceito UML, para Lima (2011) mostram conceitualmente o conjunto de funções que o sistema deve executar para atender os requisitos do cliente, o sistema é visto sob o entendimento do usuário, por isso o diagrama de caso de uso é a base para as demais visões, além de ser essencial para análise, implementação, testes e plano do desenvolvimento do sistema. Para Melo (2010) o diagrama de caso de uso, é representado por atores e seus relacionamentos que expressam as funcionalidades do sistema.

O digrama de classes, outra representação do conceito UML também foi utilizado, segundo Bezerra (2007) de todos os diagramas UML, esse é o mais rico em termos de anotação, são usados na construção do modelo de classes desde a análise até o nível de especificação. O diagrama de sequência também oferecido pela UML foi usado, conforme Bezerra (2007) tem como meta mostrar as interações entre os objetos na ordem temporal em que elas acontecem.

Em um processo de desenvolvimento de *software*, padrões de projeto também devem ser considerados. Segundo Gamma et al (2000) capturam soluções que foram

desenvolvidas e aperfeiçoadas ao longo do tempo, torna mais fácil reutilizar técnicas já testadas, projetos e arquiteturas bem sucedidas. Dentre estes padrões estão o padrão MVC. Para Gamma et al (2000), a abordagem MVC é composta por objetos ela os separa para aumentar a flexibilidade e a reutilização, são três tipos de objetos: o Modelo é o objeto de aplicação, a Visão é a apresentação na tela e o Controlador é o que define a maneira como a interface do usuário reage às entradas do mesmo.

Segundo Freeman e Freeman (2007) os objetos modelo, apresentam a razão de ser do seu aplicativo, contem dados e lógica. Segundo Freeman e Freeman (2007) os objetos visualização tendem a ser controles usado para exibir e editar, os dados fluem entre o modelo e a visualização, o controlador faz a mediação, manipula todos os valores que sofrem alteração.

Buscando-se a qualidade no desenvolvimento do *software* apresentado, optou-se pelo ciclo de vida de prototipação. De acordo com Bezerra (2007) é um modelo de uma sequência específica de fases para a construção de um sistema, existem diversos modelos de ciclo de vida, a diferença entre eles está na maneira de como as diversas fases são encadeadas.

Ainda segundo Bezerra (2007), protótipos são construídos para telas de entrada e saída ou para o sistema como um todo, para ser usado na sua validação, é revisto por um ou mais usuários, que fazem críticas de suas características, então é refinado de acordo com as críticas, esse processo de revisão continua até o protótipo seja aceito pelos usuários, esse modelo de ciclo de vida tem o objetivo de assegurar que os requisitos do sistema foram realmente bem entendidos.

Buscando-se aumentar o grau de facilidade do usuário ao interagir com o *software* desenvolvido, foi aplicada a usabilidade. Que segundo a *International Organization for Standardization* (ISO) (2014, *apud* Cybis, 2003) a maior desenvolvedora mundial de normas internacionais, a usabilidade pode ser entendida como uma das qualidades de um sistema, a capacidade que um sistema interativo oferece ao usuário, em um determinado contexto de operação, para a realização de tarefas, de modo eficaz, eficiente e agradável, que o usuário atinja suas metas com menos esforços.

Segundo Nielsen e Loranger (2007), usabilidade é um atributo de qualidade relacionado à facilidade do uso de algo, mais especificamente, à rapidez com que os usuários conseguem aprender a usar alguma coisa, a eficiência deles ao usá-la, o quanto lembram daquilo seu grau de propensão a erros e o quanto gostaram ao utilizá-la. De acordo com Cybis (2003) as interfaces difíceis, que aumentam o trabalho do usuário, trazem consequências negativas como a resistência ao uso, chegando ao abandono do sistema e a prejuízos.

Na próxima seção, Materiais e Métodos, os passos realizados para o desenvolvimento do *software* BORG serão descritos em detalhes.

## Materiais e Métodos

Seguindo o ciclo de vida de prototipação, o desenvolvimento do sistema iniciou-se com a fase de coleta dos requisitos. Nesta fase os requisitos foram levantados por meio de entrevistas com os donos da empresa, diálogos e análise de documentos. Assim obtendo um detalhamento completo do *software* a ser desenvolvido, os requisitos levantados para o sistema foram: sistema de login para controlar quem acessa o sistema com lembrete de senha por e-mail, cadastros dos clientes, fornecedores e funcionários com campos de observação para clientes e fornecedores, cadastro de pedidos com campo para saber se o produto foi já entregue, controle de compras no estoque com campo para os números da nota fiscal e observação, controle de vendas, gerador de amostragem das impressões para o gráfico com dados e imagem da impressão e a geração de relatórios, ficha da coleta de requisitos pode ser visualizada no apêndice A.

A próxima etapa do projeto foi à modelagem do sistema, elaborando os diagramas UML: diagrama de caso de uso, diagrama de sequência e diagrama de classe com a utilização da ferramenta *Astah*, os diagramas estão no apêndice B, C, D, E, F. A etapa seguinte foi à modelagem do banco de dados. Para essa fase foi selecionada a ferramenta *MySQL Workbench* que conforme o site oficial do *MySQL* (2014), é uma ferramenta visual unificada, fornece modelagem de dados, desenvolvimento de SQL e ferramentas de administração abrangentes para configuração do servidor, administração de usuários, *backup* e está disponível em Windows, Linux e Mac OS X. Ainda segundo *MySQL* (2014), ele inclui tudo o que um modelador de dados precisa para a criação de modelos ER (Entidade Relacionamento) complexas e engenharia reversa. O modelo lógico do sistema está no apêndice G.

Na sequência iniciou-se a etapa da construção do protótipo que passou por avaliação do cliente. A ficha da avaliação do protótipo está no apêndice H, houve o aceite do protótipo, não havendo a necessidade de alterações, que se manteve como inicialmente apresentado, assim, ficou decidido utilizar o paradigma *Multiple Document Interface* (MDI) para desenvolver a aplicação, segundo EIJE (2010) nas aplicações baseadas no paradigma MDI existe uma janela principal da aplicação e as demais são internas ou "filhas" da principal, e também foi utilizado para alterar a aparência do sistema o *Look and Feel* que define a forma como seus componentes serão desenhados na tela.

