

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Relatório de Estágio

Um Sistema de Gestão de Pedidos de uma Empresa de Bebidas

DIEGO TAVARES CAVALCANTI

Estagiário

ROHIT GHEYI

Orientador Acadêmico

STEPHANIE MUNIZ

Supervisora Técnica

Campina Grande-PB

2009

Julho de 2009

Um Sistema de Gestão de Pedidos de uma Empresa de Bebidas

Aprovado em _____

BANCA EXAMINADORA

Rohit Gheyi

ORIENTADOR ACADÊMICO

Tiago Massoni

MEMBRO DA BANCA

Joseana Fechine

MEMBRO DA BANCA



Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

Resumo

Este documento apresenta o relatório de Estágio Integrado realizado na Empresa Nacional de Bebidas pelo aluno Diego Cavalcanti. A referida empresa trabalha com a produção de bebidas e suas vendas são feitas por vendedores, que viajam durante toda a semana pelo interior da Paraíba e retornam à Campina Grande na sexta-feira, com os formulários de pedidos preenchidos.

Não havia uma forma de gerenciar os pedidos feitos diariamente, pois os vendedores não tinham como manter a empresa atualizada sobre os pedidos preenchidos durante o dia. Dessa forma, a empresa só tomava conhecimento do resultado do trabalho dos vendedores no fim da semana. Nesse dia, havia uma carga excessiva de trabalho para o faturista, pois ele precisava processar os pedidos que chegavam. Para solucionar esse problema, foi proposto para o estágio o desenvolvimento de um Sistema de Gestão de Pedidos. Esse sistema é acessível pela *web* e contém todos os formulários anteriormente preenchidos pelos vendedores manualmente. A partir disso, todos os dias, os vendedores acessam a Internet e repassam os dados coletados durante o dia.

Os requisitos foram levantados e o sistema foi implementado e validado junto à diretoria da empresa, que o aprovou e solicitou que fosse implantado. Antes de começar a ser usado, realizamos um treinamento com os usuários de forma a ensiná-los como utilizar a aplicação, como também recebemos as suas primeiras impressões de utilização do sistema. A partir daí, modificamos a interface do sistema para melhor adequar às necessidades do usuário. O estágio foi de uma importância ímpar para a aplicação dos conhecimentos adquiridos na universidade e agregação de outros referentes a experiências que só podem ser adquiridas num ambiente real de desenvolvimento.

Agradecimentos

A ajuda de algumas pessoas foi imprescindível para a realização deste estágio. Portanto, gostaria de agradecê-las sinceramente:

Agradecer primeiramente à Empresa Nacional de Bebidas, na pessoa de seus diretores, pela oportunidade; em especial, a Stephanie, pelo apoio e confiança demonstrados desde o início.

Agradecer também a Rohit por ser, além de um excelente orientador, um amigo. Por ser sempre dedicado e nos fazer buscar o nosso melhor. Seus conselhos sempre contribuíram para um aprendizado contínuo durante todo esse período.

A Joseana, pela preocupação e acompanhamento do trabalho como professora da disciplina, sempre contribuindo de alguma forma.

A Tiago, por ter aceito participar da banca e pelos comentários tão valiosos para o enriquecimento do trabalho.

A Lorena, por toda a amizade. O seu apoio e dedicação foram fundamentais para o desenvolvimento do trabalho. Trabalhar ao lado de amigos sempre torna tudo bem mais fácil.

Aos funcionários da ENB que nos acolheram com muita simpatia, em especial, a Maria do Carmo, Joselito, Jailton e Alisson.

E a Leonardo e os vendedores da ENB, pelo *feedback* do uso do sistema, fazendo com que melhorássemos a qualidade do nosso trabalho.

Sumário

Lista de Figuras	5
Lista de Siglas e Abreviaturas	6
1 Introdução	7
1.1 Contexto	7
1.2 Problema	8
1.3 Solução	8
1.4 Objetivos do Estágio	9
1.5 Estrutura do Documento	9
2 Ambiente de Estágio	11
3 Fundamentação Teórica	12
3.1 Processo de Desenvolvimento de Software	12
3.1.1 Etapa de Processo de Engenharia de Requisitos	13
3.1.2 Etapa de Planejamento	14
3.1.3 Etapa de Modelagem do Sistema	15
3.1.4 Etapas de Implementação e de Inspeção de Código	16
3.1.5 Etapa de Validação	16
3.1.6 Etapa de Implantação e Treinamento	16
3.1.7 Etapa de Evolução	17
3.2 Plataforma de Desenvolvimento: Grails	17
4 Ciclo de Desenvolvimento de Software	21

4.1	Levantamento de Requisitos	21
4.1.1	Estudo de Viabilidade	21
4.1.2	Elicitação	22
4.1.3	Especificação	23
4.1.4	Validação dos Requisitos	26
4.2	Planejamento de Atividades	27
4.2.1	Cronograma	27
4.2.2	Reuniões	28
4.2.3	Estimativas	28
4.3	Modelagem	29
4.3.1	Modelagem Estrutural: Diagrama de Classes	29
4.3.2	Modelagem Comportamental: Diagrama de Atividades	30
4.4	Implementação	31
4.4.1	Codificação	32
4.4.2	Inspeção de Código	39
4.5	Verificação e Validação	40
4.6	Implantação e Treinamento	41
4.7	Evolução	42
5	Considerações Finais	44
	Referências Bibliográficas	46
	Apêndices	48

Lista de Figuras

3.1	Processo de Engenharia de Requisitos	14
3.2	Arquitetura do Grails	18
3.3	Iteração sobre vetor implementada em Java e em Groovy	20
4.1	Requisitos Funcionais - Diagrama de Casos de Uso	24
4.2	Exemplo de Documentação de Requisito	26
4.3	Diagrama de Classes do Sistema	30
4.4	Diagrama de Atividades - Cadastro de Pedidos	32
4.5	Exemplo de Restrições de Atributos em Classes	33
4.6	Exemplo de Senha Alfabética Codificada	34
4.7	Tela de Login	36
4.8	Página Inicial de Administrador	37
4.9	Página Inicial de Faturista	38
4.10	Página Inicial de Vendedor	39
4.11	Formulário de Cadastro de Pedidos	40
4.12	Exemplo de Relatório de Vendas	43

Lista de Siglas e Abreviaturas

ENB	Empresa Nacional de Bebidas LTDA.
PCU	Pontos de Casos de Uso
SGBD	Sistema de Gerenciamento de Banco de Dados

Capítulo 1 - Introdução

Para ser bem sucedido no mercado, um profissional necessita, dentre outros aspectos, de conhecimentos que são adquiridos pela experiência real em empresas. Quando ainda está na universidade, o aluno raramente tem oportunidades assim, tendo que se adaptar apenas ao que é cobrado em sala de aula. Como forma de contornar esse problema, a universidade disponibiliza, por meio de estágios em empresas, uma oportunidade única de o aluno adquirir uma experiência real que agregue valor à sua formação.

Essa experiência é valiosa no que tange à absorção de conhecimentos e visa complementar o que é visto na academia. Até porque, mesmo que o professor tente simular ao máximo uma situação real, há características intrínsecas que só podem ser vivenciadas no ambiente de uma empresa.

Este documento descreve a experiência de um estágio realizado numa empresa de bebidas, com o objetivo de desenvolver um sistema para gestão de pedidos da mesma. Durante o estágio, todo o ciclo de desenvolvimento do sistema foi trabalhado, desde a fase de levantamento de requisitos até a implementação, validação, treinamento dos usuários e evolução do sistema. O plano de estágio está apresentado no Apêndice A.

A seguir, há detalhes da empresa na qual o estágio foi realizado (Seção 1.1) e da situação que ela enfrentava e que motivou o desenvolvimento do sistema (Seção 1.2). A Seção 1.3 apresenta a solução proposta e como a empresa se adequou ao sistema. Os objetivos geral e específicos traçados para o estágio podem ser vistos na Seção 1.4, enquanto que a Seção 1.5 descreve a estrutura deste documento.

1.1 Contexto

O estágio foi desenvolvido na Empresa Nacional de Bebidas Ltda., tendo sido realizado durante os meses de Março a Julho de 2009. A Empresa Nacional de Bebidas (ENB) foi fundada em 14 de julho de 1946 e atravessa os seus 63 anos de existência voltada para o engarrafamento de diversas bebidas, como a Cachaça Caranguejo Export, a Série *Premium* da Caranguejo, Ron D'Cuba, Vodka Paloff, entre outras.

As vendas da ENB são feitas por vendedores que representam a empresa tanto em Campina Grande, quanto em outras cidades. Cada vendedor é responsável por visitar um determinado número de clientes, em sua maioria, em cidades do interior da Paraíba. Em cada cliente, o vendedor preenche um formulário de pedidos e passa essa informação à fábrica. Um faturista analisa todos os pedidos entregues pelos vendedores, preenche um relatório com a compilação dos mesmos e repassa para o Departamento de Produção começar a produzir de acordo com os insumos em estoque.

1.2 Problema

Embora o processo de vendas da ENB seja aparentemente simples, a empresa enfrentava um problema em relação à gestão dos pedidos. Como os vendedores representam a empresa fora da cidade, eles passam toda a semana viajando e só retornam à Campina Grande no fim da semana (geralmente, na sexta-feira). Dessa forma, não era possível ter um controle diário dos pedidos, dado que os vendedores só poderiam entregar todos os formulários preenchidos ao retornarem.

Além disso, a empresa também não tinha controle sobre a carga horária de trabalho de cada vendedor. Por exemplo, poderia ocorrer casos em que um vendedor informasse à empresa que estava trabalhando desde a segunda-feira, mas só viajou no meio da semana. Dessa forma, estaria atrasando a visita aos clientes e não cumprindo com o cronograma estabelecido pela empresa. Outro problema enfrentado é que o faturista tinha um carga de trabalho excessiva na sexta-feira. Frequentemente, era necessário que ele fizesse horas-extras para que pudesse conferir todos os pedidos que chegavam naquele dia (quando os vendedores retornavam à cidade). Por fim, o setor de produção precisa planejar sua produção de modo a armazenar em estoque alguns lotes de produtos que serão entregues aos clientes. Com a gestão diária de pedidos, é possível que esse planejamento se adeque melhor à demanda do dia.

1.3 Solução

Como forma de resolver o problema supracitado, foi desenvolvido um **Sistema de Gestão de Pedidos**, acessível via *web*, para que fosse possível ser acessado a partir das diferentes

idades que os vendedores visitam. O sistema permite cadastrar informações dos pedidos que os vendedores recebem dos clientes. A empresa definiu que cada vendedor deve ir diariamente a uma *lan house* e repassar para o sistema todos os pedidos recolhidos durante o dia. Assim, a empresa passou a ter um controle diário sobre o trabalho dos vendedores. Para auxiliar na gestão dos pedidos, o sistema também gera relatórios com a compilação das informações cadastradas por período.

1.4 Objetivos do Estágio

Após conhecer o problema, foi possível traçar o objetivo geral do estágio: desenvolver um sistema que **melhore a gestão de pedidos** da Empresa Nacional de Bebidas. A seguir, estão elencados os objetivos específicos para alcançar o objetivo geral traçado, bem como para adquirir a experiência esperada a partir do estágio:

- Entender a rotina de trabalho da empresa e seus problemas recorrentes em relação à gestão de pedidos;
- Planejar um sistema adequado que ajude a ENB a ter um maior controle sobre os seus pedidos;
- Selecionar tecnologias e ferramentas adequadas para o desenvolvimento do sistema, indo ao encontro dos requisitos não-funcionais determinados pela empresa;
- Implementar o sistema;
- Treinar os usuários, lidando com uma possível resistência à aceitação do produto por parte deles;
- Acompanhar o uso do sistema, a fim de adequar melhor a interface com o usuário e resolver problemas que possam surgir.

1.5 Estrutura do Documento

Esse relatório de estágio está dividido em 5 capítulos que descrevem as atividades realizadas durante o estágio. No Capítulo 2, é descrito o ambiente no qual o estágio foi realizado e apresenta o orientador acadêmico e a supervisora técnica. No Capítulo 3, é apresentada

a fundamentação teórica do estágio, na qual é descrita o processo de desenvolvimento e a plataforma de desenvolvimento utilizada. O ciclo de desenvolvimento do software, com descrição das atividades desenvolvidas durante o estágio, está apresentado no Capítulo 4. Por fim, no Capítulo 5, são feitas as considerações finais sobre o trabalho desenvolvido. Em seguida, encontram-se os apêndices.

Capítulo 2 - Ambiente de Estágio

A Empresa Nacional de Bebidas (ENB), considerada empresa de médio porte, situada à Rua Benjamin Constant nº 170, Estação Velha, Campina Grande, Paraíba, é dividida nos seguintes setores: Diretoria – de onde parte a tomada de decisões, bem como ações de auditoria; Tecnologia da Informação – onde são controladas todas as informações e realizados a gerência da rede, o suporte aos usuários e desenvolvimento de soluções em software; Administração – da qual fazem parte os seguintes departamentos: contabilidade, financeiro, recursos humanos, controle; Jurídico – onde toda a parte legislativa é tratada; Comercial – do qual fazem parte os seguintes departamentos: vendas, atendimento ao cliente, faturamento, telemarketing, publicidade; Logística – constituído pelos departamentos de carregamento e de distribuição; Industrial – do qual fazem parte os departamentos de almoxarifado, produção, manutenção mecânica/elétrica/hidráulica, controle de estoques, laboratório químico e expedição; e, por fim, Segurança – setor que está em ampliação, por enquanto apenas relacionada com vigilância.

O estágio foi desenvolvido no Setor de Tecnologia da Informação da ENB. A empresa cedeu uma sala com dois computadores, cada um contendo processador Pentium Dual Core 2.5 GHz, 250 GB de espaço em disco e 2 GB de memória principal, além dos softwares necessários à realização do estágio. O trabalho foi desenvolvido em parceria com Lorena Lira, que também é aluna da disciplina Estágio Integrado, atuou como testadora do sistema e participou das discussões durante todo o processo de desenvolvimento.

De Março a Julho de 2009, foi trabalhado na empresa, em média, 20 horas semanais. Isso dá um somatório de 320 horas nos 4 meses completos trabalhados. Nesse período, foram realizadas desde reuniões com a diretoria e entre a equipe de desenvolvimento, até treinamento dos usuários, desenvolvimento do sistema, dentre outras atividades que serão descritas neste documento.

O nosso orientador acadêmico foi o professor Rohit Gheyi ¹, que também atuou como gerente da equipe. Na empresa, a supervisão técnica ficou a cargo de Stephanie Muniz ², diretora da empresa e gerente de TI.

¹rohit@dsc.ufcg.edu.br

²smuniz@enb.com.br

Capítulo 3 - Fundamentação Teórica

Neste capítulo, descreveremos a plataforma de desenvolvimento *web* utilizada e daremos detalhes sobre o processo de desenvolvimento de software aplicado para desenvolver o Sistema de Gestão de Pedidos. Na Seção 3.1, o processo de desenvolvimento de software utilizado será apresentado, enquanto que na Seção 3.2, mostramos a tecnologia de desenvolvimento *web* utilizada, o Grails.

3.1 Processo de Desenvolvimento de Software

Segundo Humphrey [Humphrey 1989], “um processo de desenvolvimento de software é um conjunto de ferramentas, métodos e práticas usados para construir um produto de software”. Assim, para definir qual o processo a ser escolhido para o desenvolvimento de um software deve-se levar em consideração a harmonia das ferramentas necessárias, métodos aplicáveis e práticas adequadas com as especificidades do projeto em questão. Dessa forma, é possível estabelecer um conjunto de atividades de processo de desenvolvimento de software a ser utilizado no desenvolvimento de um projeto em específico. Processos para projetos específicos podem, então, ser definidos a partir da instanciação do processo de software padrão, levando em consideração suas características particulares.

No intuito de atender os objetivos da disciplina de Estágio Integrado, o easY-Process foi o processo de desenvolvimento de software utilizado para inspirar a definição de um processo de desenvolvimento para o nosso estágio. O easYProcess é um processo de desenvolvimento de software simplificado, iterativo e incremental que se apóia em práticas do eXtreme Programming [Beck e Andres 2004], Rational Unified Process [Kruchten 2000] e Agile Modeling [Ambler 2002]. Tem como objetivo auxiliar tanto a gerência do desenvolvimento de aplicações em disciplinas de engenharia de software quanto a aprendizagem dos conceitos desta disciplina, podendo também ser utilizado em projetos de pequeno e médio porte em empresas.

O sistema desenvolvido possui um escopo bem definido, considerado como um projeto de médio porte. Foi desenvolvido por uma equipe pequena (3 integrantes - 2 de-

envolvedores e 1 gerente) e que atuou em todas as fases do desenvolvimento do sistema. Visando obter um software de qualidade, bem como cumprir com os prazos estabelecidos pela empresa e pela disciplina, além de atender à realidade da equipe, definimos as atividades macros que compuseram o processo seguido durante todo o estágio. Nas próximas seções, descrevemos as atividades escolhidas para o nosso processo.

3.1.1 Etapa de Processo de Engenharia de Requisitos

Essa etapa, resumidamente, objetiva criar e manter um documento de requisitos. O processo inclui quatro subprocessos: estudo de viabilidade do sistema, elicitação, análise, especificação e validação dos requisitos. Geralmente, estas atividades ocorrem simultânea e incrementalmente, em um processo evolutivo que está presente em todo o processo de desenvolvimento de software [Sommerville 2007].

O **Estudo de Viabilidade** corresponde ao primeiro contato da equipe de desenvolvimento com o contexto do problema. É desejável responder à questões tais como, se o sistema contribui para os objetivos da empresa; se o sistema pode ser desenvolvido com tecnologia atual e dentro das restrições definidas de custo e prazo; e se o sistema pode ser integrado a outros sistemas já implantados. A realização desse estudo envolve a avaliação de informações, e, pode ser elaborado um relatório a ser apresentado à empresa requisitante do sistema, podendo a equipe propor mudanças no escopo, orçamento e prazos.

Na **Elicitação e Análise de Requisitos**, é feito um contato direto com os clientes, usuários finais do sistema e todo pessoal na organização que possa ser afetado pela instalação do sistema para obtenção dos requisitos. A elicitação e análise visam compreender o domínio da aplicação, onde o sistema será aplicado; entender o negócio, como os sistemas interagem e contribuem de forma geral com os objetivos de negócio; e apreender as necessidades e limitações dos *stakeholders* do sistema.

Dentre um conjunto de técnicas existentes para descobrir (elicitar) os requisitos do sistema a ser desenvolvido, incluindo as informações do universo de informação que restringem este sistema, algumas das técnicas mais conhecidas estão relacionadas a entrevistas, questionários, participação ativa dos usuários, reuso de requisitos, etnografia [Sommerville 2007].

