

LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
ESPECIALIZAÇÃO EM DESENVOLVIMENTOS DE SISTEMAS
ADMINISTRATIVOS

ANTÔNIO ALVES DA SILVA

SISTEMA DE GESTÃO E CONTROLE DE QUALIDADE

NATAL/RN
2017

ANTÔNIO ALVES DA SILVA

SISTEMA DE GESTÃO E CONTROLE DE QUALIDADE

Trabalho de Conclusão de Curso
(Desenvolvimento de Sistemas
Corporativos) apresentado ao Centro
Universitário do Rio Grande do Norte
(UNI-RN) como requisito final para
obtenção do título de em Orientador
(a): Prof. Ms. Matheus Lucena

NATAL/RN
2017
ANTÔNIO ALVES DA SILVA

SISTEMA DE GESTÃO E CONTROLE DE QUALIDADE

Trabalho de Conclusão de Curso
(Desenvolvimento de Sistemas
Corporativos) apresentado ao
Centro Universitário do Rio Grande
do Norte (UNI-RN) como requisito
final para obtenção do título de
Especialista em Desenvolvimento de
Sistemas Corporativos.

Aprovado em: ____/____/____.

BANCA EXAMINADORA

Prof. Nome Completo
Orientador

Prof. Nome Completo
Membro

Prof. Nome Completo
Membro

Dedico este trabalho primeiramente a Deus, depois aos meus pais que me impulsionam a buscar uma nova vida a cada dia, meus agradecimentos aos que aceitaram a se privar da minha companhia pelos estudos, concedendo a mim a oportunidade de realizar ainda mais. Agradeço também ao Prof. Matheus que dedicou seu tempo para me orientar neste projeto.

RESUMO

O sistema desenvolvido tem por finalidade disponibilizar segurança e confiabilidade para o setor de gestão e controle de qualidade dentro de uma empresa. Provendo dessa forma a informatização dos processos internos melhorando a produtividade de forma ágil e segura.

PALAVRAS-CHAVE: Gestão e Controle de Qualidade, confiabilidade, segurança e produtividade.

ABSTRACT

The system developed has the purpose of providing security and reliability to the management and quality control sector within a company. Providing in this way the computerization of the internal processes, improving productivity in an agile and safe way.

KEYWORDS: Management and Quality Control, reliability, safety and productivity.

LISTA DE ILUSTRAÇÕES

- Figura 1 - Hibernate – pág. 12
- Figura 2 - Model View Controler – pág. 15
- Figura 3 - Diagrama DEER – pág. 18
- Figura 4 - Dependências do pom.xml – pág. 19
- Figura 5 - Dependências do pom.xml – pág. 20
- Figura 6 - Dependências do web.xml – pág. 20
- Figura 7 - Configuração do Hibernate.cfg – pág. 21
- Figura 8 - Classe Formulario.Domain – pág. 21
- Figura 9 - Classe Formulario.Dao – pág. 22
- Figura 10 - Classe Formulario.Bean – pág. 22
- Figura 11 - Página formulario.xhtml – pág. 23
- Figura 12 – Cadastro formulário – pág. 23
- Figura 13 – Visualização formulário cadastrado – pág. 24
- Figura 14 – Formulário salvo no banco de dados – pág. 24

Lista de Abreviaturas e Siglas

SGCQ	Sistema de Gestão e Controle de Qualidade
------	---

Sumário

1. INTRODUÇÃO.....	10
2. REFERENCIAL TEÓRICO	11
2.1 Tecnologias Utilizadas:	11
2.2 A linguagem de programação JAVA:	11
2.3 Banco de dados MYSQL:	11
2.4 PrimeFaces:	12
2.5 Hibernate:.....	12
2.6 Apache Maven:	13
2.7 Junit:.....	13
2.8 TomCat:	14
2.9 OminiFaces:	14
2.10 Model-View-Controller:	14
3 JUSTIFICATIVA.....	16
4. CRONOGRAMA DAS ATIVIDADES	16
5. DETALHAMENTO TÉCNICO	17
6. CONSIDERAÇÕES FINAIS.....	25
7. REFERÊNCIAS	26

1. INTRODUÇÃO

O setor de Qualidade tem como objetivo, controle da documentação de processos utilizados na empresa. Processos esses que são realizados setorialmente e especificando os procedimentos que são realizados no dia a dia seja pelo gestor ou sua equipe, sendo sistêmico ou operacional, procurando sempre manter o procedimento atualizado. Em caso de mudança o gestor responsável pelo seu setor entra em contato informando a mudança ocorrida no procedimento para a atualização do mesmo.

O processo hoje é feito utilizando arquivo.doc onde são salvos em pastas de rede que são disponibilizadas via browser, com níveis de permissões para os colaboradores. A organização é dividida por departamentos e seus tipos de procedimentos. A utilização deste recurso gera transtornos nas auditorias internas, pois a pessoa que está sendo auditada tem que apresentar os documentos impressos para o auditor.

O presente relatório detalha o processo de criação de um sistema de controle e gestão de qualidade, utilizando a linguagem Java, framework Primefaces, Servidor Apache TomCat e o banco de dados Mysql Workbench 6.3 para melhorar o controle dos processos possibilitando dessa forma agilidade e segurança e acesso disponível dentro e fora da empresa.

2. REFERENCIAL TEÓRICO

2.1 Tecnologias Utilizadas:

Para desenvolvimento do sistema foi utilizada a arquitetura Model-View-Controller (MVC) a linguagem de programação JAVA com o banco de dados MYSQL e os frameworks: PrimeFaces, Hibernate, Apache Maven, Junit, GlassFich e OminiFaces.

2.2 A linguagem de programação JAVA:

JAVA é uma linguagem de programação orientada a objetos utilizada para desenvolvimento de sistemas desktop, mobile ou web que foi desenvolvida por uma equipe de desenvolvedores da Sun Microsystems lançada em 1995. Ficando entre as três linguagens mais utilizadas no mundo.

2.3 Banco de dados MYSQL:

MYSQL é um dos SGBDs mais utilizados no mercado, possuindo versões gratuitas e pagas, onde atende os requisitos de vários projetos desde o mais básico ao os mais complexos.

Possuindo compatibilidade para os drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, C#, Visual Basic, Python, Perl, PHP, ASP e Ruby. Além de possuir excelente desempenho e estabilidade, o Mysql não é exigente em questões de Hardware.

2.4 PrimeFaces:

É um popular Framework utilizado para projetos JSF, que pode ser utilizado para desenvolver aplicações mais rapidamente. Com ele é possível aumentar a produtividade do desenvolvedor e a experiência do usuário com a aplicação, possuindo diversos componentes para os mais diversos fins.