Então, iniciou-se a fase de codificação do *software* seguindo o padrão MVC. Para o desenvolvimento foi utilizada a IDE *NetBeans* e a tecnologia de programação Java SE. Para aumentar a produtividade no desenvolvimento foi empregado o *framework OpenSwing* e para a geração de relatórios a ferramenta *iReport*, que de acordo com a Jaspersoft Corporation (2013) é uma ferramenta de código aberto que pode permitir criar relatórios complexos rapidamente com muita facilidade a partir de qualquer tipo de aplicação Java através da biblioteca *JasperReports*, ele está escrito em

Java, ele gera relatórios em XML, HTML, DOCX, ODT e outros formatos. Foi aplicado o *framework Hibernate* para o mapeamento objeto-relacional do banco dados.

As telas do sistema foram desenvolvidas seguindo critérios de usabilidade como visibilidade do status do sistema, controle do usuário e liberdade, consistência de padrões, prevenção de erros e flexibilidade e eficiência de uso. Tela do sistema indicando a aplicação destes critérios está no apêndice I.

Após o desenvolvimento do sistema, foi aplicado um questionário de avaliação final do *software* que foi respondido pelo cliente, conforme o apêndice J. Ao realizar o mesmo buscou-se obter uma impressão positiva, tal expectativa foi alcançada com êxito, pois não se obteve nenhum tipo de reclamação relevante referente ao sistema.

## **Resultados e Discussões**

Os resultados deste trabalho de pesquisa podem ser analisados em três aspectos: (i) os resultados obtidos com a implantação do *software* na empresa cliente e (ii) os resultados obtidos mediante as ferramentas e processos empregados durante o desenvolvimento, (iii) resultados de aprendizagem.

Dentre os resultados obtidos com a implantação do sistema na empresa cliente é possível afirmar que o emprego deste foi capaz de suprir a falta de gerenciamento automatizado dos processos da empresa cliente: Antes toda a informação da empresa estava “armazenada” em papéis, fato este que ocasionava em vários momentos perdas de dados, além da redundância e inconsistência dos dados. Assim, o sistema implementado possibilitou a empresa ter um maior controle de todos seus produtos, vendas, pedidos e cadastros auxiliando na administração dos negócios, elevando o patamar de qualidade dos serviços prestados. As ferramentas e processos empregados no desenvolvimento do *software* geraram resultados que são mencionados nos parágrafos que seguem.

Utilizando os diagramas UML foi possível construir diferentes visões do sistema garantindo a compreensão do mesmo antes da sua codificação. O uso da ferramenta *MySQL Workbench*, por meio de sua interface simples e descomplicada, para a modelagem do banco de dados do sistema permitiu trabalhar diretamente com componentes gráficos elaborando uma modelagem detalhada e eficaz das classes de persistência do sistema e visualizar a melhor maneira de relacionar os dados para seguir as regras de negócio.

Com o uso do *framework* de persistência *Hibernate* foi possível desenvolver consultas, exclusão e atualizações dos dados do sistema de forma menos complexa, reduzindo assim o tempo de desenvolvimento.

A utilização da linguagem de programação Java e seu ambiente de desenvolvimento *NetBeans*, tornou o custo do desenvolvimento praticamente zero, outras vantagens que a tecnologia Java trouxe foram: como não é vinculada a um

processador ou sistema operacional específico, possui portabilidade, rodando em várias plataformas, e o suporte dado é satisfatório, pois a comunidade Java, hoje, é extensa, com diversos fóruns de discussões, além de que existe uma variedade de *frameworks* que visam facilitar o trabalho do desenvolvedor.

Quando um desenvolvedor começa a trabalhar com componentes gráficos espera produtividade, infelizmente os componentes padrões do Java deixam a desejar, forçando o programador a desenvolver soluções, com o uso do *framework* *OpenSwing* o programador obtém um conjunto de componentes gráficos avançados, o programador não precisa gastar tempo e esforço desenvolvendo soluções e novos componentes, além que são componentes totalmente desacoplados e baseados no padrão MVC, assim o desenvolvimento se torna muito mais produtivo, rápido e eficiente.

A realização deste trabalho contribuiu para aprendizagem de inúmeros conceitos, especialmente é possível destacar: a utilização do ambiente de desenvolvimento *NetBeans* exigiu estudo do seu Gerenciadores de *Layout*; o *framework* *Hibernate* necessitou estudos sobre *criteria*, *annotations* e mapeamento de classes; o emprego da linguagem Java exigiu estudos do padrão *Singleton*, da API *JavaMail*, Classes *WebService* e *Swing* *JDialog*, métodos *concat()*, *split()*, *maskFormatter()*, *setPlaceholderCharacter()*, *setMnemonic()*, *setToolTipText()*, recursos XML e para alterar a aparência o *Look And Feel*. A utilização do *framework* *OpenSwing* exigiu o estudo sobre aplicações MDI, componentes *Form*, *Grid*, *Lookup* e métodos e classes *MDIFrame()*, *Properties()*, *ClientFacade()*, *Listener*, *ValueObject*, *SelectionMode()*, *setAccelerator()*, *ApplicationFunction*, *setMnemonic*, *KeyStroke* e *ClientSettings*; para a geração de relatórios exigiu o estudo do *framework* *IREport* e para a geração da amostra a *Application Programming Interface* (API) *IText*.

## **Considerações Finais**

A falta de gerenciamento automatizado dos processos da empresa cliente era um problema, pois toda a informação estava “armazenada” em papéis o que ocasionava perda de dados, além da redundância e inconsistência dos mesmos, além de que não existir um controle sobre eles. O *software* desenvolvido para resolver esses problemas, possibilitou à empresa ter um maior controle de todos seus produtos, vendas, pedidos e cadastros auxiliando na administração dos negócios.

Como trabalho futuro, será desenvolvido uma nova versão do *software* para *Web*, utilizado a tecnologia *Java Server Faces*, com acesso aos dados online onde o cliente poderá ter acesso aos dados da empresa de qualquer lugar, também na versão *Web* será implementado cadastro de anúncios de compras e vendas, pois a empresa possui um jornal de negócios impresso semanalmente.