A **Especificação dos Requisitos** corresponde a conversão desses requisitos em alguma forma-padrão, denominada documento de requisitos do sistema. Documento destinado a desenvolvedores, clientes e usuários do sistema. Ele deve ser usado como importante meio de comunicação entre cliente e empresa de desenvolvimento de software contratada para possível conferência de funcionalidades implementadas.

Validação dos Requisitos objetiva a descoberta de problemas levantados na fase de elicitación e análise dos requisitos. Essa validação pode ser feita através de revisões dos requisitos ou por prototipação do sistema. É uma importante etapa para minimizar, sabendo que raramente não se encontram todos os problemas, os custos com mudanças de requisitos em momentos posteriores do desenvolvimento, por questões de mal-entendidos ou omissões.

Na Figura 3.1, é resumido o Processo de Engenharia de Requisitos, citando cada subprocesso que o compõe e o artefato gerado [Sommerville 2007]. Além de representar, através das setas, o fluxo de informações e a característica incremental do processo.

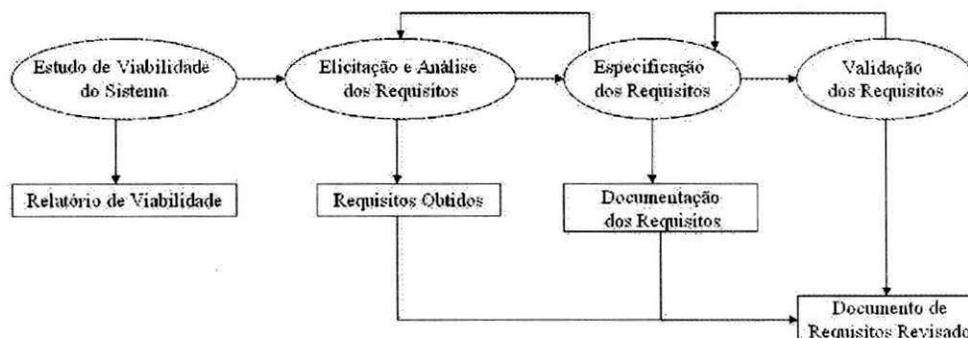


Figura 3.1: Processo de Engenharia de Requisitos

3.1.2 Etapa de Planejamento

Essa etapa é útil na organização e no planejamento a curto e em longo prazo, e no cumprimento das atividades do projeto. A partir dos requisitos definidos, é possível a realização do planejamento de todo o fluxo de desenvolvimento do sistema, de forma que essa etapa está presente em todo o processo de desenvolvimento do software.

Visando cumprir o planejamento e os prazos acordados com a empresa, é realizada a **Estimativa de Tempo** [Sommerville 2007]. Durante essa etapa, é feita uma

estimativa de quanto tempo demandará para execução das atividades e, após a realização de cada atividade, a catalogação do tempo real consumido para conclusão da mesma. Com a estimativa de tempo, é possível identificar o que será mais custoso.

A fim de desenvolver uma aproximação ou estimativa dos custos dos recursos que são necessários à implementação das atividades do projeto, foi definido dentro do planejamento do projeto a atividade de **Estimativa de Custo**. Essa estimativa pode ser feita utilizando Pontos de Caso de Uso [Karner 1993], que prediz o tamanho de um sistema de acordo com os modelos de casos de uso gerados [OMG 2009].

Outra atividade fundamental para o planejamento são as reuniões da equipe. Os objetivos podem ser os mais diversos, tendo como principal motivo o acompanhamento do desenvolvimento do sistema. A participação de todos é imprescindível e a ocorrência semanal visa minimizar impactos negativos de mudanças no planejamento.

O planejamento das atividades é documentado através de um **Cronograma de Atividades**, identificando o plano de entrega de *release*, uma tabela de estimativa de tempo e de tempo real consumido para realização das atividades e um plano de custo do projeto. A documentação do planejamento deve estar acessível a todos da equipe, pois auxilia o gerente na avaliação do andamento do projeto, os desenvolvedores a se guiarem sobre o desenvolvimento das suas atividades, além de facilitar a atualização dessa documentação por qualquer componente da equipe.

3.1.3 Etapa de Modelagem do Sistema

Os requisitos do sistema podem ser expressos de maneira mais detalhada para os desenvolvedores do sistema. Uma abordagem amplamente utilizada é documentar a especificação do sistema como um conjunto de modelos que descrevem o sistema a ser desenvolvido. É possível representar o sistema sobre várias perspectivas: externa, na qual o contexto ou o ambiente do sistema é modelado; comportamental, na qual o comportamento do sistema é modelado; e estrutural, na qual a arquitetura do sistema ou a estrutura dos dados processados pelo sistema é modelada.

Para o processo utilizado, foi definida a representação do sistema sobre as perspectivas comportamental e estrutural. Especificamente, devemos construir um diagrama estrutural, relativo aos dados processados e um diagrama comportamental, correspon-

dente às partes mais complexas do sistema.

3.1.4 Etapas de Implementação e de Inspeção de Código

A etapa de **Implementação** do software corresponde a codificação propriamente dita do sistema. É necessário que o código esteja disponível e atualizado, em um ambiente de armazenamento de código. Práticas como padrão de codificação, *Design* simples e padrões de projeto precisam ser seguidas. Mesmo com uma equipe composta por apenas um desenvolvedor principal e um secundário, a fase de manutenção e evolução do sistema pode vir a ser feito por outra equipe. Ou ainda, é difícil a própria equipe de desenvolvimento manter algo sem padrão algum de implementação.

Com o objetivo de obter uma nova visão sobre possíveis melhorias no código, foi definida uma etapa de **Inspeção de Código** após cada iteração do desenvolvimento. O desenvolvedor secundário do sistema atua nessa etapa visando encontrar os defeitos, conferir as conformidades aos padrões de implementação e identificar programação de baixa qualidade, além disso, deve documentar para que o desenvolvedor principal, autor do código, possa rever o que foi feito.

3.1.5 Etapa de Validação

Para validar o sistema é definida a utilização de testes funcionais. O teste funcional é aquele onde se coloca o sistema em execução e alguém o utiliza antes da entrega ao cliente. O objetivo dessa categoria de teste é encontrar erros de programação, inconsistências nas mensagens de erros e na interface. No contexto dessa etapa, o desenvolvedor secundário da equipe é o responsável por executar essa atividade e documentar os resultados.

3.1.6 Etapa de Implantação e Treinamento

A etapa de implantação do sistema corresponde a fase em que o sistema deve ser colocado em produção na empresa. Para tanto, é necessário que seja realizado um treinamento com os usuários do sistema. Esse treinamento objetiva ensinar os usuários a utilizarem o sistema, bem como a equipe adquirir *feedback* dos usuários com relação aos aspectos de usabilidade do produto. Em seguida, esse retorno sobre o sistema pode ser considerado

como mais um momento em que se realiza a Etapa de Validação descrita acima.

3.1.7 Etapa de Evolução

Quando um sistema é colocado em produção, inevitavelmente ele pode ser modificado para continuar útil, pois novos requisitos surgem ou os existentes se modificam. O desenvolvimento do software não pára, portanto, quando o sistema é implantado. A etapa de evolução de software no processo utilizado, visa dá suporte a fase inicial de implantação do sistema, dado que dentro do cronograma de Estágio Integrado é possível ser atingida essa etapa do processo de desenvolvimento de software. Alguma modificação, identificação de defeitos ou novos requisitos podem ser elucidados pelos usuários e então, de acordo com a análise desse levantamento pela equipe de desenvolvimento, haver modificações no software implantado.

3.2 Plataforma de Desenvolvimento: Grails

Um dos requisitos do sistema desenvolvido era que ele fosse acessível via internet. Para isso, utilizamos o Grails [Rocher e Brown 2009] para desenvolvê-lo. O Grails é uma plataforma de desenvolvimento para aplicações *web* que utiliza a linguagem de programação Groovy [Koenig et al. 2007]. Grails permite manipular de uma forma simples desde a camada de interface gráfica com o usuário até preocupações referentes à persistência dos dados. Dessa forma, é possível ter uma alta-produtividade ao desenvolver aplicações, pois o desenvolvedor não se preocupa com configurações de diversos arquivos nem com minúcias de consultas a banco de dados, por exemplo.

Como forma de simplificar o processo de desenvolvimento, sem perder em flexibilidade, o Grails segue o modelo de programação por convenção (do inglês *Convention over configuration*). Este modelo visa diminuir o número de decisões que os desenvolvedores precisam tomar, pois, para toda configuração que é feita repetidas vezes nos sistemas *web* desenvolvidos, foram criadas convenções que precisam ser seguidas pelos desenvolvedores. Por exemplo, o nome das tabelas do banco de dados é sempre o nome das classes do Grails que descrevem as entidades do sistema. Se existe uma classe “Vendedor” no modelo, a tabela correspondente no banco de dados será chamada, por padrão, “vendedor”. Dessa forma, o programador não precisa se preocupar com a criação das tabelas e

relacionamentos no banco de dados ou configurar arquivos comuns a todas as aplicações (por exemplo, configurações de arquivos XML para comunicação das páginas com controladoras). Somente se o comportamento desejado for distinto da convenção implementada é que se torna necessário elaborar configurações.

Apesar de ser mais simples de desenvolver com Grails, o sistema criado ainda utiliza todas as construções de Java e também de alguns *frameworks* conhecidos, tais como: Spring, Hibernate, Sitemesh, Quartz, dentre outras. O Grails utiliza essas tecnologias de forma transparente para o programador. Na Figura 3.2, é apresentada uma arquitetura do Grails destacando algumas tecnologias utilizadas. A figura representa que o Grails está implementado utilizando algumas outras tecnologias, tais como: Java Enterprise Edition, Spring, Hibernate e Quartz. Além disso, o *framework* é implementado utilizando o Groovy como linguagem de programação. Tudo isso é executado sobre a plataforma Java.

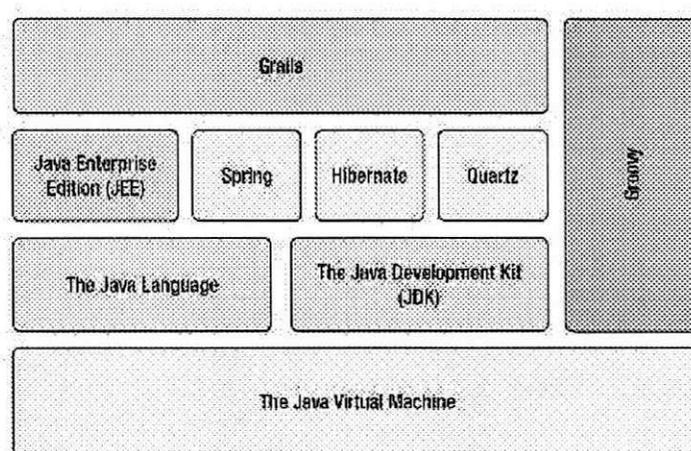


Figura 3.2: Arquitetura do Grails

Groovy

Groovy [Koenig et al. 2007] é uma linguagem dinâmica, que roda na plataforma Java, mas com construções adicionais similares às de outras linguagens, como Python, Ruby e Smalltalk. A linguagem é interpretada e são gerados *bytecodes* compatíveis com a Máquina Virtual Java (JVM). Por esse motivo, é possível uma boa integração com Java e os desenvolvedores dessa linguagem podem utilizar o Groovy com uma curva de aprendizado pequena.

Apesar de um programador Groovy também ter a possibilidade de utilizar a

mesma sintaxe Java, em Groovy, existem construções de mais alto nível que ajudam a deixar o código mais claro e menor. Na Figura 3.3, é apresentado um exemplo de código feito em Java (Listagem 3.1) e em Groovy (Listagem 3.2). Ambos os códigos contêm um laço que itera sobre um vetor e imprime o resultado do método `toString` de seus objetos. Perceba que a estrutura do laço pode ser escrita de forma mais compacta em Groovy (linha 4 da Listagem 3.2). Além disso, veja que não é seguro chamar `vetor_obj[i].toString()` sem nenhuma verificação inicial, pois, se o objeto for nulo, é lançada uma exceção acusando um erro de `NullPointerException`. Portanto, em Java, precisamos utilizar um comando de decisão (linhas 5-9 da Listagem 3.1) para verificar se o objeto é nulo. Caso seja, imprimimos apenas “null” (linha 8 da Listagem 3.1). Em Groovy, a checagem para verificar se o objeto é nulo também é feita, mas de uma forma mais sucinta, colocando apenas uma interrogação após a referência ao objeto (linha 5 da Listagem 3.2). Caso `vetor_obj[i]` seja nulo, o método `toString` não é chamado e é retornado “null” como resultado.

Utilização de *Plugins* em Grails

Grails disponibiliza um conjunto de *plugins*¹ que são úteis no desenvolvimento de sistemas *web* para adição rápida de novas funcionalidades. Por exemplo, podemos instalar um *plugin* numa aplicação para envio de e-mails pelo sistema. Para tanto, o programador apenas necessita instalar o *plugin* Mail Plugin e configurar informações como endereço e porta SMTP do servidor de e-mails e login e senha, além do destinatário, assunto e corpo do e-mail. Fora isso, não é preciso nenhuma outra configuração. Tudo é integrado ao sistema que está sendo desenvolvido de forma transparente, graças à arquitetura flexível que o Grails possui.

Na nossa aplicação, utilizamos os *plugins* Dojo, RichUI e Filterpane para adicionar ao sistema bibliotecas que possibilitam uma melhor interface com o usuário (por exemplo, o recurso de auto-completar em campos dos formulários, buscando os dados no BD). Também utilizamos o Mail Plugin para envio de e-mails diários para determinados usuários do sistema e o Functional Test Plugin para realizar testes funcionais. Além disso, foram utilizados os seguintes *plugins*: JSecurity Plugin, Quartz Plugin e Jasper Plugin.

¹Os *plugins* do Grails estão disponíveis em: <http://grails.org/plugin/home>

Figura 3.3: Iteração sobre vetor implementada em Java e em Groovy

Listagem 3.1: Exemplo de Código em Java

```
1 public class Exemplo1 {
2   Object[] vetor_obj;
3   ...
4   for (int i=1; i <= vetor_obj.length; i++) {
5     if(vetor_obj[i] != null) {
6       System.out.println(vetor_obj[i].toString());
7     } else {
8       System.out.println(null);
9     }
10  }
11 }
```

Listagem 3.2: Exemplo de Código em Groovy

```
1 class Exemplo1 {
2   def vetor_obj = []
3   ...
4   for (i in 1..vetor_obj.length) {
5     println vetor_obj[i]?.toString()
6   }
7 }
```

Capítulo 4 - Ciclo de Desenvolvimento de Software

Neste capítulo, descrevemos as atividades realizadas durante o ciclo de desenvolvimento do Sistema de Gestão de Pedidos. Mostraremos o que fizemos em cada fase do ciclo e as ferramentas utilizadas para produzir os diversos artefatos. As atividades desenvolvidas estão de acordo com o processo de desenvolvimento de software utilizado no estágio, conforme descrito na Seção 3.1.

A seguir, descrevemos na Seção 4.1 a etapa de engenharia de requisitos realizada para o desenvolvimento do Sistema de Gestão de Pedidos. Nessa etapa do processo, são realizados o estudo da viabilidade do sistema, além da elicitação, análise, especificação e validação dos requisitos. Após essa etapa, informações sobre o cronograma de atividades podem ser encontradas na Seção 4.2, bem como informações sobre as reuniões feitas e as estimativas realizadas; ao passo que na Seção 4.3, é descrita a etapa de modelagem do sistema. Detalhes sobre a implementação e inspeção do código estão descritos na Seção 4.4. Por fim, a validação do sistema está detalhada mais adiante, na Seção 4.5, enquanto que o treinamento de usuários e evolução do software estão descritos nas Seções 4.6 e 4.7, respectivamente.

4.1 Levantamento de Requisitos

O processo de engenharia de requisitos envolve 4 etapas, como mostrado na Figura 3.1. Nessa fase do processo de desenvolvimento, inicialmente, estudamos a viabilidade do sistema (Seção 4.1.1). Depois, elicítamos os requisitos (Seção 4.1.2) e os documentamos (Seção 4.1.3). Por fim, os requisitos são validados junto ao cliente (Seção 4.1.4).

4.1.1 Estudo de Viabilidade

O estudo da viabilidade do sistema se deu por meio de reuniões com alguns membros da diretoria da empresa. Nessas reuniões iniciais, conhecemos o problema que a empresa

enfrentava quanto à gestão de pedidos e, a partir disso, pudemos criar hipóteses sobre o impacto que o sistema teria na empresa.

Após reconhecida a necessidade do sistema, analisamos o seu escopo e qual seria o tempo estimado para o seu desenvolvimento e implantação. Além de analisarmos se o tempo estimado se adequava às necessidades da empresa, tivemos que verificar também se ele seria adequado ao tempo necessário para a disciplina de Estágio Integrado.

Como mencionado anteriormente, os pedidos eram cadastrados todos na sexta-feira, dia em que os vendedores voltam de suas rotas no interior da Paraíba. Esse cenário dificultava o gerenciamento tanto do almoxarifado quanto da produção. Por este motivo, a empresa definiu um requisito não-funcional para que o sistema fosse *web* (acessível de qualquer lugar). Com isso, os vendedores podem repassar diariamente os dados de seus pedidos a partir de qualquer cidade que tenha acesso à Internet.

Além disso, a empresa definiu que o sistema deveria ser implementado em uma tecnologia que fosse bem conhecida na comunidade de desenvolvedores. Com isso, eles esperam que o sistema seja de fácil manutenção para qualquer desenvolvedor com conhecimento em desenvolvimento *web*. Assim, decidimos utilizar o Grails, por ele ter uma ampla comunidade. Por este proporcionar um rápido desenvolvimento, estimamos que o tempo necessário para o desenvolvimento seria adequado ao estágio não só para implementação, como também para todas as atividades do processo, tais como: treinamento dos usuários e acompanhamento do uso do sistema durante algumas semanas.

4.1.2 Elicitação

Após concluído o estudo da viabilidade do sistema, utilizamos algumas técnicas de elicitação de requisitos: entrevistas, etnografia e questionários. Inicialmente, a fim de compreendermos o domínio da aplicação e o problema enfrentado, foram realizadas reuniões com o faturista. Por ser o funcionário mais próximo dos vendedores dentro da empresa, ele é a pessoa mais adequada para nos passar informações de como eles trabalham. Esta entrevista permitiu ver o modelo de formulário utilizado pelos vendedores para preencher os pedidos dos clientes. Como o sistema deveria conter apenas os dados que eram preenchidos nesses formulários, os requisitos do sistema tornaram-se mais claros, a partir de então. Entretanto, a empresa não queria só um sistema de gerenciamento da submissão dos pedidos. A mesma também queria, a partir dos dados cadastrados, extrair algumas

informações que fossem úteis para a gestão da empresa. Neste sentido, realizamos outra entrevista com uma das diretoras da empresa, que nos forneceu uma visão de que tipo de relatórios seriam interessantes implementarmos no sistema. Para entender melhor o dia-a-dia da empresa, usamos a etnografia [Sommerville 2007] para entender melhor como todo o processo de gestão de pedidos era feito. A etnografia é o ato de observar o dia-a-dia do trabalho da empresa para coletar dados que auxiliarão no levantamento de requisitos.