2.5 Hibernate:

É um framework criado com o intuito de facilitar a persistência de dados em Java. Para isso, traz como uma das suas principais características recursos que possibilitam o mapeamento objeto-relacional (ORM). Abstraindo do programador a tarefa de, a cada consulta, fazer a conversão de dados. Essa funcionalidade reduz a quantidade de código que precisa ser escrita, assim há um aumento de produtividade e consequentemente menos erros no projeto.

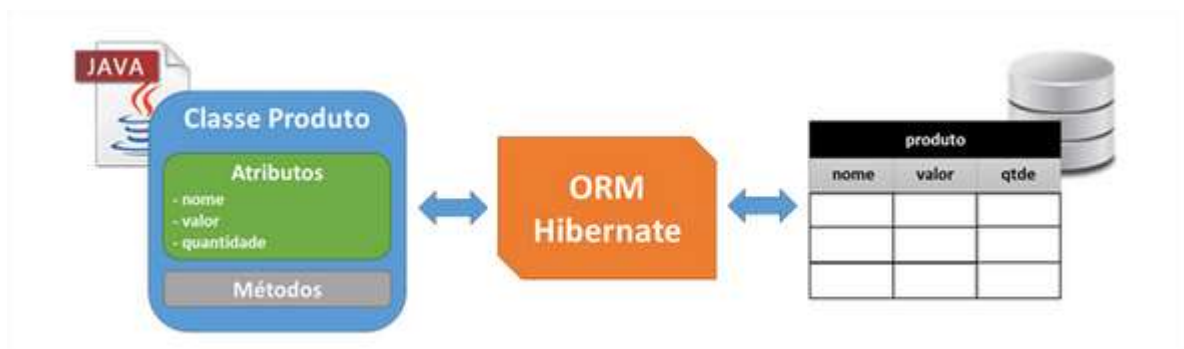


Figura 1

O mapeamento pode ser feito com anotações XML que pode ser o persistence.xml ou o hibernate.cfg.xml. Que por ser um processo mais simples é muito utilizado pelos programadores

2.6 Apache Maven:

O Apache Maven é uma ferramenta de automação e compilação utilizada em projetos Java fornecendo a possibilidade de automatizar e padronizar suas aplicações agregando agilidade e qualidade ao produto final, além de ser extremamente flexível, permite que sejam adicionados plug-ins que para estender a sua capacidade nativa.

Além do Java, o Apache Maven é compatível com outras linguagens de programação como C# e Ruby. A configuração do mesmo é feita no arquivo pom.xml onde descreve o projeto de software que está sendo construído, possuindo objetivos pré-definidos para realizar certas tarefas como como compilação do código e o empacotamento.

2.7 Junit:

Sendo uma das ferramentas mais utilizadas no mercado o Junit é um framework open-source criado por Erich Gamma e Kent Beck com o intuito de facilitar a execução de testes de unidade em Java a partir de anotações e algumas declarações, conseguimos avaliar classe e métodos para saber se eles estão apresentando algum comportamento indesejado.

Com o Junit é evitado o uso do método main() em qualquer lugar apenas para saber se o código de um método criado está funcionando perfeitamente. O teste de unidade ou teste unitário verifica a menor unidade de um software, na orientação de objetos a menor unidade é o método, checando os resultados do teste fornecendo uma resposta imediata, podendo criar uma hierarquia de teste que permitirá testar apenas uma parte do sistema ou todo ele.

2.8 TomCat:

O Apache TomCat é um servidor web open-source, sendo uma das ferramentas mais utilizadas pelos desenvolvedores foi lançado oficialmente pela Apache Software Foundation (ASF) em 1999, como uma implementação de referência para a Servlet API. Possui ferramentas para configuração e gerenciamento, o que também pode ser feito editando-se manualmente arquivos de configuração formatados em Xml.

2.9 OminiFaces:

O OminiFaces é uma biblioteca open-source desenvolvida para o framework JavaServer Faces. Desenvolvida com o objetivo de facilitar a utilização do JSF, fornecendo um conjunto de artefatos com o intuito de melhorar a funcionalidade de sua estrutura. Possuindo cooperação com bibliotecas JSF existentes como PrimeFaces, RichFaces, OpenFaces, ICEfaces

2.10 Model-View-Controller:

Model-View-Controller (MVC) é um padrão de projeto de software ou arquitetura muito utilizado em aplicações e frameworks que com o intuito de facilitar a resolução de uma situação de cunho crítico e deixar o código mais limpo, lhe possibilita particionar os recursos de seu sistema as diferentes partes que compõe um website como a vista, a lógica e o acesso ao banco de dados.

Trazendo como benefício isolar as regras de negócio desde a lógica de apresentação a interface de interação com o usuário final, a sua utilização possibilita a divisão do projeto em camadas executando apenas o que lhe é definido. Podendo ser utilizado em vários tipos de projetos como, por exemplo, desktop, web e mobile.

A comunicação entre a interface e regras de negócio é definida pelo Controller e é ele que torna possível a separação entre as camadas. Quando um evento é realizado através da View, como por exemplo, o clique em um botão, a interface irá se comunicar com o Controller, que por sua vez se comunica com as regras de negócios. A função realizada pelo clique do botão é realizada pelo Model, a regra de negócio que é executada assim que receber um pedido do Controller. Após realizar o processo requisitado o Model o envia para o Controller que por sua vez o destina para a View que irá encaminhar a resposta para o usuário, onde o único que conhece a existência das camadas Model e View é o Controller. Entre as diversas vantagens do padrão MVC estão a possibilidade de reescrita da GUI ou do Controller sem alterar o nosso modelo, reutilização da GUI para diferentes aplicações com pouco esforço, facilidade na manutenção e adição de recursos, reaproveitamento de código, facilidade de manter o código sempre limpo.

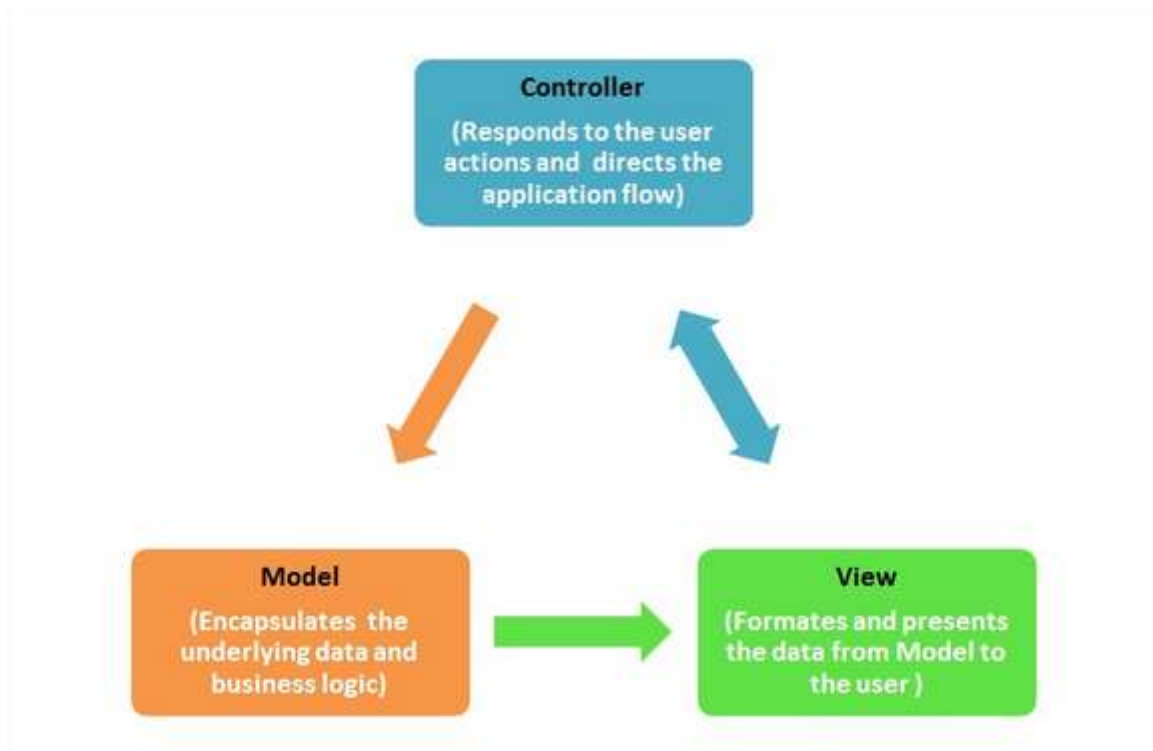


Figura 2

3 JUSTIFICATIVA

A criação do sistema está sendo realizada com o intuito de melhoria dos procedimentos do setor de qualidade onde já tivemos vários problemas desde arquivos corrompidos a quantidade demasiada de processos impressos para distribuir para os colaboradores durante o período de auditoria.

Dessa forma será economizado o uso de impressões e irá auxiliar o setor de qualidade a ter mais agilidade e segurança nos seus procedimentos, onde os próprios gestores não precisaram baixar um documento Word destinado ao setor, o mesmo irá apenas acessar o sistema onde terá acesso aos procedimentos do seu setor, fazendo as alterações necessárias com a aprovação do setor de qualidade. Optei pela criação desse sistema devido a demanda que tem aumentado nos processos da empresa e na dificuldade de organização documental tanto para períodos de auditoria quanto ao uso diário.

4. CRONOGRAMA DAS ATIVIDADES

Eventos/Atividades	Out/2017	Nov/2017	Dez/2017	Jan/2018
Elaboração do projeto	X			
Pesquisa Bibliográfica	X			
Fundamentação Teórica		X		
Etapas especificadas ao tempo de elaboração do TCC			X	
Elaboração do TCC			X	
Apresentação do TCC				X

5. DETALHAMENTO TÉCNICO

Foi necessário realizar uma reunião com o gestor do setor de qualidade antes de começar o desenvolvimento do sistema, dessa forma foi possível analisar melhor a situação atual, que antes possuía apenas uma visão superficial, e ver novos métodos com o intuito de melhora em relação aos processos. Foi percebido então que o grande problema estava nos processos de qualidade pois eram os que possuíam a maior demanda em relação a cadastros, alterações e gestão dos mesmos.

No cenário atual, no período de cada 3 meses é realizada uma revisão de todos os processos da empresa, seja ele sistêmico ou não, onde são marcadas reuniões com os gestores de cada setor para averiguar se houve mudança ou não referente aos processos realizados periodicamente ou mensalmente ou se é necessário a inclusão de um novo processo.

Após essa avaliação é enviado para o gestor copias dos processos que precisam ser alterados ou inclusos já com as alterações feitas com relação aos dados apurados na reunião, caso concorde com a alteração irá enviar um meio para o setor de qualidade aprovando a alteração, caso discorde será feita uma nova reunião apontando os pontos que precisam ser corrigidos.

Durante o período de auditoria seja ela interna ou externa, os responsáveis pelos setores auditados levam os processos documentados em mãos para a sala de auditoria. Utilizando este método a um consumo grande relacionado a utilização de papéis na empresa além de uma demanda maior no tempo entre reuniões e alterações de documentos.

Para amenizar esse procedimento resolvi desenvolver um sistema simples de cadastro de processos do setor de qualidade, onde o colaborador não terá mais que guardar documentos relacionados aos processos internos, processos esses que serão cadastrados e visualizados através do sistema de gestão e controle de qualidade (SGCQ). O gestor de qualidade irá realizar a alteração no documento via sistema e ira informar ao responsável do setor, que no mesmo instante poderá acessar os seus

processos e realizar as alterações, alterações essas que deveram ser aprovadas em conjunto com o setor de qualidade.

Foi realizado um diagrama DEER como base para a criação da estrutura, comas tabelas relacionadas as classes de usuário, grupo e cadastrado do formulário de processo, sempre se baseando na real necessidade para a criação do sistema onde pode ocorrer alterações até a conclusão do mesmo.