## Referências

- ABREU, Felipe Machado Mauricio. “Projeto de Banco de Dados”: Uma Visão Prática. 8. ed. São Paulo: Ed. Érica, 2002.
- ARNOLD, Ken; GOSLING, James; HOLMES, David. “A Linguagem de Programação Java”. 4. ed. Porto Alegre: Ed. Bookman, 2007.
- AUDY, J. L. N.; ANDRADE, G. K.; CIDRAL, A. “Fundamentos de Sistemas de Informação”. Porto Alegre: Ed. Bookman, 2005.
- BEZERRA, Eduardo. “Princípios de Análise e Projeto de Sistemas com UML”. 2. ed. Rio de Janeiro: Ed. Campus, 2007.
- CLARO, Daniela Barreiro; SOBRAL, João Bosco Manguiera. “Programação em JAVA”. Florianópolis: Ed. Copley Pearson Education, 2008.
- CYBIS, W. A. “Engenharia da Usabilidade”: Uma Abordagem Ergonômica”. Florianópolis, 2003.
- EIJE, Albert. “Dominando o OpenSwing”. Rio de Janeiro: Ed. Ciência Moderna, 2010.
- ELLIOT, James; O'BRIEN, Timothy M.; FOWLER, Ryan. “Dominando Hibernate”. Rio de Janeiro: Ed. atlas books, 2009.
- FREEMAN, Eric. FREEMAN, Elisabeth. “Use a Cabeça! Padrões de Projeto”. 2. ed. Rio de Janeiro: Ed. Alta Books, 2007.
- GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. “Padrões de projeto: soluções reutilizáveis de software O.O”. Porto Alegre: Ed. Bookman, 2000.
- HEUSER, Carlos A. “Projeto de Banco de Dados”. 4. ed. Porto Alegre: Ed. Sagra e Luzzato, 1998.
- HORSTMANN, Cay, S; CORNELL, Gary. “Core Java: Fundamentos - Volume I. 8”. ed. São Paulo: Ed. Pearson Prentice Hall, 2010.
- ISO. (2014) “About ISO”. Disponível em: < <http://www.iso.org/iso/home/about.htm>>. Último acesso: Abril, 2014.
- Jaspersoft Corporation (2013) “iReport Ultimate Guide”.
- KING, Gavin et al. “Documentação de Referência Hibernate”. Disponível em: <<http://www.hibernate.org/docs>>. Acesso em 15 nov. 2013.
- LAUDON, Kenneth Craig; LAUDON, Jane Price. “Sistemas de informação gerenciais”. 7. ed. São Paulo: Ed. Pearson Prentice Hall, 2007.
- MELO, Ana Cristina. “Desenvolvendo aplicações com UML 2.2: do conceitual à implementação”. 3. Ed. Rio de Janeiro: Ed. Brasport, 2010.
- LIMA, Adilson da Silva. “UML 2.3: Do Requisito à Solução”. 1. ed. São Paulo: Ed. Érica, 2011.

- MENDES, Douglas Rocha. “Programação Java: com Ênfase em Orientação a Objeto”. São Paulo: Ed. Novatec, 2009.
- MySQL. (2014). “MySQL Workbench 5.2”. Disponível em: <<https://www.mysql.com/products/workbench/>>. Último acesso: Março, 2014.
- MYSQL AB. “Manual de referência do Mysql 4.1. Tradução do manual oficial”. Disponível em: <<http://downloads.mysql.com/docs/refman-4.1-pt.a4.pdf>>. Acesso em: 19 nov. 2013.
- Netbeans. (2012). “Netbeans IDE Features”. Disponível em: <<http://netbeans.org/features/index.html>>. Último acesso: Março, 2014.
- NIELSEN, Jakob e LORANGER, Hoa. “Usabilidade na Web”. Rio de Janeiro: Ed. Elsevier, 2007.
- O’BRIEN, James A. “Sistemas de informação: E as decisões gerenciais na era da internet”. 2. Ed. São Paulo: Ed. Saraiva, 2004.
- OLIVEIRA, Celso Henrique Poderoso de. “SQL: Curso Prático”. São Paulo: Ed. Novatec, 2002.

## Apêndices

### Apêndice A: Ficha da coleta de requisitos

**ENTREVISTA**

**Nome do entrevistado:** Helena Koçowski  
**Nome da empresa:** Grandes Negócios  
**Cargo:** Dona, Gerente e Orçamentista.  
**Data:** 13/09/2013

**PARTE I – ESTABELECENDO O PERFIL DO CLIENTE**

**1 - Quem é o cliente?**  
É uma Editora e indústria Gráfica, de sociedade Ltda-me, proprietária do jornal de classificados semanais Grandes Negócios, Lista Telefônica Disk Certo e Gráfica com o nome fantasia Grandes Negócios.

**2 - Quais são as suas responsabilidades na empresa?**  
Atender com qualidade e agilidade para satisfazer todos os clientes e usuários dos nossos produtos, através do controle de orçamentos, qualidade dos produtos, financeiro, estoque e controle.

**3 - Quais são os produtos?**  
Jornal Semanal de Classificados, Criação e Produção de Artes Gráficas, Publicidade e Propaganda no Jornal, Site, Lista Telefônica, impressos gráficos (sacolas, pastas, caixas e embalagens, envelopes, panfletos, folders, agendas, cartões, blocos, papéis timbrados, convites, cartazes).

**4 - Para quem você gera esses produtos?**  
Empresas e leitores de todas as idades e classes, das cidades de Prudentópolis, Irati, Imbituva e Guamiranga.

**5 - Quais são as necessidades da empresa?**  
**Um sistema com as seguintes funcionalidades:**

1. Sistema de Login: Para controlar quem acessa o sistema com lembrete de senha por e-mail caso o gerente não esteja disponível para trocar.
2. Controle de Vendas;
3. Controle de Compras no Estoque: Com campo para os números da nota fiscal e observação.
4. Cadastro de Pedidos: Com campo para saber se o produto foi já foi entregue.
5. Relatórios;
6. Cadastros de Cliente, Fornecedores e Funcionários: Campos de observação para clientes e fornecedores.
7. Amostragem: Gerador de amostragem das impressões para o gráfico com dados e imagem da impressão

**6. Quais são os problemas que interferem para o sucesso de seu trabalho?**  
Hoje toda a informação da empresa está presa no papel, ocasionando em perdas por estarem armazenadas localmente e redundância por não armazenar os dados de maneira eficiente. Falta de automatização, cadastros das pessoas envolvidas com a empresa, o estoque, pedidos e a amostragem, uma amostra do trabalho final, para o gráfico.



Continua na próxima página...

Continuação...