Por fim, diante do que foi observado no estudo de viabilidade do sistema, nos preocupamos em estudar o requisito não-funcional de usabilidade mais a fundo. Como os vendedores geralmente trabalham viajando e não usam computadores da empresa em sua rotina semanal, ficamos preocupados se a utilização de computadores na nova rotina que o sistema iria impor na empresa poderia ser um empecilho para a aceitação do mesmo. Para averiguarmos isso, elaboramos um questionário com o objetivo de entendermos melhor o perfil dos usuários que iriam utilizar o sistema no tangente à familiaridade com a Internet e com sistemas computacionais. Cada usuário informou sua frequência de uso da Internet e principais atividades feitas por ele no computador. Assim, pudemos entender melhor o requisito não-funcional da usabilidade do sistema, como também já começar a planejar o treinamento. O modelo do questionário aplicado pode ser visto no Apêndice B. Os dados dos questionários respondidos mostraram que 75% dos usuários acessam a Internet diariamente para ler e-mails e acessar páginas na *web*. Dessa forma, como 25% dos usuários são menos experientes no uso do computador, a interface foi planejada para que eles pudessem utilizar o sistema sem complicações.

4.1.3 Especificação

Ao final do processo de elicitação, documentamos os requisitos levantados e modelamos os requisitos funcionais em um diagrama de casos de uso. A seguir, detalhamos os requisitos elicitados.

Requisitos Funcionais

Na fase de elicitação de requisitos, levantamos 27 requisitos funcionais. Eles foram modelados em um diagrama de casos de uso (Figura 4.1). Esse diagrama contém todos os requisitos funcionais divididos por níveis de um usuário (requisito não-funcional).

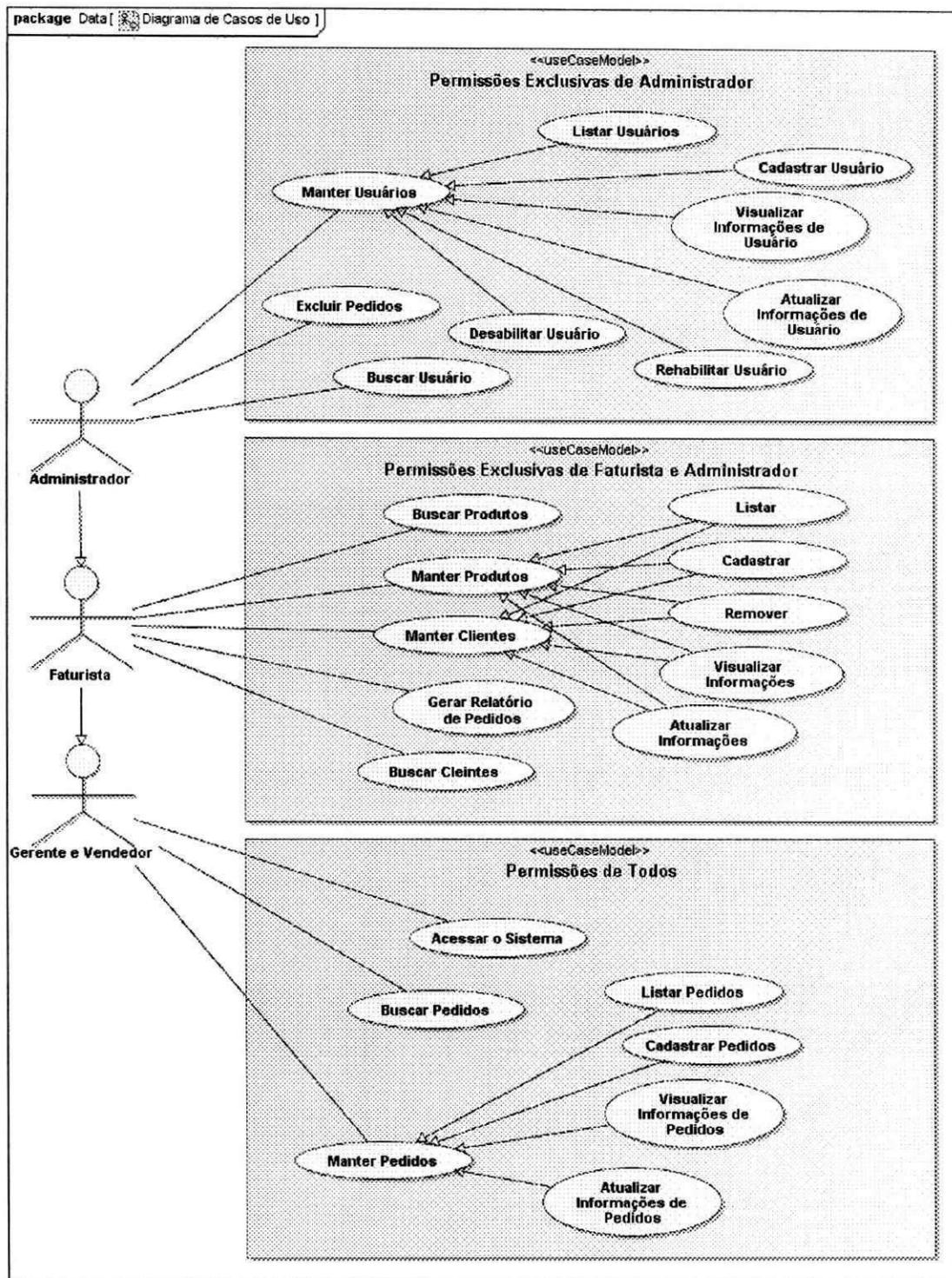


Figura 4.1: Requisitos Funcionais - Diagrama de Casos de Uso

Para a manipulação dos registros de cada entidade, o sistema oferece a opção de criação, atualização, remoção e visualização dos dados de cada módulo (Usuários, Clientes, Produtos e Pedidos). Além disso, permite a listagem de todos os registros cadastrados.

Também é possível gerar relatórios refinados por vendedor e período. Assim,

a empresa pode obter estatísticas de vendas que auxiliem na produção. A fim de manter a diretoria sempre informada sobre a situação de vendas da empresa, um relatório com um resumo diário das vendas é enviado diariamente, de forma automática, por e-mail.

Requisitos Não-Funcionais

Além de levantar os requisitos funcionais, fizemos o levantamento dos requisitos não-funcionais. Inicialmente, eles se resumiam a quatro requisitos. O primeiro referia-se a divisão dos usuários por diferentes níveis de acesso com permissões distintas, de acordo com sua função na empresa. Além disso, o sistema deveria ser acessível via *web*, como também para o seu acesso deveria ser feita autenticação por meio de senha. Por fim, as tecnologias e ferramentas utilizadas deveriam ser gratuitas.

Apesar dos 4 requisitos que já haviam sido levantados, na validação dos requisitos, houve uma preocupação em relação à proteção dos dados, já que o sistema seria acessado de computadores públicos (*lan houses*). Por exemplo, se cada vendedor tivesse apenas uma senha a ser digitada no momento da autenticação, um programa malicioso instalado no computador poderia capturar o que foi digitado e a senha daquele vendedor seria descoberta. Dessa forma, o sistema poderia ser acessado indevidamente, expondo dados sigilosos da empresa. Sendo assim, foram acrescentados mais alguns requisitos não-funcionais referentes a medidas de segurança, com o intuito de proteger os dados do sistema (mais detalhes sobre isso podem ser encontrados na Seção 4.4.1).

Documentação

A fim de documentar os requisitos elicitados, elaboramos um Documento de Requisitos do Usuário com todos os requisitos funcionais e não-funcionais levantados. Os requisitos foram cadastrados seguindo um modelo de documento.

A fim de ilustrar como os mesmos foram documentados, a seguir descrevemos o requisito funcional mais importante do sistema: cadastrar pedidos. É a partir dessa funcionalidade que os vendedores irão diariamente cadastrar cada pedido feito pelos clientes. Para isso, é preciso cadastrar, para cada pedido, a razão social do cliente, a condição de pagamento de como o cliente deseja pagar o pedido, a tabela de preços utilizada, a operação do pedido (se foi venda, troca ou bonificação), a quantidade de cada produto

solicitado e o seu valor unitário.

4.20 Cadastrar pedido

Identificação:	Anexo:	Nome:
20	-	Cadastrar pedido
Descrição:		
<p>Um usuário, seja qual for o seu nível de acesso, poderá cadastrar um novo pedido no sistema. Este pedido terá a identificação do usuário que o criou.</p> <p>Para adicionar um pedido, é necessário cadastrar as seguintes informações:</p> <ul style="list-style-type: none"> • Razão social do cliente; • Condição de pagamento; • Tabela de preços utilizada; • Operação do pedido: venda, bonificação ou troca; • Quantidade de cada produto solicitado; • Valor unitário de cada produto, de acordo com a tabela de preços utilizada. <p>O pedido também contém a informação de qual foi o usuário que o registrou e a data em que esse registro foi feito.</p>		

Figura 4.2: Exemplo de Documentação de Requisito

A versão final do Documento de Requisitos do Sistema pode ser encontrada no Apêndice C e contém detalhes dos 27 requisitos funcionais e 10 não-funcionais implementados no sistema. Este documento foi modificado a cada validação feita (Seção 4.1.4), para se adequar às novas solicitações da empresa.

4.1.4 Validação dos Requisitos

O processo de desenvolvimento utilizado é iterativo e incremental. Sendo assim, após o final de cada iteração, validamos os requisitos implementados com a diretoria da empresa. O fato de um dos membros da diretoria estar sempre presente permitiu que não perdêssemos tempo com trabalho duplicado oriundo da má interpretação dos requisitos.

A primeira versão implementada possuía os principais requisitos do sistema (formulários de cadastro de clientes e pedidos, além de autenticação de usuários). No geral, a diretoria ficou satisfeita com o resultado, o que fez aumentar o seu interesse e apoio ao desenvolvimento do sistema. Entretanto, nos foi solicitado que apresentássemos propostas de medidas de segurança a serem implementadas para melhorar a proteção do software. Para isso, buscamos ideias em sistemas de bancos e de comércio eletrônico. Após termos reunido um número razoável de propostas, nos reunimos novamente com a

diretoria para apresentá-las. Para cada solução proposta, também mostramos o problema que ela iria atacar. Todas as medidas de segurança foram aceitas e implementadas no sistema.

Além das medidas de segurança, também foi solicitada a remoção de imagens e nome da empresa do sistema. Essa solicitação deriva do fato de que na primeira versão implementada foi elaborada uma interface contendo o logotipo e nome da ENB, bem como imagens dos produtos vendidos, e isso poderia chamar a atenção de pessoas que estivessem próximo dos vendedores nas *lan houses*. Dessa forma, a diretoria pediu que a interface fosse reformulada e todas as referências à empresa fossem removidas. Assim, além de chamar menos atenção, mesmo que alguém obtivesse acesso ao endereço do sistema, não saberia a qual empresa ele pertence.

4.2 Planejamento de Atividades

Nesta seção, mostramos o cronograma planejado (Seção 4.2.1), além de apresentarmos uma visão geral sobre as reuniões que participamos durante o estágio (Seção 4.2.2) e as estimativas feitas (Seção 4.2.3).

4.2.1 Cronograma

Dado que o nosso processo de desenvolvimento é iterativo e incremental, foi preciso definir um cronograma de atividades que foi atualizado ao longo do estágio. Inicialmente, após definidos os primeiros requisitos do sistema, um cronograma de atividades foi elaborado de forma a guiar o processo de implementação e validação dos mesmos. Para isso, marcamos a nossa primeira *release* para o dia 30 de abril. Nessa data, teríamos uma reunião com a diretoria da empresa, de forma a apresentar o sistema implementado até então e validar os requisitos.

Na data marcada, os principais requisitos levantados até o momento estavam implementados e foram validados com a diretoria, conforme descrito na Seção 4.1.4. A partir da resposta recebida na reunião e dos requisitos adicionais levantados posteriormente, marcamos a data da segunda *release* para o dia 01 de junho. Com todos os requisitos implementados e validados pela diretoria, começamos a trabalhar no planejamento do treinamento, que ficou marcado para o dia 12 de junho. Após isso, o sistema

ficaria funcionando em caráter experimental durante duas semanas. A partir de então, nosso trabalho seria evoluir o software de acordo com as observações feitas (Seção 4.7).

Como é possível notar, não foi definido um planejamento detalhado por requisitos. As *releases* foram definidas e, visando cumprir o prazo para as suas entregas, os requisitos relacionados a cada uma foram implementados. Na implementação, era dada prioridade aos principais requisitos e que tivessem dependências entre si. Todas as datas de *release* e treinamento foram cumpridas. Por fim, o cronograma planejado na Proposta de Estágio (Apêndice A) não previa um processo no qual a fase de planejamento era incrementada iterativamente. Portanto, tivemos que adequar o cronograma planejado no Plano de Estágio ao processo definido.

4.2.2 Reuniões

Semanalmente, nos reunimos com o gerente a fim de realizarmos o acompanhamento da execução das atividades e discutirmos problemas que surgiram durante a implementação. Era também nesses momentos que era verificado o cumprimento do cronograma.

Além das reuniões entre a equipe de desenvolvimento e o gerente e das já citadas na Seção 4.1, também nos reunimos com a diretoria da empresa para discutirmos detalhes do treinamento que seria realizado e com o faturista para obtermos informações sobre o uso do sistema e problemas enfrentados.

4.2.3 Estimativas

Para que pudéssemos estimar o porte – e, conseqüentemente, o custo – do sistema a ser desenvolvido, utilizamos o método de **Pontos de Casos de Uso (PCU)** [Karner 1993]. Esse método estima o tamanho de sistemas orientados a objeto a partir dos casos de uso gerados, se baseando no modo como os usuários utilizarão o sistema, a complexidade das ações requeridas por cada tipo de usuário e uma análise de alto nível dos passos necessários para a realização de cada tarefa.

Utilizamos o diagrama de casos de uso construído na fase de validação dos requisitos (Figura 4.1). De acordo com o PCU, mensuramos a complexidade dos 3 atores do nosso sistema e dos 27 casos de uso, além dos fatores técnicos e ambientais do projeto. A estimativa dada foi de 23,67 meses, assumindo 20h semanais de trabalho. Entretanto,

sabendo que iríamos utilizar o Grails para o desenvolvimento, definimos um fator de redução para 30% dessa estimativa, dado que este *framework* possibilita alta produtividade. Esse valor foi obtido de forma empírica, a partir de analogia com os tempos real e estimado de um projeto que implementamos com Grails para a disciplina de Laboratório de Engenharia de Software.

Além disso, com o *framework* utilizado, os casos de uso referentes a manutenção simples (criação, visualização, atualização, remoção e listagem), sem lógica adicional, são implementados de uma só vez, já que o Grails gera boa parte do código e o desenvolvedor precisa apenas adaptá-lo ao seu sistema e modificar a interface. Dessa forma, para a estimativa com PCU, reduzimos os nossos casos de uso de 27 para 11, pois agrupamos os casos de uso citados em apenas um caso de uso por módulo. Com essa redução, a nova estimativa foi de 10,96 meses. Por fim, com o fator de redução estipulado, a estimativa do sistema foi de 3,29 meses. A tabela contendo detalhes da mensuração feita para o cálculo do PCU pode ser encontrado no Apêndice D.

A estimativa feita serviu para termos uma ideia sobre a viabilidade de o sistema ser implementado no tempo previsto. Além disso, aumentou a nossa confiança em relação ao fator de redução anteriormente utilizado sistemas desenvolvidos com o Grails.

4.3 Modelagem

Nesta seção, descreveremos os dois tipos de modelos criados com o objetivo de facilitar o entendimento e implementação do sistema. Modelamos dois tipos de diagramas: um estrutural (Seção 4.3.1), que representa o diagrama de classes implementado; e outro comportamental (Seção 4.3.2), com o fluxo de atividades de uma das etapas mais importantes do sistema.

4.3.1 Modelagem Estrutural: Diagrama de Classes

Na etapa de modelagem do processo de desenvolvimento, detalhamos o sistema num nível estrutural por meio de um diagrama de classes [OMG 2009]. A Figura 4.3 apresenta o diagrama de classes modelado para o Sistema de Gestão de Pedidos.

Ao total, o sistema contém 7 classes que representam as suas entidades e 2

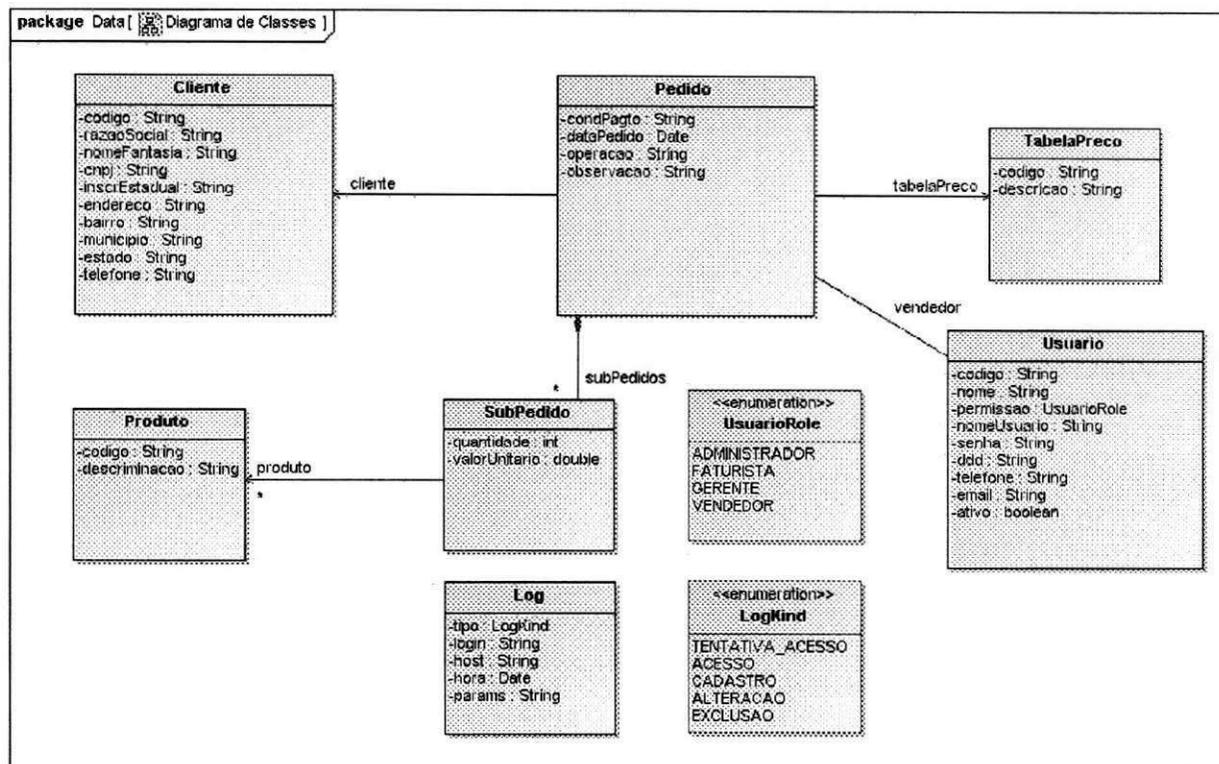


Figura 4.3: Diagrama de Classes do Sistema

enumerações representando os tipos criados para o sistema. Cada Usuário contém, além de dados pessoais, informações sobre seu nível de acesso ao sistema. Além disso, cada Pedido contém vários SubPedidos, como também referências para o Cliente a quem o pedido pertence e para o Usuário que o cadastrou no sistema. Cada Pedido também contém uma referência à Tabela de Preços utilizada. Um SubPedido tem uma referência a um Produto e informações da quantidade solicitada pelo cliente e valor unitário do produto, de acordo com a tabela de preços. Por fim, cada ação feita no sistema é gravada em formato de *log*. Cada Log contém informações sobre a ação efetuada e identificação de quem a realizou.