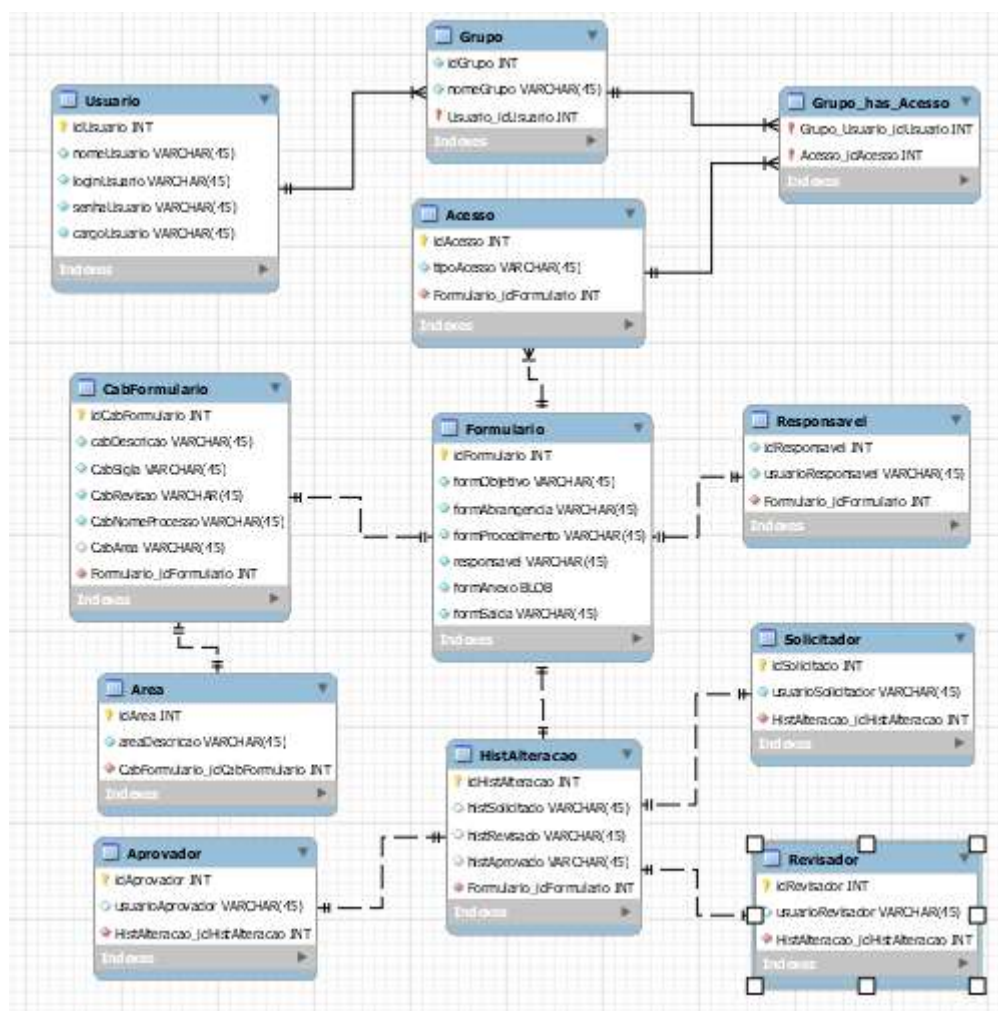


Figura 3

Foi criado um projeto Java web com Maven, Hibernate e Primefaces onde no pom.xml (figura 2) e (figura 3) do projeto foi inserido todas as dependências utilizadas, tive preferência pela versão 5.2 do Primefaces pois a considero mais estável, assim como o Hibernate onde utilizei a versão 4.3.11, a opção pelo banco de dados Mysql Workbench foi devido a uma maior familiaridade com o mesmo. No web.xml (figura 4) Foram realizadas configurações básicas para dar andamento ao projeto.

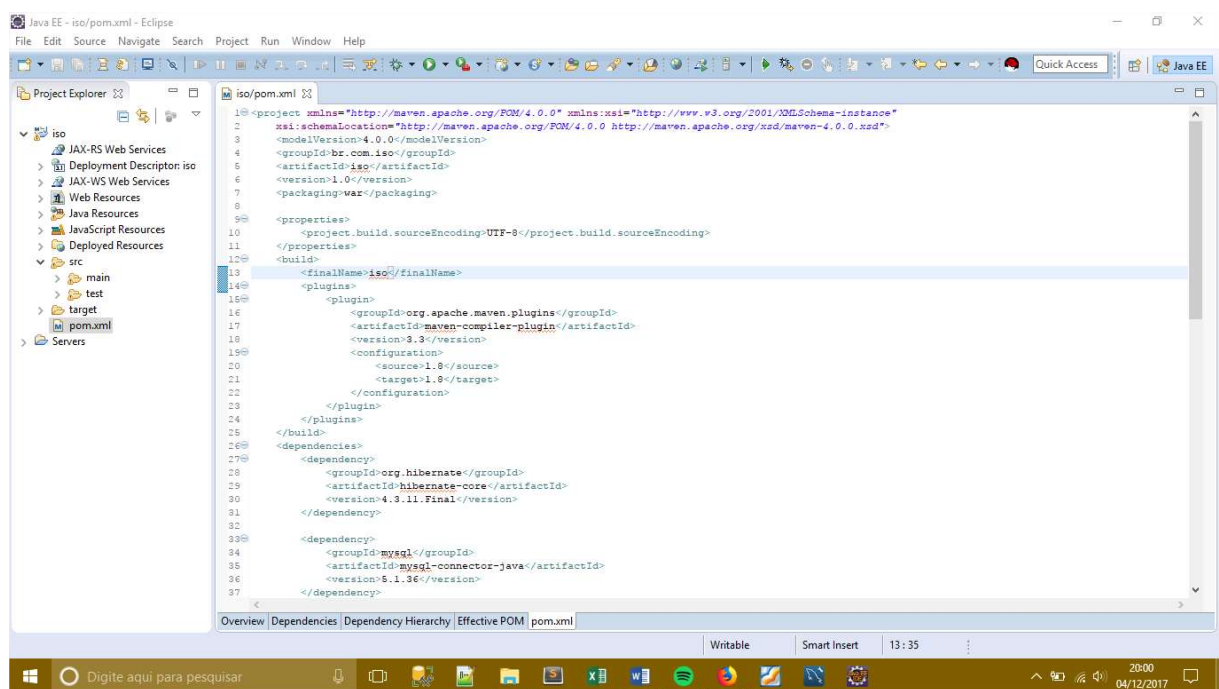


Figura 4

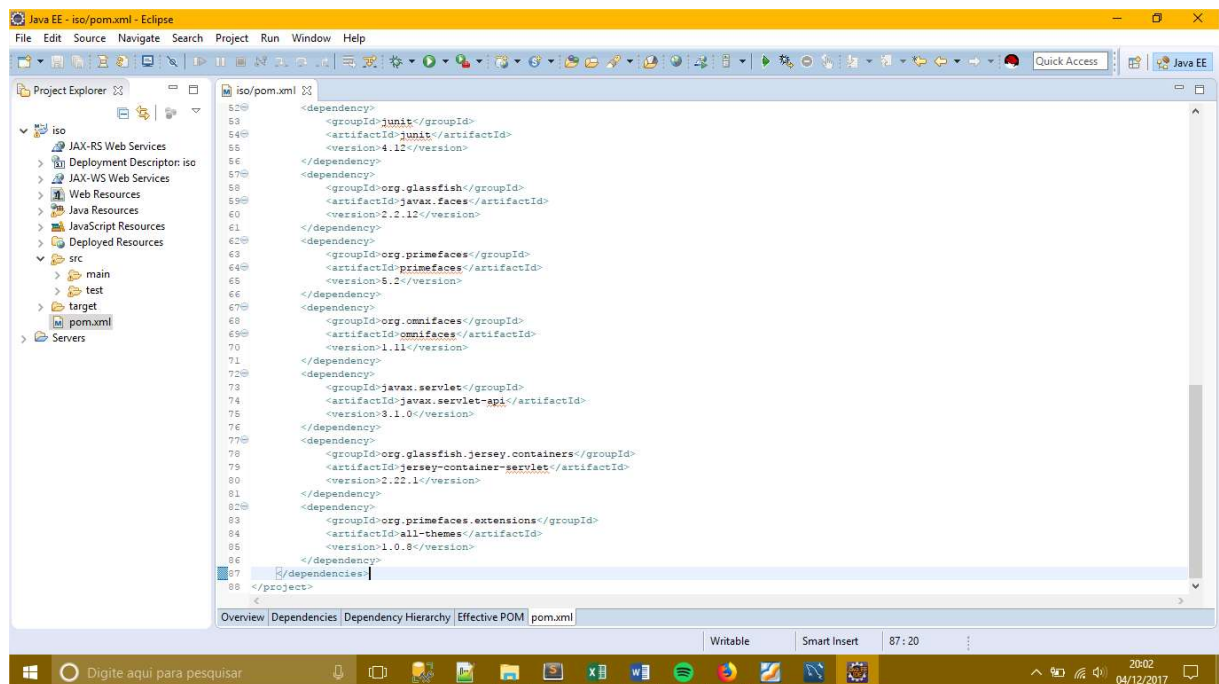


Figura 5

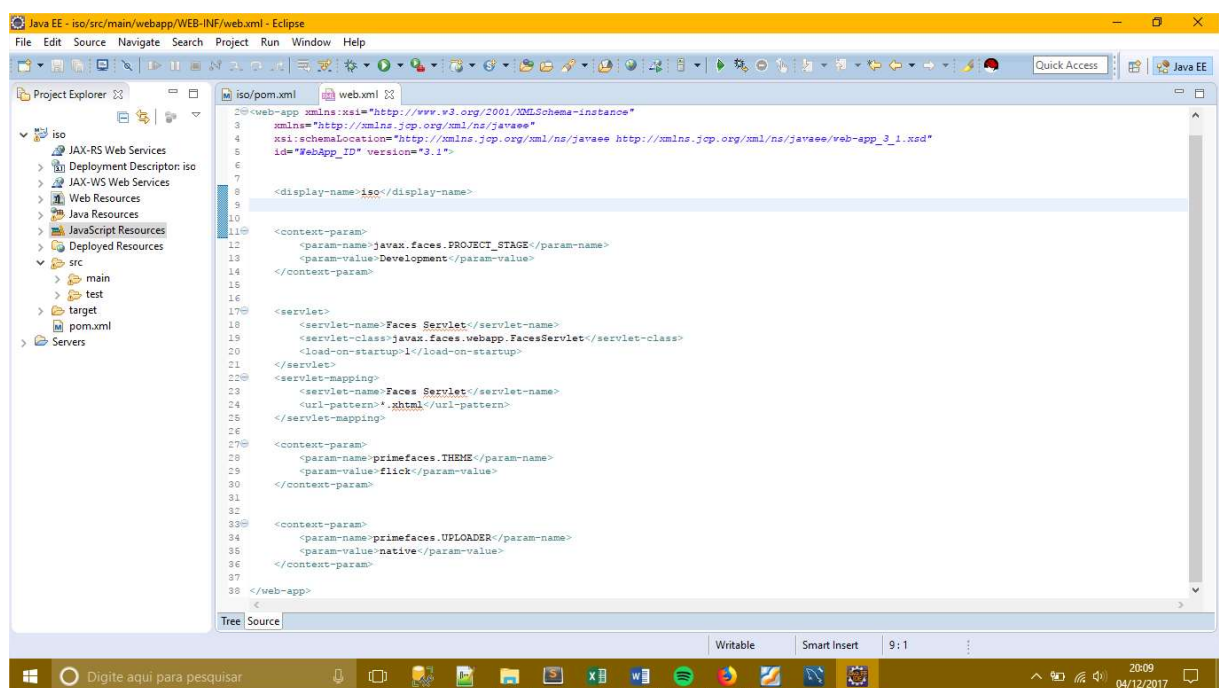


Figura 6

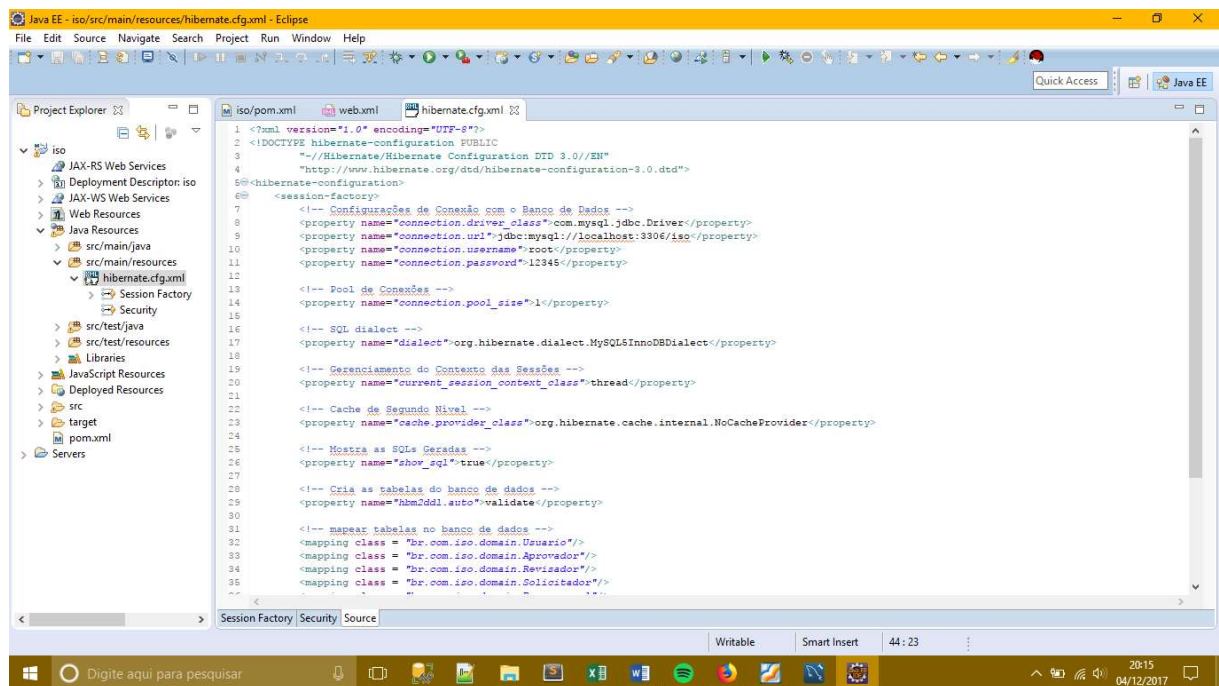


Figura 7

Foram criadas classes de Domain, Dao e MBean com Hibernate annotations (figura 6), (figura 7) e (figura 8) por fim arquivo.xhtml para criar a página web(figura 9).

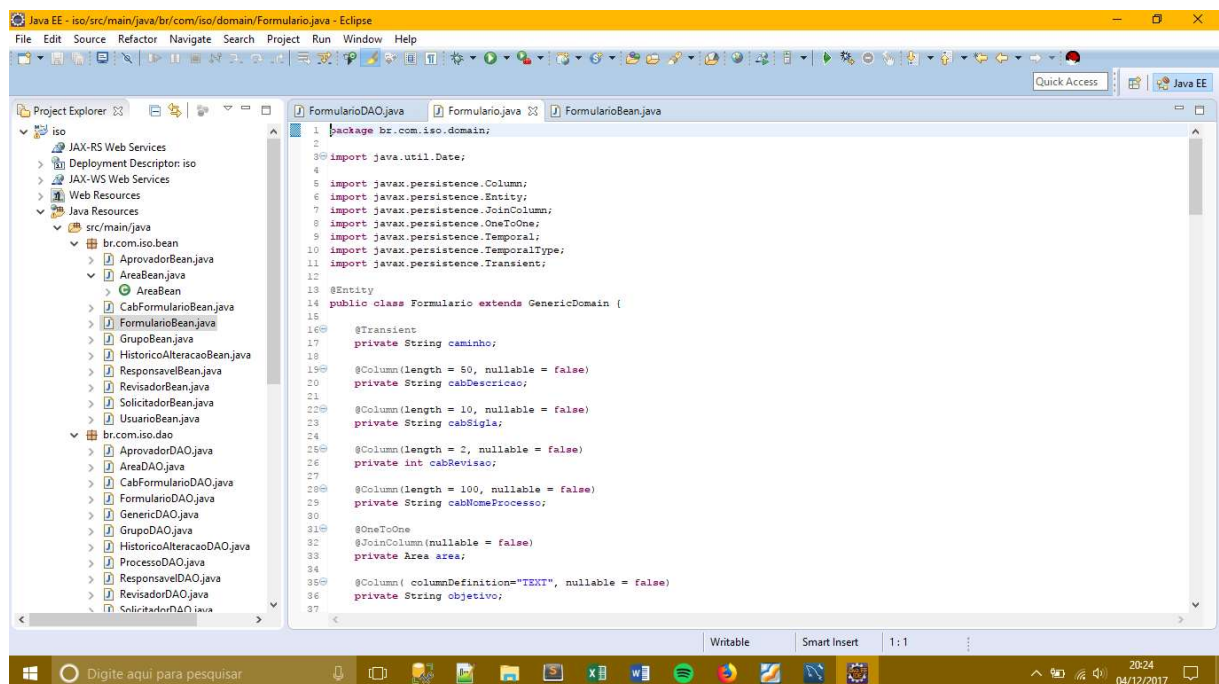


Figura 8

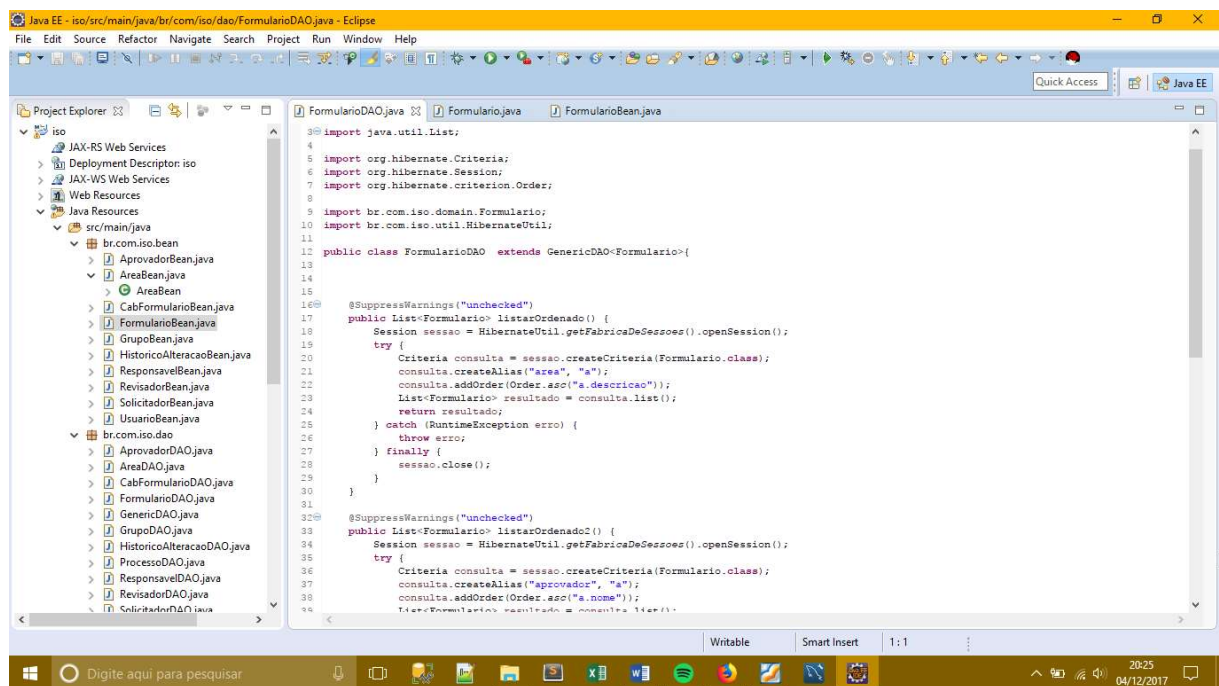


Figura 9

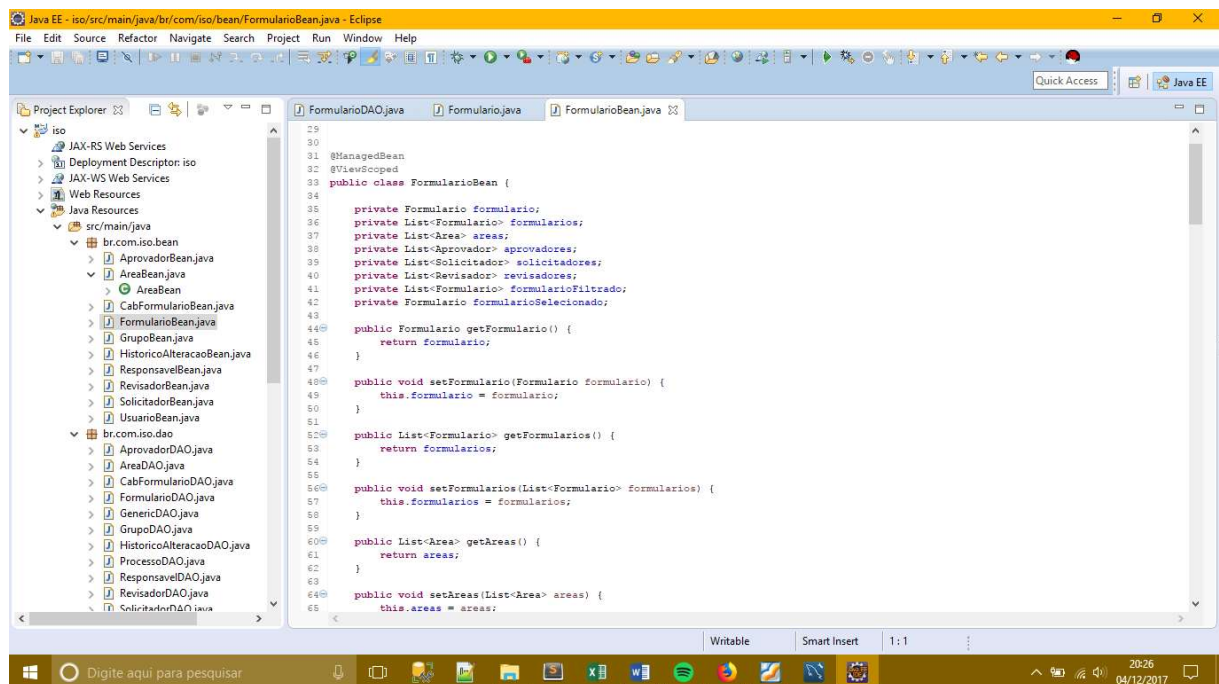


Figura 10

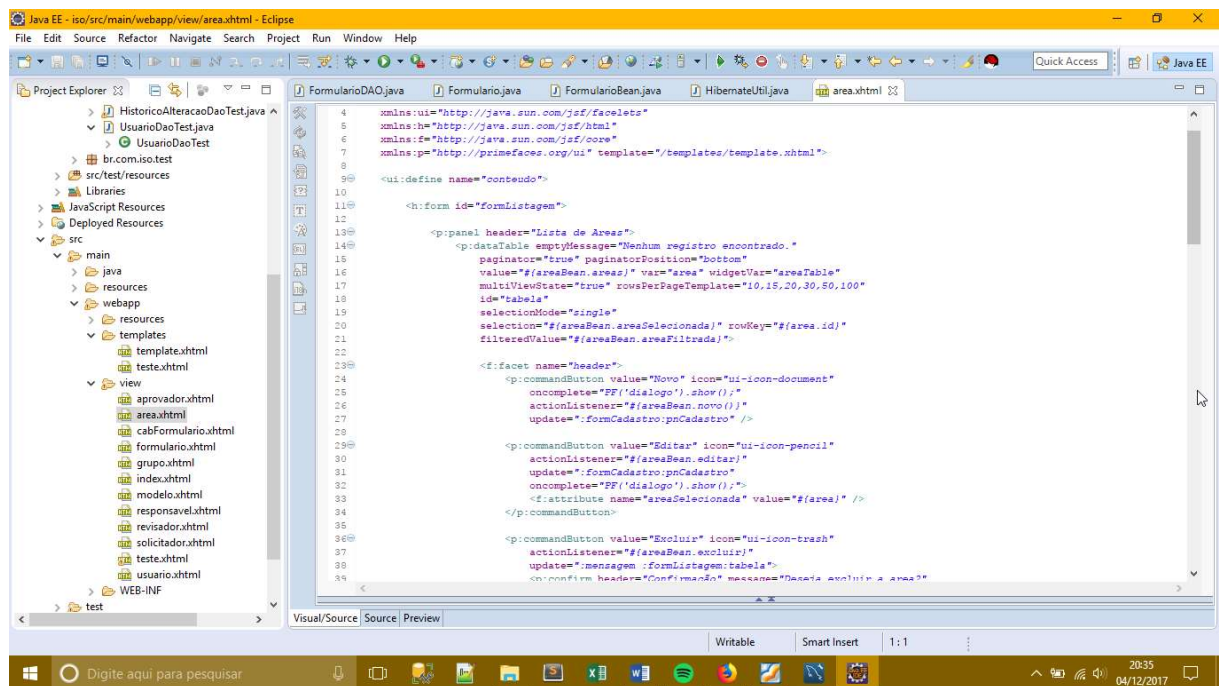


Figura 11

Cadastro de área na (figura 10) a (figura 12) onde é apresentado o cadastro salvo no banco de dados.

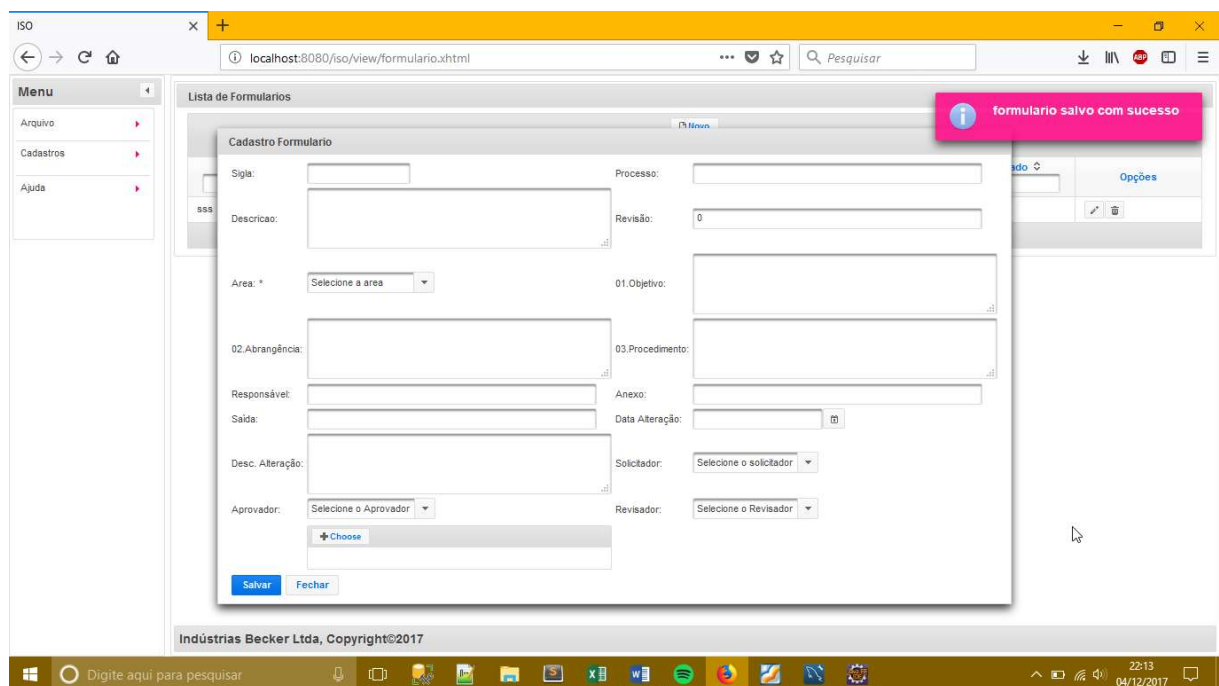


Figura 12

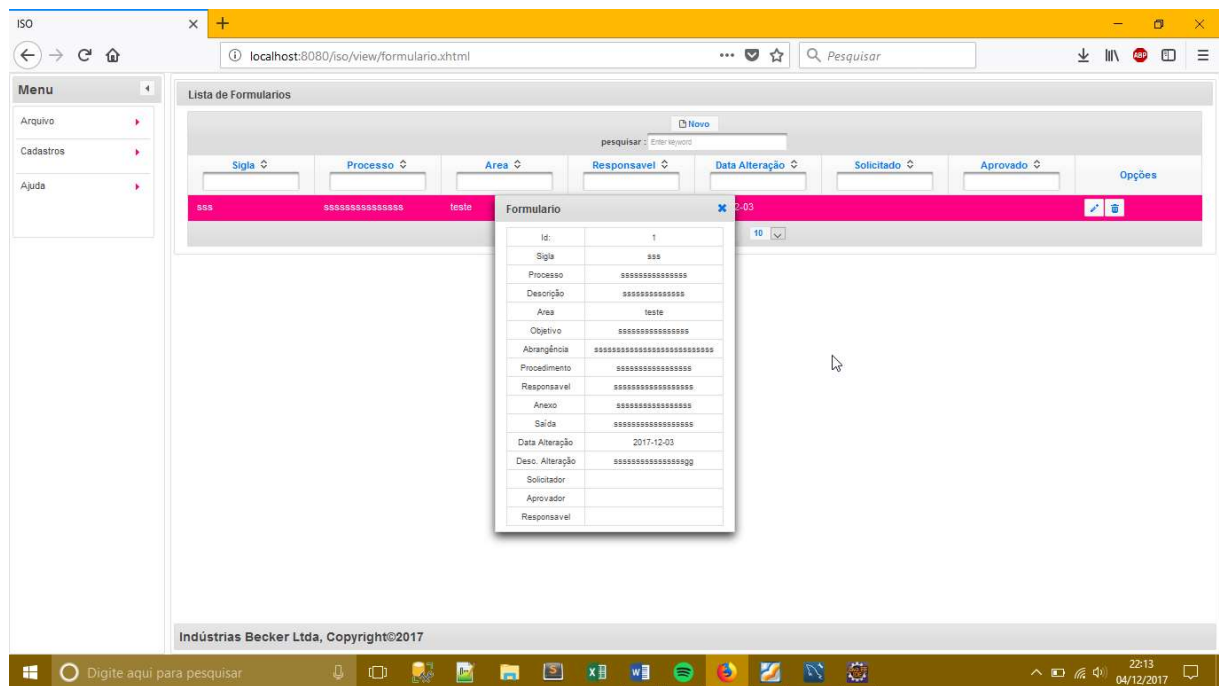


Figura 13

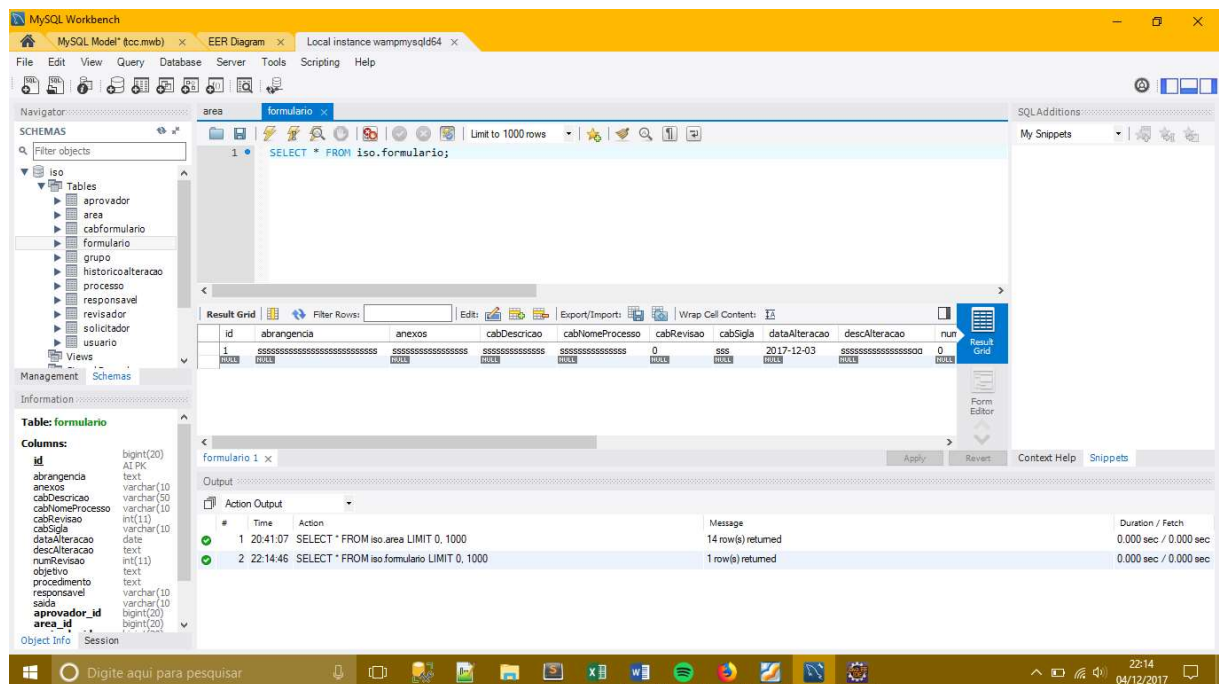


Figura 14

6. CONSIDERAÇÕES FINAIS

Ao concluir este relatório técnico procurei refletir sobre os desafios superados até o presente momento, observando o nível de conhecimento adquirido desde o começo do projeto, visto que o mesmo começou devido a uma carência e necessidade do setor de qualidade para o desenvolvimento deste sistema.

7. REFERÊNCIAS

HIBERNATE DOCUMENTATION, disponível em:
<<http://hibernate.org/orm/documentation/4.3/>>. Acesso em 10 de setembro de 2017.

APACHE TOMCAT, disponível em:< <http://tomcat.apache.org/>>. Acesso em: 11 de setembro de 2017.

CAELUM, disponível em: < <https://www.caelum.com.br/apostila-java-web/bancos-de-dados-e-jdbc/>>. Acesso em 11 de setembro de 2017.

DEVEMEDIA, disponível em: < <https://www.devmedia.com.br/crud-completo-com-hibernate-e-jpa/32711>>. Acesso em 13 de setembro de 2017.

DEVEMEDIA, disponível em: < <https://www.devmedia.com.br/introducao-ao-jsf-managed-bean/29390>>. 18 de setembro de 2017.

DEVEMEDIA, disponível em:< <http://www.devmedia.com.br/jpa-como-usar-a-anotacao-generatedvalue/38592>>. Acesso em: 18 de setembro de 2017.

PRIMEFACES, disponível em:< <https://www.primefaces.org/showcase/index.xhtml>>. Acesso em: 10 de outubro de 2017.

PRIMEFACES, disponível em:
https://www.primefaces.org/docs/guide/primefaces_user_guide_5_2.pdf>. Acesso em: 22 de outubro de 2017.