## **PARTE II: ENTENDENDO O AMBIENTE DO USUÁRIO**

### **1 - Quem são os usuários?**

- Atendente
- Gerente
- Auxiliar Gráfico
- Gráfico
- Arte- Finalista

### **2 - Qual é a experiência dos usuários em computação?**

- Atendente - Nível Básico.
- Gerente – Nível Avançado
- Auxiliar Gráfico – Nível Médio
- Gráfico – Nível Básico
- Arte-Finalista - Nível Avançado

### **3 - Qual sistema operacional é usado?**

Windows XP/7.

### **5 – Existem planos para uma futura mudança de sistema operacional?**

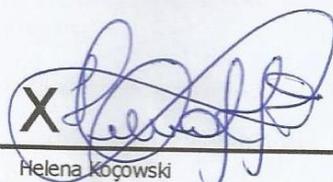
Não.

### **6 - Quais são as suas expectativas para a usabilidade e desempenho do produto?**

Previsível, preciso, rápido e interativo.

### **7 - Quais são as suas expectativas para o tempo de treinamento?**

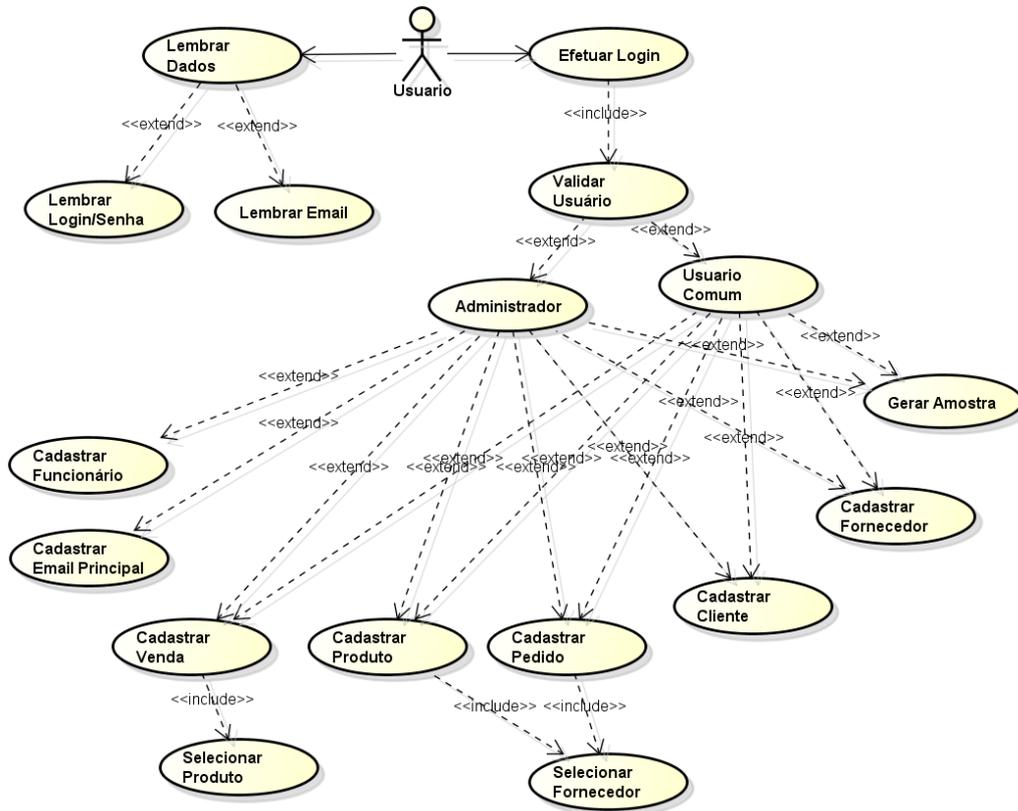
Dois dias.