4.3.2 Modelagem Comportamental: Diagrama de Atividades

Após descrever a parte estrutural do sistema, agora iremos mostrar um exemplo de como foi modelada a parte comportamental do mesmo. Para isso, modelamos um Diagrama de Atividades [OMG 2009] para mostrar o fluxo de atividades da funcionalidade central do sistema, que é cadastrar pedidos.

A Figura 4.4 apresenta o diagrama de atividades modelado, mostrando as atividades que devem ser feitas pelos usuários a fim de cadastrar um pedido. Basicamente, as atividades são as seguintes:

1. O usuário acessa a página de login e se autentica digitando seu nome de usuário e senha;
2. Caso os dados estejam corretos, ele é redirecionado para a página principal referente ao seu nível de acesso;
3. Para cadastrar um pedido, o usuário deve escolher a opção correspondente no menu principal;
4. Novamente será solicitada a sua autenticação, de modo a reforçar a segurança;
5. Caso a senha digitada seja a correta, o usuário acessa o formulário de cadastro de novo pedido e o preenche com as informações solicitadas (razão social do cliente, condição de pagamento, tabela de preço, tipo da operação e quantidade dos produtos e valor unitário de cada um);
6. Após clicar em Salvar, o sistema verifica se todos os dados foram inseridos corretamente (por exemplo, se a razão social do cliente refere-se a um cliente cadastrado no sistema). Caso esteja tudo correto, o pedido é salvo e os dados cadastrados são mostrados ao usuário.
7. Caso algum dado inserido esteja incorreto, o usuário é redirecionado novamente ao formulário, que contém as informações que já foram preenchidas por ele e as mensagens de erro informando quais dados foram preenchidos incorretamente.

4.4 Implementação

Após as três fases iniciais do processo de desenvolvimento, começamos a fase de implementação do Sistema de Gestão de Pedidos ¹. Essa fase é dividida em duas etapas: codificação e inspeção de código. A primeira está especificada na Seção 4.4.1 e trata de aspectos referentes à codificação e ao que foi implementado no sistema. Já a Seção 4.4.2 refere-se à inspeção de código feita no código implementado.

¹Uma demonstração do sistema está disponível em: <http://www.diegocavalcanti.com/ControlePedidos>

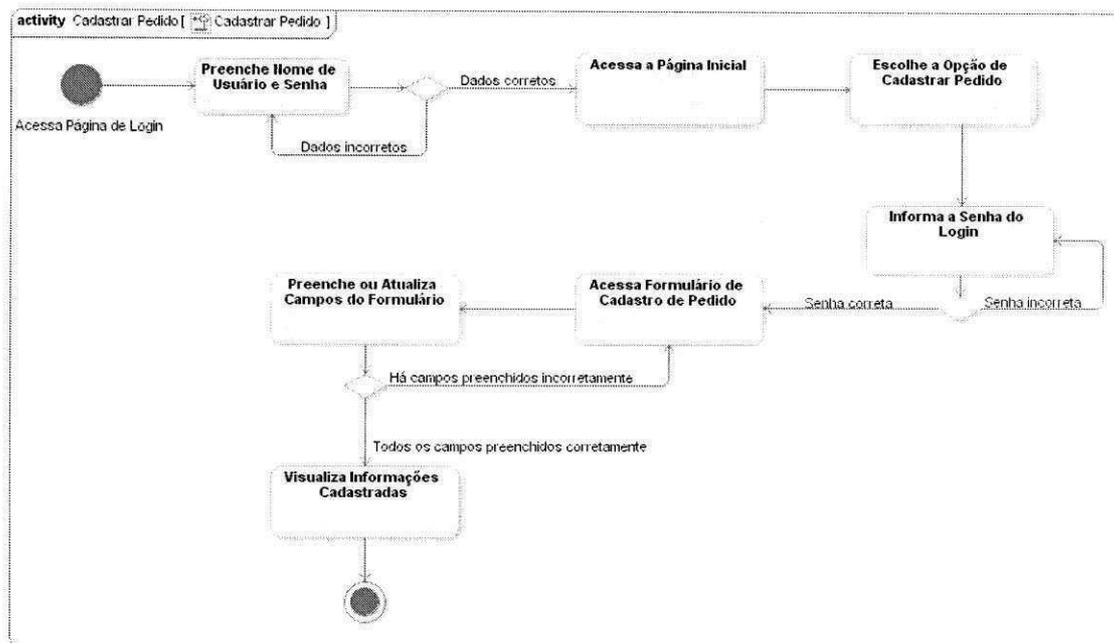


Figura 4.4: Diagrama de Atividades - Cadastro de Pedidos

4.4.1 Codificação

Após a fase de elicitação de requisitos, iniciou-se a implementação das funcionalidades requeridas. Para isso, o desenvolvimento da aplicação seguiu a arquitetura definida pelo Grails. Foram implementadas 9 classes com atributos e restrições próprios de cada entidade.

Cada classe contém restrições que indicam pré-condições que devem ser satisfeitas pelos usuários ao realizar um cadastro ou atualização. Tomemos como exemplo o código apresentado na Figura 4.5. Ele representa as restrições da classe `SubPedido`. Como é possível ver no diagrama de classes criado (Figura 4.3), essa classe contém três atributos: `produto`, `quantidade` e `valor unitário`. As restrições aplicadas indicam que nenhum dos atributos pode ser nulo e que o valor mínimo de `quantidade` e `valorUnitario` deve ser zero (não pode ser negativo). Indica ainda que o valor unitário deve conter uma precisão de dois dígitos decimais. Essas restrições são importantes para manter a integridade dos dados no banco de dados.

```
static constraints = {  
    produto(nullable:false)  
    quantidade(nullable:false, min:0)  
    valorUnitario(nullable:false, scale:2, min:0.0 as Double)  
    pedido(nullable:false)  
}
```

Figura 4.5: Exemplo de Restrições de Atributos em Classes

Medidas de Segurança Implementadas no Sistema

Conforme dito anteriormente, após a etapa de validação dos requisitos (Seção 4.1.4), foi visto que seria necessário a implementação de algumas medidas de segurança a fim de aumentar a proteção do sistema. A seguir, citamos cada medida de segurança implementada e qual o problema atacado por cada uma.

Criptografia de Senha dos Usuários. Um dos primeiros requisitos não-funcionais solicitados foi o controle de acesso ao sistema por meio de nome de usuário e senha. Mas uma preocupação que surgiu foi que as senhas seriam armazenadas no banco de dados e qualquer pessoa que tivesse acesso a elas, poderia acessar o sistema. Dessa forma, propusemos criptografar as senhas dos usuários, de modo que, mesmo que alguém tivesse acesso aos dados do banco de dados, não haveria a possibilidade de descobrir a senha do usuário.

Certificação Digital. Por trafegarem na *web*, os dados enviados entre a máquina do cliente e o servidor, podem ser facilmente interceptados. Principalmente quando se trata de computadores públicos. Dessa forma, foi implementada uma proteção dos dados durante essa comunicação entre cliente e servidor. A nossa proposta foi de utilizar certificação digital para criptografar os dados durante a comunicação. Mesmo havendo um custo adicional para a empresa na implementação desse requisito, ele foi aceito, dado que é de extrema relevância.

Evitar Auto-completar de Campos nos Navegadores de Internet. O sistema seria acessado a partir de computadores aos quais muita gente tem acesso. Então, a diretoria da empresa alertou para o fato de que os navegadores de internet armazenam os dados fornecidos pelos usuários em formulários, a fim de facilitar o preenchimento do mesmo formulário em um outro acesso. Isso poderia fazer com que outros clientes das *lan houses* utilizadas obtivessem alguma informação a respeito dos dados dos pedidos

cadastrados. Como forma de evitar esse problema, o sistema implementado solicita aos navegadores que não armazenem dados digitados nos formulários. Isso é feito a partir de um atributo de configuração do formulário. Os navegadores mais utilizados (por exemplo, Mozilla Firefox e Internet Explorer) respeitam essa configuração e não armazenam os dados dos formulários para auto-completar posteriormente.

Restrição à Utilização de Navegadores. Havia a necessidade de garantir que os navegadores utilizados para acesso ao sistema fossem seguros e confiáveis. Para tanto, o sistema foi configurado para redirecionar qualquer requisição vinda de um navegador diferente do Mozilla Firefox e do Internet Explorer para uma página de erro. Ou seja, nenhum usuário consegue acessar o sistema a partir de um navegador diferente dos citados.

Senha Alfabética Codificada. Por acessar o sistema a partir de *lan houses*, os vendedores estão suscetíveis a utilizar computadores infectados e com programas maliciosos instalados. Assim, é possível que alguém mal-intencionado instale um programa que capture o que é digitado no computador, inclusive as senhas. Então, não seria seguro criarmos uma senha para cada vendedor e ele ter que digitá-la sempre da mesma forma ao acessar o sistema. Portanto, nossa proposta foi de criar senhas alfabéticas codificadas para cada vendedor. Cada senha contém três letras e, ao solicitá-la, o sistema apresenta uma tabela com mapeamentos aleatórios de letras e números (exemplo na Figura 4.6).

C, P - 82	A, Y - 52	E, S - 49
J, R - 84	G, Q - 78	B, Z - 95
L, V - 55	D, O - 18	I, U - 50
K, W - 40	F, T - 53	H, N - 28

Figura 4.6: Exemplo de Senha Alfabética Codificada

O vendedor deve buscar na tabela os números que correspondem à sua senha alfabética e digitar os números nos campos apropriados para isso. No exemplo da Figura 4.6, se a senha do vendedor for RTJ, ele deveria digitar: 84 53 84. Como o mapeamento da tabela é gerado aleatoriamente, mesmo que alguém grave o que foi digitado, é impossível descobrir qual é a senha alfabética real do vendedor.

Encerramento Automático de Sessão. É possível que um vendedor utilize o sistema e, após o seu uso, esqueça de encerrar a sessão, deixando-o autenticado. Dessa

forma, alguém poderia acessar os dados cadastrados no sistema, mesmo sem o poder da senha do vendedor. Para evitar essa situação, o sistema encerra automaticamente a sessão após 10 minutos de inatividade.

Confirmação de Autenticação para Operações Críticas. Se o sistema for esquecido autenticado, alguém pode utilizá-lo antes de completar os 5 minutos de inatividade necessários para que a sessão se encerre automaticamente. Dessa forma, a medida de segurança anterior não inibiria a ação de uma pessoa mal-intencionada. Assim, nossa proposta foi de solicitar confirmação de autenticação antes de operações críticas. Por exemplo, visualizar a lista de pedidos cadastrados no sistema é considerada uma operação crítica, dado que ela contém informações sigilosas da empresa (valor total de vendas, por exemplo). Então, antes da visualização dos pedidos, o sistema solicita novamente ao usuário que coloque a sua senha.

Limite de Tentativas de Autenticação. Embora pouco provável, por meio de tentativas e erros um usuário mal-intencionado poderia descobrir a senha de um usuário do sistema. Dessa forma, limitamos o número de tentativas frustradas de autenticação para três. Assim, após errar a quarta vez a sua senha, o usuário tem seu acesso ao sistema bloqueado e é necessário que um administrador efetue o desbloqueio.

Registro de Operações Realizadas. A fim de manter o controle das operações realizadas, as principais ações dos usuários são mantidas registradas no banco de dados. Assim, é possível saber exatamente qual usuário efetuou cada ação, como também o horário e o endereço IP de onde o sistema foi utilizado.

Interface com o Usuário

Um ponto importante no sistema é a sua usabilidade. Por isso, a preocupação com a interface do usuário foi constante durante a implementação do software. A seguir, mostraremos como ficou a interface do sistema após a validação com o cliente.

Página de Login. Ao acessar o sistema, é apresentada a página de login, com o formulário para se autenticar. A Figura 4.7 apresenta a tela de *login* do sistema. Conforme dito anteriormente, como medida de segurança, cada usuário contém uma senha alfabética e, na tela de login, é apresentada uma tabela com mapeamentos aleatórios de letras em números.

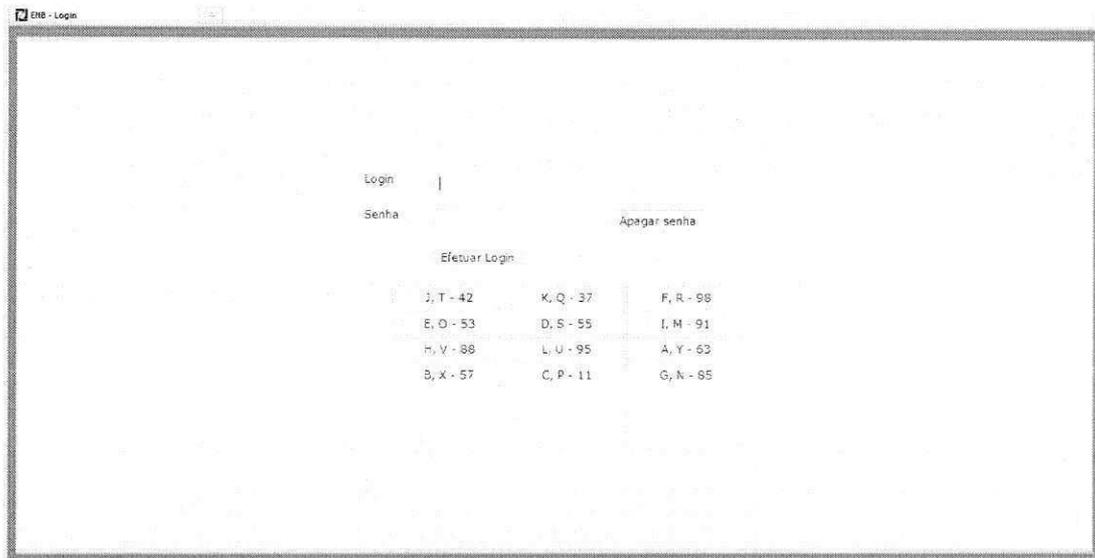


Figura 4.7: Tela de Login

Permissões de Usuários e Página Inicial. Os usuários do sistema podem ser classificados em quatro níveis de acesso diferentes: administrador, faturista, gerente e vendedor. O administrador tem acesso a todas as funcionalidades do sistema. Já o faturista, tem acesso a todas as funcionalidades, exceto a manutenção de cadastros de usuários. Enquanto isso, os vendedores e o gerente têm acesso apenas a manutenção dos pedidos. A diferença entre vendedor e gerente é que o vendedor só pode listar e alterar os pedidos feitos por ele, enquanto que o gerente lista e altera todos os pedidos cadastrados. Uma visão geral sobre as permissões de acesso de cada nível de usuário, pode ser vista no diagrama de casos de uso modelado (Figura 4.1).

Como cada nível, exceto os dois últimos, tem permissões diferentes, a aplicação contém diferentes menus para fornecer acesso às suas páginas. A Figura 4.8 apresenta a página inicial do administrador. Por ter acesso a todas as funcionalidades do sistema, o seu menu permite acesso a todos os módulos da aplicação (cadastros de: clientes, usuários, produtos e pedidos, além dos relatórios). O faturista também tem acesso a todas essas funcionalidades, exceto o Cadastro de Usuários. Sua página inicial é apresentada na Figura 4.9. Os vendedores e gerente têm páginas iniciais iguais (Figura 4.10), dado que ambos tem acesso às mesmas funcionalidades.

Formulários. A maior parte do sistema desenvolvido é composta de formulários para cadastro e atualização dos dados. A Figura 4.11 mostra um exemplo do

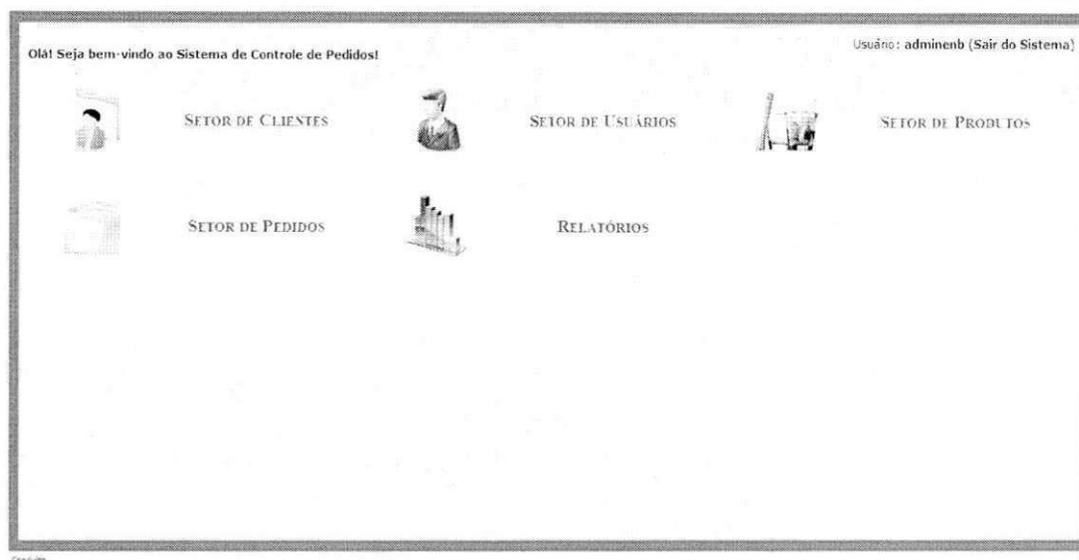


Figura 4.8: Página Inicial de Administrador

formulário preenchido pelos vendedores para cadastro de pedidos. Esse formulário em especial é dividido em duas partes. A primeira parte contém a identificação do cliente e informações sobre os dados financeiros do pedido. Na segunda parte é onde são cadastradas as informações principais do pedido: quantidade e valor unitário de cada produto solicitado.

