

UNIVERSIDADE FEDERAL DO PARANÁ

HENRIQUE JOSÉ ZEFERINO
MARCOS PAULINO DA ROCHA
MARIA DANIELA CAPURRO
RICARDO JULIANI
WILLIAM VIEIRA

SISTEMA SEPELNET: SISTEMA DE PREGÃO ELETRÔNICO

CURITIBA

2007

HENRIQUE JOSÉ ZEFERINO
MARCOS PAULINO DA ROCHA
MARIA DANIELA CAPURRO
RICARDO JULIANI
WILLIAM VIEIRA

SISTEMA SEPELNET: SISTEMA DE PREGÃO ELETRÔNICO

Trabalho de conclusão de curso de graduação
apresentado à disciplina de Projetos do Curso
Superior de Tecnologia em Informática, Setor
Escola Técnica da Universidade Federal do
Paraná.

Orientador: Prof. Dr. Mauro José Belli

CURITIBA
2007

TERMO DE APROVAÇÃO

HENRIQUE JOSÉ ZEFERINO
MARCOS PAULINO DA ROCHA
MARIA DANIELA CAPURRO
RICARDO JULIANI
WILLIAM VIEIRA

SISTEMA SPELNET: SISTEMA DE PREGÃO ELETRÔNICO

Trabalho de Conclusão de Curso aprovado como requisito para obtenção da graduação no Curso Superior de Tecnologia em Informática, Setor Escola Técnica da Universidade Federal do Paraná, pela seguinte banca examinadora:

Orientador:

Prof. Dr. Mauro José Belli
Departamento de Tecnologia, UFPR

Prof. Dieval Guizelini
Departamento de Tecnologia, UFPR

Prof^a MsC. Sandramara S. K. de Paula Soares
Departamento de Tecnologia, UFPR

Curitiba, 14 de Dezembro de 2007

RESUMO

Este documento apresenta todo o processo de desenvolvimento de software aplicado ao Projeto de Conclusão de Curso, direcionado a atender o cliente pessoa física Claudinei Vieira Grittem.

Foi feita uma análise sobre a necessidade do cliente para agilizar o processo de compras de empresas do setor alimentício visando automatizar alguns processos manuais. Existem controles sobre o processo de compras, entretanto são controles manuais ou baseados em planilhas de cálculos, que possuem uma menor segurança da informação, tanto no aspecto de acesso, rastreabilidade e disponibilidade.

Este documento tem foco principal abordar toda a metodologia empregada durante o desenvolvimento de um projeto, desde a definição do cliente, elaboração de um plano de projeto, do entendimento dos objetivos do sistema a ser desenvolvido, até as etapas subseqüentes de análise, implementação e documentação.

Para o desenvolvimento do sistema foi utilizada a linguagem Java, por se tratar de uma tecnologia atual e multi-plataforma, assim como o banco de dados PostgreSQL. Para o gerenciamento de versões foi utilizado o Concurrent Version System (CVS). Por fim é apresentado o resultado do desenvolvimento, ou seja, o produto final, e as possíveis implementações futuras que foram identificadas.

Palavras-chave: Pregão eletrônico. Compra on-line. Java. UML. Comércio Eletrônico. Alimentação.

ABSTRACT

This document all presents the process of development of software applied to the Project of Conclusion of Course, directed to take care of to the customer natural person Claudinei Vieira Grittem. An analysis on the necessity of a system was made that supplied the necessity to speed the process of purchases of companies of the nourishing sector in specifics, automating some manual processes.

Controls exist on the process of purchases, however are manual or the controls are based in spread sheets of calculations, which have a minor security of the information, as much in the access aspect, track-ability and availability. As main focus, the methodology used during the development of the project was boarded all, since the acquisition of the customer, elaboration of a project plan, the agreement of the objectives of the system to be developed, until the subsequent stages of analysis, implementation and documentation.

For the development of the system it was used language JAVA, for if dealing with a current technology and multiple-platform, as well as the data base PostgreSQL. For the management of versions was used Concurrent Version System (CVS).

Finally it is presented the result of the development, or either, the end item, and the possible future implementations that had been identified.

Key words: E-Procurement. Purchase online. Java. UML. Eletronic Commerce. Feeding

LISTA DE SIGLAS

AJAX	- Asynchronous Javascript And XML
API	- Application Programming Interface
CASE	- Computer-Aided Software Engineering
CREA-PR	- Conselho Regional de Engenharia e Arquitetura do Paraná
CVS	- Concurrent Version System
DER	- Diagrama de Entidade Relacionamento
HTML	- HyperText Markup Language
IDE	- Integrated Development Environment
JSON	- JavaScript Object Notation
JSP	- Java Server Pages
JUDE	- Java and UML Developer's Environment
LGP	- Livro Geral do Projeto
MVC	- Modeo View Controller
PGP	- Plano de Gerenciamento de Projetos
RUP	- Rational Unified Process
SQL	- Structured Query Language
UML	- Unified Modeling Language
WBS	- Work Breakdown Structure

SUMÁRIO

1 INTRODUÇÃO	7
2 ESPECIFICAÇÃO DO PROJETO	8
2.1 CARACTERIZAÇÃO DO CLIENTE	8
2.2 CENÁRIO ATUAL	8
2.3 SOLUÇÃO PROPOSTA	8
3 METODOLOGIA PARA O DESENVOLVIMENTO DO PROJETO	10
3.1 FASES DA METODOLOGIA	10
3.1.1 Levantamento de requisitos	10
3.1.2 Análise e projeto	11
3.1.3 Implementação	12
3.1.4 Teste	12
3.1.5 Implantação	12
3.2 GERENCIAMENTO DO PROCESSO DE SOFTWARE	12
3.3 TÉCNICAS E FERRAMENTAS PARA MODELAGEM FUNCIONAL E NÃO-FUNCIONAL	13
3.4 TÉCNICAS E FERRAMENTA PARA MODELAGEM DE DADOS	15
3.5 FERRAMENTAS PARA IMPLEMENTAÇÃO	15
3.5.1 Requisitos determinados pelo cliente	15
3.5.2 Métodos de trabalho adotados pela equipe	15
3.6 TÉCNICAS E FERRAMENTAS PARA PROCESSO DE HOMOLOGAÇÃO	16
3.6.1 Homologação interna	16
3.6.2 Homologação pelo cliente	16
4 PROCESSO DE DESENVOLVIMENTO DO SISTEMA SPELNET	17
4.1 APLICAÇÃO DA METODOLOGIA ADOTADA	17
4.2 APLICAÇÃO DO GERENCIAMENTO DO PROCESSO DE SOFTWARE	18
4.3 APLICAÇÃO DAS TÉCNICAS E FERRAMENTAS PARA MODELAGEM FUNCIONAL E NÃO-FUNCIONAL	19
4.4 APLICAÇÃO DAS TÉCNICAS DE MODELAGEM DE DADOS	19
4.5 APLICAÇÃO DAS FERRAMENTAS DE IMPLEMENTAÇÃO	20
4.6 APLICAÇÃO DAS TÉCNICAS DE HOMOLOGAÇÃO	21
4.6.1 Homologação interna	21
4.6.2 Homologação pelo cliente	22
5 TRABALHOS FUTUROS	23
6 CONCLUSÃO	24
REFERÊNCIAS	25
APÊNDICES	26

INTRODUÇÃO

O processo de compras é complexo na maioria das empresas, sendo um procedimento lento, podendo envolver muitas pessoas, gerar altos custos operacionais e, muitas vezes, são perdidas boas opções de compra ou comprados produtos com qualidade deficiente.

Para as empresas a automatização desse processo proporcionará a redução dos fatores citados acima e até mesmo sua eliminação, além de favorecer as tomadas de decisões, pois as informações serão analisadas de forma mais rápida e precisa.

O sistema será gerenciado pelas empresas, entretanto os fornecedores terão acesso a ele para realizar lances e, também serão beneficiados com a divulgação dos seus produtos e serviços.

Portanto, para todos os envolvidos, o Spelnet irá gerar uma economia de recursos no processo de compra/venda, controle informatizado e maior transparência no processo.

2 ESPECIFICAÇÃO DO PROJETO

2.2 CARACTERIZAÇÃO DO CLIENTE

O Sr. Claudinei Vieira Grittem, possui formação em Administração de Empresas e há muitos anos trabalha na concepção de sistemas voltados para a área de alimentação.

Nestes anos, acumulou muita experiência no setor e verificou a necessidade de um sistema específico para compras on-line, e essa necessidade acabou dando origem ao Spelnet.

2.2 CENÁRIO ATUAL

Perante a necessidade de efetuar compras, muitas empresas do setor de prestação de serviços em alimentação ainda trabalham utilizando métodos tradicionais, onde, a partir da sua lista de compras, efetuam suas negociações com os fornecedores via telefone, fax e/ou e-mail.

Este procedimento é lento, pode envolver muitas pessoas, gera altos custos operacionais e, muitas vezes, são perdidas boas opções de compra ou comprados produtos com qualidade deficiente.

Já para o fornecedor, o Spelnet facilitará na divulgação dos seus produtos e serviços. Já para o comprador, permitirá uma maior rapidez e aumento no número de ofertas recebidas. Portanto, para ambos, o sistema irá gerar uma economia de recursos no processo de compra/venda, além de um controle informatizado.

2.3 SOLUÇÃO PROPOSTA

Analisando a situação atual, decidiu-se pelo desenvolvimento de uma aplicação informatizada para prover meios que facilitem o processo de negociação na hora da compra, tanto para o comprador quanto para o fornecedor, focando empresas privadas prestadores de serviços de alimentação.

Tendo como base o artigo publicado na Revista CREA-PR Economia e Agilidade (2007, p.34) sobre o Sistema de Pregão Eletrônico Público, implantado pelo governo do Estado do Paraná, estimamos que o Spelnet possa gerar uma

diferença de 15% entre o preço de referência e o de aquisição após a disputa entre os fornecedores, além de um aumento de 25% no leque de ofertas recebidas de fornecedores.

Espera-se que através da utilização do Spelnet o processo de compra, das empresas prestadoras de serviços do setor de alimentação, seja agilizado, e as necessidades das empresas no que diz respeito à quantidade, qualidade e preço sejam formalizadas. E ainda, que o fornecedor, parte ativa do processo, possa acompanhar claramente suas oportunidades de negócios.

3 METODOLOGIA PARA O DESENVOLVIMENTO DO PROJETO

A metodologia de desenvolvimento abrange todo o ciclo de vida do projeto incluindo as regras criadas, os padrões adotados, as ferramentas utilizadas, e todos os pontos relevantes para a definição do processo de desenvolvimento completo.

No projeto Spelnet optou-se por utilizar o *Rational Unified Process (RUP)*, por se tratar de uma metodologia com etapas e objetivos bem definidos. Para a *Rational Software Corporation*, “[...] Ele oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis”.

O *RUP* não foi aplicado em sua plenitude, por se tratar de uma metodologia bastante extensa e com grande número de artefatos gerados, leiam-se documentos, relatórios ou diagramas. Alguns deles foram considerados desnecessários, fazendo uso somente dos relevantes e fundamentais.

Baseando-se no *RUP* e filtrando os seus principais aspectos, foi desenvolvido o seguinte modelo, executado como metodologia de desenvolvimento:

- a) Levantamento de requisitos;
- b) Análise e projeto;
- c) Implementação;
- d) Teste;
- e) Implantação.

3.1 FASES DA METODOLOGIA

3.1.1 Levantamento de Requisitos

Nesta fase são listados todos os requisitos do sistema, baseando-se na solicitação do cliente, dando origem ao documento de requisitos. Através desta etapa as funcionalidades e os dados necessários s listados e descritos, proporcionando uma base para o desenvolvimento de seu detalhamento e o avanço para as próximas fases.

A elaboração inicial dos casos de uso negociais para permitir o entendimento dos objetivos e do escopo do sistema pelas partes envolvidas ocorre neste momento, onde as regras de negócio são bem definidas e os casos de uso são priorizados.

Sendo aprovados os casos de uso negociais, a próxima fase pode ser iniciada.

3.1.2 Análise e Projeto

Com os casos de uso negociais finalizados, iniciou-se o detalhamento dos casos de uso, diagrama de seqüência e diagrama de classes.

Seguindo as recomendações da UML, a partir do documento de levantamento de requisitos, foram especificados os casos de uso, que contém os *data views*, atores, pré-condições, pós-condições, fluxos principais, alternativos e de exceções, além das regras de negócios.

Com o auxílio de ferramentas CASE foi montado o diagrama de seqüência que descreve a maneira como os grupos de objetos colaboram em algum comportamento ao longo do tempo.

Em seguida desenvolvemos o diagrama de classes, que se propõe a demonstrar a estrutura das classes, com seus atributos, métodos e relações existentes entre as classes do sistema.

Em se tratando de arquitetura de software, o padrão adotado foi *Model View Controller* (MVC), utilizando na prática o modelo em três camadas. Estas foram especificadas em camada de apresentação ou interface com o usuário, camada de negócio e camada de acesso a dados. A camada de apresentação é a responsável pelas validações de tela e apresentação de dados ao usuário. A de acesso a dados realiza a comunicação direta com os dados armazenados no banco de dados. A de negócio realiza a comunicação entre essas duas camadas, recebendo as requisições da interface, realizando todo o processamento lógico do sistema. Ela entrega e busca informações da camada de acesso a dados e, tratando-as conforme seja necessário, devolve os dados para a camada de interface. A camada de interface nunca realiza solicitações diretas à camada de acesso a dados, e vice-versa.

3.1.3 Implementação

Esta fase corresponde à elaboração e preparação dos módulos necessários à execução do sistema, baseando-se na documentação gerada nas fases anteriores.

Nesta fase foram aplicados conhecimentos sobre as regras de negócio especificadas, e feita a modularização do sistema para possibilitar a melhor divisão de tarefas, visando também facilitar a futura integração entre os módulos desenvolvidos.

Foram analisados os casos de uso e os diagramas detalhadamente, e foi implementado o que estava especificado. Quando da ocorrência de inconsistências na documentação, o assunto foi discutido e, quando necessário, a documentação foi alterada para se adaptar a real necessidade identificada.

3.1.4 Testes

A metodologia foi dividir os testes do sistema entre os membros da equipe, cada um testando o módulo do outro membro e documentando as falhas encontradas.

Antes de colocar o sistema em produção, uma equipe ou um grupo restrito de usuários deve realizar os testes do cliente, visando garantir o correto funcionamento do sistema e também verificar se o produto final foi o que o cliente esperava.

3.1.5 Implantação

Finalmente, o sistema deve ser implantado no cliente, levando em conta o ambiente necessário e os pré-requisitos definidos inicialmente para seu correto funcionamento.

3.2 GERENCIAMENTO DO PROCESSO DE SOFTWARE

“Um processo de software é uma seqüência coerente de práticas que objetiva o desenvolvimento ou evolução de sistemas de software. Estas práticas

englobam as atividades de especificação, projeto, implementação, testes e caracterizam-se pela interação de ferramentas, pessoas e métodos”.

Pressman (2006, p.52) afirma que “os processos de software devem ter uma forte gerência, através de planos detalhados, estimativas, medições, avaliações e controle”.

As atividades técnicas - análise e projeto, algoritmos, estruturação de dados, linguagens etc. - devem ser tratadas com métodos, técnicas e ferramentas específicos, mas a gerência do processo como um todo precisa se utilizar de conceitos e métodos idênticos aos de outras áreas: a definição de objetivos claros para a organização, o comprometimento da alta administração, o envolvimento e o preparo das pessoas, a sincronização de ações em uma mesma direção. As ferramentas para avaliação e controle gerencial também são as mesmas.

O principal objetivo de gerenciar o processo de software é obter qualidade no produto final. Um processo com qualidade resulta em um produto com qualidade.

Segundo Pressman (2006, p.52), qualidade de software se define em “Conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente definido”.

A gerência do processo visa estabelecer uma infra-estrutura para suportar e guiar os trabalhos dos diversos projetos de maneira uniforme. A gerência inclui:

- a) Definição do processo: estabelecer um padrão para implementação, avaliação e melhoria de cada tarefa;
- b) Execução do processo: definir os métodos e técnicas usadas para produzir produtos com qualidade;
- c) Coleta de dados e análise: tratar as medições realizadas dos produtos e processos de software, e o uso desses dados;
- d) Controle do processo: estabelecer mecanismos para certificar o desempenho do processo definido; monitorar e ajustar o processo onde melhorias forem necessárias.

3.3 TÉCNICAS E FERRAMENTAS PARA MODELAGEM FUNCIONAL E NÃO-FUNCIONAL

Primeiramente, deve-se entender o que é um requisito funcional e um requisito não-funcional. Segundo Wazlawick (2004, p.102), “um requisito funcional é o que o sistema deve fazer, e os requisitos não funcionais são as restrições sobre como o sistema deve desempenhar suas funções”. Ainda segundo Wazlawick (2004, p.102), “a modelagem funcional especifica as funções externas do sistema, e descreve as Operações do Sistema como entradas, e as Consultas do Sistema como saídas”.

Visando atender às necessidades da modelagem funcional, recomenda-se o desenvolvimento de diagramas da *Unified Modeling Language (UML)*, por possuir diagramas padronizados que facilitam o entendimento do desenvolvedor do sistema, tanto na fase de implantação como para manutenções futuras. Segundo definição de Guedes:

A UML (Unified Modeling Language) é uma Linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos. Essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem de software adotada internacionalmente pela indústria de Engenharia de Software (...) Seu objetivo é auxiliar os engenheiros de software a definir as características do software, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e até mesmo suas necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implantado. Todas essas características são definidas por meio da UML antes do software começar a ser realmente desenvolvido. (GUEDES; 2004, p.22).

A modelagem não-funcional do sistema é descrita nos casos de uso através das regras de negócio, que descrevem restrições e regras propriamente dita. A modelagem não-funcional pode definir os requisitos mínimos para utilizar o sistema, tanto em nível de ambiente, como em nível e conhecimento que os usuários deverão possuir sobre a utilização do mesmo.

De acordo com Guedes (2004, p.27), “O diagrama de classes é o diagrama mais utilizado e o mais importante da *UML*, servindo de apoio para a maioria dos outros diagramas”. Sobre o diagrama de seqüência, Guedes em sua publicação afirma que:

O diagrama de seqüência preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um caso de uso definido pelo diagrama

de mesmo nome e apóia-se no diagrama de classes para determinar os objetos das classes envolvidas em um processo. (GUEDES; 2004, p.28).

3.4 TÉCNICAS E FERRAMENTAS PARA MODELAGEM DE DADOS

A modelagem de dados visa preparar uma estrutura para a montagem do banco de dados. Nela são definidas as tabelas, chaves primárias, relacionamentos e todos os itens necessários para uma correta representação dos dados.

Segundo a enciclopédia on-line Wikipédia, a abordagem da modelagem de dados:

Atende a três perspectivas: Modelagem Conceitual, Modelagem Lógica e Modelagem Física. A primeira é usada como representação de alto nível e considera exclusivamente o ponto de vista do usuário criador do dado, a segunda já agrega alguns detalhes de implementação e a terceira demonstra como os dados são fisicamente armazenados.

Quanto ao objetivo, podemos identificar as seguintes variações: modelagem de dados entidade-relacionamento (leitura, construção e validação dos modelos); modelagem de relacionamentos complexos, grupos de dados lógicos e ciclo de vida das entidades; modelagem de dados corporativa; modelagem de dados distribuídos (cliente/servidor); modelagem e re-engenharia de dados legados e modelagem de dados para Data Warehouse.

3.5 FERRAMENTAS PARA IMPLEMENTAÇÃO

3.5.1 Requisitos determinados pelo cliente

Preferências ou exigências estipuladas pelo cliente referente às ferramentas que deverão ser utilizadas para desenvolver o projeto, tendo em vista a viabilidade do uso das mesmas.

3.5.2 Métodos de trabalho adotado pela equipe

Seguindo as solicitações de requisitos de ferramentas do cliente, avaliando os conhecimentos da equipe e analisando o software a ser desenvolvido, a equipe deverá determinar as melhores ferramentas para utilização no projeto, buscando, principalmente, qualidade e produtividade.

3.6 TÉCNICAS E FERRAMENTAS PARA PROCESSO DE HOMOLOGAÇÃO

3.6.1 Homologação interna

Homologação interna de sistema são os testes realizados pela equipe responsável pelo projeto, onde são testadas as funcionalidades desenvolvidas. Esses testes devem levar em conta características básicas de funcionamento, assim como verificar se as regras de negócios e a lógica do sistema estão implementadas de acordo com a documentação.

3.6.2 Homologação pelo cliente

É a fase aonde o cliente irá efetuar os testes do sistema, verificando se o produto desenvolvido atende suas expectativas. A partir dessa fase tem-se a solicitação de ajustes e o aceite final do cliente.

4 PROCESSO DE DESENVOLVIMENTO DO SISTEMA SPELNET

4.1 APLICAÇÃO DA METODOLOGIA ADOTADA

A metodologia de desenvolvimento do projeto Spelnet foi seguida em todas as suas etapas, porém com alguns desvios durante o processo.

A fase de levantamento de requisitos foi realizada sem maiores dificuldades, a partir de reuniões com o cliente.

Após esse levantamento de requisitos foram elaborados os casos de uso em sua versão inicial. Nesses casos de usos, foi adotada a sistemática de adicionar as telas de protótipos, que tinha por objetivo facilitar a programação e o entendimento do cliente.

Através da análise mais específica juntamente com o cliente desses casos de uso, constatou-se algumas divergências de entendimento das regras de negócio, o que ocasionou um ajuste de alguns itens dos casos de uso e do modelo de dados inicial.

A diagramação do sistema também sofreu impacto, já que os casos de uso foram alterados. Os diagramas iniciais não representavam a real implementação que seria feita no sistema, já que a equipe optou pelo desenvolvimento em três camadas, sendo elas a camada de apresentação, negócio e acesso a dados. Com isso, os diagramas do sistema também foram modificados, visando representar de forma mais próxima possível a implementação.

Durante a implementação do sistema, chegou-se à conclusão de que alguns itens ainda ficaram pendentes nos casos de uso e nos diagramas, bem como na modelagem dos dados. Esta constatação ocasionou numa nova reformulação de alguns itens, para que a documentação ficasse condizente com o produto final e com as necessidades estipuladas pelo cliente.

Os testes do sistema foram realizados sem dificuldades, utilizando os casos de testes desenvolvidos pela equipe, e após cada ciclo de testes o responsável corrigia as falhas encontradas, devolvia os resultados e se reiniciava o ciclo até que todos os problemas fossem solucionados.

Durante todo o processo, foram realizadas reuniões formais registradas no nosso Livro Geral do Projeto, em forma de ata. Nele eram documentadas as

discussões e definições tomadas para a seqüência do projeto, sendo que ao final da reunião todos assinavam.

4.2 APLICAÇÃO DO GERENCIAMENTO DO PROCESSO DE SOFTWARE

No projeto Spelnet o gerenciamento do processo de desenvolvimento foi definido no Plano de Gerenciamento de Projetos (*PGP*), disponível no apêndice 1, sendo que nele estão detalhados os objetivos, e as ferramentas utilizadas para atingi-los.

Foi feito o uso de métodos auxiliares, como o *Work Breakdown Structure (WBS)* para auxílio no desenvolvimento do cronograma do projeto. Também foi realizado o levantamento de riscos e sua medição de impacto, com o objetivo de minimizar ou extinguir possíveis problemas durante o andamento do projeto.

E para o controle como um todo foi utilizado o *Microsoft Project 2003*, onde todas as tarefas foram relacionadas com intenção de controlar as atividades dos componentes da equipe, bem como a definição de datas de entregas de atividades, visando o acompanhamento do cumprimento do cronograma estipulado.

Apesar de todo esse controle e do cumprimento do prazo previsto, durante o desenvolvimento do projeto o cronograma estipulado não foi cumprido, devido à concretização de riscos levantados no *PGP*, como:

- Falhas na Análise de Requisitos: não foi possível manter um dialogo constante com o cliente, muitas vezes tendo resolver duvidas por e-mail, o que ocasionaram alguns atrasos.
- Problemas com o desempenho técnico no decorrer do projeto: um risco que acreditávamos ser baixo se tornou médio, pois o conhecimento técnico equipe não se mostrou adequando no inicio do desenvolvimento. Isso acarretou atraso na programação.

Fatores que aumentaram a complexidade do processo foram o controle de acesso ao sistema, e o uso do *Asynchronous Javascript And XML (AJAX)*, que é o uso sistemático de tecnologias providas por navegadores, como *Javascript*, *JavaScript Object Notation (JSON)* e *XML*, para tornar páginas mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações.

4.3 TÉCNICAS E FERRAMENTAS PARA MODELAGEM FUNCIONAL E NÃO-FUNCIONAL

Como ferramenta para o desenvolvimento da modelagem funcional, optou-se pelo *Java and UML Developer's Environment (JUDE)*, pelo fato de ser uma ferramenta que atende à *UML 2.0*, bem estruturada, que possibilita a interação entre os diagramas criados e a melhor organização de todos os diagramas que possam existir no sistema, além de ser um software livre.

A seqüência temporal do desenvolvimento da modelagem iniciou-se com o levantamento dos requisitos, especificação dos casos de uso, modelo de dados, e para atender as necessidades da modelagem funcional foi desenvolvido também o diagrama de seqüência, o diagrama de classes e o diagrama de estado para a classe pregão. Nos diagramas de casos de uso estão presentes as pré-condições, pós-condições, fluxo principal, fluxos alternativos e as exceções.

As pré-condições do sistema definiram o que deve existir ou estar presente para que determinada funcionalidade do sistema seja executada corretamente. As pós-condições são as características que o sistema apresentará após a execução do caso de uso. O fluxo principal especifica o que normalmente acontece quando o caso de uso é executado, enquanto os fluxos alternativos e de exceção descrevem comportamentos opcionais ou excepcionais, respectivamente.

Os diagramas confeccionados durante esta fase podem ser verificados no apêndice 4, que apresenta o diagrama de casos de uso, apêndice 5 com o diagrama de seqüência, apêndice 6 que contém o diagrama de classes, além do diagrama de estado para a classe pregão no apêndice 7.

4.4 APLICAÇÃO DAS TÉCNICAS E FERRAMENTAS DE MODELAGEM DE DADOS

A ferramenta utilizada para a modelagem de dados foi o *Microsoft Visio*, por ser de fácil utilização, atender todas as necessidades e permitir a geração de scripts para a futura criação do banco de dados. É importa ressaltar também o uso da ferramenta *microOLAP Database Designer for PostgreSQL* no início do processo. Entretanto, devido a maior facilidade e conhecimento do uso do *Visio*, utilizamos o *microOLAP* mais em caráter de aprendizado.

Com relação à modelagem, buscou-se realizar a modelagem completa antes do início do desenvolvimento, para o aproveitamento do recurso da ferramenta de geração automática dos scripts do banco. Porém durante o desenvolvimento alguns campos e tabelas foram alterados, de acordo com as necessidades verificadas, o que fez com que após o término da construção do sistema fosse gerada uma nova versão do Diagrama de Entidade Relacionamento (*DER*).

Como artefatos produzidos nessa fase do projeto têm o *DER*, disponível no apêndice 8, o Dicionário de dados, que segue no apêndice 9 e o script do banco no apêndice 10.

4.5 APLICAÇÃO DAS FERRAMENTAS DE IMPLEMENTAÇÃO

Referente às ferramentas de implementação, o cliente deixou totalmente em aberto a utilização dessas ferramentas, apenas comentando a sua preferência por ferramentas open source.

Seguindo a preferência do cliente, e também da equipe, optamos por utilizar a linguagem Java para o desenvolvimento do projeto, assim como o desenvolvimento voltado a internet através do uso de *Java Server Pages (JSP)*. Entre as maiores vantagens de utilizar Java, encontramos:

- i. É multi-plataforma;
- ii. É uma arquitetura aberta, extensível, com várias implementações, o que a torna independente do fornecedor.
- iii. É uma linguagem poderosa, rodeada de *Application Programming Interface (API's)* eficazes e completas.
- iv. É uma linguagem descomprometida, aceite inclusive nos meios universitários como uma boa linguagem para a aprendizagem, o que facilita o recrutamento de técnicos.
- v. As tecnologias Java são desenvolvidas e apoiadas por dezenas de empresas, dispensando assim os seus utilizadores da dependência de um único fornecedor. Este fato traz enormes vantagens, quer em termos econômicos, quer de qualidade de produtos e serviços.

Para o desenvolvimento, foi definido o Eclipse Europa como *Integrated Development Environment (IDE)*, por ser uma ferramenta de maior conhecimento da equipe e ser open source.

Durante o desenvolvimento existiram algumas situações que causaram atraso no desenvolvimento, sendo, em nossa opinião, a pouca experiência da equipe em trabalhar com orientação a objeto a que mais influenciou.

O método de trabalho utilizando controle de versão mostrou-se muito vantajoso por centralizar as alterações e possibilitar que, sempre que necessário, o programador resgatasse a versão mais recente da classe desejada.

Na utilização da *Structured Query Language (SQL)*, existiam 2 métodos que atenderiam a necessidade do projeto. Um deles seria a utilização da sintaxe desenvolvida diretamente na ferramenta, já no outro haveria modularização dos códigos gerados, utilizando-se funções (*stored procedures*). Optou-se pela utilização dos dois métodos, procurando minimizar a utilização de funções.

4.6 APLICAÇÃO DAS TÉCNICAS DE HOMOLOGAÇÃO

4.6.1 Homologação interna

Durante toda a etapa de desenvolvimento, cada programador realizou os testes iniciais no módulo desenvolvido (teste unitário). Após o término do processo, foram preparados os casos de testes por funcionalidade. A elaboração dos casos de testes foi realizada buscando atender os seguintes requisitos:

- a) Os parâmetros estipulados nos casos de uso deveriam ser observados, tanto nos fluxos principais quanto alternativos;
- b) Verificação da implementação das regras de negócio;
- c) Verificações nos dados alterados no banco.

Foram realizados dois ciclos de testes internos com base nos casos de testes, alternando-se os responsáveis pelos testes entre os membros da equipe, garantindo-se que o desenvolvedor não realizasse os testes no código-fonte gerado por ele. Os ajustes e um último ciclo de testes foram realizados, tendo todos os itens sido atendidos conforme esperado.

Como produtos dessa fase têm os casos de testes por funcionalidade do sistema, que seguem no apêndice 11.

4.6.2 Homologação pelo cliente

O cliente iniciou os testes do sistema em outubro deste ano, à medida que a equipe de desenvolvimento disponibilizava os casos de uso. Devido à falta de estrutura do cliente, os testes iniciais foram realizados em nosso servidor de páginas internet. Estimamos que os testes sejam finalizados até o final de fevereiro do próximo ano. A homologação será validada com uma carta de aceite do cliente assim como a disponibilização do ambiente para instalação e configuração do sistema em produção.

Tendo ciência da necessidade da entrega do projeto, o cliente elaborou uma declaração de intenção de uso do sistema, disponível no apêndice 3.

5 TRABALHOS FUTUROS

Durante o planejamento e a execução do projeto Spelnet, surgiram algumas possibilidades de novas funcionalidades ou uma nova forma de implementar as funcionalidades já estipuladas.

Como o planejamento já havia sido realizado, as funcionalidades e suas implementações definidas e aprovadas, não houve a possibilidade de implementar tais funcionalidades. Por este motivo, planejou-se que, havendo uma nova versão, estas funções seriam avaliadas pelo cliente para inclusão ou modificação.

Entre as possibilidades de implementações futuras, encontram-se:

- a) Substituição de itens nas ordens de compras;
- b) Criação do help on-line do sistema;
- c) Relatórios mais específicos;
- d) Generalização do módulo de pregão, para atender a outros segmentos empresariais.
- e) Interrupção temporária do pregão.
- f) Contratos entre as partes envolvidas;

6 CONCLUSÃO

O Sistema de Pregão Eletrônico colabora para o bom andamento dos trabalhos de uma organização ao agilizar o processo de compras das empresas, garantido uma melhor eficiência no serviço.

O processo de compra, das empresas prestadoras de serviços do setor de alimentação, será agilizado, e as necessidades das empresas no que diz respeito à quantidade, qualidade e preço serão formalizadas. E ainda, o fornecedor, parte ativa do processo, poderá acompanhar com transparência suas oportunidades de negócios.

A inserção da equipe em todo este processo trouxe aos membros menos experientes uma nova visão do mercado de desenvolvimento de software e a experiência muito enriquecedora do trabalho em equipe, tanto do ponto de vista das vantagens quanto das desvantagens. Já aos mais experientes agregaram-se conhecimentos específicos de metodologia e ferramentas de desenvolvimento.

REFERÊNCIAS

DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**. 6. ed. Porto Alegre: Bookman, 2006.

GUEDES, G. T. A. **UML: Uma abordagem pratica**. 1 ed. São Paulo:Novatec, 2004. p. 22, p.27, p.28.

WIKIPÉDIA. **Modelagem de Dados**. Disponível em: <
http://pt.wikipedia.org/wiki/Modelagem_de_dados/> Acesso em 10/10/2007.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. Rio de Janeiro: McGraw-Hill Interamericana do Brasil, 2006. p.52, p.65.

Processo de Software. Disponível em: <
http://pt.wikipedia.org/wiki/Engenharia_de_software#Processo_de_Software>
Acesso em 02/11/2007.

RATIONAL SOFTWARE CORPORATION. **Rational Unified Process: Visão Geral**. Disponível em: <<http://www.wthreex.com/rup/>> Acesso em 16/10/2007.

WAZLAWICK, R. S. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. 1 ed. São Paulo: Campus, 2004. p.102.

APÊNDICES

APÊNDICE 1 – PLANO DE GERENCIAMENTO DE PROJETOS	27
APÊNDICE 2 – ADENDO AO PLANO DE GERENCIAMENTO DO PROJETO	38
APÊNDICE 3 – DECLARAÇÃO DE INTENÇÃO DE UTILIZAÇÃO	40
APÊNDICE 4 – DIAGRAMAS DE CASOS DE USO	42
APÊNDICE 5 – DIAGRAMAS DE SEQUÊNCIA.....	44
APÊNDICE 6 – DIAGRAMA DE CLASSES	45
APÊNDICE 7 – DIAGRAMA DE ESTADO – CLASSE PREGÃO.....	47
APÊNDICE 8 – DIAGRAMAS DE ENTIDADE-RELACIONAMENTO.....	49
APÊNDICE 9 – DICIONÁRIO DE DADOS	51
APÊNDICE 10 – SCRIPTS DO BANCO	60
APÊNDICE 11 – CASOS DE TETES	61
APÊNDICE 12 – CÓDIGOS-FONTE DO APLICATIVO	62
APÊNDICE 13 – DIAGRAMA DE COMPONENTES	63

APÊNDICE 1 – PLANO DE GERENCIAMENTO DE PROJETOS

1 SITUAÇÃO ATUAL

Perante a necessidade de efetuar compras, muitas empresas do setor de prestação de serviços em alimentação ainda trabalham utilizando métodos tradicionais, onde, a partir da sua lista de compras, efetuam suas negociações com os fornecedores via telefone, fax e/ou e-mail.

Este procedimento é lento, pode envolver muitas pessoas, gera altos custos operacionais e, muitas vezes, são perdidas boas opções de compra ou comprados produtos com qualidade deficiente.

Para o fornecedor, o Spelnet facilitará na divulgação dos seus produtos e serviços. Já para o comprador, o Spelnet permitirá uma maior rapidez e aumento no número de ofertas recebidas. Portanto, para ambos, o Spelnet irá gerar uma economia de recursos no processo de compra/venda, além de um controle informatizado.

2 OBJETIVO

Analisando a situação atual, decidiu-se pelo desenvolvimento de uma aplicação informatizada para prover meios que facilitem o processo de negociação na hora da compra, tanto para o comprador quanto para o fornecedor, focando empresas privadas prestadores de serviços de alimentação.

Tendo como base o artigo publicado na Revista CREA-PR Economia e Agilidade (2007, p.34) sobre o Sistema de Pregão Eletrônico Público, implantado pelo governo do Estado do Paraná, estimamos que o Spelnet possa gerar uma diferença de 15% entre o preço de referência e o de aquisição após a disputa entre os fornecedores, além de um aumento de 25% no leque de ofertas recebidas de fornecedores.

Espera-se que através da utilização do Spelnet o processo de compra, das empresas prestadoras de serviços do setor de alimentação, seja agilizado, e as necessidades das empresas no que diz respeito à quantidade, qualidade e preço sejam formalizadas. E ainda, que o fornecedor, parte ativa do processo, possa acompanhar claramente suas oportunidades de negócios.

3 ESCOPO

Após reunião com o cliente, definiram-se as necessidades/funcionalidades que o Spelnet deverá suprir e desempenhar. A decisão mais relevante é a utilização de uma base de dados única para o Spelnet, que permita padronizar os dados do sistema para todos os usuários.

No Spelnet, os compradores poderão montar um pregão com sua lista de produtos, selecionar os fornecedores e convidá-los a participarem. Por outro lado, os fornecedores acessarão os pregões em aberto e poderão dar lances de preços.

O Spelnet enviará automaticamente convites com as informações necessárias para os fornecedores cadastrados e habilitados a participarem do pregão. Os pregões terão seu período de duração previamente estipulados, entretanto ultrapassado o tempo mínimo de duração, o comprador pode encerrá-lo a qualquer momento.

Durante a realização do pregão, o Spelnet permitirá que os participantes troquem informações publicamente, ou seja, se algum fornecedor precisar tirar uma dúvida, ele poderá fazê-lo no seu decorrer, e os outros que estiverem participando terão total acesso a essas informações, conferindo total transparência ao processo.

A partir da conclusão do pregão, o Spelnet permitirá a emissão de ordens de compra ou relatórios necessários para a documentação da transação.

Serão contempladas no sistema informações adicionais e optativas, como a taxa de variação do lance, observações gerais com especificações extras e, após o total encerramento do processo, ou seja, o fechamento do pregão e entrega dos produtos, existirá possibilidade do comprador realizar uma avaliação da qualidade do atendimento do fornecedor.

Visando permitir uma maior integração dos participantes, a aplicação estará disponível em ambiente Web. Para o controle de acesso, os compradores e fornecedores, possuirão um usuário e uma senha de identificação que lhes permitirá acesso ao sistema.

4 ANÁLISE DE REQUISITOS

Após reuniões com o cliente, foram levantados os seguintes requisitos:

- 1) O Spelnet contará com um banco de dados único. Constarão no banco de dados os cadastros em geral, que terão como principais funcionalidades:

- i. Cadastro de Usuários: Perfil de usuário e permissões (pregoeiro, fornecedor, administrador).
 - ii. Cadastro de Clientes: Dados básicos dos clientes (compradores e fornecedores).
 - iii. Cadastro de itens (Produtos): Dados dos itens a serem ofertadas nos pregões. O sistema vai permitir propostas para inclusão de produtos para usuários com perfil “pregoeiro” e “fornecedor”, que deverão ser homologadas pelos usuários com perfil “administrador”. Somente após esta homologação os produtos estarão disponíveis para uso nos pregões.
 - iv. Cadastro de pregões: Dados dos pregões.
 - v. Cadastro de Categorias: Categorias de itens. O nível inicial contém duas categorias principais “perecíveis” e “não perecíveis”. Dentro destes níveis poderão ser criadas subcategorias de produtos.
 - vi. Unidades de Medida: Unidade de medida dos itens (quilograma, cento, litro, unidade, metro, milheiro). Será fixo.
 - vii. Cadastro de Embalagens: Embalagens dos itens com fator de conversão para uso em relatórios.
 - viii. Lances: Vai armazenar os lances dos fornecedores, por pregão, com o preço unitário e o código do fornecedor.
 - ix. Ordens de compra: Vão armazenar os dados do fornecedor ganhador por pregão.
 - x. Avaliação: Vai armazenar as avaliações dos compradores sob os fornecedores.
- 2) O sistema não registrará dados relativos à entrega dos produtos cotados, uma vez que as empresas trabalharão com o conceito de “Central de Distribuição” (CD), ou seja, as compras serão entregues em um local e a partir dali, o próprio cliente realizará a distribuição.
- 3) Quando um usuário com perfil “pregoeiro” acessar o sistema, ele terá liberada a opção de “CRIAR PREGÃO”. Deverão ser fornecidos os seguintes dados ao criar um pregão:
- i. Data do pregão
 - ii. Hora de início do pregão
 - iii. Tempo mínimo de duração do pregão.
 - iv. Lista de itens, embalagens, quantidade e preço unitário base (este último

- não obrigatório).
- v. Lista de fornecedores cadastrados. O sistema vai apresentar a lista de fornecedores habilitados para participar do pregão, o pregoeiro, se necessário, poderá excluir da lista algum fornecedor desqualificando-o.
 - vi. Taxa de variação do lance (não obrigatório)
 - vii. Observações gerais. Nesta opção o pregoeiro informará dados adicionais como: validade de produtos, data máxima para entrega, e outros detalhes que considerar necessário esclarecer aos fornecedores participantes.
- 4) O cadastro do pregão poderá ser dividido em etapas, facilitando a recuperação de dados, ou ainda poderão ser utilizados modelos (templates).
 - 5) Após a confirmação do pregoeiro, o Spelnet enviará por e-mail convites a título de edital para os fornecedores habilitados a participar do processo e para os demais usuários do sistema com o perfil pregoeiro, pertencentes à empresa responsável do pregão.
 - 6) O Spelnet vinculará ao pregão um código único de identificação e realizará a sua abertura de forma automática na data e hora informada pelo pregoeiro.
 - 7) Após atingido o seu tempo mínimo de duração, o pregão poderá ser encerrado pelo comprador a qualquer momento.
 - 8) O Spelnet permitirá a consulta “on the fly” dos lances por produto, tanto de usuários com perfil “fornecedor” como “pregoeiros”. Os fornecedores serão identificados por um código que irá preservar sua identidade até a finalização do pregão.
 - 9) O sistema não vai permitir o acesso a um pregão de dois usuários com perfil “fornecedor” pertencentes à mesma empresa.
 - 10) Durante o pregão, o Spelnet disponibilizará um chat para que os participantes possam sanar dúvidas. Todas as conversações serão públicas.
 - 11) Após a finalização do pregão, estará disponível aos usuários do Spelnet a classificação final do pregão ordenada por lance, do mais baixo ao mais alto, com o nome verdadeiro dos fornecedores.
 - 12) Será enviado um aviso, mensagem eletrônica ou SMS, ao(s) fornecedor(es) vencedor (es) com a ordem de compra dos itens, os preços do lance, o local e data de entrega e as observações gerais.
 - 13) O Spelnet disponibilizará consultas e relatórios referentes a pregões anteriores.

14)O sistema contará com um módulo de acompanhamento da Qualidade (opcional) onde poderão ser realizadas avaliações de fornecedores nos quesitos prazo de entrega e qualidade do produto.

5 ORGANIZAÇÃO DO PESSOAL

5.1 ESTRUTURA DA EQUIPE

5.1.1 GERENTE DO PROJETO: WILLIAM

Responsável por delegar e cobrar tarefas relativas ao andamento de projetos. Foi escolhido pela própria equipe e acompanhará os trabalhos, os atrasos e os problemas que ocorrerem no projeto.

5.1.2 ADMINISTRADOR DE BANCO DE DADOS (DBA): WILLIAM / RICARDO

Responsável pelo servidor do banco de dados, no âmbito do desenvolvimento. É o profissional destinado a ajudar na otimização de consultas SQL, na implementação de recursos para aceleração de processos a respeito da estrutura do banco de dados, tais como criação de índices ou confecção de stored procedures. Políticas de backup, segurança de acesso no nível de banco, instalação e configuração dos parâmetros operacionais do banco são algumas das tarefas do DBA.

5.1.3 SEGURANÇA E REDES: RICARDO

Responsável pelo planejamento e controle operacional do servidor, que realizará a instalação e configuração de hardwares e serviços necessários, manutenção e backup. Cuidará da segurança, gerenciando acessos, senhas e firewalls. Deverá zelar pela integridade e disponibilidade do servidor e suas conexões.

5.1.4 ANALISTA DE NEGÓCIOS: WILLIAM / RICARDO

Profissional com maior contato com o cliente e também com maior conhecimento na área em que o software será desenvolvido.

5.1.5 ANALISTA DE SISTEMAS: DANIELA / MARCOS

Responderá pelos levantamentos dos requisitos e modelagem em alto nível do sistema, com o uso de linguagens como a UML.

5.1.6 PROGRAMADOR: MARCOS / DANIELA / WILLIAM

Profissional responsável pela codificação dos sistemas. Seu trabalho está intimamente ligado à utilização de ferramentas de desenvolvimento.

5.1.7 HOMOLOGADOR: HENRIQUE

Profissional com a tarefa de testar o software antes que ele seja disponibilizado para o cliente.

A homologação final será efetuada pelo cliente, sendo possível sua homologação via e-mail.

5.1.8 DOCUMENTADOR: HENRIQUE

Responsável pela documentação do software, criação de manuais de operação e treinamento do software, sistema de ajuda on-line do software e atualização da documentação existente quando da criação de novas features ou quando da modificação da forma de operação do software.

5.1.9 ANALISTA DE SUPORTE: WILLIAM

É o profissional responsável por resolver dúvidas surgidas no ambiente de operação do cliente.

Essa distinção de funções é apenas para determinar responsabilidades, sendo que no desenvolvimento do projeto, todos estarão envolvidos em todas as áreas.

5.2 RELATÓRIOS ADMINISTRATIVOS

5.2.1 RELATÓRIO DE REUNIÕES

Livro ATA utilizado para documentar as reuniões e suas respectivas decisões.

6 ANÁLISE DE RISCOS

Identificação	Probabil.	Ações Preventivas	Ações Corretivas
1. Falha na estimativa do cronograma	Média	Planejar cronograma detalhadamente.	Readequar o cronograma.
2. Imprevistos com o Cliente	Baixa	Manter diálogos constantes.	Prosseguir com os conhecimentos adquiridos pela equipe e encontrar outro cliente.
3. Imprevistos com um membro da equipe	Baixa	Procurar fazer um acompanhamento de pessoal.	Redistribuir as tarefas através da revisão do cronograma.
4. Problemas com o Servidor	Baixa	Seguir com rigor o Plano de contingência.	Restaurar o backup e solucionar o problema original.
5. Problemas de conexão com a internet.	Baixa	Seguir com rigor o Plano de contingência.	Entrar em contato com o provedor para resolver o problema. Caso o problema persista, trocar de provedor de internet banda larga.
6. Falhas na Análise de Requisitos	Baixo	Manter diálogos constantes com o cliente.	Revisar requisitos com o cliente e readequar o cronograma.
7. Problemas com o desempenho técnico no decorrer do projeto	Baixo	Planejar a distribuição das tarefas de forma adequada.	Realizar reuniões e treinamentos para aprimorar os conhecimentos e resolver os problemas técnicos.
8. Falha na segurança do Servidor	Médio	Atualizar o sistema constantemente.	Reconfiguração de mecanismos de segurança.

7 CRONOGRAMA

7.1 WBS

Será utilizado para facilitar o entendimento do projeto, pois este nos permite uma melhor visualização para a definição de prazos para o projeto.

O WBS encontra-se disponível no cd-rom.

7.2 GANTT

Permitirá ter uma visão geral do planejamento e possibilitará um acompanhamento mais eficaz do projeto.

A lista de tarefas do gráfico de Gantt encontra-se disponível no cd-rom.

7.2 PERT

Será utilizado para mostrar os inter-relacionamentos e interdependências entre as tarefas. A partir do PERT, estaremos visualizando o caminho crítico, que são as seqüências de etapas que mais consomem tempo até a conclusão do projeto.

O diagrama PERT encontra-se disponível no cd-rom.

8 RECURSOS DO PROJETO

8.1 PESSOAL

Alunos: Henrique Zeferino
Marcos Paulino
Maria Daniela Capurro
Ricardo Juliani Ferreira
William Vieira

Professor Orientador: Mauro José Belli
Cliente do Projeto: Claudinei V. Grittem

8.2 HARDWARE E SOFTWARE

8.2.1 HARDWARE

- Servidor:
 - Processador: 1.5 GHz;
 - RAM: 256 MB;

- Hds: 2 Ata 80 GB com Raid 0 via software;
- Interfaces de rede: 10/100 Mb/s;
- Conexão com a internet (para o servidor): 300 Kb/s;
- Impressora: Jato de tinta;
- Estações de trabalho (05 computadores pessoais):
- Processadores: entre 2.5 Ghz à 3.0 Ghz ;
- RAM: entre 512 MB à 1 GB;
- Hds: entre 80 GB à 120 GB;

8.2.2 SOFTWARE

- Servidor
 - Sistema Operacional Ubuntu 6.06 (distribuição linux baseada em Debian, com vários componentes de controle de segurança pré-instalados, e freqüentes atualizações). A opção pelo software livre foi feita pelos seguintes motivos:
 - Versão LTS (*Long Time Support*), com suporte até 2011;
 - Bom suporte para aplicações WEB;
 - Projeto sem custos com aquisição de licenças de softwares proprietários;
 - Desenvolvimento de uma cultura anti-pirataria;
 - Ótimas configurações de segurança, incluindo firewall.
 - Banco de dados: PostgreSQL 8.1:
 - Java;
 - Firewall: Netfilter, gerenciado através do Iptables;
 - Servidor HTTP/Container JSP: Apache TomCat;
 - Controle de Versionamento: CVS.
- Estações de trabalho
 - Sistema Operacional: Ubuntu 6.10 ou Windows Xp Sp2;
 - Ferramenta de desenvolvimento: Eclipse 3.2;
 - DB Designer 4;
 - Jude;
 - MicroOLAP;

- Microsoft Visio;
- Dia;
- PG Admin
- Suíte Office e OpenOffice;
- Linguagem de Programação: Java / JSP;
- Acompanhamento de Projeto: MS Project.

8.3 RECURSOS ESPECIAIS

- É necessária a previsão de recursos financeiros para:
 - Conexão ADSL para o servidor;
 - Hds de backup do servidor;
 - Impressões, encadernamentos;

9 MÉTODOS DE ACOMPANHAMENTO DO PROJETO

- Acompanhamento do Cronograma
 - Constantes verificações/ajustes no cronograma através do MS Project e/ou software similar. Esse acompanhamento deixará claro o andamento do processo com base no cronograma descrito nesse plano de projeto.

APÊNDICE 2 – ADENDO AO PLANO DE GERENCIAMENTO DO PROJETO

1 ADENDO AO ITEM 4, SUBITEM 10 DO PGP

Nesse item é mencionado a criação da funcionalidade chat no sistema :

- Durante o pregão, o SPELNET disponibilizará um chat para que os participantes possam sanar dúvidas. Todas as conversações serão públicas.

Entretanto, após análise entre a equipe e o cliente foi decidido sua alteração de chat para fórum. Isso aconteceu tendo em vista o tempo de resposta das duvidas, pois com o chat entende-se por uma conversa continua, enquanto o fórum é para esclarecer dúvidas, onde podem não serem respondidas imediatamente.

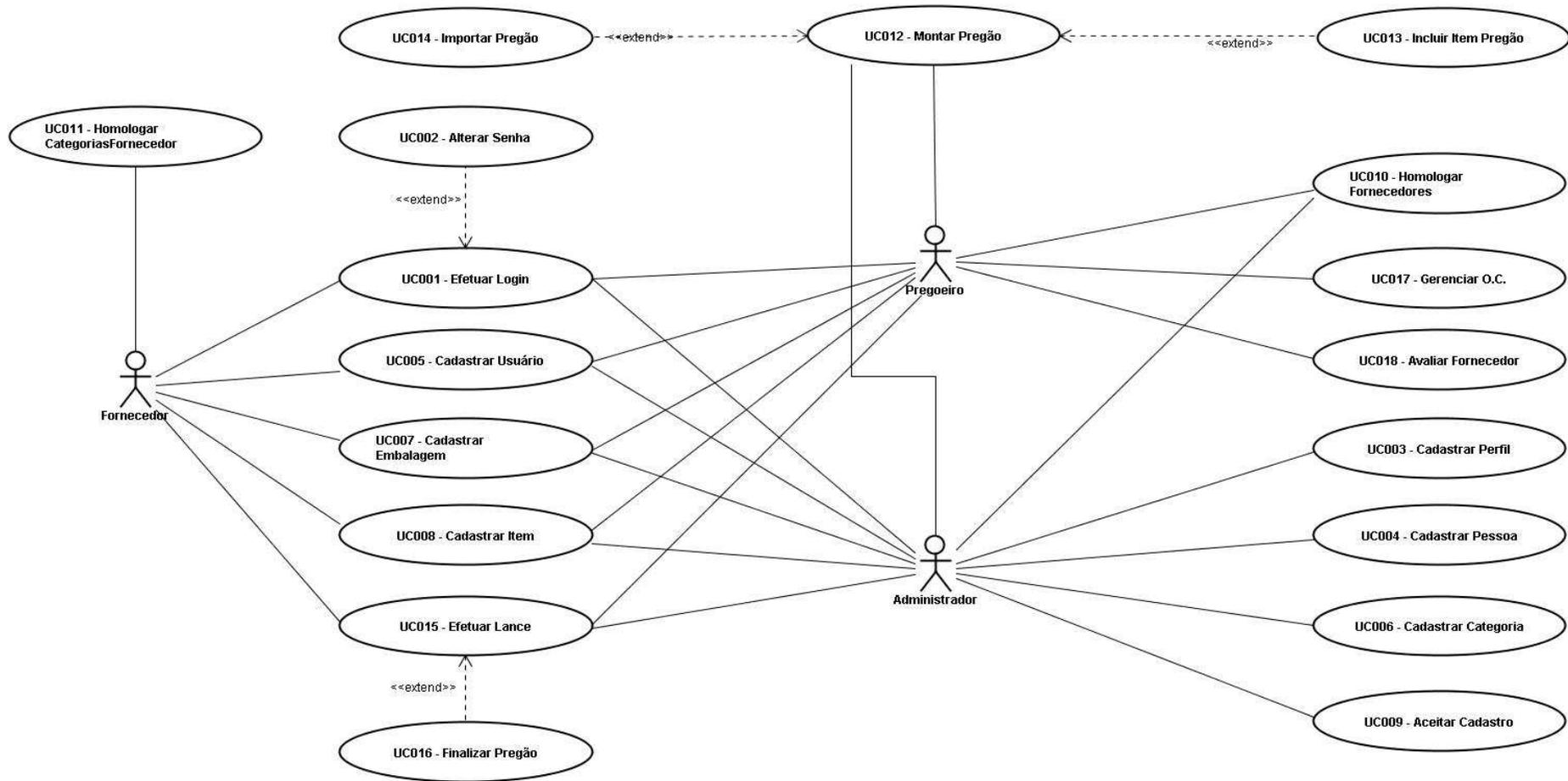
APÊNDICE 3 – DECLARAÇÃO DE INTENÇÃO DE UTILIZAÇÃO

1 DECLARAÇÃO DE INTENÇÃO DE UTILIZAÇÃO

O cliente CLAUDINEI VIEIRA GRITTEM, pessoa física, CPF 727.552.109-15, residente a Rua Atílio Druszc, 491, município de Araucária, após a realização dos testes e da homologação do sistema, através deste documento declara sua intenção de utilização do sistema Spelnet.

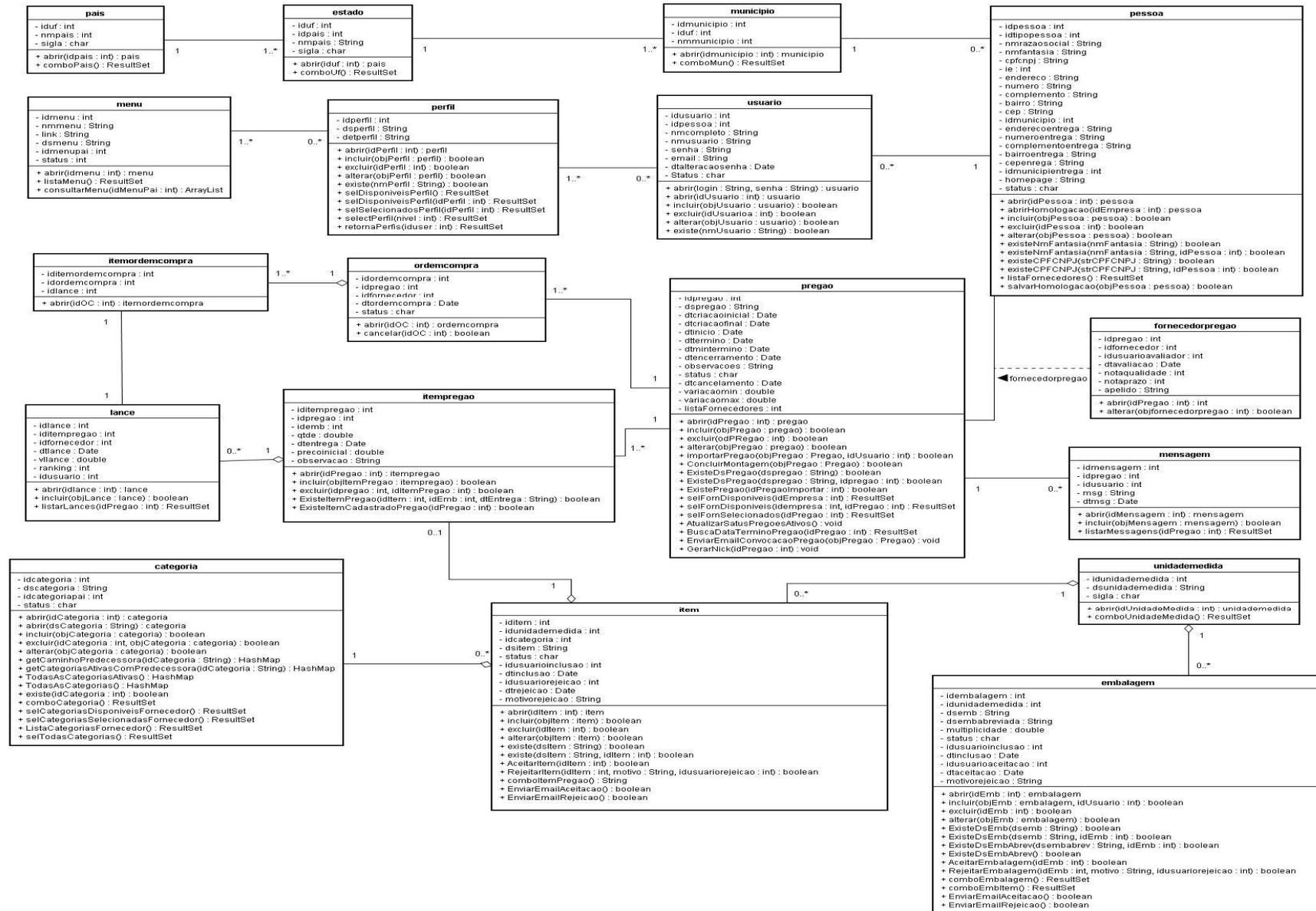
Curitiba, 01 de Novembro de 2007.

APÊNDICE 4 – DIAGRAMA DE CASO DE USO



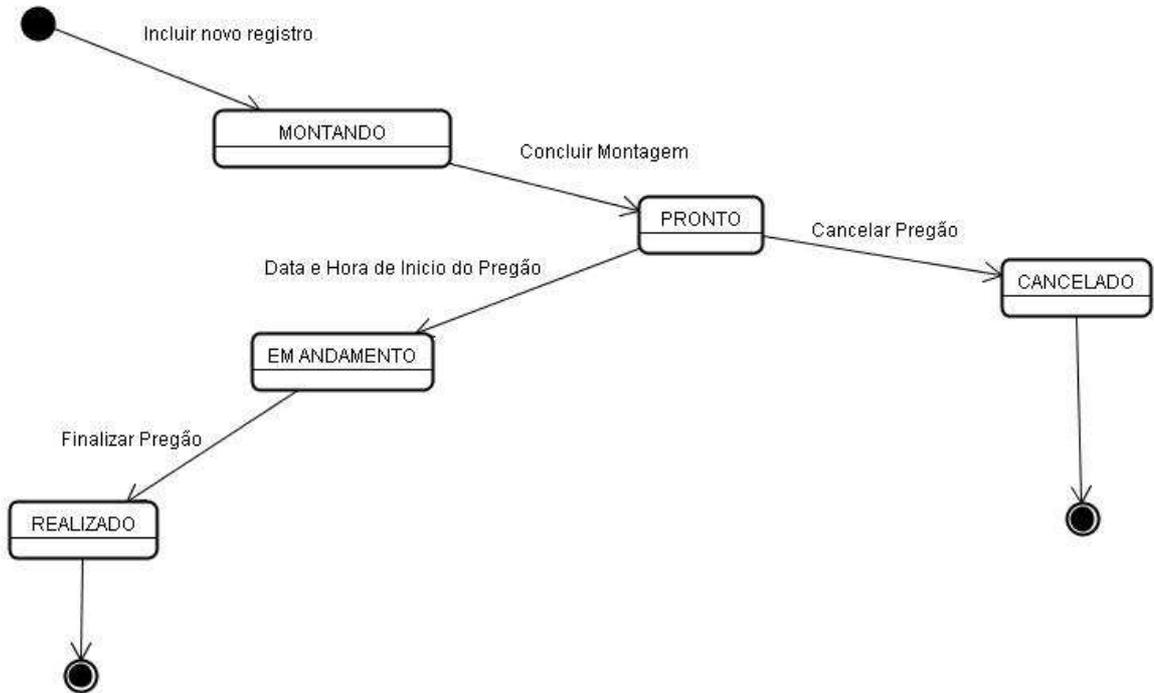
APÊNDICE 5 – DIAGRAMAS DE SEQUÊNCIA
DISPONÍVEL NO CD – PASTA “APÊNDICES/DIAGRAMAS DE SEQUÊNCIA”

APÊNDICE 6 – DIAGRAMA DE CLASSES

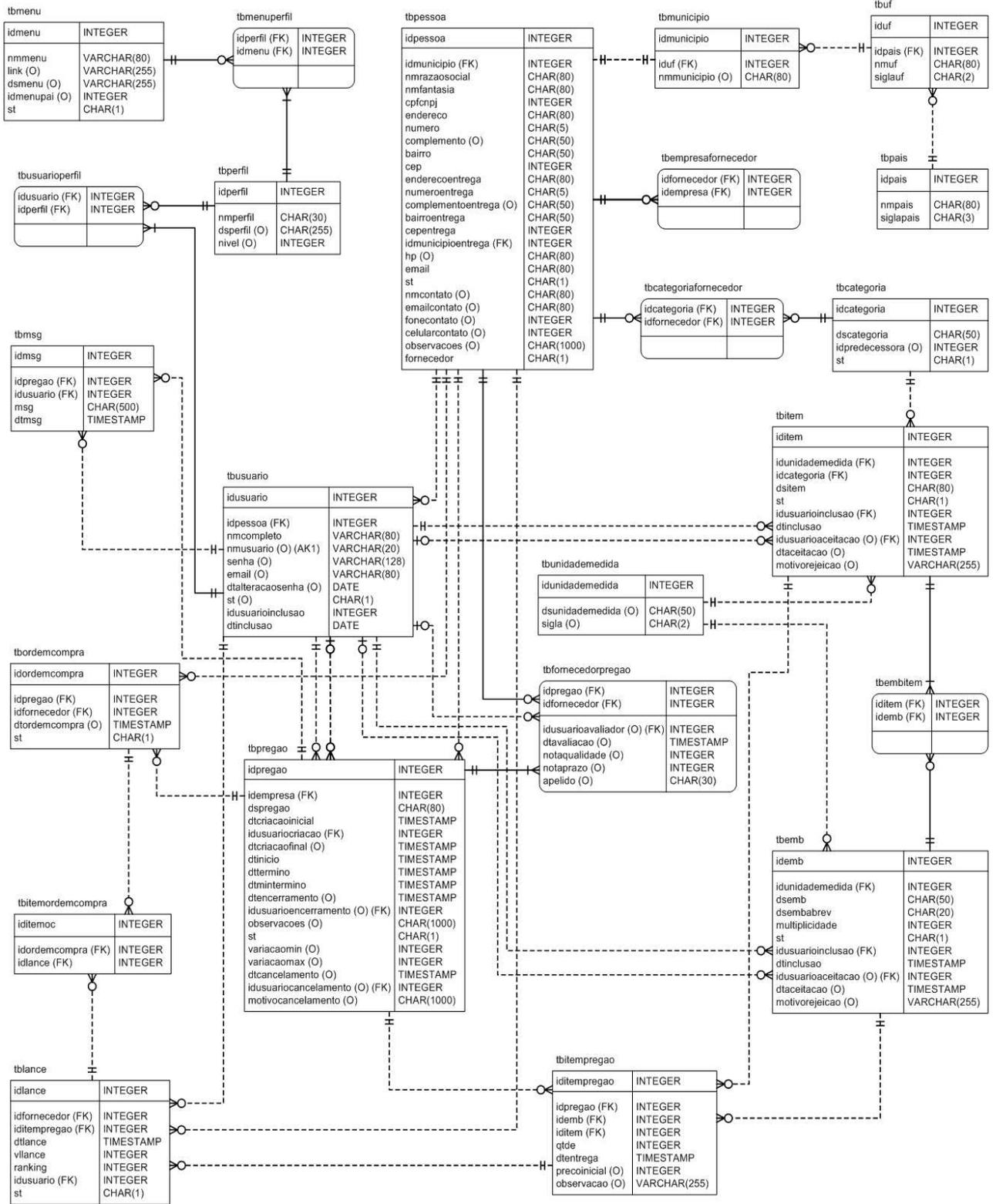


APÊNDICE 7 – DIAGRAMA DE ESTADO – CLASSE PREGÃO

DIAGRAMA DE ESTADO PARA A CLASSE PREGÃO



APÊNDICE 8 – DIAGRAMA DE ENTIDADE-RELACIONAMENTO



- H- : exatamente um.
- o : zero ou um.
- | : um ou mais.
- o| : zero ou mais.

APÊNDICE 9 – DICIONÁRIO DE DADOS

PK: Primary Key
FK: Foreign Key

Tabela: tbpais

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idpais	integer		Not null	ID do país
	nmpais	char	80	Not null	Nome do país
	siglapais	char	3	Not null	Sigla do país

Tabela: tbuf

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	iduf	integer		Not null	ID da Unidade da Federação (Estado)
FK	idpais	integer		Not null	ID do país que pertence o Estado
	nmuf	char	80	Not null	Nome do Estado
	siglauf	char	2	Not null	Sigla do Estado

Tabela: tbmunicipio

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idmunicipio	integer		Not null	ID do Município
FK	iduf	integer		Not null	ID do estado que pertence o Município
	nmmunicipio	char	80	Not null	Nome do Município

Tabela: tbpessoa

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idpessoa	Serial		Auto-Increment	ID do Município
FK	idmunicipio	integer		Not null	ID do estado que pertence o Município
	nmrazaosocial	char	80	Not null	Nome do município
	nmfantasia	char	80	Not null	Nome fantasia da pessoa
	cpfcnpj	bigint		Not null	cpf ou cnpj da pessoa
	endereco	char	80	Not null	Endereço da pessoa
	numero	char	5	Not null	número do endereço
	complemento	char	50		complemento do endereço
	bairro	char	50	Not null	bairro
	cep	integer		Not null	CEP
	enderecoentrega	char	80	Not null	Endereço de entrega da pessoa
	numeroentrega	char	5	Not null	Número do endereço de entrega
	complementoentrega	char	50		complemento do endereço de entrega
	bairroentrega	char	50	Not null	bairro de entrega
	cepentrega	integer		Not null	CEP de entrega
FK	idmunicipioentrega	integer		Not null	ID do Município de entrega
	hp	char	80		Home page da pessoa
	email	char	80	Not null	email da pessoa
	st	char	1	Not null	status da pessoa. 'A' = Ativo 'I' = Inativo
	nmcontato	char	80		Nome do contato da pessoa
	emailcontato	char	80		email do contato
	fonecontato	bigint			Telefone do contato
	celularcontato	bigint			Telefone celular do contato
	observacoes	varchar	1000		Observações gerais da pessoa
	fornecedor	char	1	Not null	Identificação se a pessoa é um fornecedor. 'S' = Sim; 'N' = Não; default = 'N'

Tabela: tbempresaforneecedor

	Campo	Tipo	Tam.	Propriedade	Descrição
PK; FK	idfornecedor	integer		Not null	ID do Município
PK; FK	idempresa	integer		Not null	ID do estado que pertence o Município

Tabela: tbusuario

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idusuario	serial		Auto-Increment	ID do usuário.
FK	idpessoa	integer		Not null	ID da pessoa (empresa) que pertence o usuário.
	nmcompleto	varchar	80	Not null	Nome completo do usuário
	nmusuario	varchar	20	Not null	nome de login para o usuario
	senha	varchar	128	Not null	senha do usuário
	email	varchar	80	Not null	email do usuário que esta sendo cadastrado
	dtalteracaosenha	date		Not null	Data em que o usuário deve alterar sua senha
	st	char	1	Not null	Status do usuário. 'A' = Ativo; 'I' = Inativo; 'B' = Bloqueado. Default = 'B'
	idusuarioinclusao	integer		Not null	ID do usuário que fez a inclusão no banco.
	dtinclusao	date		Not null	Data em que o usuário foi incluído. Default = CURRENT_TIMESTAMP(0)

Tabela: tbperfil

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idperfil	serial		Auto-Increment	ID do perfil.
	nmperfil	char	30	Not null	Nome do perfil.
	dsperfil	char	255		Descrição do perfil.
	nivel	integer			nível do perfil.

Tabela: tbusuarioperfil

	Campo	Tipo	Tam.	Propriedade	Descrição
PK; FK	idperfil	integer		Not null	ID do perfil.
PK; FK	idusuario	integer		Not null	ID do usuário.

Tabela: tbmenu

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idmenu	integer		Not null	ID do menu.
	nmmenu	varchar	80	Not null	Nome do menu.
	link	varchar	255		Link da página relativa ao menu.
	dsmenu	varchar	255		Descrição do menu.
	idmenupai	integer			ID do menu pai.
	st	char	1	Not null	Status do menu. 'A' = Ativo; 'I' = Inativo. Default = 'A'

Tabela: tbmenuperfil

	Campo	Tipo	Tam.	Propriedade	Descrição
PK; FK	idperfil	integer		Not null	ID do perfil.
PK; FK	idmenu	integer		Not null	ID do menu.

Tabela: tbcategoria

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idcategoria	serial		Auto-Increment	ID do perfil.
	dscategoria	char	50	Not null; Unique	Descrição da categoria.
	idpredecessora	integer			ID da categoria predecessora (pai).
	st	char	1	Not null	Status da categoria. 'A' = Ativo; 'I' = Inativo. Default = 'A'

Tabela: tbcategoriafornecedor

	Campo	Tipo	Tam.	Propriedade	Descrição
PK; FK	idcategoria	integer		Not null	ID da categoria
PK; FK	idfornecedor	integer		Not null	ID do fornecedor

Tabela: tbunidademedida

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idunidademedida	integer		Not null	ID da unidade de medida
	dsunidademedida	char	50	Not null	Descrição da unidade de medida.
	sigla	char	2	Not null	Sigla da unidade de medida.

Tabela: tbemb

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idemb	serial		Auto-increment	ID da embalagem.
FK	idunidademedida	integer		Not null	ID da Unidade de medida.
	dsemb	char	50	Not null	Descrição da embalagem.
	dsembabrev	char	20	Not null	Descrição abreviada da embalagem.
	multiplicidade	numeric		Not null	Fator de multiplicidade da embalagem em relação a unidade de medida.
	st	char	1	Not null	Status da embalagem. 'A' = Ativo; 'I' = Inativo; 'H' = Homologar. Default 'A'.
	dtinclusao	timestamp		Not null	Data de inclusão da embalagem. Default CURRENT_TIMESTAMP(0)
FK	idusuarioinclusao	integer		Not null	ID do usuário que incluiu a embalagem.
	dtinclusao	timestamp		Not null	Data de inclusão da embalagem.
FK	idusuarioaceitacao	integer			ID do usuário que aceitou ou não a embalagem.
	dtaceitacao	timestamp			Data de aceite ou não da embalagem.
	motivorejeicao	varchar	255		Motivo da rejeição (não aceite) da embalagem.

Tabela: tbitem

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	iditem	serial		Auto-increment	ID do item.
FK	idunidademedida	integer		Not null	ID da Unidade de medida.
	idcategoria	integer		Not null	ID da categoria.
	dsitem	char	80	Not null	Descrição do item.
	st	char	1	Not null	Status do item. 'A' = Ativo; 'I' = Inativo; 'H' = Homologar. Default 'A'.
FK	idusuarioinclusao	integer		Not null	ID do usuário que incluiu o item.
	dtinclusao	timestamp		Not null	Data de inclusão do item. Default CURRENT_TIMESTAMP(0)
FK	idusuarioaceitacao	integer			ID do usuário que aceitou ou não o item.
	dtaceitacao	timestamp			Data de aceite ou não do item.
	motivorejeicao	varchar	255		Motivo da rejeição (não aceite) do item.

Tabela: tbembitem

	Campo	Tipo	Tam.	Propriedade	Descrição
PK; FK	iditem	integer		Not null	ID do Item.
PK; FK	idemb	integer		Not null	ID da Embalagem.

Tabela: tbpregao

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idpregao	serial		Auto-increment	ID do pregão.
FK	idempresa	integer		Not null	ID da empresa que esta montando o pregão.
	dspregao	char	80	Not null	Descrição do pregão.
	dtcriacaoinicial	timestamp		Not null	Data de criação do pregão.
FK	idusuariocriacao	integer		Not null	ID do usuário que criou o pregão.
	dtcriacaofinal	timestamp		Not null	Data em que a criação do pregão foi concluída.
	dtinicio	timestamp		Not null	Data de inicio do pregão.
	dttermino	timestamp			Data de termino do pregão.
	dtmintermino	timestamp			Data mínima de termino do pregão.
	dtencerramento	timestamp			Data de encerramento do pregão.
FK	idusuarioencerramento	integer			ID do usuário que encerrou o pregão.
	observacoes	char	1000		Observações referente ao pregão.
	st	char	1	Not null	Status do item. 'A' = Andamento; 'M' = Montando; 'C' = Cancelado; 'R' = Realizado; 'P' = Pronto. Default 'M'.
	variacaomin	numeric			Taxa de variação mínima para um lance.
	variacaomax	numeric			Taxa de variação máxima para um lance.
	dtcancelamento	timestamp			Data de cancelamento do pregão.
	idusuariocancelamento	integer			ID do usuário que cancelou o pregão.
	motivocancelamento	char	1000		Motivo de cancelamento do pregão.

Tabela: tbitempregao

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	iditempregao	serial		Auto-increment	ID do item do pregão.
FK	idpregao	integer		Not null	ID do pregão.
FK	iditem	integer		Not null	ID do item.
FK	idemb	timestamp		Not null	ID da embalagem.
	qtde	integer		Not null	Quantidade do item, que deve ser maior que zero.
	dtentrega	timestamp		Not null	Data de entrega do item.
	precoinitial	numeric			Preço inicial estipulado como máximo pago para o item.
	observacao	char	255		Observações referentes ao item.

Tabela: tbfornecedorpregao

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idpregao	integer		Not null	ID do pregão.
FK	idfornecedor	integer		Not null	ID do fornecedor.
FK	idusuarioavaliador	integer			ID do usuário que fez a avaliação do fornecedor.
	dtavaliacao	timestamp			Data da avaliação.
	notaqualidade	numeric			Nota referente a avaliação da qualidade do fornecedor, que deverá ser 1,2,3,4,5
	notaprazo	numeric			Nota referente a avaliação do cumprimento de prazo, que deverá ser 1,2,3,4,5
	apelido	char	30		Apelido estipulado ao fornecedor para preservar a identidade do fornecedor durante o pregão.

Tabela: tblance

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idlance	serial		Auto-increment	ID do lance.
FK	idtempregao	integer		Not null	ID do item do pregão.
FK	idfornecedor	integer		Not null	ID do fornecedor.
	dtlance	timestamp		Not null	Data do lance. Default = CURRENT_TIMESTAMP(0)
	vllance	numeric		Not null	Valor do lance para o item, que deve ser maior que zero.
	ranking	integer		Not null	ranking do lance.
FK	idusuario	integer		Not null	ID do usuário que efetuou o lance.

Tabela: tbmsg

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idmsg	serial		Auto-increment	ID do lance.
FK	idpregao	integer		Not null	ID do item do pregão.
FK	idusuario	integer		Not null	ID do fornecedor.
	msg	varchar	500	Not null	Data do lance. Default = CURRENT_TIMESTAMP(0)
	dtmsg	timestamp		Not null	Data de envio da mensagem. Default = CURRENT_TIMESTAMP(0)

Tabela: tbordemcompra

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	idordemcompra	serial		Auto-increment	ID da ordem de compra.
FK	idpregao	integer		Not null	ID do pregão.
FK	idfornecedor	integer		Not null	ID do fornecedor.
	dtordemcompra	timestamp		Not null	Data da geração da ordem de compra. Default = CURRENT_TIMESTAMP(0)
	st	char	1	Not null	Status item. 'A' = Ativo; 'C' = Cancelada. Default = 'A'.

Tabela: tbitemordemcompra

	Campo	Tipo	Tam.	Propriedade	Descrição
PK	iditemordemcompra	serial		Auto-increment	ID do item da ordem de compra.
FK	idordemcompra	integer		Not null	ID da ordem de compra.
FK	idlance	integer		Not null	ID do lance.

ANEXO 10 – SCRIPTS DO BANCO
DISPONÍVEL NO CD – PASTA “APENDICES/SCRIPTS DO BANCO”

APÊNDICE 11 – CASOS DE TESTES
DISPONÍVEL NO CD – PASTA “APÊNDICES/CASOS DE TESTES”

APÊNDICE 12 – CÓDIGOS-FONTE DO APLICATIVO
DISPONÍVEL NO CD – PASTA “APENDICES/CODIGO FONTE”

APÊNDICE 13 – DIAGRAMA DE COMPONENTES