X 

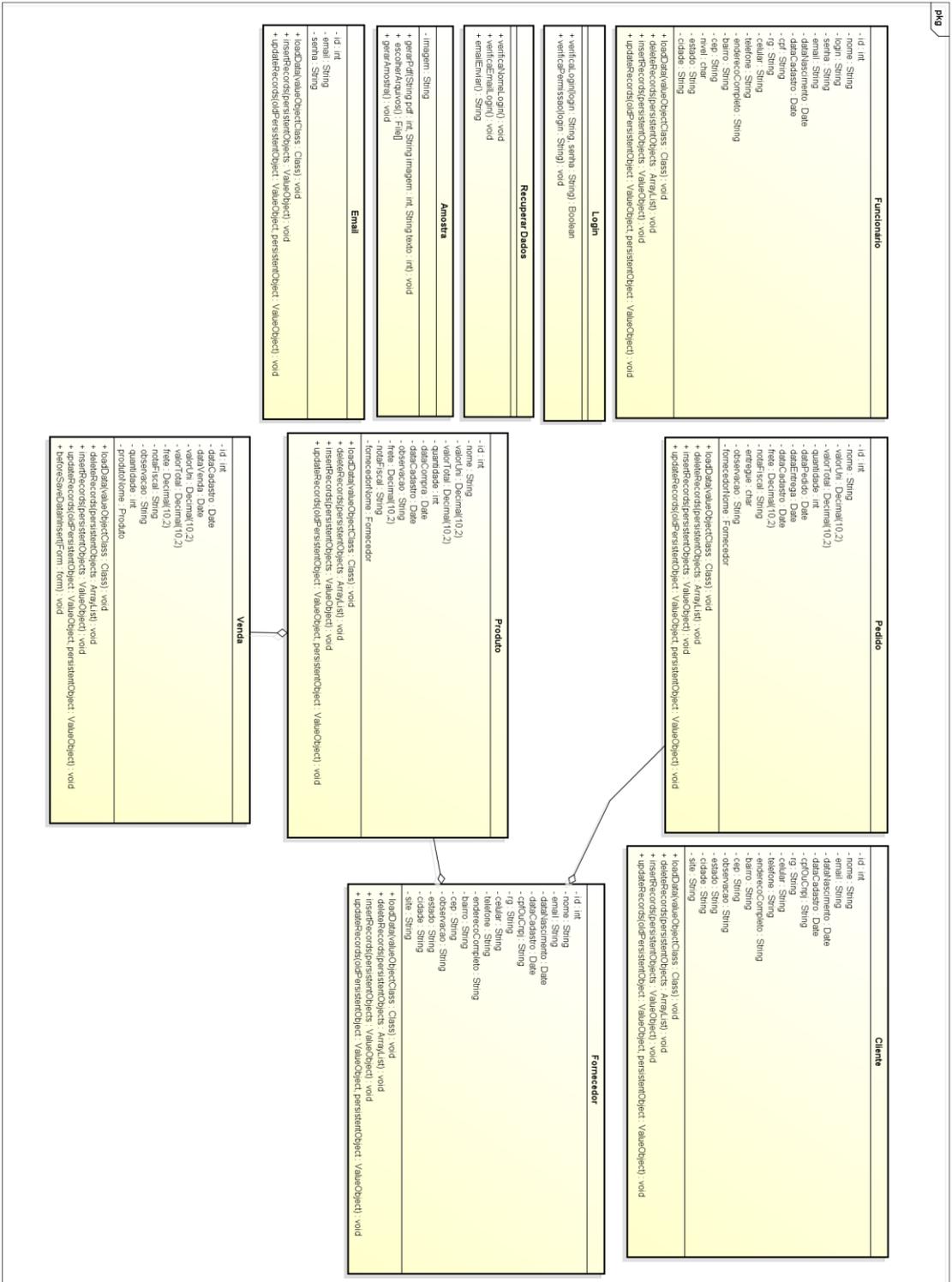
---

Helena Koçowski  
Gerente e Orçamentista

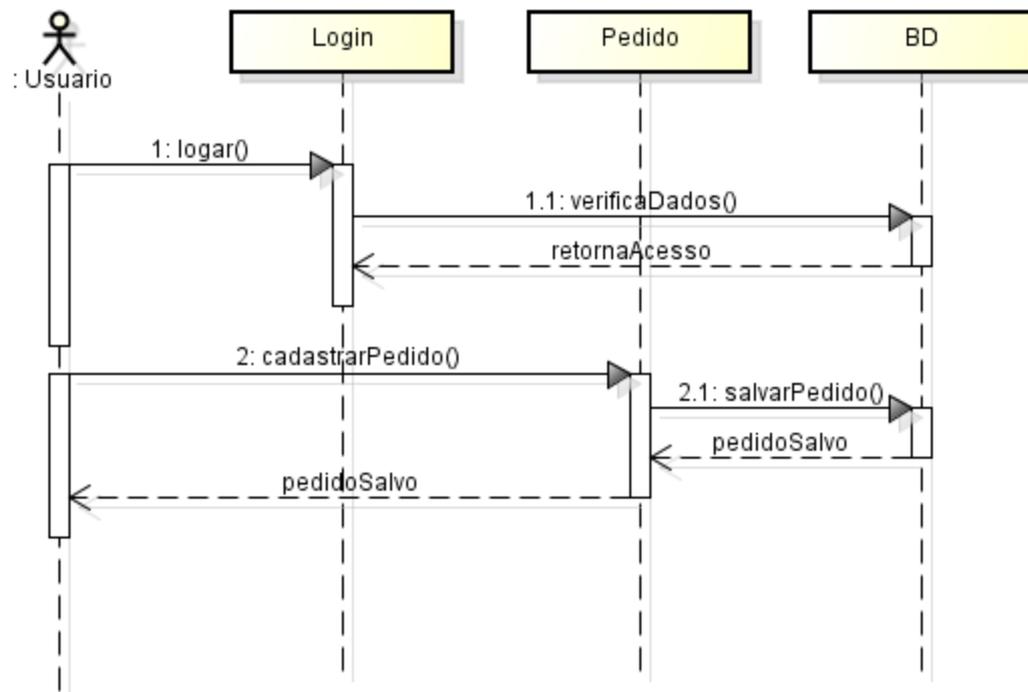
## Apêndice B: Diagrama de Caso de Uso



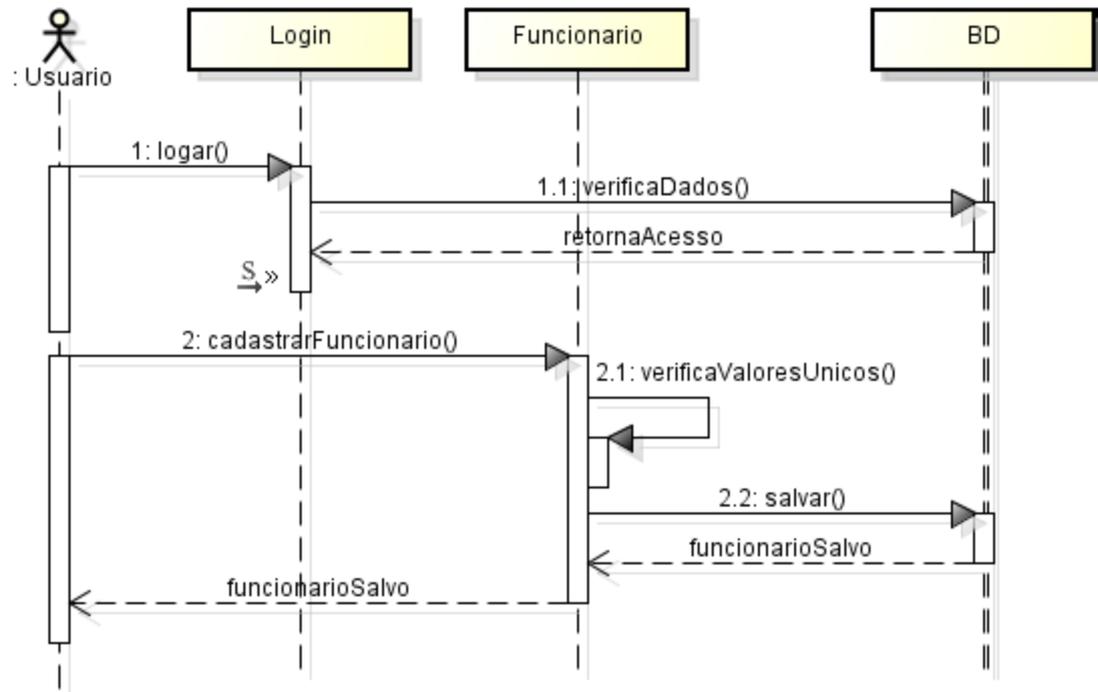
# Apêndice C: Diagrama de Classes



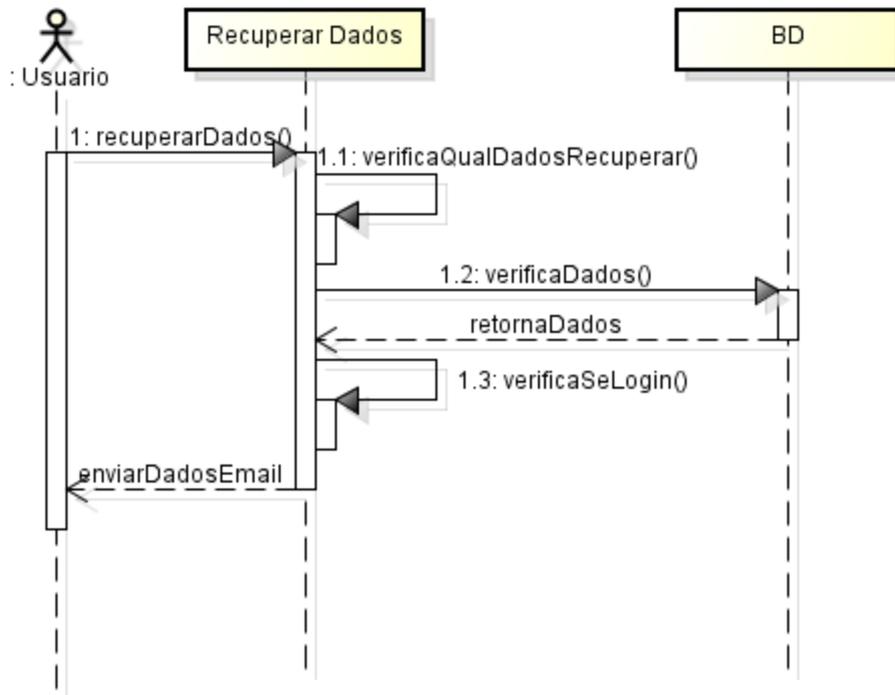
Apêndice D: Digrama de Sequência “Cadastrar Pedido”



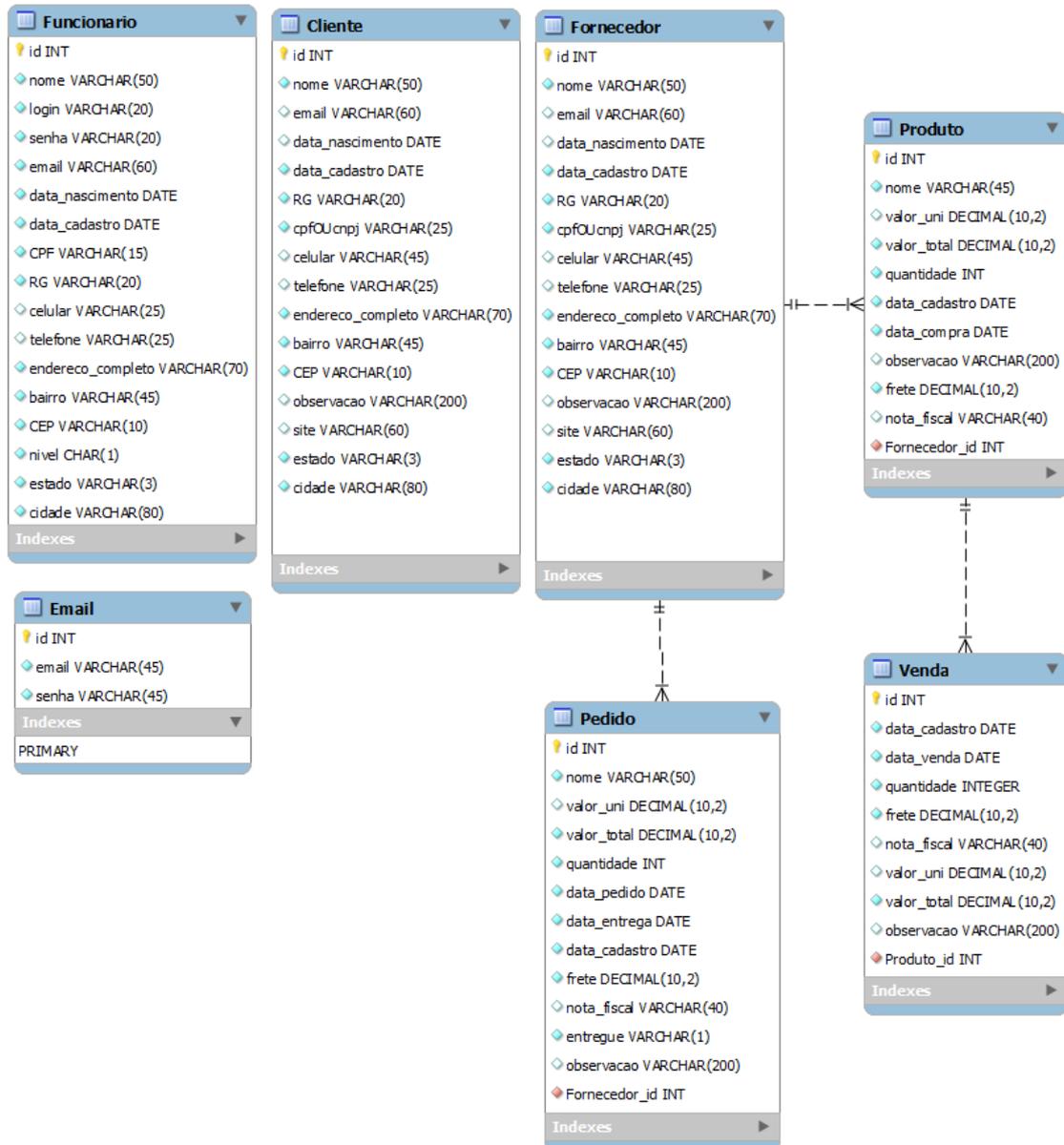
Apêndice E: Digrama de Sequência “Cadastrar Funcionário”



Apêndice F: Digrama de Sequência “Recuperar Dados”



## Apêndice G: Modelo lógico do sistema



Apêndice H: Ficha da avaliação do protótipo

**FICHA DA AVALIAÇÃO DO PROTÓTIPO**

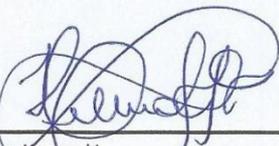
Data: 20/03/2014

<b>Critério</b>	<b>Atende</b>
<b>Aprendizado</b>	
É fácil aprender a utilizar o Sistema?	(X) Sim ( ) Parcialmente ( ) Não
É fácil a memorização de como executar as Tarefas?	(X) Sim ( ) Parcialmente ( ) Não
<b>Interface</b>	
A Interface é agradável?	(X) Sim ( ) Parcialmente ( ) Não
Os botões e ícones são de fácil visualização?	(X) Sim ( ) Parcialmente ( ) Não
Navegar pelo sistema é fácil?	(X) Sim ( ) Parcialmente ( ) Não
As possibilidades para seleção de dados são fáceis?	(X) Sim ( ) Parcialmente ( ) Não
As janelas e menus estão bem organizados?	(X) Sim ( ) Parcialmente ( ) Não
<b>Eficiência</b>	
O tempo de resposta é satisfatório?	(X) Sim ( ) Parcialmente ( ) Não
<b>Funcionalidades</b>	
As funcionalidades estão de acordo com a coleta de requisitos?	(X) Sim ( ) Parcialmente ( ) Não
Corresponde as expectativas esperadas?	(X) Sim ( ) Parcialmente ( ) Não