Relatórios. Os relatórios disponíveis no sistema podem ser refinados por vendedor e período. Eles contêm os nomes dos clientes e valores de cada pedido feito num determinado intervalo de datas. A Figura 4.12 apresenta um exemplo de relatório de vendas cadastradas pelo vendedor hipotético José da Silva.

Tecnologias e Ferramentas Utilizadas

O **Grails** (Seção 3.2) foi utilizado como plataforma de desenvolvimento *web*. Além disso, durante o estágio, foram utilizadas outras tecnologias e algumas ferramentas que auxiliaram no processo de desenvolvimento, como também ferramentas para gerenciamento de versões, controle de documentos, gerenciamento de banco de dados, gerenciamento da segurança da aplicação, agendamento de tarefas e geração de relatórios. Dado que um requisito não-funcional solicitado pela empresa foi que não houvesse custos adicionais com compras de licenças de software, utilizamos apenas ferramentas e tecnologias gratuitas. A seguir, apresentamos mais detalhes sobre algumas tecnologias e ferramentas que foram

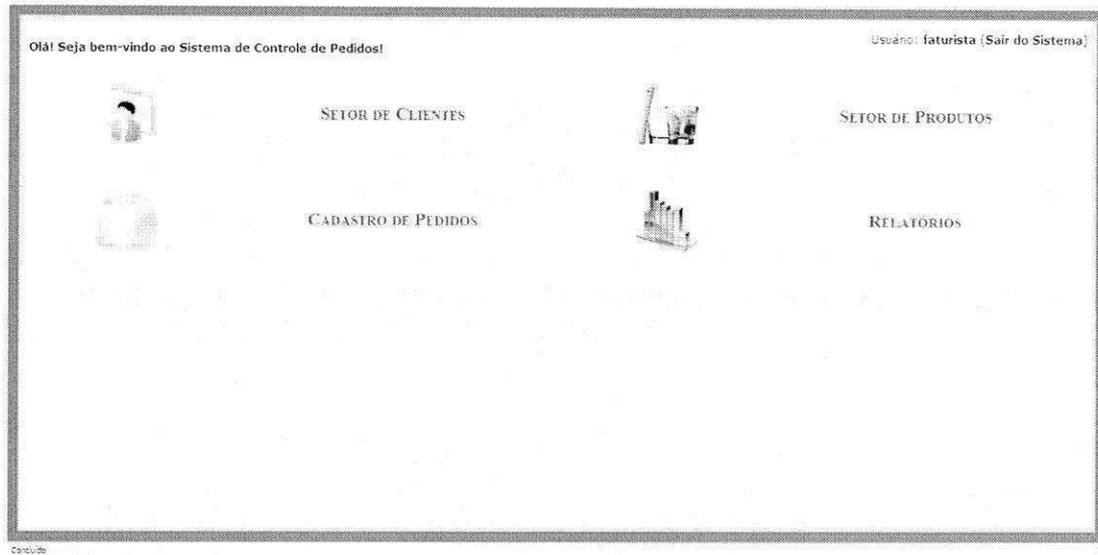


Figura 4.9: Página Inicial de Faturista

utilizadas.

Para a fase de codificação, utilizamos o ambiente de desenvolvimento Netbeans [Sun Microsystems 2009]. Atualmente, ele é o que oferece melhor integração com o Grails. Além disso, como forma de gerenciar versões do código e possibilitar um desenvolvimento colaborativo, utilizamos o **Subversion (SVN)** [Tigris.org 2009]. O **Google Docs** [Google Inc. 2009] também foi utilizado para controle de documentos, tais como: planilhas de acompanhamento do trabalho, apresentações das reuniões e arquivamento de erros reportados.

Para o armazenamento dos dados, foi utilizado o banco de dados relacional **MySQL** [MySQL, AB 2009] versão 5.1.34. O MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD) otimizado para aplicações *web*, multiplataforma e com baixa exigência de processamento em comparação com outros SGBDs. Quanto à segurança, possui um sistema de privilégios e senhas que é flexível e seguro, além disso, o tráfego de senhas entre a máquina cliente e servidor é criptografado. É um banco de dados *open source* e utilizado por grandes empresas, a exemplo da NASA [MySQL, AB 2009].

Também utilizamos o *framework* de segurança **JSecurity** [Jsecurity.org 2009], para implementação da segurança da aplicação. Ele visa prover segurança para autenticação, controle de acesso, criptografia e gerência de sessão da aplicação *web*. Foi possível utilizar o JSecurity no Grails por meio de um *plugin*. O JSecurity consiste de duas ca-

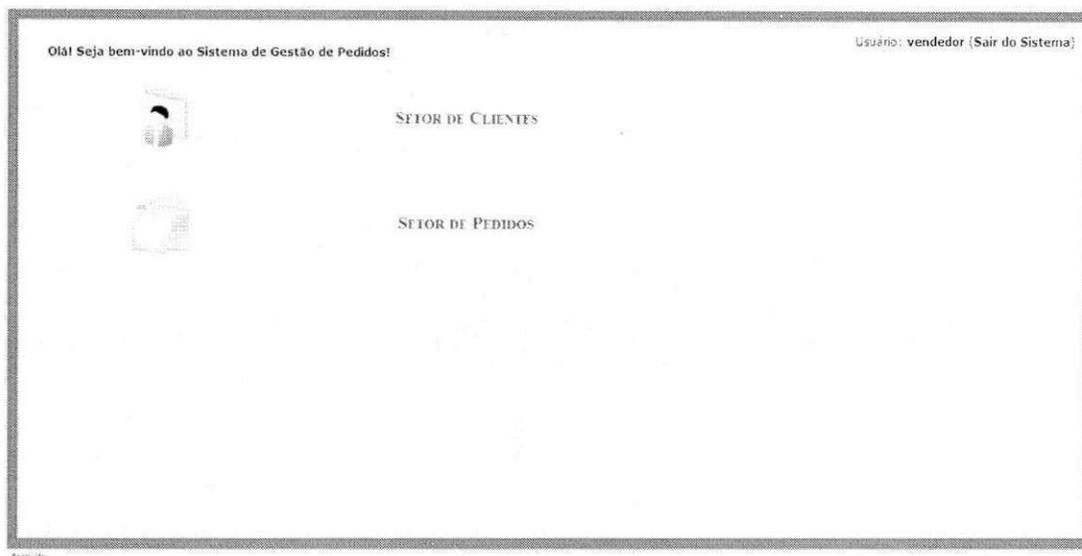


Figura 4.10: Página Inicial de Vendedor

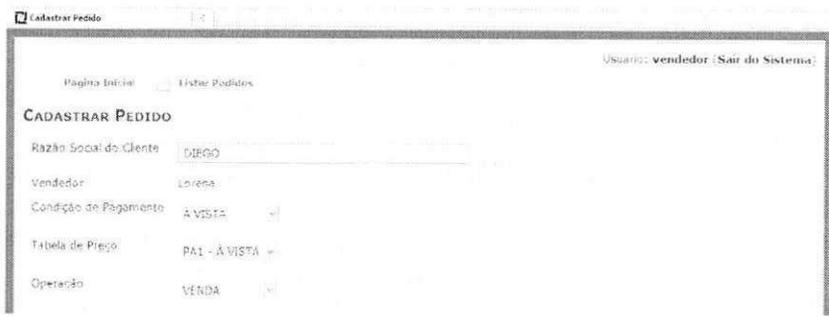
madam basicamente: gerenciador de permissões, que determina se um usuário possui uma regra ou permissão particular; e controle de acesso, que restringe o acesso a algumas partes da aplicação baseado nas regras e permissões existentes.

O **Quartz** [OpenSymphony 2009] é utilizado para realizar agendamento de tarefas que são executadas em um intervalo específico de tempo. Por fim, o **JasperReports** foi utilizado para geração de relatórios. Ao passo que para construção e edição desses relatórios, a ferramenta utilizada foi o **iReport** versão 2.0.2 [Jaspersoft Corporation 2009].

4.4.2 Inspeção de Código

Paralelo à etapa de implementação, foi feita uma revisão do código pelo desenvolvedor secundário, de forma a encontrar e corrigir defeitos na implementação e melhorar a qualidade do código. Para isso, após cada módulo implementado, o revisor analisava o código implementado e o comparava com o descrito no Documento de Requisitos do Usuário (Seção 4.1.3). Caso fosse encontrado algum erro ou sugestão de melhoria, ele próprio o corrigia e documentava, a fim de que o desenvolvedor principal verificasse o que estava errado e/ou poderia ter sido mais bem implementado.

1ª Parte



CADASTRAR PEDIDO

Página Inicial | [Fazer Pedidos](#) | Usuário: vendedor (Sair do Sistema)

Razão Social do Cliente: DIEGO

Vendedor: Lorena

Condição de Pagamento: À VISTA

Tabela de Preço: PA1 - À VISTA

Operação: VENDA

2ª Parte

Pedido	Código	Descrição	Quantidade	Valor Unitário
	0301003	CARANGUEJO PET 480M PT 12		
	0301026	GOSTOSA PET 480ML PT 12		
	0301004	CARANGUEJO 600ML CX 12 SD		
	0301007	CARANGUEJO 600ML GR 24 CD		
	0301008	CARANGUEJO DURO 600ML CX 6 SD		
	0301009	CARANGUEJO DURO 600ML CX 1 SD		
	0301010	CARANGUEJO 880ML CX 12 SD		
	0301012	CARANGUEJO 880ML CX 12 CD		
	0301015	CARANGUEJO 880ML GR 12 CD		
	0301024	CARANGUEJO PRATA 350ML PT 06		

Figura 4.11: Formulário de Cadastro de Pedidos

4.5 Verificação e Validação

Para verificar falhas no sistema, realizamos testes funcionais para cada módulo implementado. Essa atividade é feita em conjunto com a Inspeção de Código (Seção 4.4.2). Nela, o desenvolvedor secundário executa testes funcionais visando encontrar erros de programação e inconsistências nas mensagens de erros e na interface. Os erros encontrados são corrigidos e reportados ao desenvolvedor principal.

Os testes são feitos buscando explorar situações nas quais os formulários falhem ao validar o conteúdo inserido, de acordo com as restrições dos atributos das classes. Por exemplo, no cadastro de pedidos, não é possível salvar um pedido sem indicar a razão social do cliente. Assim, o testador pode verificar se o erro reportado está claro o suficiente para que o usuário entenda onde está errado.

Não implementamos testes funcionais automáticos porque, na época da implementação, a versão do Grails utilizada estava com problemas de integração com a ferramenta que executa esse tipo de teste. Mesmo assim, o processo de verificação supriu as nossas necessidades, dado que o sistema está há mais de 1 mês em funcionamento e, ape-

nas no começo do seu uso, foram encontrados alguns poucos problemas de implementação, que foram rapidamente corrigidos.

4.6 Implantação e Treinamento

Após a implementação e validação do sistema, ele está pronto para ser colocado em produção. Para tanto, foi realizado um treinamento com seus usuários, de forma a apresentá-los o que foi desenvolvido e mostrar como utilizar. O treinamento também foi importante para sondarmos as primeiras impressões dos usuários sobre o sistema e, a partir disso, adaptarmos a interface do mesmo às suas necessidades.

O treinamento do Sistema de Gestão de Pedidos foi realizado com 5 vendedores, além do faturista e toda a diretoria. A equipe de desenvolvimento do software (ambos os desenvolvedores e gerente) também estava presente. No primeiro momento do treinamento, foi realizada uma apresentação oral, com demonstração de como utilizar o sistema. Após isso, foi entregue a cada usuário um roteiro de atividades (Apêndice E) descrevendo, passo-a-passo, como realizar as principais operações. Cada vendedor utilizou um computador para acessar o sistema e seguir o roteiro entregue. A partir de observação do uso feito nesse momento do treinamento, pudemos coletar algumas sugestões que foram incorporadas ao sistema. Por exemplo, após 5 minutos de inatividade, a sessão era encerrada e o usuário era redirecionado à tela de login. Entretanto, vimos que muitos vendedores não tinham habilidade com digitação. Dessa forma, 5 minutos eram pouco tempo. Assim, aumentamos o tempo de encerramento da sessão por inatividade para 10 minutos.

Ao final do treinamento, foi aplicado um questionário anônimo de sondagem de satisfação do usuário (Apêndice F). O questionário solicitava notas de 0 a 10 para o sistema e para o treinamento, como também pedia opinião sobre a usabilidade do software, além de sugestões e/ou críticas. Dos 5 vendedores que responderam ao questionário, apenas um deles deu nota abaixo de 10,0 para o sistema (nota 6,0). Ele também marcou que o sistema era **razoavelmente fácil de usar**, enquanto que os demais afirmaram que o sistema era **fácil** de utilizar. Como forma de justificar os conceitos dados, ele deixou uma sugestão para que houvesse mais treinamento, devido a sua pouca habilidade com computador. Por fim, todos os vendedores deram nota 10,0 ao treinamento e disseram

que foi fácil cadastrar um pedido.

4.7 Evolução

Após o treinamento, o sistema passou por um período de maturação, sendo utilizado pelos vendedores em caráter experimental. Durante duas semanas, os vendedores preenchiam os formulários no sistema mas também entregavam os pedidos preenchidos nos formulários em papel, conforme era feito antes. Isso serviu para que o faturista comparasse os dados cadastrados no software com os dados escritos no papel. Assim, foi possível verificar se os vendedores estavam sabendo preencher os dados sem introduzir erros e se o sistema era realmente confiável.

Após as duas semanas de experimentos, o sistema foi aceito pela diretoria da empresa e começou realmente a ser utilizado como única fonte de informação sobre os pedidos recebidos. Desde então, ele tem se mantido estável e funcionando sem problemas de implementação. Apenas algumas mudanças foram feitas de forma a melhorar a usabilidade do mesmo.

Durante o período de observação, o sistema ficou fora do ar em alguns momentos mas foi detectado que o problema é um estouro de memória no servidor de hospedagem contratado. Para resolver isso, buscamos formas de diminuir o consumo de memória, mas o problema persiste, embora com menos frequência. A solução encontrada foi contratar mais memória para o servidor da aplicação. Atualmente, estamos aguardando que essa contratação seja feita, para observarmos se o problema será completamente solucionado.

Empresa Nacional de Bebidas Ltda.
Rua Benjamin Constant, 170 - Estação Velha
65105-195 - Campina Grande - PB
Fones: (+55 83) 3310-3777
Fax: (+55 83) 3310-3757

Relatório de Pedidos

Vendedor: José da Silva

Código : 001

	Data	Cliente	Valor
1	20/07/2009	Bar Dois Irmãos Ltda.	R\$ 2554,34
2	21/07/2009	Supermercado Melhor Preço	R\$ 6912,04
3	22/07/2009	Bar Dois Irmãos Ltda.	R\$ 5500,00

Valor Total: R\$ 14976,38

Figura 4.12: Exemplo de Relatório de Vendas

Capítulo 5 - Considerações Finais

Este relatório descreveu o estágio desenvolvido na Empresa Nacional de Bebidas (ENB), cujo objetivo geral foi o de desenvolver um sistema que melhorasse a gestão de pedidos da referida empresa. Para isso, seguimos um processo de desenvolvimento que nos guiou nas diversas etapas da Engenharia de Software, desde o levantamento de requisitos até a implantação, treinamento e evolução do software. Para a implementação do **Sistema de Gestão de Pedidos**, utilizamos o *framework* Grails e algumas outras tecnologias adicionais, a fim de satisfazer os requisitos do sistema.

No ambiente acadêmico, apesar de desenvolvermos diversos projetos de disciplina, raramente, vemos um sistema nosso sendo utilizado na prática por usuários reais. Por isso, é inimaginável para um aluno que não passou por uma experiência de trabalho real, qual é o impacto causado por um sistema dentro de uma empresa. Na universidade, aprendemos bastante sobre os possíveis problemas técnicos que possam surgir no desenvolvimento de uma aplicação. Entretanto, raramente tomamos conhecimento dos desafios enfrentados pela equipe de desenvolvimento para a aceitação de seu produto por parte da diretoria e usuários, que têm que modificar sua rotina para se adequar a uma nova filosofia de trabalho.

A partir da observação da rotina de trabalho da empresa, pudemos aprender como funciona a gestão de pedidos da ENB. Ao iniciarmos o estágio, detectamos alguns problemas que poderiam ser resolvidos pelo Sistema de Gestão de Pedidos. Um problema crítico detectado foi que o faturista geralmente fazia hora-extra nas sextas-feiras, saindo apenas após todos os pedidos terem sido processados. Atualmente, com o sistema em funcionamento, seu horário de trabalho diminuiu consideravelmente. Isso gerou uma economia para a empresa, de forma que agora não é mais necessário pagá-lo por horas extras. Outro problema enfrentado era a falta de informação contínua sobre os pedidos feitos durante a semana. Atualmente, a diretoria recebe, de forma automática, um e-mail diário com o resumo dos pedidos cadastrados durante o dia. Assim, é possível manter o controle diário sobre o trabalho dos vendedores.

É importante ressaltar que a ENB já possui um software que auxilia no controle

de pedidos. Entretanto, eles queriam algo que fosse atualizado pelos vendedores, via *web*, diariamente. Dessa forma, todos os dias, o faturista copia os dados do nosso sistema para o sistema deles. A integração do dois sistemas será feito como trabalho futuro. Não fizemos ainda porque a empresa desejava usar o nosso sistema antes para ver como os seus funcionários o aceitavam e como o mesmo se adequava à filosofia da empresa.

Outro ponto forte do nosso trabalho foi a utilização do Grails para o desenvolvimento do sistema. Com ele, pudemos implementar todo o sistema de forma ágil, possibilitando, assim, a nossa participação em todas as etapas do processo dentro do tempo previsto para o estágio. Por fim, o treinamento dos usuários e acompanhamento do uso do sistema, sempre ouvindo as opiniões dos usuários, foram fundamentais para a implementação de melhorias no mesmo.

Este estágio caracterizou-se como uma experiência enriquecedora. Além de ser possível aplicar na prática os conhecimentos de Engenharia de Software aprendidos na universidade, ele nos possibilitou o amadurecimento para defender o nosso produto diante da empresa, como também nos ensinou a lidar com os usuários, entendendo as suas preocupações e necessidades.

Como trabalho futuro, pretendemos integrar o sistema desenvolvido durante o estágio com o sistema que já é utilizado na empresa para gerenciar as vendas. Além disso, há perspectivas de adicionar funcionalidades do sistema no tocante ao controle de trabalho de cada vendedor. Mas são requisitos que ainda devem ser melhor elicitados com a diretoria.

Referências Bibliográficas

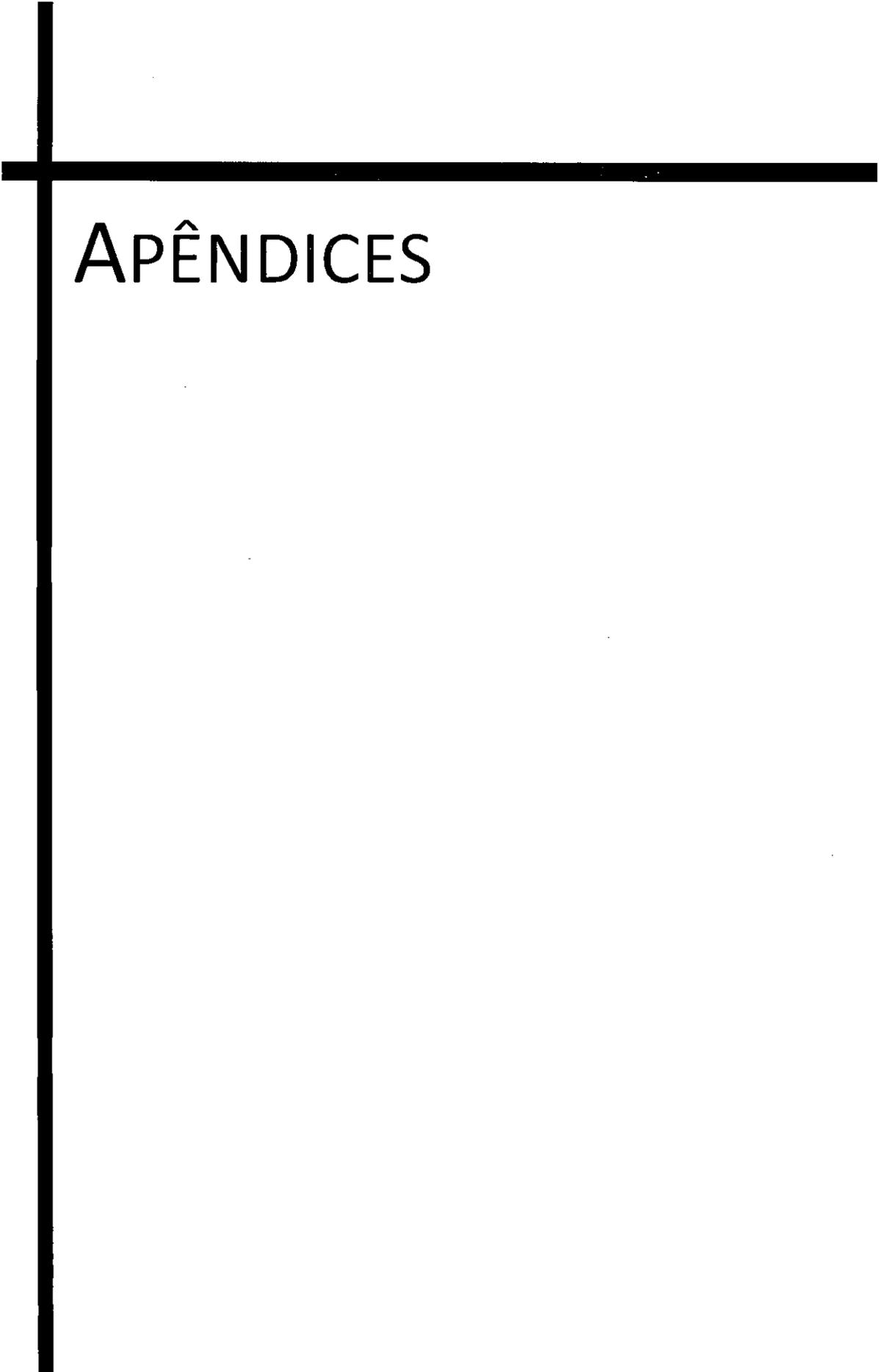
- [Ambler 2002]AMBLER, S. *Agile modeling*. [S.l.]: Wiley, 2002.
- [Beck e Andres 2004]BECK, K.; ANDRES, C. *Extreme programming explained: embrace change*. [S.l.]: Addison-Wesley Professional, 2004.
- [Google Inc. 2009]Google Inc. Google Docs. <http://docs.google.com/>, 2009.
- [Humphrey 1989]HUMPHREY, W. S. *Managing the Software Process*. [S.l.]: Addison-Wesley Professional, 1989.
- [Jaspersoft Corporation 2009]Jaspersoft Corporation. iReport Project Home. <http://jasperforge.org/projects/ireport>, 2009.
- [Jsecurity.org 2009]Jsecurity.org. JSecurity — Easy Java Security. <http://www.jsecurity.org>, 2009.
- [Karner 1993]KARNER, G. Resource estimation for Objectory projects. *Objective Systems SF AB (copyright owned by Rational Software)*, 1993.
- [Koenig et al. 2007]KOENIG, D. et al. *Groovy in action*. [S.l.]: Manning Publications Co. Greenwich, CT, USA, 2007.
- [Kruchten 2000]KRUCHTEN, P. *The rational unified process: an introduction*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2000.
- [MySQL, AB 2009]MySQL, AB. MySQL: The world's most popular open source database. <http://www.mysql.com>, 2009.
- [MySQL, AB 2009]MySQL, AB. Referências e Cases — MySQL Brasil. <http://www.mysqlbrasil.com.br/?q=node/4>, 2009.
- [OMG 2009]OMG, O. M. G. UML Resource Page. <http://www.uml.org/>, 2009.
- [OpenSymphony 2009]OPENSYPHONY. Quartz — Enterprise Job Scheduler. <http://www.opensymphony.com/quartz/>, 2009.

[Rocher e Brown 2009]ROCHER, G.; BROWN, J. *The Definitive Guide to Grails (Definitive Guide)*. [S.l.]: Apress, 2009. ISBN 9781590599952.

[Sommerville 2007]SOMMERVILLE, I. *Software Engineering*. [S.l.]: Addison-Wesley, 2007.

[Sun Microsystems 2009]Sun Microsystems. Netbeans IDE. <http://www.netbeans.org/>, 2009.

[Tigris.org 2009]TIGRIS.ORG. Subversion. <http://subversion.tigris.org/>, 2009.



APÊNDICES

Apêndice A



UFCG – UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CEEI – CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DSC – DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

Plano de Estágio Integrado

Empresa Nacional de Bebidas LTDA – ENB

Processo de Automatização de Pedidos

Diego Tavares Cavalcanti

Mat.: 20611122

Orientador: Prof. Dr. Rohit Gheyi

Março 2009

Ambiente do Estágio

O estágio será realizado no laboratório de desenvolvimento de software da Empresa Nacional de Bebidas LTDA, situada à Rua Benjamin Constant nº 170, Estação Velha, Campina Grande, Paraíba.

O trabalho será realizado num grupo de 3 (três) pessoas, sendo dois desenvolvedores e um gerente.

Supervisão

Supervisão Acadêmica

Nome: Rohit Gheyi

Endereço: Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Avenida Aprígio Veloso, 882 – CEP: 58.109-970
Bodocongó, Campina Grande, PB – Brasil.

Email: rohit@dsc.ufcg.edu.br

Supervisão Técnica

Nome: Stephanie Bezerra Muniz

Endereço: Benjamin Constant, 170
Estação Velha, Campina Grande, PB – Brasil.

Email: smuniz@caranguejo.com.br

Resumo do Problema

A Empresa Nacional de Bebidas é uma fábrica situada em Campina Grande, com mais de 60 anos de tradição, proprietária de oito marcas de bebidas famosas, tais como: Cachaça Caranguejo, Ron D'Cuba, Vodka Paloff. As vendas são feitas por vendedores que representam a empresa em outras cidades. Cada vendedor deve visitar os clientes, preencher os pedidos e os entregar à fábrica. Após análise dos pedidos feita pelo faturista, a produção deve ser planejada de acordo com os insumos em estoque e, então, iniciada.

Atualmente, um problema enfrentado pela empresa é a falta de controle da atividade dos vendedores, pois, como eles passam a maior parte do tempo viajando, a empresa não tem como controlar a quantidade de horas trabalhadas em cada dia, pois os pedidos são entregues ao faturista apenas uma vez por semana. Além disso, a produção tem uma carga de trabalho excessiva em apenas no fim da semana, pois só é possível iniciar a produção quando os pedidos são entregues.

Como não há um controle automatizado dos pedidos, também não é possível fazer um planejamento da produção de acordo com o histórico de vendas de cada vendedor.

Proposta da Solução

O objetivo do estágio é desenvolver um módulo automatizado de pedidos da fábrica. Esse módulo será implementado como um sistema Web, com o framework Grails [1] e abrangerá todo o processo de pedidos: cadastro de produtos disponíveis para revenda, cadastro de clientes, formulário de pedidos por clientes, aprovação dos pedidos e geração de relatórios com o histórico de vendas de cada vendedor. Os cadastros incluem, além da inserção, edição, atualização, remoção e busca.

Metodologia

O sistema será desenvolvido de maneira iterativa e incremental, seguindo práticas de desenvolvimento de software adaptadas para a nossa realidade e baseadas em processos de desenvolvimento existentes no mercado, tais como: eXtreme Programming [2] e easYProcess [3].

A equipe será responsável pelas práticas concernentes ao levantamento de requisitos, planejamento, modelagem, desenvolvimento e testes do sistema, implantação e treinamento, além de adequar o processo de desenvolvimento que será seguido à realidade da empresa.

Atividades Planejadas

Devem ser desenvolvidas as seguintes atividades no estágio:

Atividades	Horas estimadas
Levantamento de requisitos	10
Cronograma detalhado de trabalho	5
Planejamento de <i>releases</i>	2
Estimativas de custo	3
Análise e Modelagem da aplicação	10
Implementação	120
Testes de unidade e ajustes na ferramenta	30
Criação da documentação	30
Implantação e treinamento	20
Escrita de relatório técnico	70
Total de Horas	300

Resultados Esperados

Espera-se que, ao fim do estágio, o sistema esteja finalizado, conforme o cronograma apresentado, e esteja sendo utilizado na Empresa Nacional de Bebidas LTDA, resolvendo o problema identificado.

Com isso, o aluno pretende ter a oportunidade de se deparar com um desenvolvimento para uma empresa, com as atribuições e responsabilidades que não são encontradas integralmente no meio acadêmico.

Cronograma de Atividades

O cronograma de atividades proposto é o seguinte:

Tarefa	Mar	Abr	Mai	Jun	Jul
Levantamento de requisitos	X				
Cronograma detalhado de trabalho	X				
Planejamento de <i>releases</i>	X				
Estimativas de custo	X				
Análise e Modelagem da aplicação		X			
Implementação		X	X	X	
Testes de unidade e ajustes na ferramenta			X	X	X
Criação da documentação	X	X	X	X	
Implantação e treinamento				X	
Escrita de relatório técnico	X	X	X	X	X
Defesa do Estágio					X

Bibliografia

- [1] ROCHER, G.. The definitive guide to Grails: 1 ed. Apress, 2006. 364 p. ISBN 1-59059-758-3.
- [2] GARCIA, F. P. et al. easYProcess: um processo de desenvolvimento de software para uso no ambiente acadêmico. In: XII Workshop de Educação em Informática - XXIV Congresso da Sociedade Brasileira de Computação, 2004, Salvador.
- [3] BECK, K., Extreme Programming Explained: Embrace Change. 2000, Reading, Massachusetts: Addison-Wesley.

Apêndice B

Suporte de Informática ENB Questionário Avaliativo ao Usuário de Informática

Nome: _____

Setor: _____

Função: _____

O questionário abaixo tem como objetivo principal identificar os níveis de conhecimento em informática que o nosso funcionário possui, para que o mesmo possa ter apoio através da equipe de Suporte para melhorar os seus conhecimentos e assim ter um melhor desenvolvimento na sua rotina de trabalho. Portanto, pedimos que seja respondido com total seriedade.

Instruções:

Assinale uma alternativa em cada questão.

Questionário:

1. Utiliza computador no seu trabalho?

Sim

Não

2. Qual a frequência de uso do computador?

Nunca utilizei

Poucas vezes no ano

Uma vez por semana

Duas vez por semana

Outros: _____

3. Utiliza a internet?

Nunca utilizei

Poucas vezes por mês

Uma vez por semana

Duas vez por semana

Três vez por semana

Outros: _____

4. Qual o tipo de atividade feita por você no computador (jogar, digitar texto, navegar na internet, ver email, etc.)?

5. Possui email?

Sim. Qual? _____

Não

Apêndice C

Documento de Requisitos do Usuário *Sistema de Gestão de Pedidos* *Empresa Nacional de Bebidas*

Índice

1. **Objetivos do documento**
2. **Visão geral**
3. **Glossário Básico do Domínio**
4. **Funcionalidades do Sistema**
5. **Mecanismos de Segurança do Sistema**

1. Objetivos do Documento

Este documento é parte integrante do Sistema de Controle de Pedidos da Empresa Nacional de Bebidas - ENB. Ele deve ser usado como importante meio de comunicação entre a ENB e a contratada para possível conferência de funcionalidades implementadas.

2. Visão Geral

O Sistema de Controle de Pedidos visa armazenar toda informação a respeito das transações relacionadas aos pedidos de produtos da ENB.

3. Glossário

- **Funcionalidades do Sistema** – Os serviços disponibilizados no sistema.
- **Usuários** – Usuários que possuem nome de usuário e senha para utilizar o sistema.
- **Nível de acesso por cargo** – Quantidade de funcionalidades que estarão disponíveis para um usuário que possui determinado cargo cadastrado.
- **Cliente** – Alvo das vendas dos produtos da ENB.
- **Administrador** – Usuário que administrará o Controle de Pedidos.
- **Faturista** – Usuário que foi cadastrado no cargo de faturista no sistema.
- **Gerente** – Usuário que foi cadastrado no cargo de gerente no sistema.
- **Vendedor** – Usuário que foi cadastrado no cargo de vendedor no sistema.
- **Endereço IP** – Endereço numérico (acrônimo de *Internet Protocol*) que identifica o computador de onde o sistema foi acessado.
- **Efetuar Transações** – Trocar informações com o servidor via Internet.
- **Criptografia** - Técnica de converter (cifrar) uma mensagem utilizando um código secreto com o propósito de segurança.

4. Funcionalidades do Sistema Controle de Pedidos

A seguir, as funcionalidades do Controle de Pedidos.

4.1 Cadastrar usuário

Identificação:	Anexo:	Nome:
01	-	Cadastrar usuário
Descrição:		
Um administrador poderá cadastrar um novo usuário no sistema.		
Para cadastrar um usuário, é necessário cadastrar as seguintes informações:		
<ul style="list-style-type: none">• Código único que o identifica no sistema;• Nome do usuário;• Login para acesso ao sistema;• Cargo que identifica seu nível de acesso no sistema;• Telefone;• E-mail.		
Um usuário também possui uma senha, que é gerada automaticamente pelo sistema.		

4.2 Atualizar cadastro de usuário

Identificação:	Anexo:	Nome:
02	-	Atualizar cadastro de usuário
Descrição:		
Um administrador poderá atualizar todas as informações cadastradas de um usuário do sistema.		

4.3 Listar usuários

Identificação:	Anexo:	Nome:
03	-	Listar usuários
Descrição:		
Um administrador poderá listar todos os usuários cadastrados no sistema.		

4.4 Buscar usuário

Identificação:	Anexo:	Nome:
04	-	Buscar um usuário
Descrição:		
Um administrador poderá buscar um usuário cadastrado no sistema.		

4.5 Visualizar usuário

Identificação:	Anexo:	Nome:
05	-	Visualizar usuário
Descrição:		
Um administrador poderá visualizar os dados de um usuário cadastrado no sistema.		

4.6 Desabilitar um usuário

Identificação:	Anexo:	Nome:
06	-	Desabilitar um usuário
Descrição:		
Um administrador poderá desabilitar um usuário cadastrado no sistema. Ao ter seu cadastro desabilitado, um usuário não poderá mais acessar o sistema.		
Mesmo desabilitado, o cadastro do usuário é mantido para futuras consultas pelo administrador.		

4.7 Habilitar um usuário

Identificação:	Anexo:	Nome:
07	-	Habilitar um usuário
Descrição:		
Um administrador poderá habilitar um usuário cadastrado no sistema, dado que ele estava desabilitado.		

4.8 Cadastrar cliente

Identificação:	Anexo:	Nome:
08	-	Cadastrar cliente
Descrição:		
Um usuário (administrador ou faturista) poderá cadastrar um novo cliente no sistema.		
Para cadastrar um cliente, é necessário cadastrar as seguintes informações:		
<ul style="list-style-type: none">• Código que o identifica de forma única no sistema;• Razão social;• Nome Fantasia;• CNPJ;• Inscrição Estadual;• Endereço;• Bairro;• Município;• Estado;• Telefone.		

4.9 Atualizar cadastro de cliente

Identificação:	Anexo:	Nome:
09	-	Atualizar cadastro de cliente
Descrição:		
Um usuário (administrador ou faturista) poderá atualizar o cadastro de um cliente cadastrado no sistema.		

4.10 Listar clientes

Identificação:	Anexo:	Nome:
10	-	Listar clientes
Descrição:		
Um usuário, seja qual for o seu nível de acesso, poderá listar todos os clientes cadastrados no sistema.		

4.11 Buscar cliente

Identificação:	Anexo:	Nome:
11	-	Buscar um cliente
Descrição:		
Um usuário, seja qual for o seu nível de acesso, poderá buscar um cliente cadastrado no sistema.		

4.12 Visualizar cliente

Identificação:	Anexo:	Nome:
12	-	Visualizar cliente
Descrição:		
Um usuário, seja qual for o seu nível de acesso, poderá visualizar os dados de um cliente cadastrado no sistema.		

4.13 Remover cliente

Identificação:	Anexo:	Nome:
13	-	Remover um cliente
Descrição:		
Um usuário (administrador ou faturista) poderá remover um cliente cadastrado no sistema.		

4.14 Cadastrar produto

Identificação:	Anexo:	Nome:
14	-	Cadastrar produto
Descrição:		
Um usuário (administrador ou faturista) poderá cadastrar um novo produto no sistema.		
Para cadastrar um produto, é necessário cadastrar as seguintes informações:		
<ul style="list-style-type: none">• Código que o identifica de forma única no sistema;• Discriminação do produto. Lorena Lira•		

4.15 Atualizar cadastro de informações de produto

Identificação:	Anexo:	Nome:
15	-	Atualizar cadastro de informações de um produto
Descrição:		
Um usuário (administrador ou faturista) poderá atualizar o cadastro de um produto cadastrado no sistema.		

4.16 Listar produtos

Identificação:	Anexo:	Nome:
16	-	Listar produtos
Descrição:		
Um usuário (administrador ou faturista) poderá listar todos os produtos cadastrados no sistema.		

4.17 Buscar produto

Identificação:	Anexo:	Nome:
17	-	Buscar um produto
Descrição:		
Um usuário (administrador ou faturista) poderá buscar um cliente cadastrado no sistema.		