**Sugestão para melhorias futuras (exemplo: funcionalidades):**

A empresa possui um jornal de negócios impresso semanalmente, funcionalidades por internet: como cadastrar anúncios de compras e vendas e acesso aos dados online do cliente, ligar o sistema BORG com serviços web seria algo muito útil à empresa.

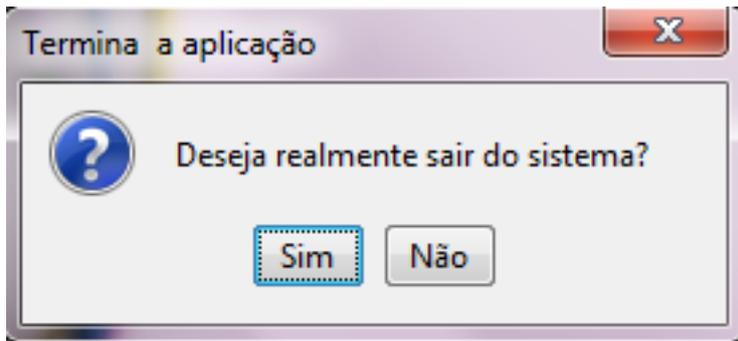
X

  
Helena Kogowski  
Gerente e Orçamentista

Apêndice I: Critérios de usabilidade

- **Visibilidade do status do sistema**

Exemplo 1: Feedback imediato ao tentar fechar o sistema:

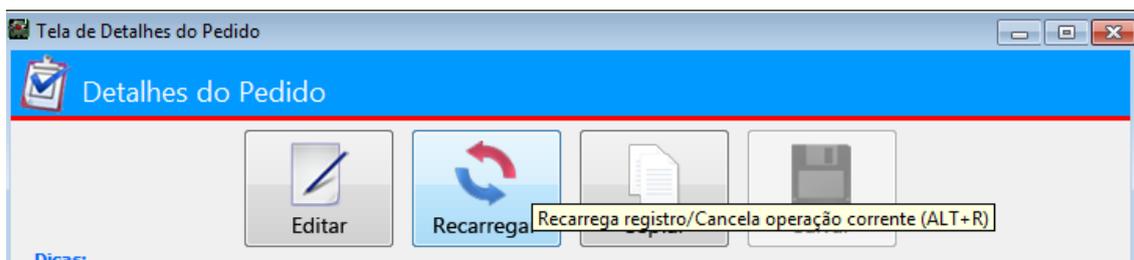


Exemplo 2: Ao abrir qualquer janela mostra que o carregamento está em curso:



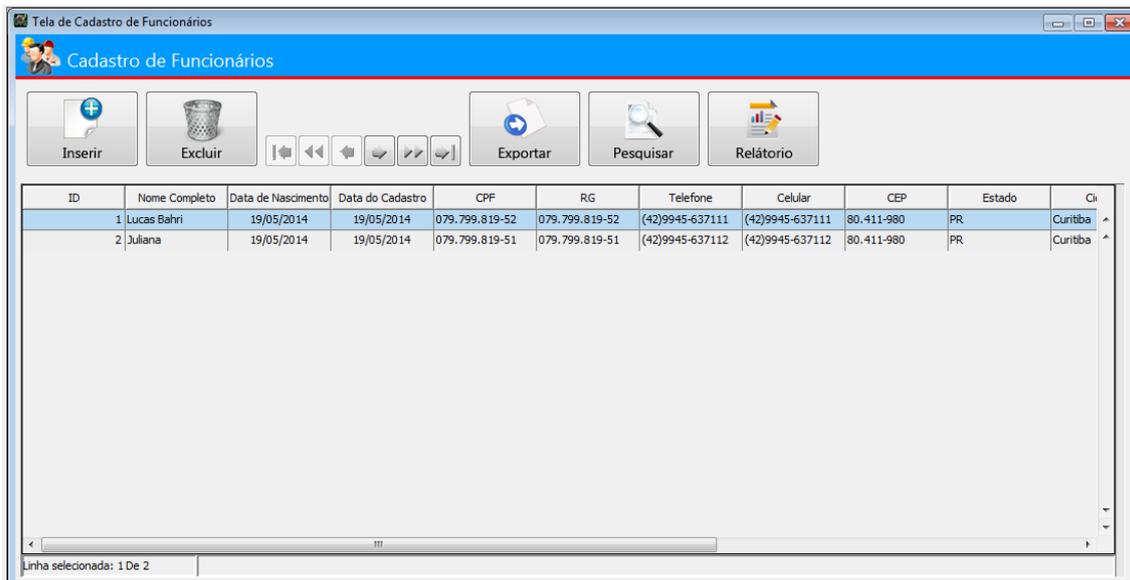
- **Controle do usuário e liberdade**

Exemplo: Com o botão recarregar toda ação feita é cancelada e os dados voltam a estar como eram:

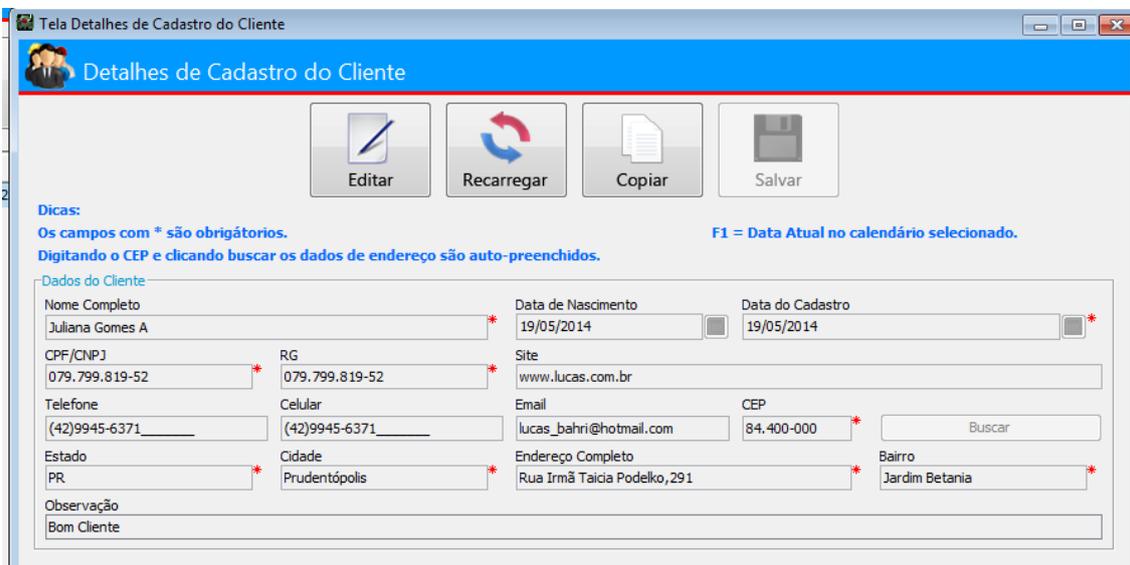


- **Consistência de padrões**

Exemplo 1: As telas do sistema seguem-se o mesmo padrão e formato:

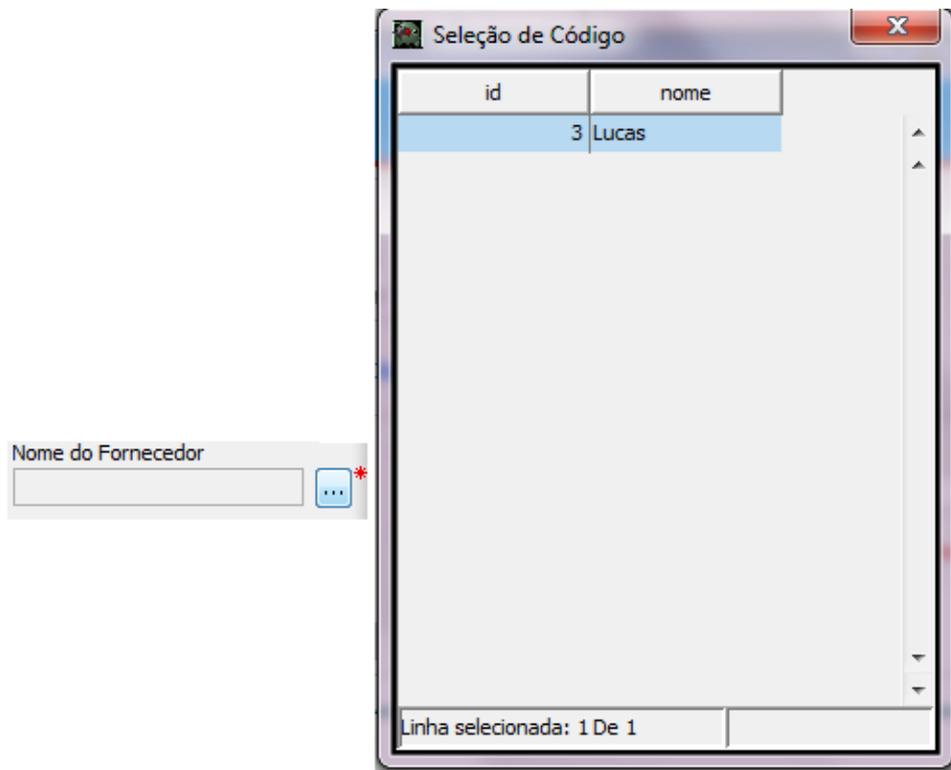


Exemplo 2: As telas do sistema seguem-se o mesmo padrão e formato:



- **Prevenção de erros**

Exemplo: Seleção de opção no lugar de edição:



- **Flexibilidade e Eficiência de uso**

Exemplo: Teclas de Atalho:

Amostra	F1
Cientes	F2
Fornecedores	F3
Funcionários	F4
Produtos	F5
Pedidos	F6
Vendas	F7
Email	F8

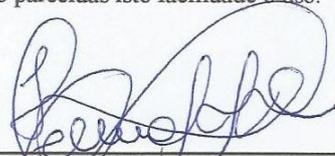
Apêndice J: Questionário de avaliação do sistema

**QUESTIONÁRIO DE AVALIAÇÃO DO SISTEMA**

**Data:** 28/05/2014

- 1) **O sistema poderá ser útil para a empresa?**  
 SIM ( ) NÃO  
Comentário: Possibilitou à empresa ter um maior controle de todos seus produtos, vendas, pedidos e cadastros auxiliando na administração dos negócios.
- 2) **O uso do Sistema BORG é?**  
 Fácil ( ) Razoável ( ) Difícil  
Comentário:
- 3) **É mais fácil realizar tarefas diárias através do BORG em vez do manual?**  
 SIM ( ) NÃO  
Comentário:
- 4) **Você precisou solicitar a ajuda do desenvolvedor do sistema para efetuar algumas tarefas?**  
 SIM  NÃO  
Comentário:
- 5) **Durante a execução do BORG ocorreu alguma falha?**  
 SIM  NÃO  
Comentário:
- 6) **A interface é agradável?**  
 SIM ( ) NÃO  
Comentário: A troca de fundo do sistema é uma função muito interessante.
- 7) **Classifique os itens abaixo de acordo com a tabela:**
  - a. **Péssimo**
  - b. **Ruim**
  - c. **Razoável**
  - d. **Bom**
  - e. **Ótimo**
  - **Disponibilidade das informações na tela (E)**  
Comentário:
  - **Navegação pelas telas do sistema (E)**  
Comentário: A função de minimizar, organizar e fechar ajuda muito.
  - **Resultado esperado com a relação à execução de um serviço desejado (E)**  
Comentário:
  - **Tempo de resposta do Sistema (E)**  
Comentário:
  - **Facilidade de uso do Sistema (E)**  
Comentário: As teclas de atalho exemplo: (F1, F2 e ALT +I) são muito úteis.
  - **Compreensão do que está na tela (E)**  
Comentário: As telas são muito parecidas isto facilita o uso.

X

  
\_\_\_\_\_  
Helena Kacowski  
Gerente e Orçamentista