4.18 Visualizar produto

Identificação:	Anexo:	Nome:
18	-	Visualizar produto
Descrição:		
Um usuário (administrador ou faturista) poderá visualizar os dados de um produto cadastrado no sistema.		

4.19 Remover produto

Identificação:	Anexo:	Nome:
19	-	Remover um produto
Descrição:		
Um usuário (administrador ou faturista) poderá remover um produto cadastrado no sistema.		

4.20 Cadastrar pedido

Identificação:	Anexo:	Nome:
20	-	Cadastrar pedido
Descrição:		
<p>Um usuário, seja qual for o seu nível de acesso, poderá cadastrar um novo pedido no sistema. Este pedido terá a identificação do usuário que o criou.</p> <p>Para adicionar um pedido, é preciso cadastrar as seguintes informações:</p> <ul style="list-style-type: none">• Razão social do cliente;• Condição de pagamento;• Tabela de preços utilizada;• Operação do pedido: venda, bonificação ou troca;• Quantidade de cada produto solicitado;• Valor unitário de cada produto, de acordo com a tabela de preços utilizada. <p>O pedido também contém a informação de qual foi o usuário que o registrou e a data em que esse registro foi feito.</p>		

4.21 Atualizar cadastro de pedido

Identificação:	Anexo:	Nome:
21	-	Atualizar cadastro de pedido
Descrição:		
<p>Um usuário, seja qual for o seu nível de acesso, poderá atualizar o cadastro de um pedido criado por ele próprio.</p> <p>Um usuário com nível de acesso de Administrador ou Faturista poderá atualizar o cadastro de todos os pedidos cadastrados no sistema.</p>		

4.22 Listar pedidos

Identificação:	Anexo:	Nome:
22	-	Listar pedidos
Descrição:		
<p>Um usuário, seja qual for o seu nível, poderá listar os pedidos criados por ele próprio.</p> <p>Um usuário com nível de acesso de Administrador, Faturista ou Gerente poderá listar todos os pedidos cadastrados no sistema.</p>		

4.23 Buscar pedido

Identificação:	Anexo:	Nome:
23	-	Buscar um pedido
Descrição:		
Um usuário, seja qual for o seu nível, poderá buscar pedidos cadastrados por ele próprio.		
Um usuário com nível de acesso de Administrador, Faturista ou Gerente poderá buscar todos os pedidos cadastrados no sistema.		

4.24 Visualizar pedido

Identificação:	Anexo:	Nome:
24	-	Visualizar pedido
Descrição:		
Um usuário, seja qual for o seu nível, poderá visualizar os dados dos pedidos cadastrados por ele próprio.		
Um usuário com nível de acesso de Administrador, Faturista ou Gerente poderá visualizar todos os pedidos cadastrados no sistema.		

4.25 Remover pedido

Identificação:	Anexo:	Nome:
25	-	Remover um pedido
Descrição:		
Um administrador poderá remover um pedido cadastrado no sistema.		

4.26 Gerar relatório de pedidos

Identificação:	Anexo:	Nome:
26	-	Gerar relatório de pedidos realizados
Descrição:		
Um usuário (administrador ou faturista) poderá gerar relatório dos pedidos realizados por um determinado vendedor ou por todos os vendedores em um intervalo de data especificado		

4.27 Acessar o sistema

Identificação:	Anexo:	Nome:
27	-	Acessar o sistema
Descrição:		
Um usuário, seja qual for o seu nível de acesso, poderá acessar o sistema, desde que esteja cadastrado e seu cadastro esteja habilitado.		

5. Mecanismos de Segurança do Sistema

5.1 Criptografia da senha

Identificação:	Anexo:	Nome:
28	-	Criptografia da Senha
Descrição:		
Para garantir a segurança das senhas dos usuários no banco de dados é utilizado o algoritmo de criptografia SHA-1.		

5.2 Inibição do armazenamento dos dados pelo navegador

Identificação:	Anexo:	Nome:
29	-	Inibição do armazenamento dos dados pelo navegador
Descrição:		
Os navegadores de Internet armazenam o que o usuário digita nos campos de formulário, a fim de facilitar o preenchimento futuro, sugerindo o que foi digitado anteriormente.		
O sistema de Controle de Pedidos impede que esse armazenamento seja feito por qualquer navegador que siga a padronização da Internet.		

5.3 Utilização de navegadores específicos

Identificação:	Anexo:	Nome:
30	-	Utilização de navegadores específicos
Descrição:		
O Sistema Controle de Pedidos só pode ser utilizado nos seguintes navegadores de Internet:		
<ul style="list-style-type: none">• Internet Explorer• Mozilla Firefox		

5.4 Senha alfabética

Identificação:	Anexo:	Nome:
31	-	Utilização de senha alfabética para acesso ao sistema
Descrição:		
<p>A senha de acesso ao sistema que cada usuário possui ao ser cadastrado no Controle de Pedidos será composta de 3(três) letras.</p> <p>Nas operações para as quais é necessário informar a senha, será apresentada ao usuário uma tabela com um mapeamento aleatório de letras e números, de forma que o usuário digitará apenas os números referentes às letras da sua senha alfabética.</p>		

5.5 Fechamento automático de sessão (timeout)

Identificação:	Anexo:	Nome:
32	-	Fechamento automático de sessão (timeout)
Descrição:		
<p>Se o usuário ficar mais de 10 minutos sem efetuar transações no sistema, será solicitado que o usuário digite seu nome de usuário e senha novamente para voltar a utilizá-lo.</p>		

5.6 Redirecionamento para a página inicial

Identificação:	Anexo:	Nome:
33	-	Redirecionamento para a página inicial do sistema
Descrição:		
<p>Após ocorrer um fechamento automático de sessão, o Controle de Pedidos é imediatamente redirecionado para a página inicial.</p>		

5.7 Acessar novamente o sistema para efetuar operações críticas

Identificação:	Anexo:	Nome:
34	-	Acessar novamente o sistema para efetuar operações críticas
Descrição:		

Ao executar operações críticas no Controle de Pedidos, o usuário deverá se identificar (acessar) novamente no sistema.

A seguir, as operações para as quais será solicitada novamente a senha:

- Listar pedidos
- Criar pedido

5.8 Bloqueio após limite de tentativas de acesso ao sistema

Identificação:	Anexo:	Nome:
35	-	Bloqueio após limite de tentativa de acesso ao sistema
Descrição:		
O usuário que exceder o limite de 03 (três) tentativas de acesso ao sistema, terá o seu acesso ao mesmo bloqueado. Neste caso, o seu acesso só poderá ser desbloqueado pelo administrador do sistema.		

5.9 Registro de operações

Identificação:	Anexo:	Nome:
36	-	Registro de operações
Descrição:		
As seguintes operações são registradas no sistema:		
<ul style="list-style-type: none">• Acessar o sistema• Criar um usuário, cliente, produto ou pedido• Remover um usuário, cliente, produto ou pedido• Atualizar um usuário, cliente, produto ou pedido• Desabilitar um usuário• Habilitar um usuário		
Esse registro contém informações sobre a data, o horário, o endereço IP e o nome do usuário que realizou a operação.		

5.10 Utilização de certificação digital

Identificação:	Anexo:	Nome:
37	-	Utilização de certificação digital
Descrição:		
O sistema poderá utilizar certificação digital, garantindo que os dados transmitidos entre a máquina do usuário e o servidor de dados estarão incompreensíveis para alguém que consiga interceptar esses dados durante a transmissão.		
Obs: a certificação digital deve ser provida pelo servidor onde o sistema estará hospedado.		

Apêndice D

Estimativa de esforço baseado em pontos de casos de uso

Referências

Resource Estimation for Objectory Projects de Gustav Kerner (artigo)
Applying Use Cases - A Practical Guide de Geri Schneider e Jason P. Winters (livro)

1.0. Mensurando Complexidade dos Atores

Ator	Interface	Peso
Simple	Interface de programa (API)	1
Médio	Protocolo (Ex.:TCP/IP) ou interface em modo texto	2
Complexo	Interface gráfica	3

	Qtd. Atores	Valor
	0	0
	0	0
	3	9
Total	3	9

2.0. Mensurando Complexidade dos Use Cases

Caso de U	Descrição	Peso
Simple	< 3 transações ou < 5 classes de análise	5
Médio	4-7 transações ou 5 a 10 classes de análise	10
Complexo	> 7 transações ou > 10 classes de análise	15

	Qtd. Casos de Uso	Valor
	10	50
	1	10
		0
Total	11	60

PCUNA	69
-------	----

PCUNA = PCU Não Ajustados

3.0. Considerando Fatores Técnicos do Projeto

Fator	Descrição	Peso
T1	Sistema distribuido	2
T2	Objetivos de performance	1
T3	Eficiência on-line	1
T4	Complexidade de processamento	1
T5	Código reusável em outras aplicações	1

Atribuido	Valor
1	2
3	3
4	4
0	0
0	0

T6	Facilidade de instalação	0,5
T7	Facilidade de uso	0,5
T8	Portabilidade	2
T9	Facilidade de alterações (<i>changeability</i>)	1
T10	Concorrência	1
T11	Segurança	1
T12	Acesso direto a terceiros	1
T13	Necessidade de facilidades especiais de treinamento para usuários	1

2	1
5	2,5
2	4
2	2
1	1
5	5
0	0
5	5
FatorT	29,5

FCT	0,895
-----	-------

FCT = Fator de Complexidade Técnica

4.0. Considerando Fatores Ambientais

Fator	Descrição	Peso
F1	Familiaridade da equipe com o processo utilizado	1,5
F2	Experiência da equipe	0,5
F3	Experiência da equipe em OO	1
F4	Capacidade dos analistas da equipe	0,5
F5	Motivação	1
F6	Estabilidade dos requisitos	2
F7	Estagiários ou funcionários em tempo parcial	-1
F8	Domínio da tecnologia e configuração do ambiente	-1,5

Atribuído	Valor
5	7,5
4	2
4	4
3	1,5
5	5
4	8
5	-5
0	0
FatorA	23

FA	0,71
----	------

FA = Fator Ambiental

5.0. Pontos de Caso de Uso

PCU	PCUNA *FCT*FA	43,85
Pessoa-hora por unidade de PCU		20 pessoa-hora/PCU
Estimativa em pessoa-hora		876,92
Tamanho da equipe		1 pessoas
Estimativa em horas		876,92 horas
Estimativa em meses		10,96 meses
Estimativa em meses com Grails (30% do valor estimado)		3,29 meses

Apêndice E



Treinamento para o Sistema de Controle de Pedidos
Roteiro de Atividades – Vendedores

Diego Cavalcanti
Lorena Lira
Rohit Gheyi
Stephanie Muniz

1ª atividade: Abrir o Sistema de Gestão de Pedidos

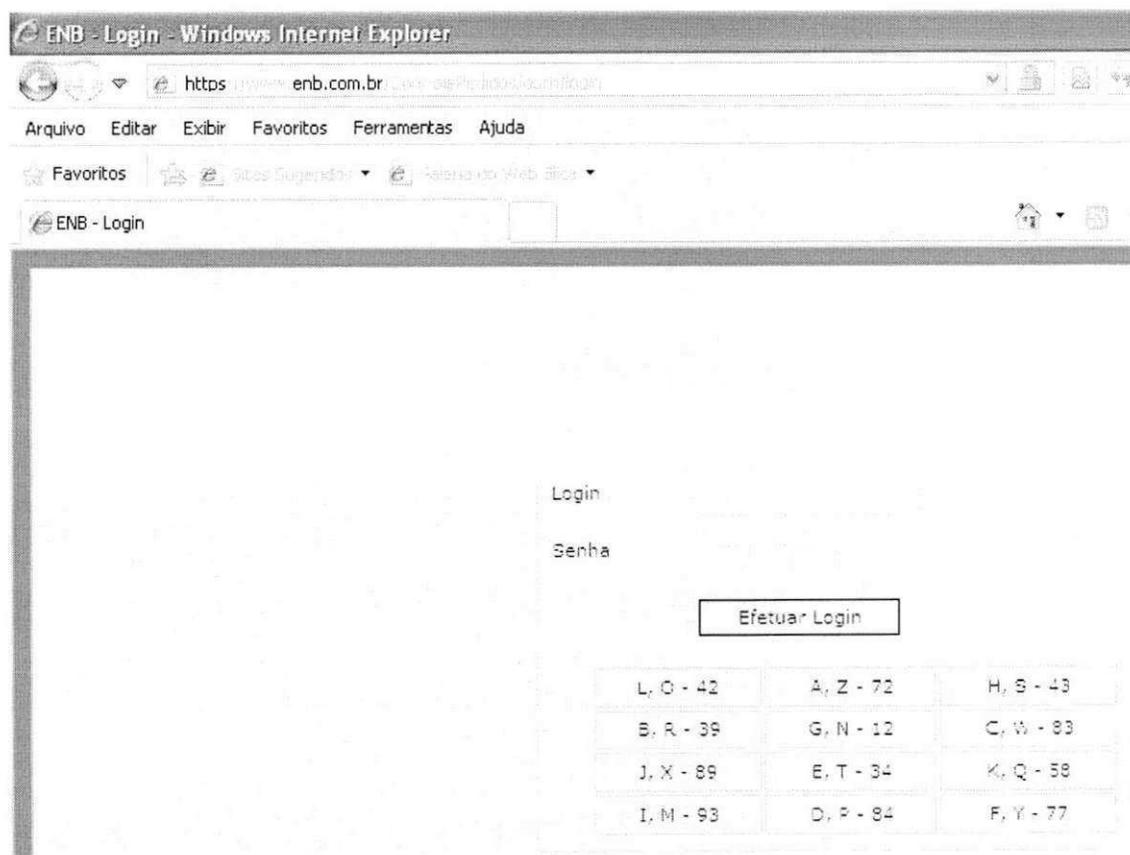
→ Digite o seguinte link no navegador de Internet (**Internet Explorer** ou **Mozilla Firefox**).

○ www.pedidos.enb.com.br

→ Aperte a tecla ENTER e aguarde.



→ Para que você tenha certeza que está acessando o site correto do Controle de Pedidos verifique se você foi redirecionado para a seguinte tela:



2ª atividade: Acessar o Sistema de Gestão de Pedidos

- Após a primeira atividade, você estará na página de acesso ao sistema:
- Insira o seu login *no campo*
- Insira os **números correspondentes à sua senha alfabética** no campo **Senha**:
 - *Senha alfabética*
A sua senha alfabética é única e intransferível
 - *Números correspondentes a serem inseridos*
Códigos numéricos serão apresentados e você **substituirá** as letras de sua senha alfabética **pelos números correspondentes** que aparecerem na tela. Cada letra da sua senha é associada a um número e esta **associação muda aleatoriamente** a cada novo acesso.
- Aperte o botão **Efetuar Login**.

Exemplo: Vendedora Lorena Lira.

Login: lorena

Senha alfabética: K B E

Números a serem digitados, para esta tabela: **19 23 56**

The image shows a login form with the following fields and elements:

- Login:** A text input field containing the name "lorena".
- Senha:** A password input field with three dots and a cursor, indicating a masked password.
- Efetuar Login:** A button to submit the login information.
- Key Table:** A table below the form that maps letters to numbers. The letters K, B, and E from the password are circled, and their corresponding numbers (19, 23, and 56) are also circled.

I, P - 87	E, S - 56	E, W - 23
H, Q - 20	G, N - 32	F, U - 41
J, X - 85	L, T - 54	K, V - 19
A, Z - 71	D, M - 70	C, Y - 93

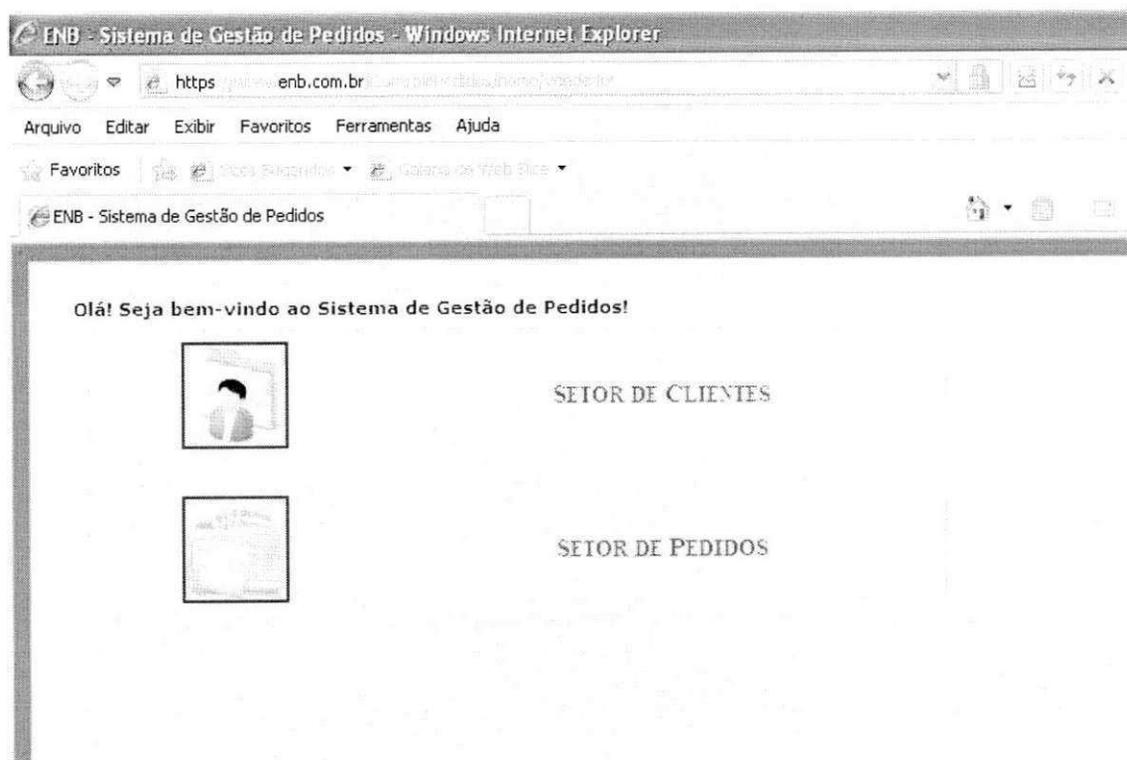
3ª atividade: Visualizar a sua Página Inicial

→ Após realizar a segunda atividade, você será encaminhado para a sua **página inicial** (imagem abaixo).

→ Perceba que sua página inicial possui duas opções:

- *Setor de Clientes*
- *Setor de Pedidos*

Exemplo: **Página Inicial** da vendedora Lorena Lira.



Setor de Pedidos

Parte do sistema responsável por cadastrar um pedido, visualizar um pedido e listar todos os seus pedidos.

4ª atividade – Parte A: Cadastrar um pedido

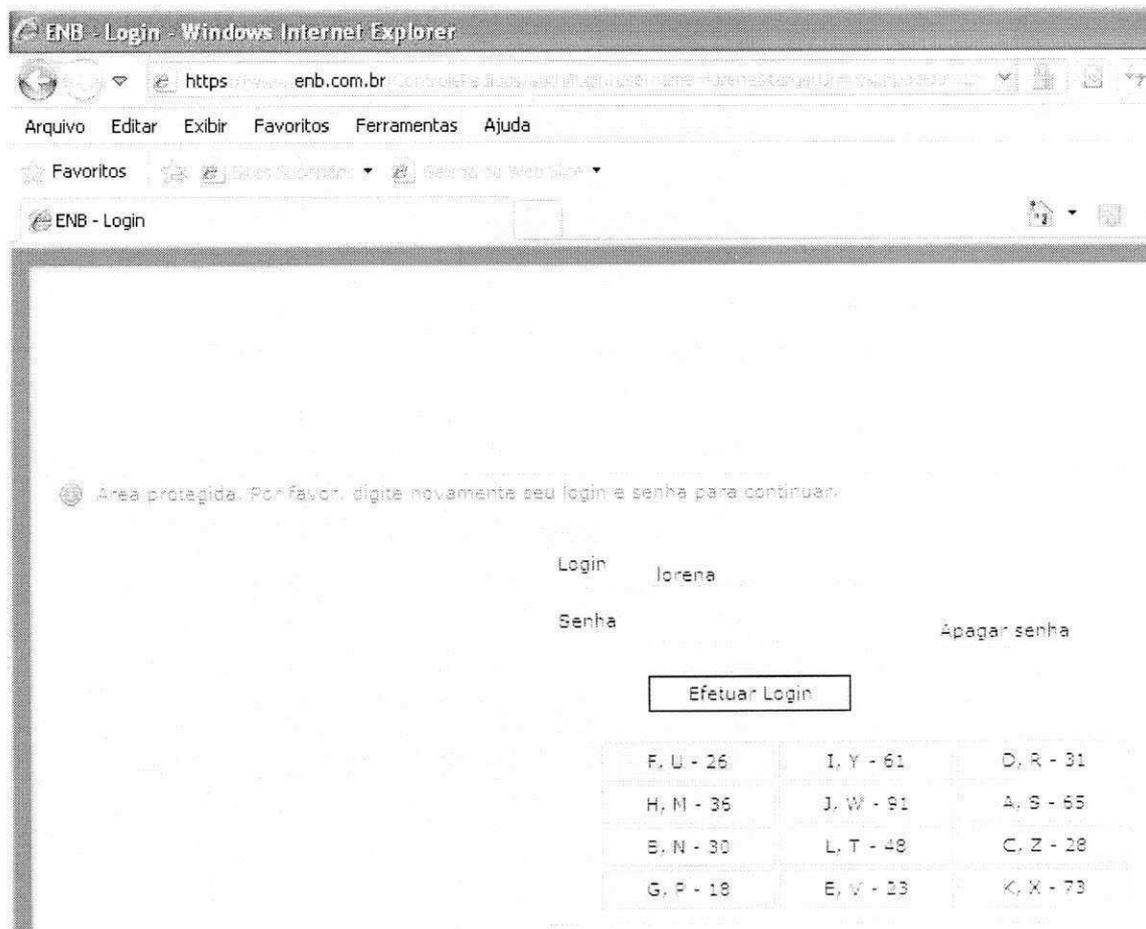
Área Protegida

→ A partir da sua **página inicial**, você pode acessar o setor de pedidos clicando em **Setor de Pedidos**.

→ Em seguida, o sistema solicitará que você digite a sua senha novamente, pois o **Setor de Pedidos** é uma **área protegida**, por conter dados importantes.

Exemplo:

Após clicar em Setor de Pedidos, é necessário que Lorena Lira insira sua senha novamente (**área protegida**).



4ª atividade – Parte B: Cadastrar um pedido

Informações Gerais

- Após ter inserido sua senha novamente, a criação de pedido será efetivamente iniciada. A criação é composta por duas partes:
 - Inserção de Informações Gerais
 - Inserção dos pedidos

- Você verá a parte de inserção de Informações Gerais sobre o pedido.

Exemplo de pedido feito pela vendedora Lorena Lira

Inserção das Informações gerais

Vendedor: Você!
Data: **22/05/2009**
Cliente: **Francisco de Sousa Pires**

Após negociação com o senhor Francisco, ficou definido o que ele deseja: a quantidade e o preço por unidade:

Desejo do Cliente	Quantidade	Valor Unitário
Vodka Paloff 980ML CX 12 SD	150	R\$ 57,20

Condição de Pagamento
45 dias no Cheque

Observações feitas
Bonificação:
O cliente receberá 45 caixas de vodka. Dessa forma, o valor por caixa ficará por R\$ 44,00.

Cadastrar Pedido - Windows Internet Explorer
https://enb.com.br
Arquivo Editar Exibir Favoritos Ferramentas Ajuda
Favoritos
Cadastrar Pedido

Página Inicial | Listar Pedidos

CADASTRAR PEDIDO

Razão Social do Cliente: FRANCISCO DE SOUZA PIRES

Vendedor: Lorena

Condição de Pagamento: CHEQUE 45 DIAS

Tabela de Preço: PÁG - A VISTA

Operação: VENDA

Opção: T

Observação: O cliente receberá 45 caixas de vodka. Assim, o valor por caixa ficará por R\$ 44,00.

Adicionar Pedidos

4ª atividade – Parte C: Cadastrar um pedido

Adicionar Pedidos

→ Após ter inserido as informações gerais sobre o pedido, você definirá os produtos do pedido.

→ Defina a quantidade (número

Adicionar Pedidos

Cadastrar Pedido

Página Inicial > Listar Pedidos

Usuário: **lorena** (desconectar)

CADASTRAR PEDIDO

Razão Social do Cliente: DIEGO MORAIS SILVA SANTOS

Pedido	Código	Descrição	Quantidade	Valor Unitário
	0301003	CARANGUEJO PET 480ML PT 12		
	0301005	GOSTOSA PET 480ML PT 12		
	0301004	CARANGUEJO 600ML CX 12 SD		
	0301007	CARANGUEJO 600ML GR 24 CD		
	0301008	CARANGUEJO OURO 900ML CX 6 SD		
	0301009	CARANGUEJO OURO 900ML CX 1 SD		
	0301010	CARANGUEJO 980ML CX 12 SD		
	0301012	CARANGUEJO 980ML CX 12 CD		
	0301015	CARANGUEJO 980ML GR 12 CD		
	0301024	CARANGUEJO PRATA 350ML PT 08		
	0301030	CARANGUEJO LIMÃO 490ML PT 12		
	0301020	GOSTOSA 980ML CX 12 SD		
	0301023	GOSTOSA 980ML GR 12 CD		

6ª atividade: Listar todos os seus pedidos

- Após selecionar **Cadastro de Pedidos** (4ª atividade), clique em **Listar Pedidos** no menu superior da página.
- Essa área também é protegida. Então, coloque novamente sua senha, quando for solicitado.
- Depois de colocar o login e a senha corretamente, a listagem dos pedidos realizados por você será exibida.

Exemplo: Listagem dos pedidos realizados pela vendedora Lorena Lira.

Data do Pedido	Cliente	Vendedor	
29-05-2009	30350899420 - BAR E RESTAURANTE RIACHAO LTDA	2 - Lorena	Visualizar Informações
20-05-2009	12-0659426 - CARLOS ALEXANDRE DA SILVA	2 - Lorena	Visualizar Informações

7ª atividade: Visualizar os dados de um pedido

➔ Após listar todos os pedidos (6ª atividade), se você quer visualizar as suas informações, clique em [Visualizar Pedido](#) na linha do pedido que você deseja visualizar.

Exemplo: Visualização de um pedido realizado pela vendedora Lorena Lira.

Visualizar Pedido - Windows Internet Explorer

https://enb.com.br

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Favoritos

Visualizar Pedido

Usuário: **lorena (desconectar)**

Página Inicial Cadastrar Pedido Listar Pedidos

VISUALIZAR PEDIDO

Data do Pedido: 20-05-2009

Cliente: 92359899426 - BAR E RESTAURANTE RIACHAO LTDA

Vendedor: 2 - Lorena

Condição de Pagamento: CHEQUE 14DD

Tabela de Frete: FFA - 01 0129

Operação: VENDA

Observação: Teste teste

SubPedidos:

Produto	Quantidade	Valor Unitario	SubTotal
0301009 - CARANGUEJO CURIO SEMILIX 1 ED	10	R\$ 20,00	R\$ 20,00
0302010 - KIT VODKA PALOFF + FANTA CLTS	120	R\$ 1,00	R\$ 120,00
Valor Total			R\$ 120,00

Editar Remove Imprimir esta página

Concluído Internet 100%

Cadastro de Clientes

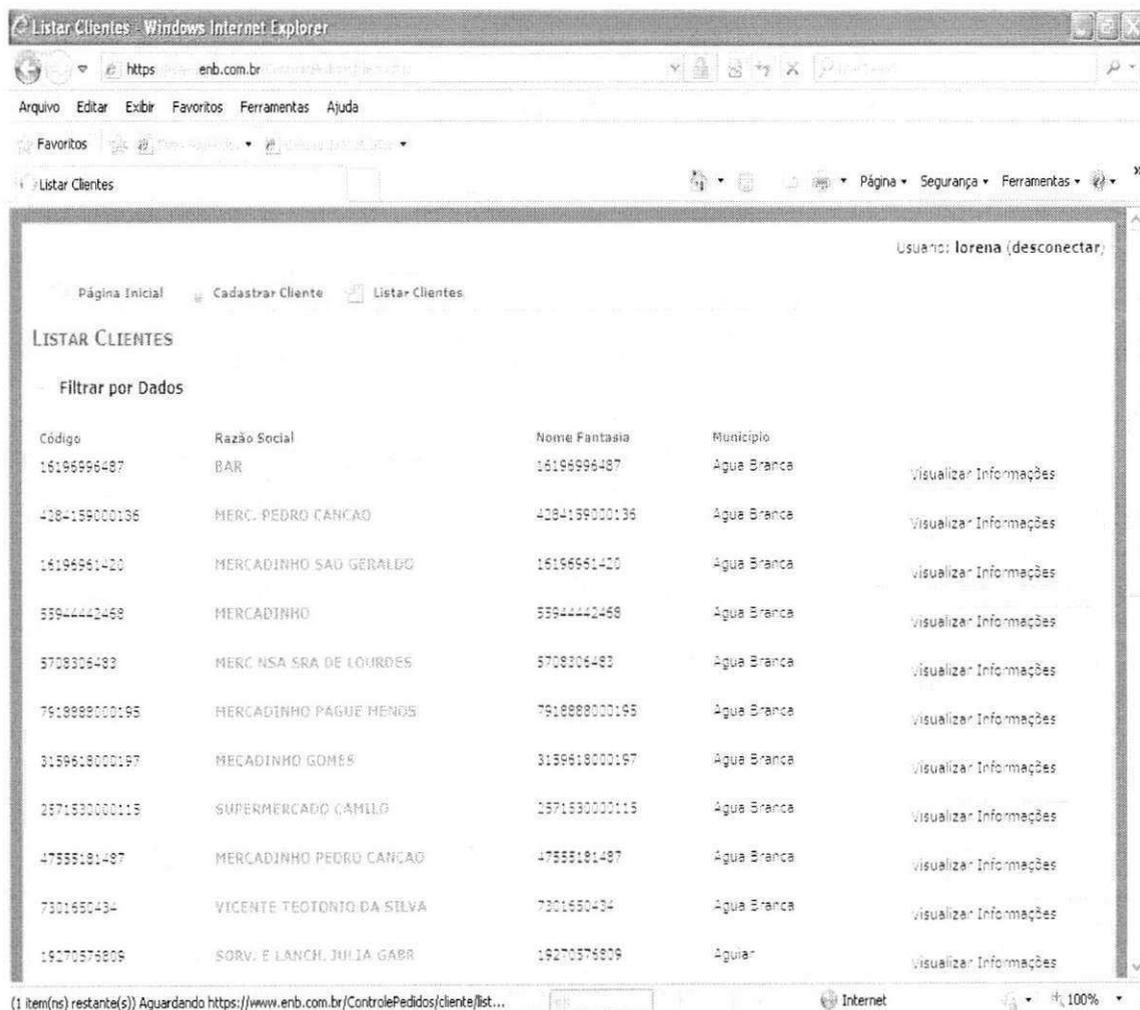
Parte do sistema responsável por visualizar um cliente e listar todos os clientes cadastrados no sistema.

8ª atividade: Cadastro de clientes

→ Na sua página inicial (3ª atividade), você pode visualizar a lista de clientes clicando em **Cadastro de Clientes**.

→ Veja que estão listadas as principais informações de cada cliente: **Código do cliente no sistema, Razão Social, Nome Fantasia, Município**.

Exemplo:



The screenshot shows a web browser window titled "Listar Clientes - Windows Internet Explorer". The address bar shows "https://www.enb.com.br/ControlePedidos/cliente/list...". The browser interface includes a menu bar (Arquivo, Editar, Exibir, Favoritos, Ferramentas, Ajuda) and a toolbar. The main content area displays a table of clients with the following columns: Código, Razão Social, Nome Fantasia, Município, and a link to "Visualizar Informações".

Código	Razão Social	Nome Fantasia	Município	Visualizar Informações
16196996487	BAR	16196996487	Água Branca	Visualizar Informações
4284159000136	MERC. PEDRO CANCAO	4284159000136	Água Branca	Visualizar Informações
16196961420	MERCADINHO SAO GERALDO	16196961420	Água Branca	Visualizar Informações
55944442468	MERCADINHO	55944442468	Água Branca	Visualizar Informações
5708306483	MERC NSA SRA DE LOURDES	5708306483	Água Branca	Visualizar Informações
7918888000195	MERCADINHO PAGUE MENOS	7918888000195	Água Branca	Visualizar Informações
3159618000197	MERCADINHO GOMES	3159618000197	Água Branca	Visualizar Informações
2571530000115	SUPERMERCADO CAMILO	2571530000115	Água Branca	Visualizar Informações
47555181487	MERCADINHO PEDRO CANCAO	47555181487	Água Branca	Visualizar Informações
7301650434	VICENTE TEOTONIO DA SILVA	7301650434	Água Branca	Visualizar Informações
19270576809	SORV. E LANCH. JULIA GARR	19270576809	Aguiar	Visualizar Informações

At the bottom of the browser window, there is a status bar showing "(1 item(ns) restante(s)) Aguardando https://www.enb.com.br/ControlePedidos/cliente/list...", a search box, and the text "Internet" and "100%".

9ª atividade: Visualizar os dados de um cliente

- Após selecionar [Visualizar Cliente](#) (8ª atividade), a listagem de clientes previamente cadastrados no sistema será exibida.
- Para visualizar os dados de um cliente, selecione o cliente clicando em [Visualizar](#) na linha que contém o cadastro do cliente que você deseja visualizar.

Exemplo: Visualização de um cliente que apareceu na listagem.

Visualizar Cliente - Windows Internet Explorer

https://enb.com.br

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Favoritos

Visualizar Cliente

Usuário: lorena (desconectar)

Página Inicial Cadastrar Cliente Listar Clientes

VISUALIZAR CLIENTE

Código	16196996487
Razão Social	BRR
Nome Fantasia	16196996487
Inscrição Estadual	16196996487
CNPJ	16196996487
Endereço	RUA ODON FLORENCIO S/N
Bairro	RUA ODON FLORENCIO S/N
Município	Agua Branca
Estado	PB
Telefone	

Editar Remove

Concluído Internet 100%

10ª atividade: Desconectar-se do sistema

- Para sair do sistema, clique no nome **Desconectar**, em qualquer página.
- Lembre-se que você deverá desconectar-se sempre que acabar de usar o sistema.

Exemplo: **Desconectar** a partir de uma visualização de pedido.

The screenshot shows a web browser window titled 'Visualizar Pedido: Windows Internet Explorer'. The address bar shows 'https://enb.com.br'. The page content includes a navigation menu with 'Página Inicial', 'Cadastrar Pedido', and 'Listar Pedidos'. The main section is titled 'VISUALIZAR PEDIDO' and displays the following details:

Data do Pedido: 20-05-2009
Cliente: 02050099420- BAR E RESTAURANTE RIACHAO LTDA
Vendedor: 2 - Lorena
Condição de Pagamento: CHEQUE 14DD
Tabela de Freço: PAA + 21 DIAS
Operação: VENDA
Observação: Teste teste

Sub-Pedidos:	Produto	Quantidade	Valor Unitário	SubTotal
	0301009 - CARANGUEJO GURU 900ML CX 1 SD	10	R\$ 0,00	R\$ 0,00
	0302010 - KIT VODKA BALOFF + PANTA 2LTS	120	R\$ 1,00	R\$ 120,00
			Valor Total	R\$ 120,00

At the bottom of the page, there are buttons for 'Editar', 'Remover', and 'Imprimir esta página'. The user name 'lorena (desconectar)' is circled in the top right corner of the page content.

Exercícios

Criar Pedidos no Sistema de Gestão de Pedidos

PEDIDO 01

Vendedor: Você!

Data: **22/05/2009**

Cliente: **Francisco de Sousa Pires**

Após negociação com o senhor Francisco, ficou definido o que ele deseja: a quantidade e o preço por unidade:

Desejo do Cliente	Quantidade	Valor Unitário
Vodka Paloff 980ML CX 12 SD	150	R\$ 57,20

Condição de pagamento negociada:

Condição de Pagamento
45 dias no Cheque

Algumas observações foram feitas:

Observações feitas
Bonificação: O cliente receberá 45 caixas de vodka. Dessa forma, o valor por caixa ficará por R\$ 44,00.

PEDIDO 02

Vendedor: Você!

Data: **22/05/2009**

Cliente: **Luiz Guedes**

Após negociação com o senhor Luiz Guedes, ficou definido o que ele deseja: a quantidade e o preço por unidade:

Desejo do Cliente	Quantidade	Valor Unitário
Caranguejo 980ML cx 6 SD	05	R\$ 41,44
Vinho Mão Santa 4,6LT	20	R\$ 11,52
RON DCuba Cristal 980ML cx 12 SD	07	R\$ 78,65
RON DCuba Limon 980ML cx 12 SD	03	R\$ 76,15

Condição de pagamento negociada:

Condição de Pagamento
28 dias no Boleto

Algumas observações foram feitas:

Observações feitas
Bonificação:
O cliente receberá 02 caixas de Ron Cristal.

Apêndice F

Sondagem de Satisfação do Usuário

1. Uma nota para o sistema (0-10): _____

2. O que achou do sistema?

- Fácil de usar
- Razoavelmente fácil de usar
- Razoavelmente difícil de usar
- Difícil de usar

3. Foi fácil cadastrar um pedido?

- Sim
- Não, por que?

4. Dê uma nota ao treinamento (0 -10): _____

5. Fique a vontade para deixar sugestões/críticas que possam permitir facilitar a utilização do sistema:
