

FACULDADE CAMPO LIMPO PAULISTA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Trabalho de Conclusão de Curso

Atividade Prática de Desenvolvimento de *Software*

Eduardo Aparecido Cassinotte – 12846

André Marcos Silva (Orientador)

Atividade Prática de Desenvolvimento de Software

**Desenvolvimento Web Ágil
Utilizando Framework**

Eduardo Aparecido Cassinotte

FACULDADE CAMPO LIMPO PAULISTA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Desenvolvimento Web Ágil Utilizando Framework

Trabalho submetido à Coordenação de Ciência da Computação da Faculdade Campo Limpo Paulista como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Campo Limpo Pta. (SP), 08 de Dezembro de 2014.

Eduardo Aparecido Cassinotte

Banca examinadora

Prof. Me. André Marcos Silva (Orientador)

Prof. Dr. Rodrigo Bonacin

Prof. Dr. Osvaldo Luís de Oliveira (Suplente)

Agradeço aos meus pais que fizeram parte da minha formação e intelecto, também agradeço a Sandra, pelo apoio dado durante momentos difíceis que enfrentei na minha vida pessoal.

Sem o apoio deles a conclusão desta etapa na minha vida não seria realidade.

FACULDADE CAMPO LIMPO PAULISTA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Desenvolvimento Web Ágil Utilizando Framework

SUMÁRIO

RESUMO	5
1. PROPOSTA	5
Motivação.....	5
Objetivos	5
Metodologia de Trabalho	5
2. RESULTADOS.....	5
Apresentação Funcional	6
Especificação Técnica	8
Problemas Encontrados.....	9
3. CONCLUSÕES	11
4. REFERÊNCIAS ÚTEIS	11
ANEXO A – MANUAL DE USUÁRIO	13
ANEXO B – PROJETO DE <i>SOFTWARE</i>	36
ANEXO C – MANUAL DE OPERAÇÃO E IMPLANTAÇÃO	40

RESUMO

Com o intuito de demonstrar a utilização de *frameworks* para o desenvolvimento ágil de aplicações, este trabalho tem como objetivo a demonstração de uma aplicação Web.

O sistema desenvolvido é uma proposta fictícia e servira como motivador de negócio para o uso das tecnologias propostas no trabalho. Trata-se de um servidor Repositório de Perguntas para auxilio em atividades educacionais buscando em uma arquitetura web-cliente e servidor de aplicação.

1. PROPOSTA

Motivação

Devido à grande demanda de tempo e custo financeiro no desenvolvimento de aplicações para Web, cada vez mais as empresas de Software buscam meios para agilizar o processo. Com a intenção pessoal de obtenção de conhecimentos para a formação pessoal de técnicas que será tendência no mercado, este trabalho tem como principal motivador a demonstração da utilização de *frameworks* para a auto formação pessoal.

Objetivos

Este trabalho tem como objetivo o desenvolvimento de um sistema fictício de um Repositório de Perguntas. O sistema possuirá as seguintes funcionalidades básicas:

- Cadastro, alteração, consulta e remoção de questões com múltiplas alternativas;
- Cadastro, alteração, consulta e remoção de gêneros para as questões;
- Cadastro, alteração, remoção, consulta e Associação/Dissociação de *Tag's* para as questões.

Metodologia de Trabalho

Para o desenvolvimento do sistema proposto serão utilizadas as seguintes ferramentas:

- NetBeans 8.0.1 – Ambiente de Desenvolvimento Integrado (IDE) para desenvolvimento de software nas linguagens Java, C, C++, PHP e entre outras;
- MS SQL 2012 – Sistema Gerenciador de Banco de Dados Relacional (SGBD);
- JSF – Java Server Faces 2.2 - *Framework* modelo, visão, controle (MVC) baseado em Java para de aplicações Web;

- Hibernate 4.2.6 – *Framework* para o mapeamento objeto-relacional escrito na linguagem Java para persistência automática um Banco de Dados;
- PrimeFaces 5.0 – *Framework* de conjunto de componentes *Open Source User Interface* (UI) para aplicações baseadas em Java Server Faces (JSF).

O sistema foi desenvolvido baseado no levantamento dos requisitos e casos de usos necessários para o pleno funcionamento do problema proposto.

Apresentação Funcional

O sistema protótipo desenvolvido é um sistema Repositório de Perguntas restrito a usuários previamente cadastrados. Entre as funcionalidades disponíveis podem ser citadas o cadastramento de perguntas, cadastramento de Tag's e cadastramento de gêneros conforme representadas nas Figuras 1, 2 e 3 respectivamente.

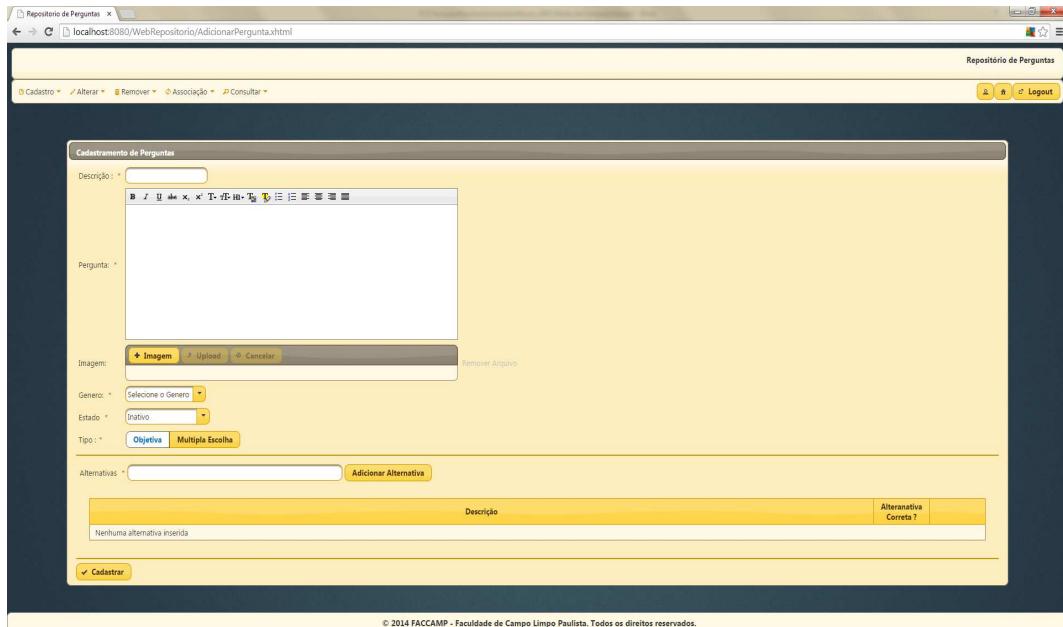


Figura 1: Cadastramento de Pergunta

A Figura 1 mostra a interface responsável pelo cadastramento de perguntas. Nesta tela, os usuários poderão incluir pergunta com suas alternativas e definir seu gênero, estado (ativo ou inativo), seu tipo (objetiva ou múltipla escolha) e uma imagem para a pergunta.

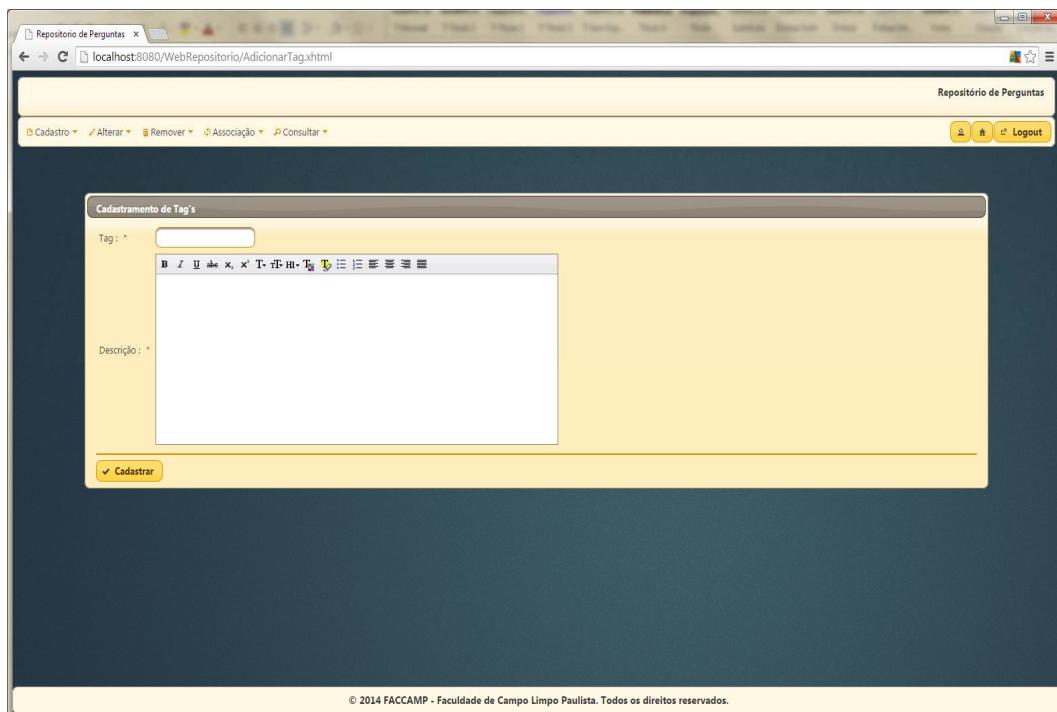


Figura 2: Cadastramento de Tag's

A Figura 2 mostra a interface responsável pelo cadastramento de *Tag's*. Nesta tela, os usuários poderão incluir *Tag's* que posteriormente serão associadas a perguntas. Uma *tag*, ou em português etiqueta, é uma palavra-chave (relevante) ou termo associado com uma informação que o descreve e permite uma classificação da informação baseada em palavras-chave, usualmente, escolhidas informalmente por usuários.

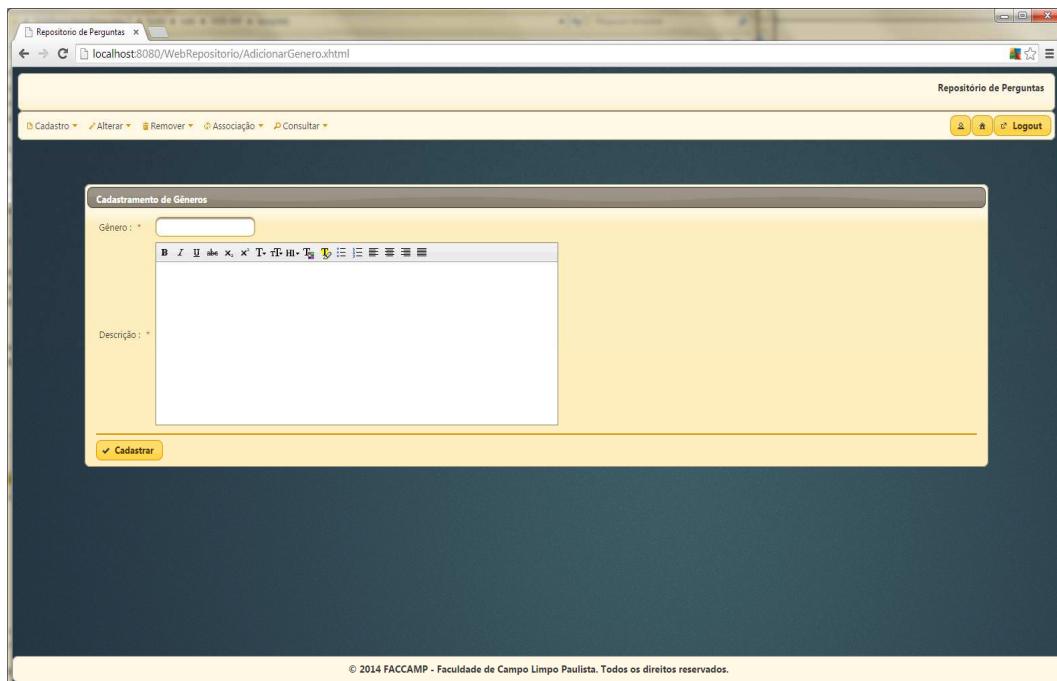


Figura 3: Cadastramento de Gênero

A Figura 3 mostra a interface responsável pelo cadastramento de gêneros. Nesta tela, os usuários poderão incluir gêneros. Gênero é um conceito que engloba todas as características básicas que possuem um determinado grupo ou classe de seres ou coisas.

Diferente de gêneros, as Tag's são englobamento de características definidas por senso comum entre usuários.

Para mais informações sobre o sistema consulte o anexo A (Manual de Usuário)

Especificação Técnica

Este tópico apresenta as informações sobre as ferramentas e as tecnologias empregadas no desenvolvimento do sistema.

Para o desenvolvimento da aplicação, foram utilizadas as seguintes ferramentas e tecnologias:

NetBeans IDE 8.0.1: Ambiente de desenvolvimento integrado (*IDE*) gratuito e de código aberto para desenvolvedores de software nas linguagens Java, C, C++, PHP, Groovy, Ruby, entre outras. O *IDE* é executado em muitas plataformas, como Windows, Linux, Solaris e MacOs. O NetBeans *IDE* oferece aos desenvolvedores ferramentas necessárias para criar aplicativos profissionais de *desktop*, empresariais, *web* e móveis multiplataformas.

Microsoft SQL Server 2012: Sistema gerenciado de Banco de Dados relacional (SGBD) desenvolvido pela Microsoft.

GlassFish Server Open Source Edition 4.1: Servidor de aplicação *open source* liberado pela Sun Microsystems para a plataforma J2EE, suporta todas as especificações da API Java EE.

JSF – Java Server Faces 2.2: *Framework* MVC (Modelo, Visão e Controle) baseado em Java para construção de interfaces de usuários baseadas em componentes para aplicações web. Possui um modelo de programação dirigido a eventos, abstraindo os detalhes da manipulação de eventos organização dos componentes, permitindo que o programador se concentre na lógica da aplicação.

Ciclo de vida do JSF é o nome dado à sequência de processamento realizada na implementação JSF para a geração das visões. O JSF, diferente de outros *frameworks*, pois possui um processamento de requisição dividido em seis fases, conforme representado na Figura 4.

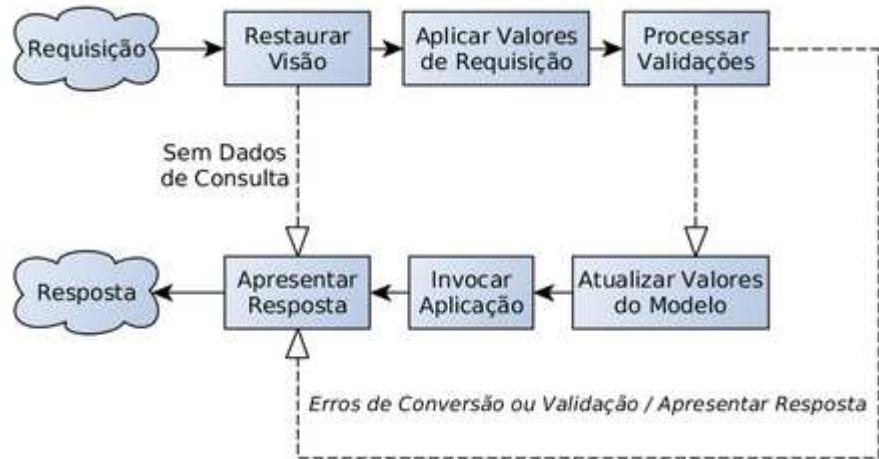


Figura 4: Diagrama mostrando o ciclo de vida do JSF (Fonte: Wikipédia)

Hibernate 4.2.6: Framework para mapeamento objeto-relacional escrito na linguagem Java, que facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (*XML - Extensible Markup Language*) ou anotações Java. O objetivo do Hibernate é diminuir a complexidade entre programas Java, baseado no modelo orientado a objetos, que precisam trabalhar com um banco de dados modelo relacional. Sua principal característica é a transformação das classes em Java para tabelas de dados. O Hibernate gera as chamadas SQL e libera o desenvolvedor do trabalho manual de conversão dos dados resultantes, mantendo o programa portável para qualquer banco de dados SQL, porém causando um pequeno aumento no tempo de execução.

PrimeFaces 5.0: Framework de componentes de código aberto para JSF 2.0 (*Java Server Faces*) com mais de 100 componentes, permitindo criar interfaces ricas para aplicações *web* de forma simplificada e eficiente. Os componentes já foram construídos para trabalhar de forma nativa com a metodologia *AJAX* (*Asynchronous JavaScript and XML*).

Problemas Encontrados

A ferramenta de desenvolvimento integrado (*IDE – Interface Development Environment*) Eclipse não possui integrado os *frameworks* utilizados neste trabalho e entre outras funcionalidades que facilitam o trabalho de desenvolvimento, sendo necessário instalação de *plug-ins* de terceiros para obter estes recursos, para tanto por questões de agilidade no qual se baseia este trabalho, foi utilizado o NetBeans que diferente do concorrente Eclipse já possui todas as funcionalidades necessárias.

Os componentes do framework PrimeFaces já possuem por padrão metodologia *AJAX* (*Asynchronous JavaScript and XML*) embutidas em seus componentes, não sendo necessário a utilização de outras bibliotecas em conjunto com os

componentes para fazer requisições *Ajax*, porém existe um problema, a utilização do próprio componente *PrimeFaces* com o recurso *Ajax* torna necessário especificar por completo a localização do componente na página *XHTML*. Exemplificando, caso numa requisição feita via *Ajax* deseje especificar que determinado componente que deverá ser reavaliado pelo servidor, e supondo que este componente esteja localizado dentro de um form, para que o componente do *PrimeFaces* o localize deve ser colocar a caminho completo até componente conforme representado abaixo:

- :form:nomeDoComponente

O componente PkList do *PrimeFaces* funciona perfeitamente para tipos primitivos (*double*, *int*, *boolean* e entre outros), porém para tipos que são objetos se faz necessário a implementação do método de conversão de tipos, para tanto que o projeto possui uma classe denominada *ListaConversão* que implementa a conversão para o tipo de classe “Pergunta”.

O Java possui a implementação de filtros para aplicações Web. Porém a utilização de tal recurso para controle de usuário se torna impossível pelo seguinte fato, com o filtro e possível barrar o que pode ou não ser enviado ao usuário conforme seu nível de acesso, como por exemplo somente usuários logados podem acessar o sistema, no caso de usuários não logado sempre e direcionado para pagina de login, porém o filtro sempre barra os efeitos visuais que a pagina possui, como por exemplo *CSS* e imagens, gerando no final a página de *login* para usuário sem os efeitos visuais. Para contornar esse problema foi utilizado uma implementação que existe no *JSF (Java Server Faces)* que é *PhaseListener*, que consegue lidar de melhor forma com esse problema.

O componente do *PrimeFaces* OpenDialog, conforme utiliza a funcionalidade do sistema consulta pergunta s realmente funciona se informado no arquivo *facesconfig.xml* a seguinte anotação:

```
<Application>
    <Action-listener>
        org.primefaces.application.DialogActionListener
    </action-listener>
    <Navigationhandler>
        org.primefaces.application.DialogNavigationHandler
    </navigation-handler>
    <View-handler>
        org.primefaces.application.DialogViewHandler
    </View-handler>
</Application>
```

Para este mesmo componente caso seja necessário definir determinadas propriedades do *form* (Modal, Scroll, Altura, Largura e entre outros), na sua chamada é necessário passar ao método de criação um objeto do tipo *Map<String, Object>* contendo as propriedades.

2. CONCLUSÕES

Este trabalho desenvolvido agregou conhecimento e experiência no desenvolvimento de aplicações *web* ágeis através da utilização de *frameworks*. A utilização de tais *frameworks* contribui positivamente para a produtividade no desenvolvimento do sistema. O desenvolvimento do sistema proporcionou um ganho profissional de um recurso que futuramente será utilizado no mercado de trabalho. Vale lembrar que fica aberto a possibilidade futura de continuidade do sistema base desenvolvido neste trabalho, tais como:

- Geração de provas, histórico de geração de provas e exportação de gabaritos em diversos formatos (por exemplo *html, pdf, doc* e etc);
- Distribuição de avaliações *on-line*.

3. REFERÊNCIAS ÚTEIS

PrimeFaces [on-line] *Site oficial do framework PrimeFaces*. Disponível em: <http://www.primefaces.org/showcase/index.xhtml>. Acessado em 08 de Novembro de 2014.

DevMedia [on-line] *Fórum de dicas e tutoriais em desenvolvimento*. Disponível em: <http://www.devmedia.com.br/criando-um-converter-padroao-para-picklist-do-primefaces/29419> / <http://www.devmedia.com.br/manytomany-hibernate-variacoes-unidirecional-e-bidirecional/29535>. Acessado em 08 de Novembro de 2014.

Helvete Av Silas Ribas [on-line] *Fórum de dicas e tutorias em desenvolvimento*. Disponível em: <http://silas.theducks.com.br/2014/05/16/doctrine-2-consulta-com-dql-envolvendo-tabela-many-to-many-nos-joins/>. Acessado em 08 de Novembro de 2014.

Rodrigo Lazoti's Blog [on-line] *Fórum de dicas e tutorias em desenvolvimento*. <http://rodrigolazoti.com.br/2008/09/01/filtrando-usuarios-logados-em-jsf-com-phaselistener/>. Acessado em 08 de Novembro de 2014.

GUJ [on-line] *Fórum de dicas e tutorias em desenvolvimento*. Disponível em: <http://www.guj.com.br/java/181327-transformar-arquivo-em-array-de-bytes> / <http://www.guj.com.br/java/246666-converter-de-byte-para-javaiofile>. Acessado em 08 de Novembro de 2014.

JournalDev [on-line] *Fórum de dicas e tutoriais em desenvolvimento*. Disponível em: <http://www.journaldev.com/3229/primefaces-fileupload-component-example-tutorial>. Acessado em 08 de Novembro de 2014.

K19 Treinamentos [on-line] *Apostila K-12 – Desenvolvimento Web com JSF 2 e JPA 2*. Disponível em: <http://online.k19.com.br/libraries/handouts>. Acessado em 08 de Novembro de 2014.

Caelum [on-line] *Apostila F-21 – Desenvolvimento Java Web*. Disponível em: <http://www.caelum.com.br/download/caelum-java-web-fj21.pdf>. Acessado em 08 de Novembro de 2014.

Laura Liparulo [on-line] *Fórum de dicas e tutoriais em desenvolvimento*. Disponível em: <http://lauraliparulo.altervista.org/primefaces-dialog-framework-how-to-make-it-work/>. Acessado em 08 de Novembro de 2014.

Wikipédia [on-line] Enciclopédia livre sobre o *Framework JSF*. Disponível em: http://pt.wikipedia.org/wiki/JavaServer_Faces. Acessado em 08 de Novembro de 2014.

FACULDADE CAMPO LIMPO PAULISTA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Desenvolvimento Web Ágil Utilizando Framework

ANEXO A – MANUAL DE USUÁRIO

Para acessar a aplicação *web*, basta acessar a *URL* (*Uniform Resource Locator*) “localhost:8080/WebRepositorio”

Ao acessá-la, será direcionado para a tela de *login*, caso seja a primeira utilização do sistema será exibido uma tela para cadastramento de Usuário para acesso as funcionalidades do sistema, conforme Figura A1.

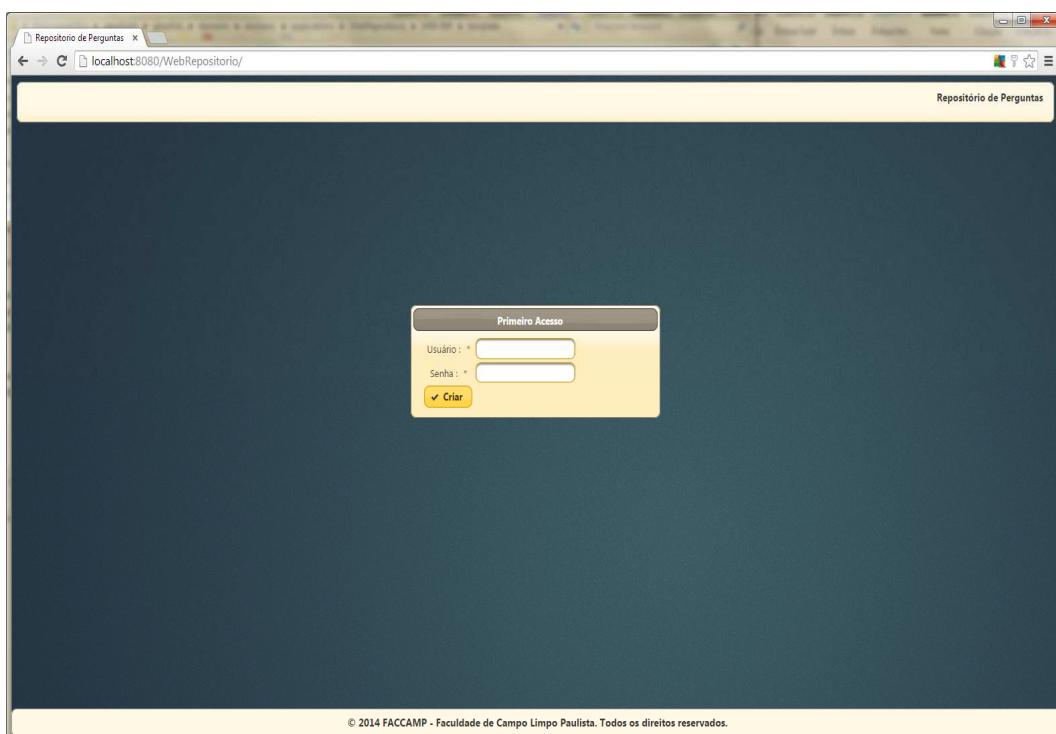


Figura A1: Tela de primeiro Acesso

Após o cadastramento do primeiro acesso, uma mensagem lateral na extrema direita informará a criação com sucesso do usuário, conforme demonstrado na Figura A2, e direcionará para o *login* no sistema.

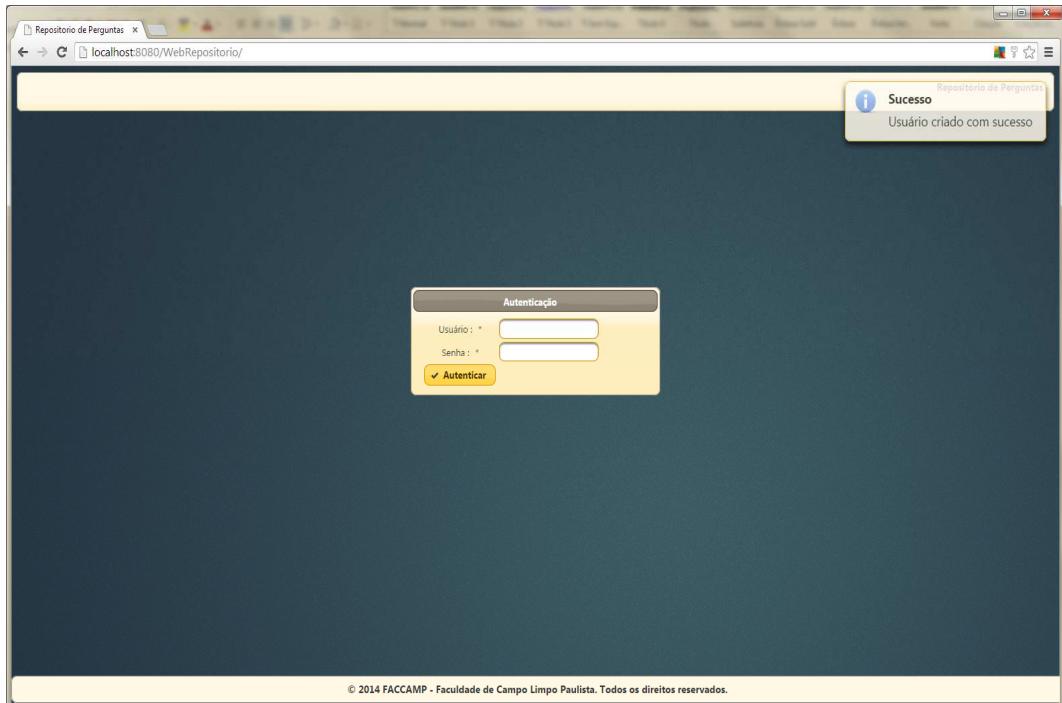


Figura A2: Mensagem de criação de Usuário

Na tela *home*, conforme representada na Figura A3, possuirá as seguintes funcionalidades:

- Cadastramento de Pergunta, conforme Figura A4;
- Cadastramento de Gênero, conforme Figura A7;
- Cadastramento de *Tag's*, conforme Figura A10;
- Alteração de Pergunta, conforme Figura A13;
- Alteração de Gênero, conforme Figura A16;
- Alteração de *Tag's*, conforme Figura A19;
- Remoção de Pergunta, conforme Figura A22;
- Remoção de Gênero, conforme Figura A25;
- Remoção de *Tag's*, conforme Figura A28;
- Associação de *Tag's* a Perguntas, conforme Figura A31;
- Desassociação de *Tag's* a Perguntas, conforme Figura A34;
- Consulta Perguntas, conforme Figura A37;
- Consulta Gêneros, conforme Figura A38;
- Consulta *Tag's*, conforme Figura A40;
- Consulta *Tag's* associadas a Perguntas conforme Figura A41;
- Usuários, conforme Figura A42;

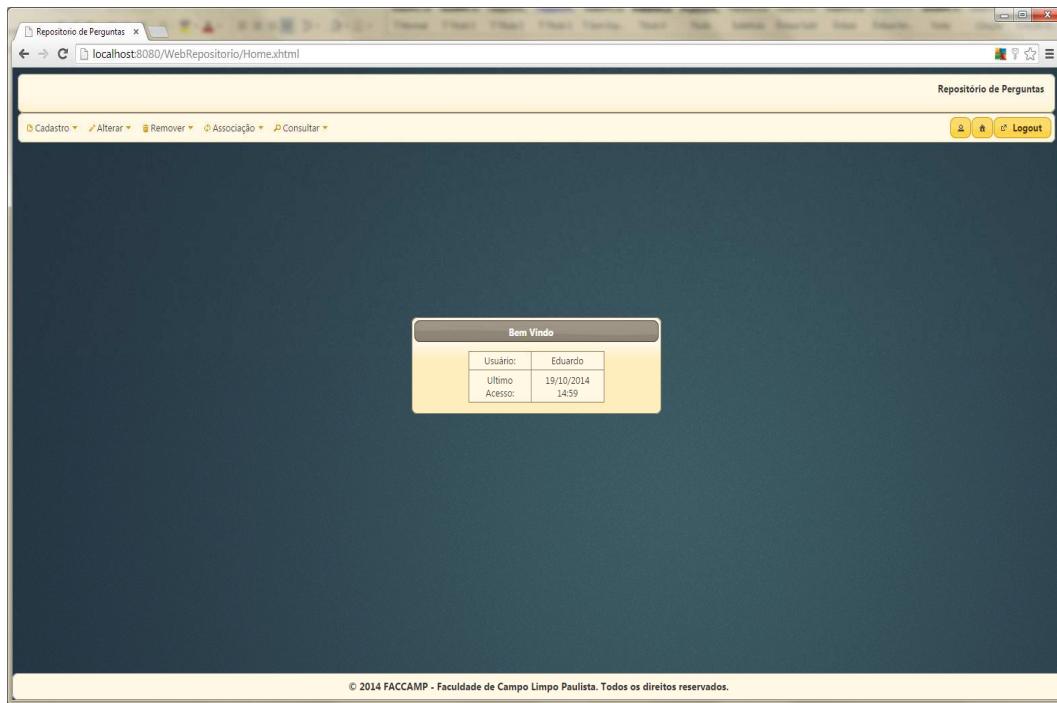


Figura A3: Tela Home

Cadastramento de Gêneros

Através desta funcionalidade, conforme demonstrado nas Figuras A4, A5 e A6, é possível fazer o cadastramento dos Gêneros que posteriormente poderão ser associados(s) a(s) pergunta(s).

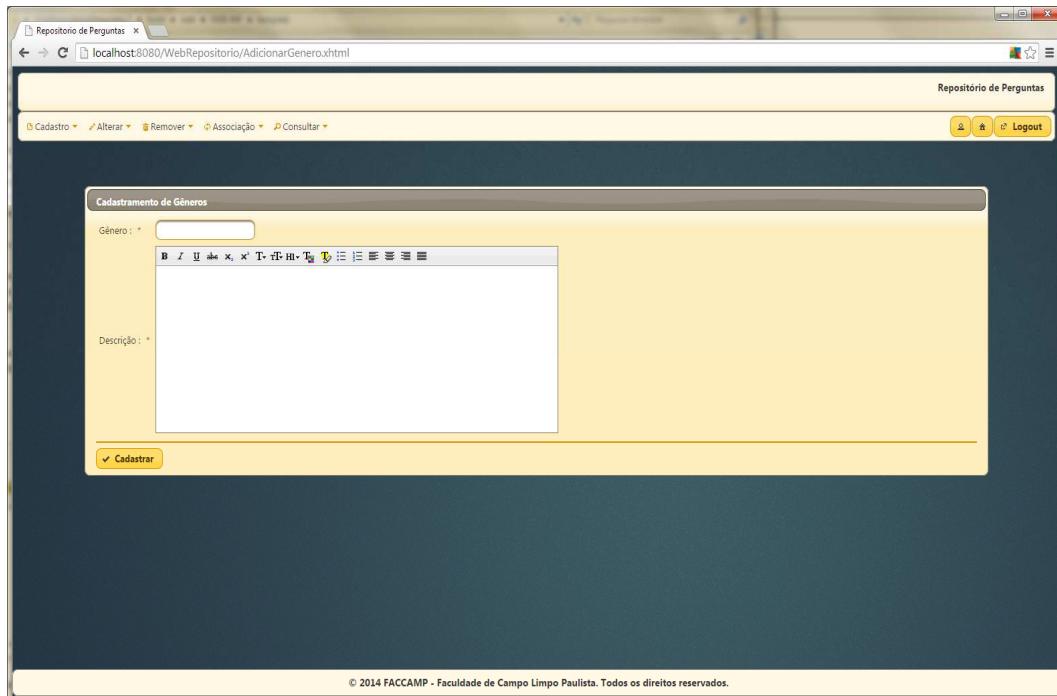


Figura A4: Tela cadastramento Gênero

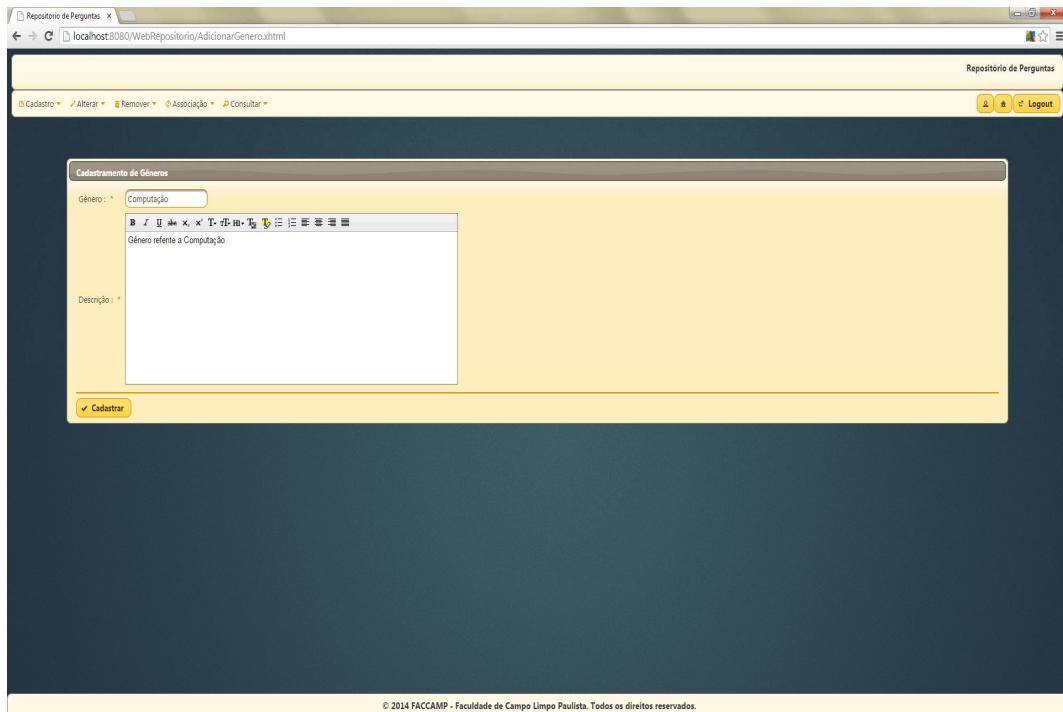


Figura A5: Tela Demonstração cadastramento Gênero

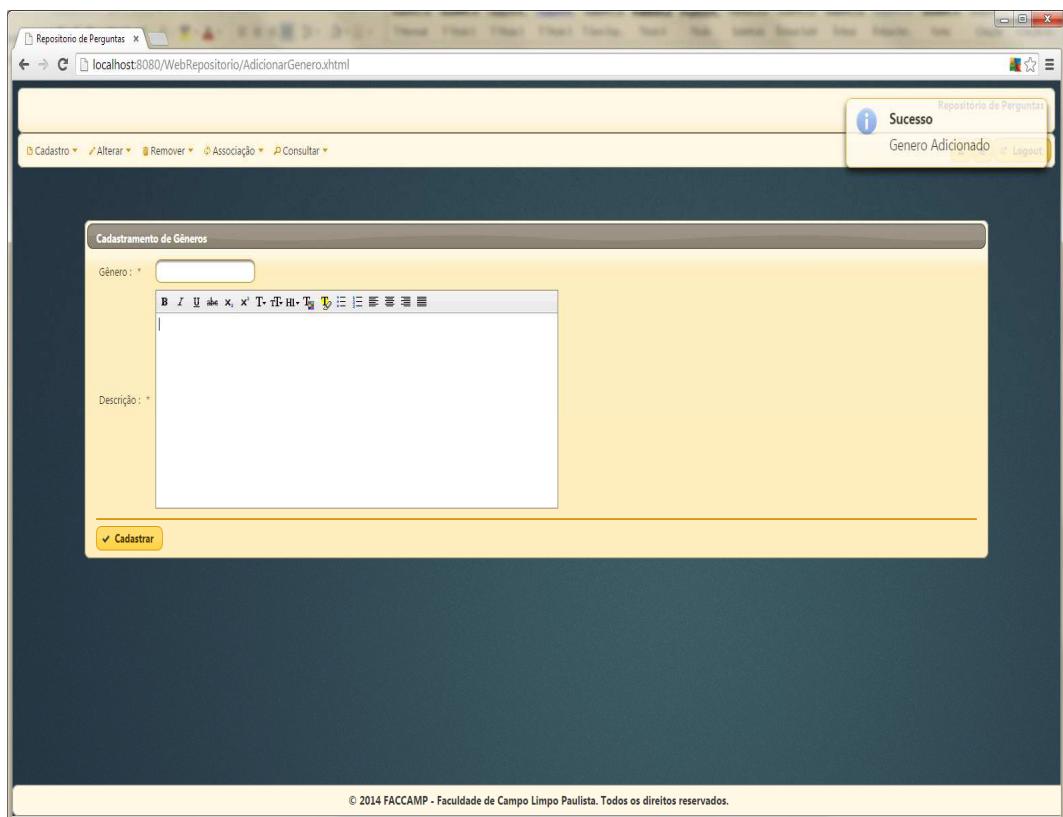


Figura A6: Tela mensagem de sucesso cadastramento Gênero

Cadastramento de Tag's

Através desta funcionalidade é possível efetuar o cadastramento de Tag's, no qual poderá ser associada as perguntas cadastradas previamente no sistema, conforme demonstrado nas Figuras A7, A8 e A9.

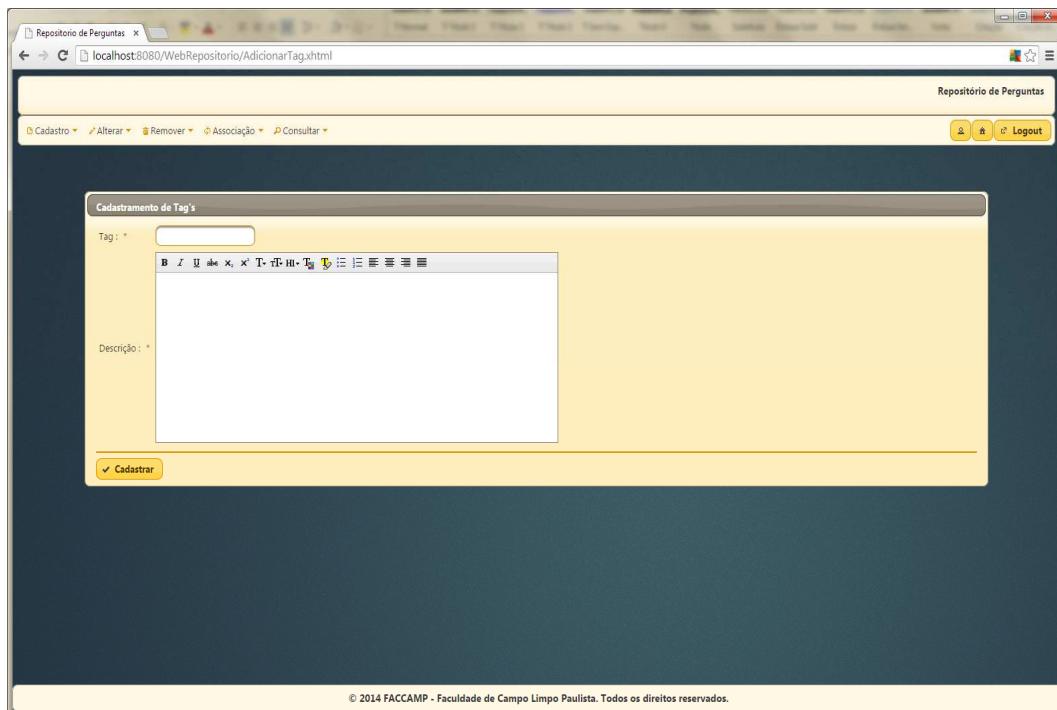


Figura A7: Tela cadastramento de Tag's

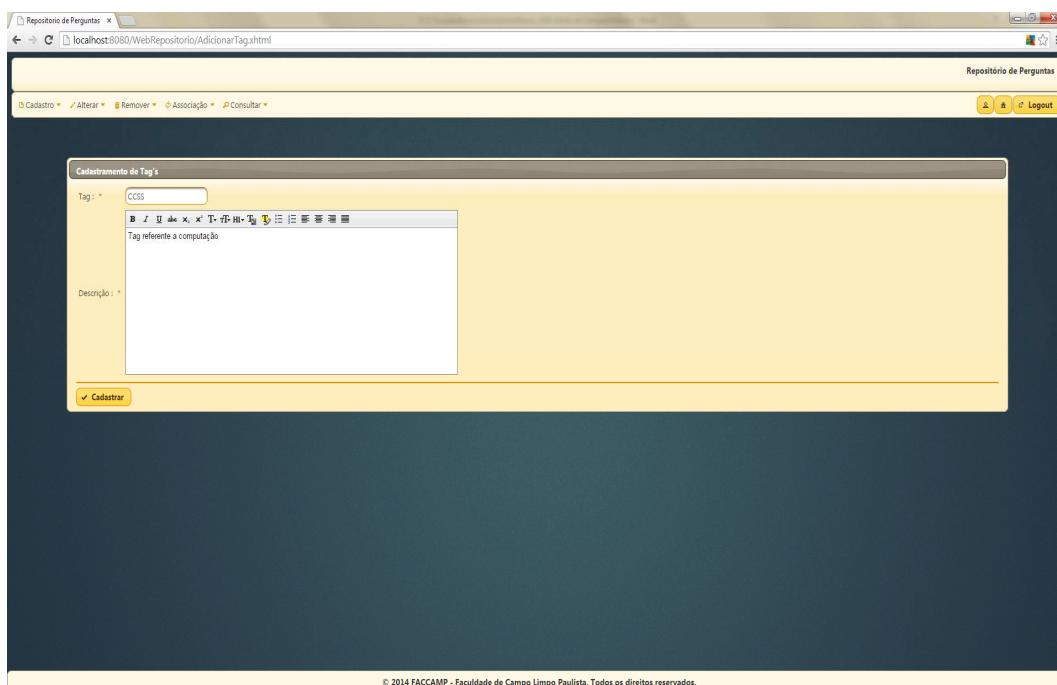


Figura A8: Tela demonstração do cadastramento de Tag's

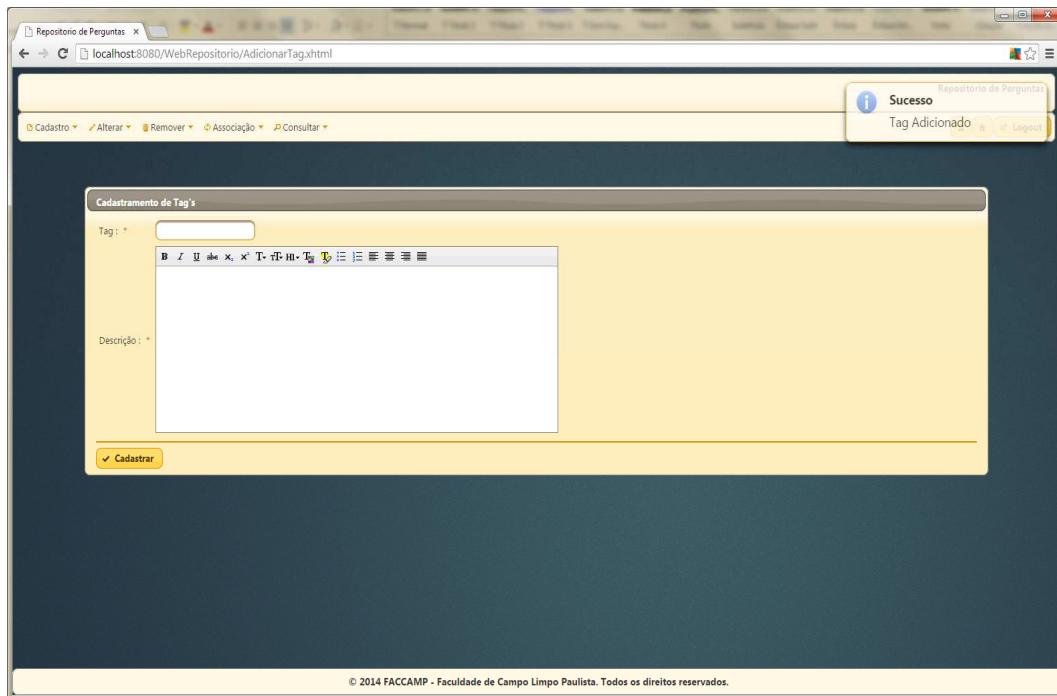


Figura A9: Tela mensagem de sucesso do cadastramento de *Tag's*

Cadastramento de Perguntas

Através desta funcionalidade, é possível efetuar o cadastramento de perguntas e suas respectivas alternativas, sendo permitido o upload de uma imagem para cada pergunta nos seguintes formatos *PNG*, *GIF*, *JPG* e *JPEG*, conforme demonstrado nas seguintes Figuras A10, A11 e A12.

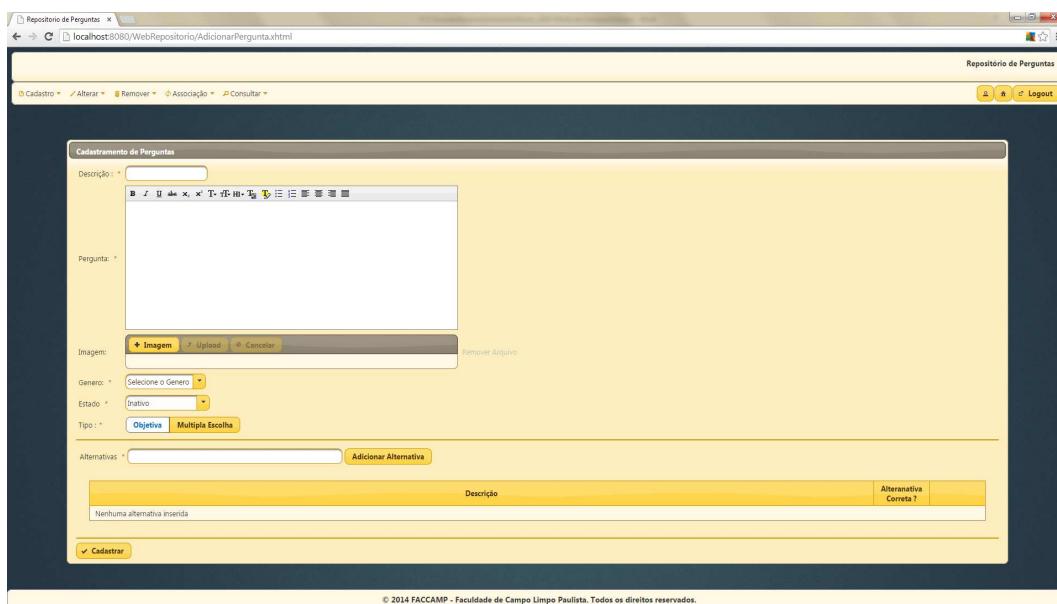


Figura A10: Tela cadastramento de Perguntas

Figura A11: Demonstração cadastramento de Perguntas

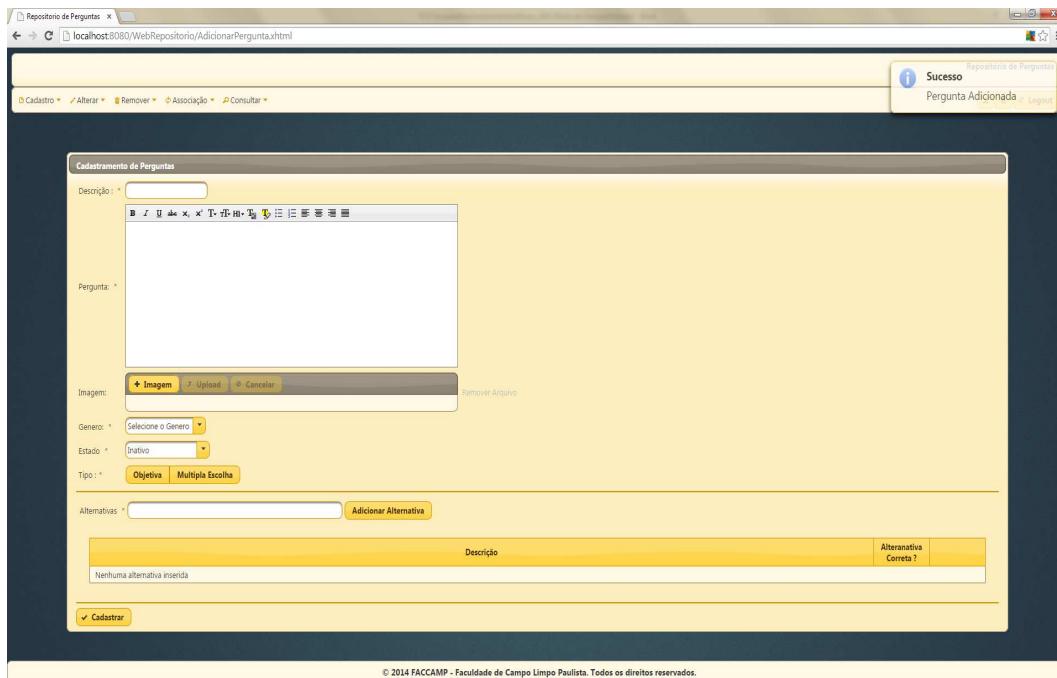


Figura A12: Mensagem de sucesso de cadastramento de Perguntas

Alteração de Gêneros

Através desta funcionalidade, é possível fazer a alteração dos gêneros anteriormente cadastrados no sistema, conforme demonstrado nas Figuras A13, A14 e A15 respectivamente.

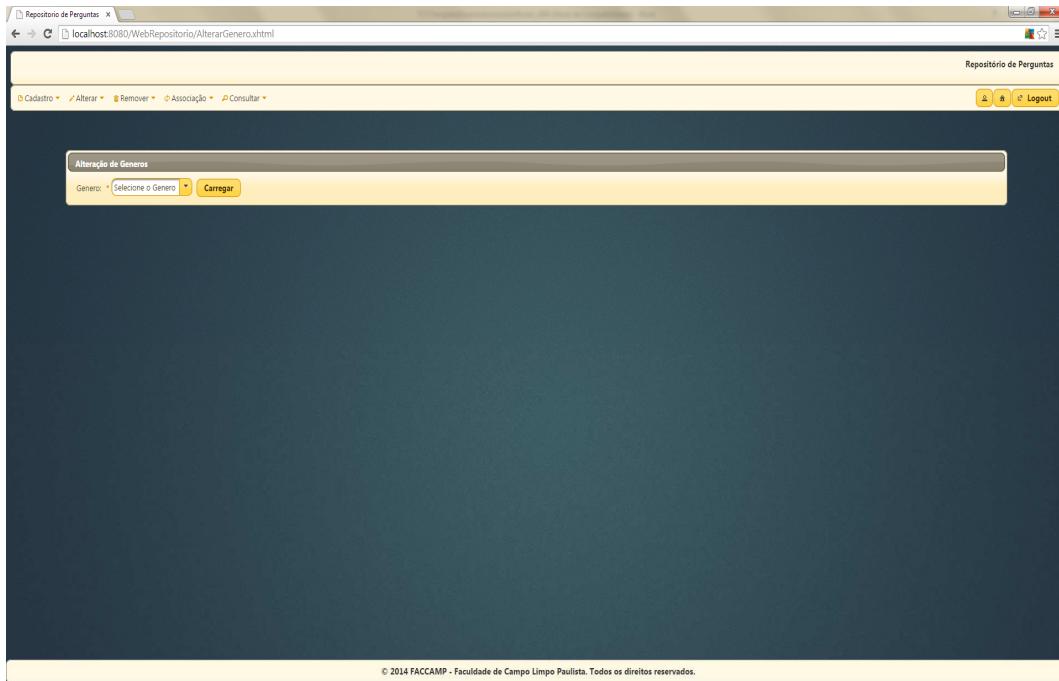


Figura A13: Tela alteração de Gênero

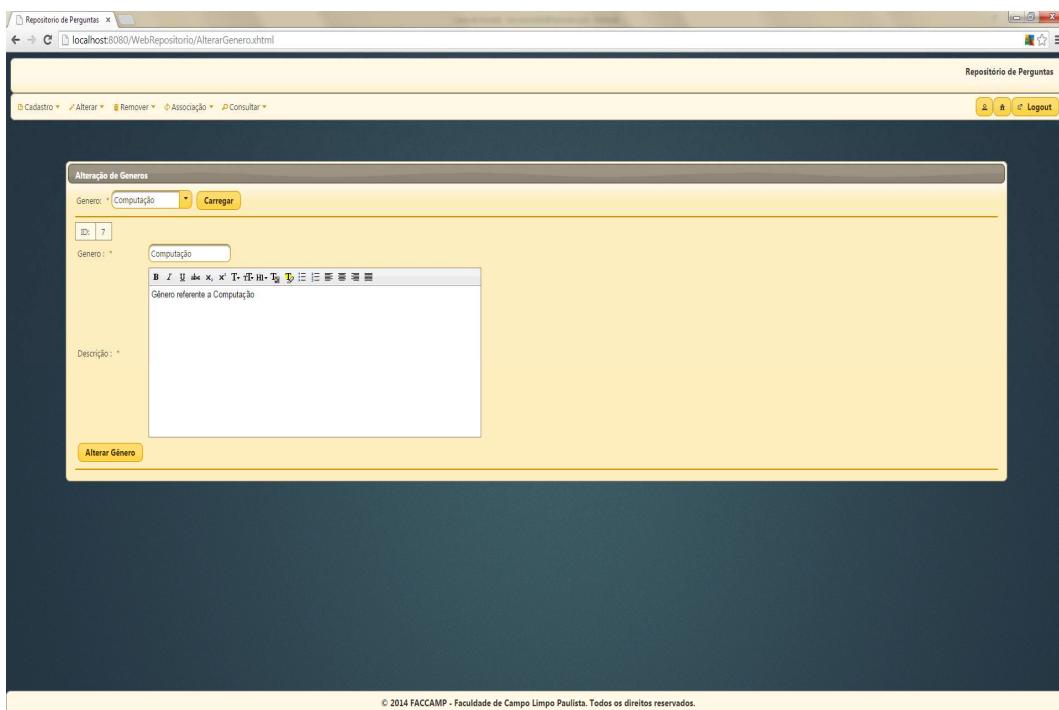


Figura A14: Demonstração alteração de Gênero

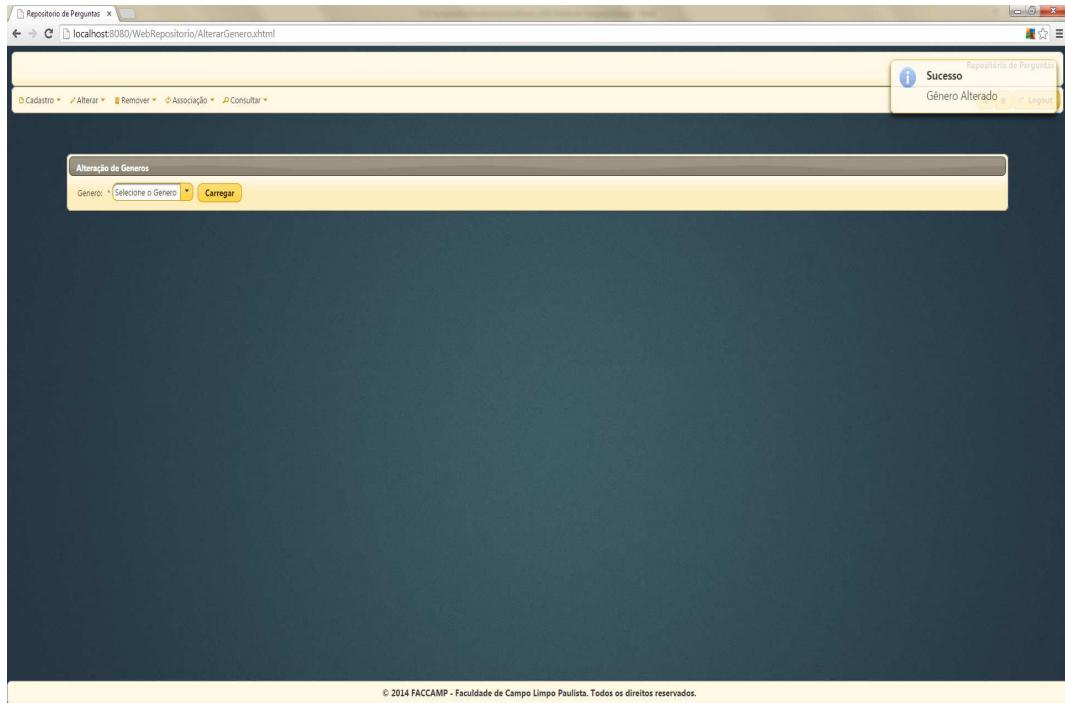


Figura A15: Mensagem de sucesso de alteração de Gênero

Alteração de Pergunta

Através desta funcionalidade, é possível fazer a alteração de perguntas anteriormente cadastradas, com possibilidades de alteração das alternativas e sua imagem associada, conforme demonstrado nas Figuras A16, A17 e A18.

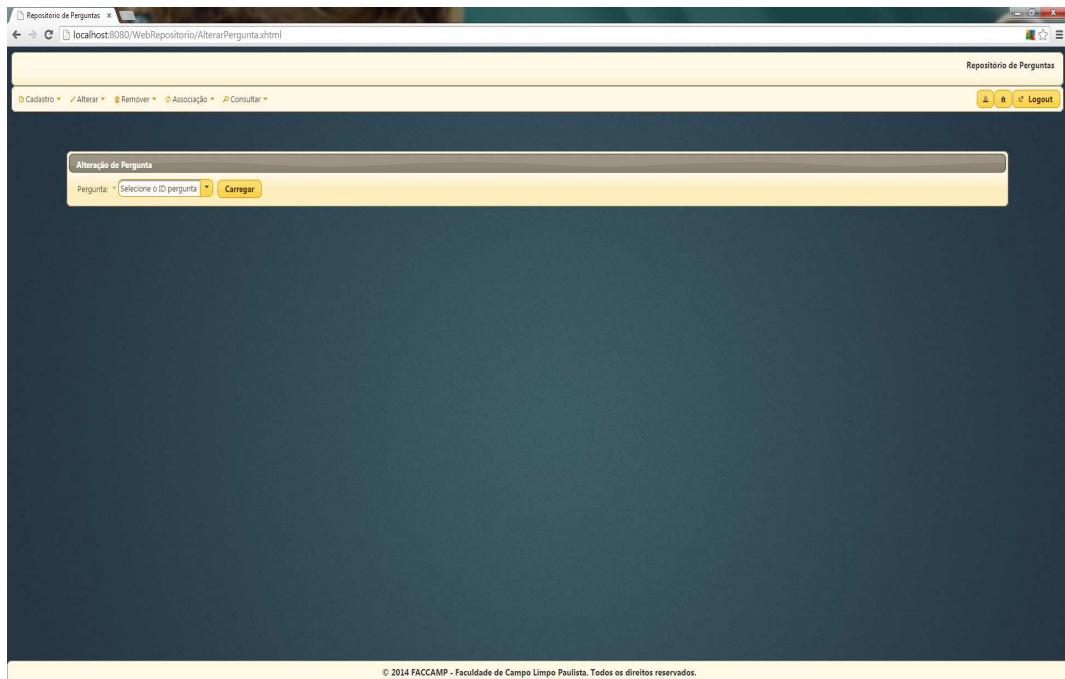


Figura A16: Tela alteração de Perguntas

Pergunta: * 9 Carregar

Id: 9

Descrição: * Pergunta 1

A figura abaixo ilustra uma imagem binária com pixels brancos formando retas sobre um fundo preto. Com relação à aplicação de transformadas sobre essa imagem, assinale a opção correta.

Pergunta: *

Imagem:

Genero: * Computação

Estado: * Ativo

Tipo: * Objetiva Multipla Escolha

Respostas		Alternativas	
ID	Resposta	Tipo	
42	A transformada de Fourier, quando aplicada à imagem descrita, produz como resultado um mapa de frequências que equivale ao histograma dos níveis de cinza das retas presentes.	Falsa	<input checked="" type="checkbox"/> Remover
43	A transformada de Hadamard da imagem apresentada tem resultado equivalente à aplicação de um filtro passa-baixa, o que desfoca as retas existentes.	Falsa	<input checked="" type="checkbox"/> Remover
44	Ao se aplicar a transformada da distância à imagem binária, considerando pixels brancos como objetos, é possível identificar a distância das retas presentes e o centro da imagem, o que permite identificar as equações das retas formadas na imagem.	Falsa	<input checked="" type="checkbox"/> Remover
45	O uso da transformada dos cosenos produz uma lista dos coeficientes lineares e angulares das diversas retas existentes nessa imagem binária.	Falsa	<input checked="" type="checkbox"/> Remover
46	O resultado da aplicação da transformada de Hough usando parametrização de retas é um mapa cujos picos indicam os pixels colineares, permitindo que sejam identificados coeficientes que descrevem as diversas retas formadas na imagem.	Verdadeira	<input checked="" type="checkbox"/> Remover

Alternativas: Adicionar Alternativa

Alterar

© 2014 FACCAMP - Faculdade de Campo Limpo Paulista. Todos os direitos reservados.

Figura A17: Demonstração alteração de Perguntas

Cadastro Alterar Remover Associação Consultar

Alteração de Pergunta

Pergunta: * Selecione o ID pergunta Carregar

Sucesso

Pergunta Alterada

© 2014 FACCAMP - Faculdade de Campo Limpo Paulista. Todos os direitos reservados.

Figura A18: Mensagem de sucesso alteração de Pergunta

Alteração de Tag's

Através desta funcionalidade, é possível fazer alteração das Tag's anteriormente cadastradas, conforme demonstrado nas Figuras A19, A20 e A21

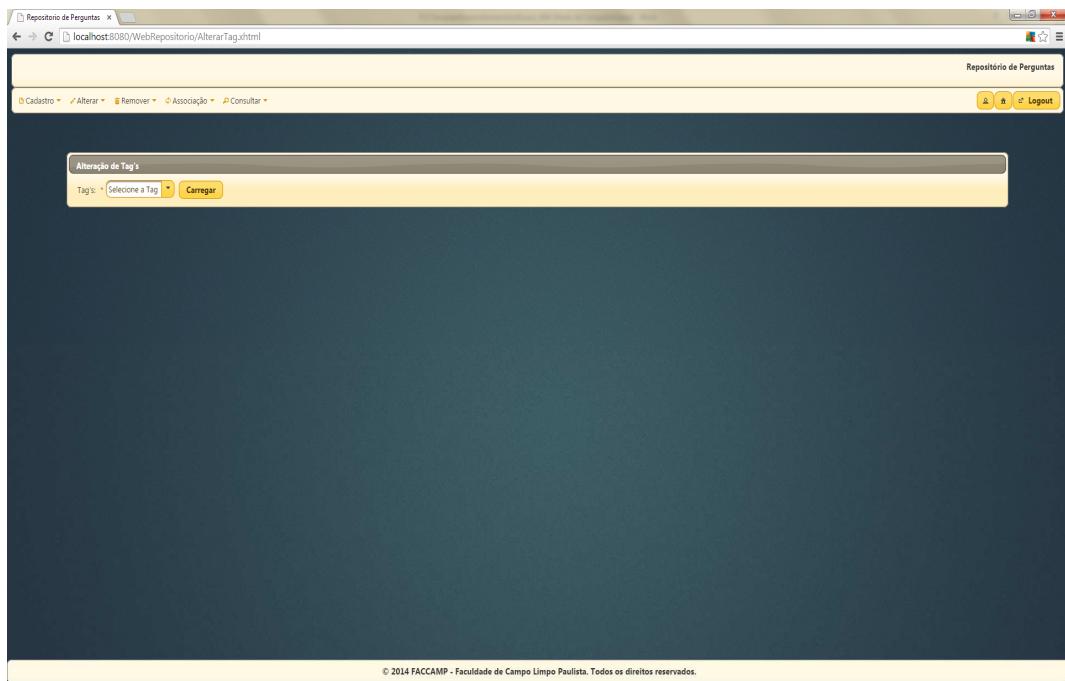


Figura A19: Tela de alteração de *Tag's*

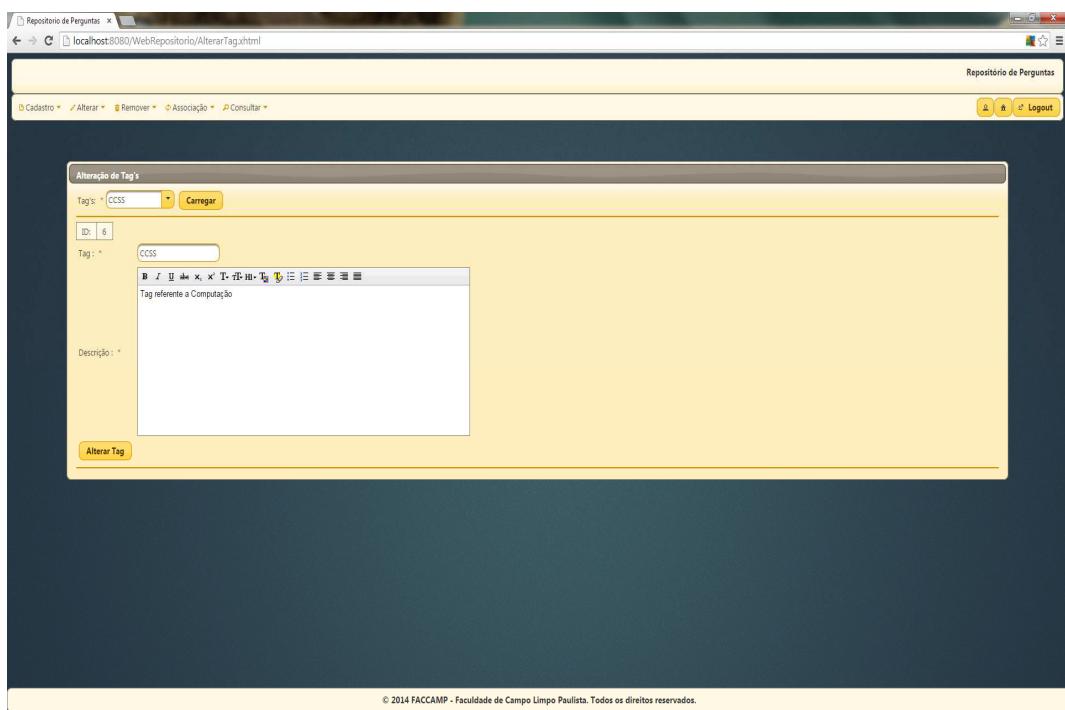


Figura A20: Demonstração de alteração de *Tag's*

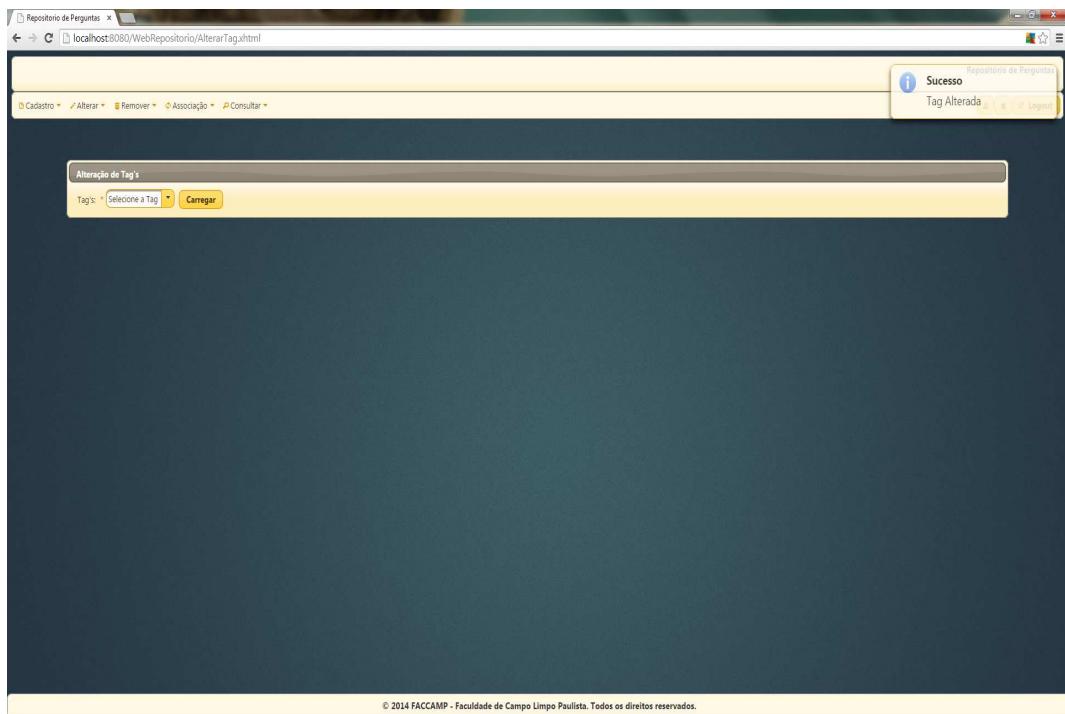


Figura A21: Mensagem de sucesso alteração de Tag's

Remoção de Gênero

Através desta funcionalidade, é possível fazer a remoção de gêneros, deste que não estejam associadas a nenhuma pergunta cadastrada no sistema, conforme demonstrado nas Figuras A22, A23 e A24.

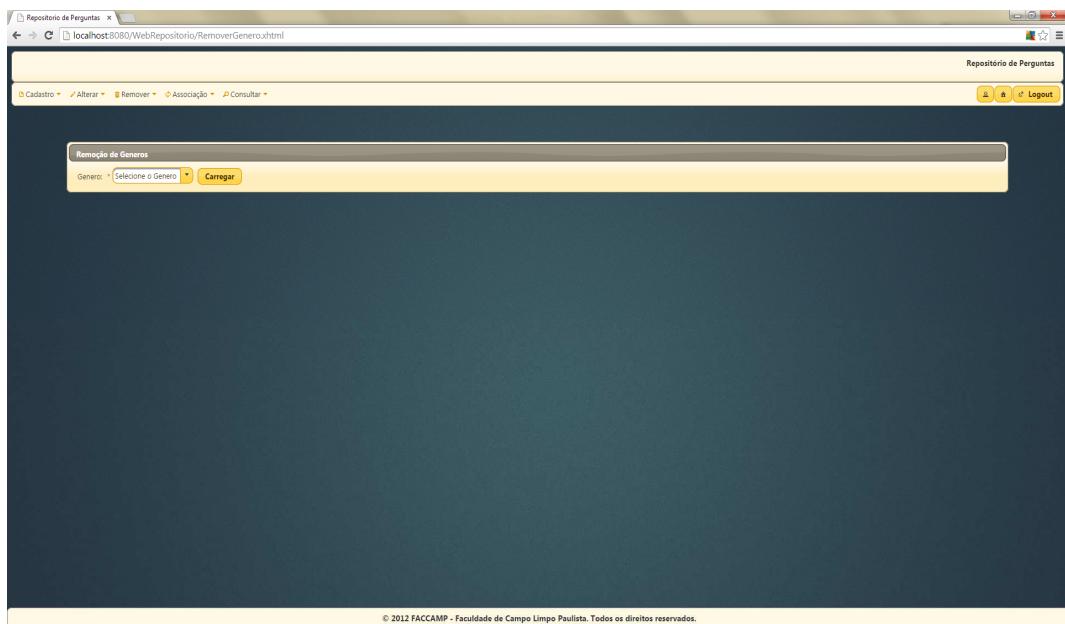


Figura A22: Tela de remoção de gênero

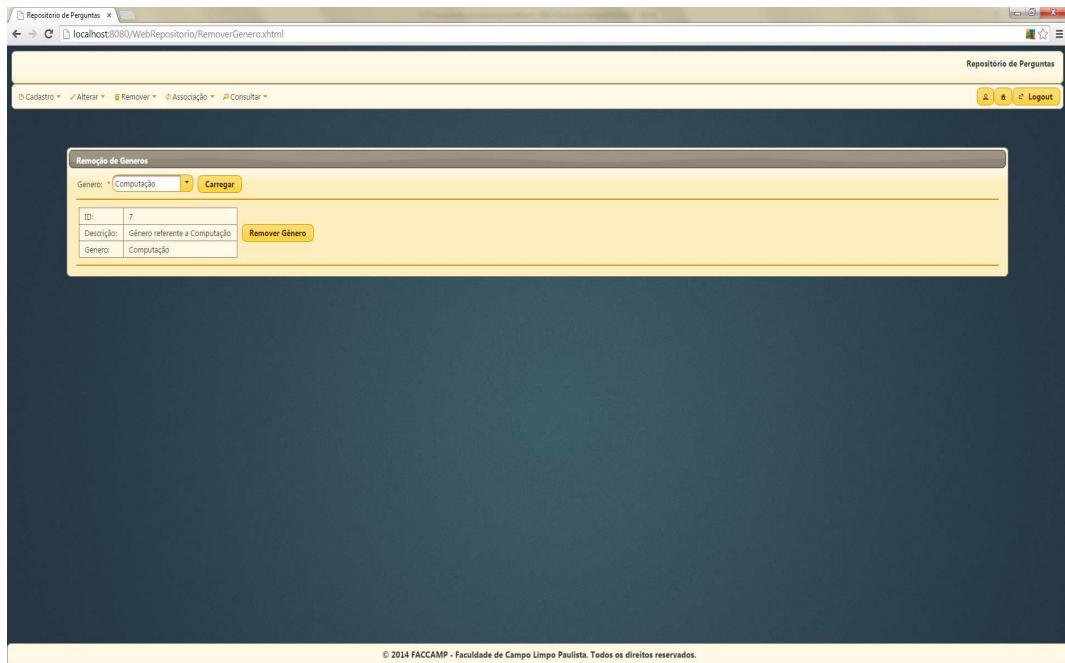


Figura A23: Demonstração remoção de gênero

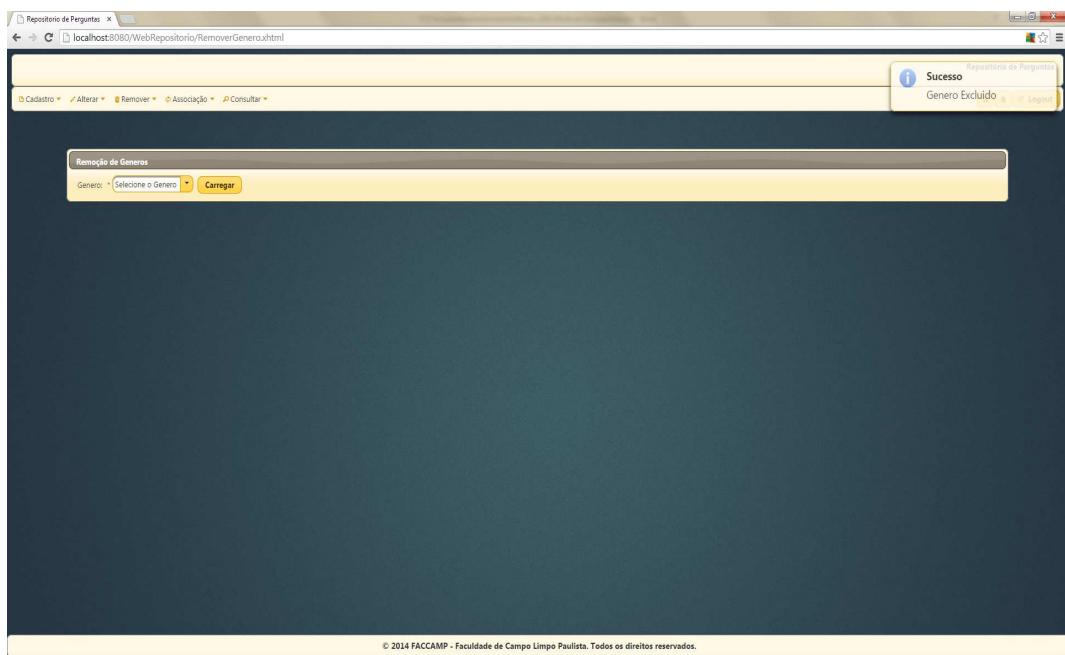


Figura A24: Mensagem de sucesso de remoção de gênero

Remoção de Pergunta

Através desta funcionalidade, é possível fazer a remoção das perguntas anteriormente cadastradas em cascata com suas alternativas associadas e também qualquer imagem associada, conforme demonstrado nas Figuras A25, A26 e A27

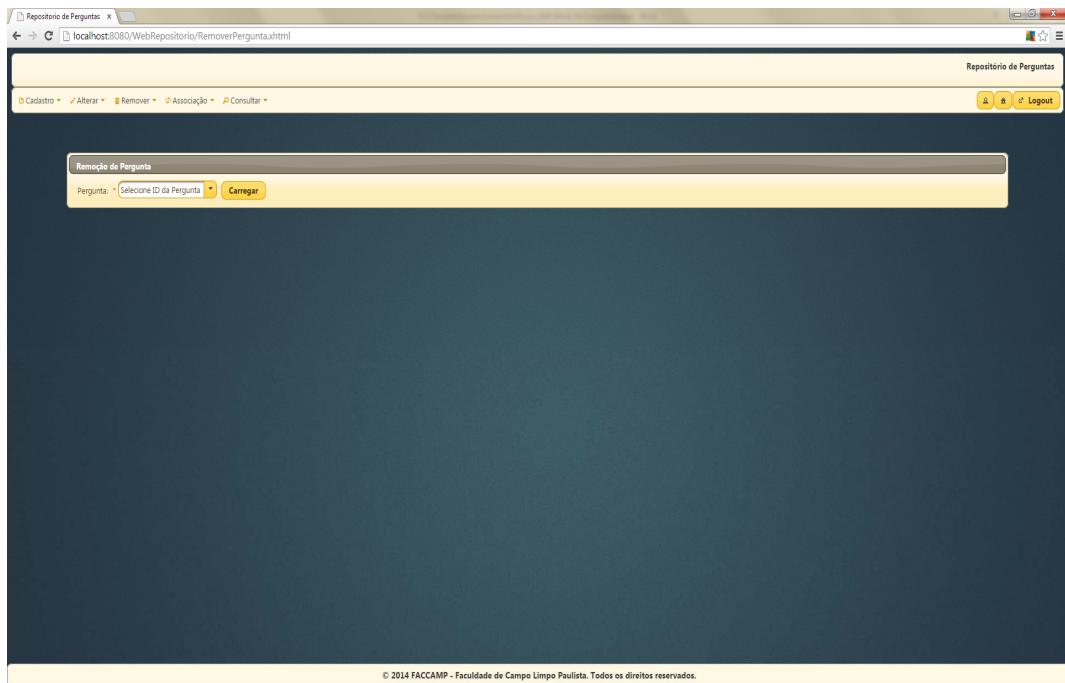


Figura A25: Tela remoção de pergunta

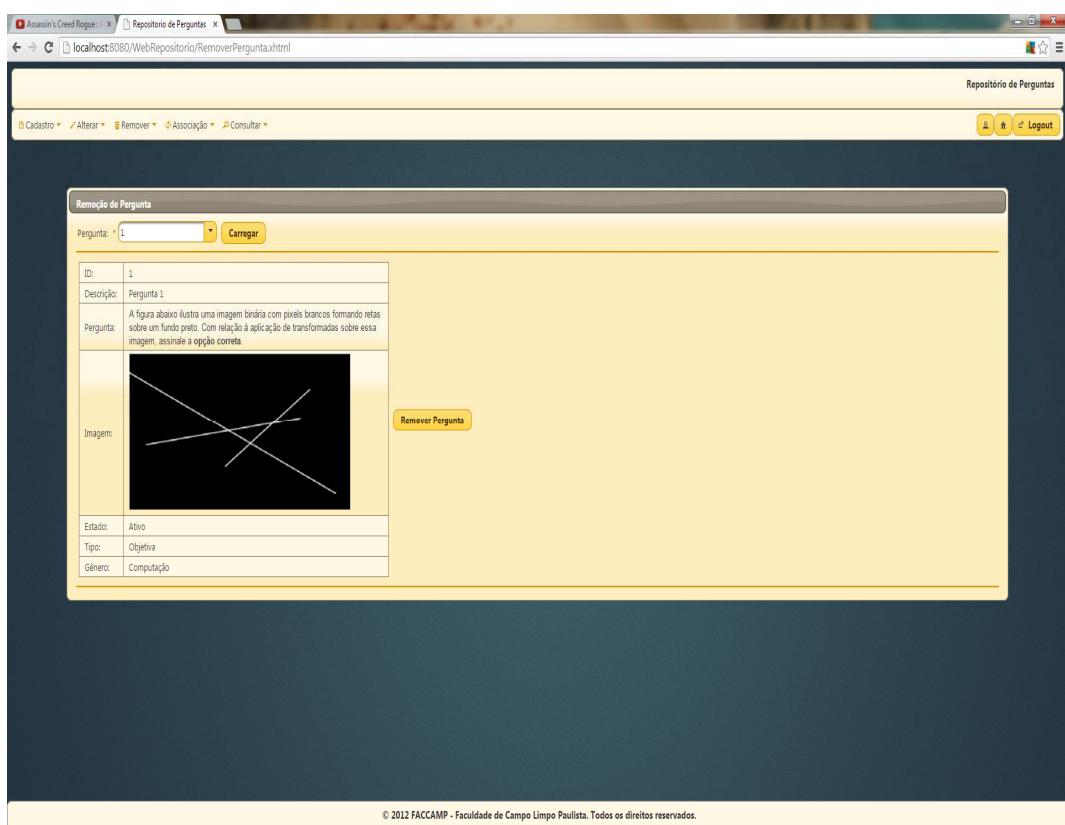


Figura A26: Demonstração remoção de pergunta

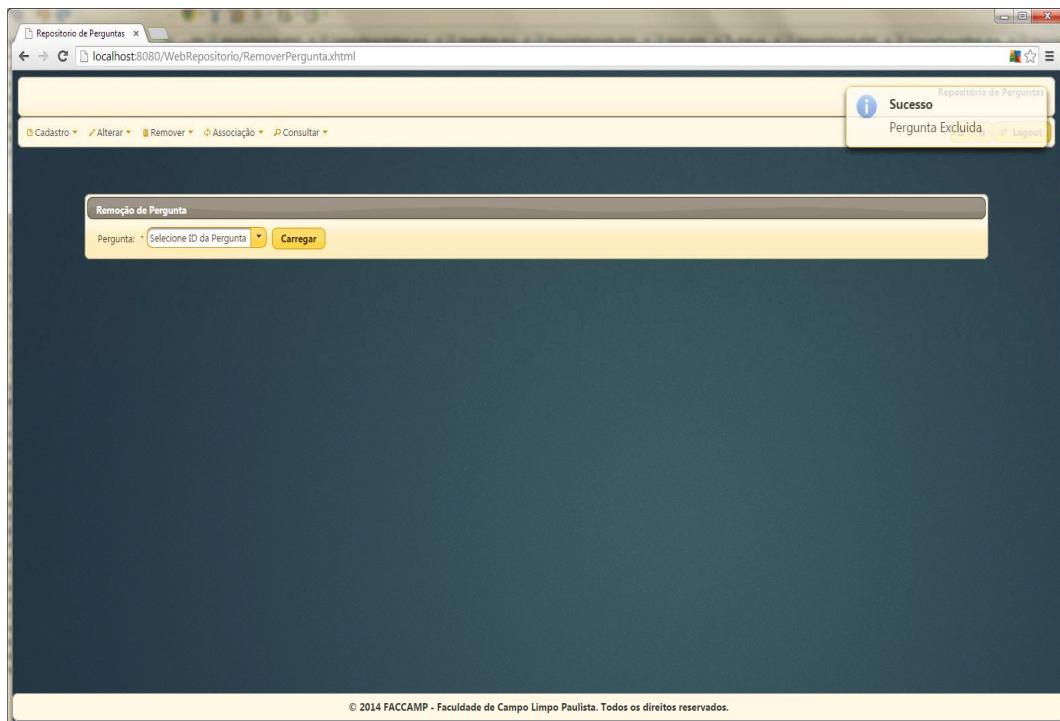


Figura A27: Mensagem de sucesso de remoção de pergunta

Remoção de Tag's

Através desta funcionalidade, é possível efetuar a remoção de Tag's previamente cadastradas no sistema, independentemente de estar ou não associada a pergunta(s) conforme demonstrados nas Figuras A28, A29 e A30.

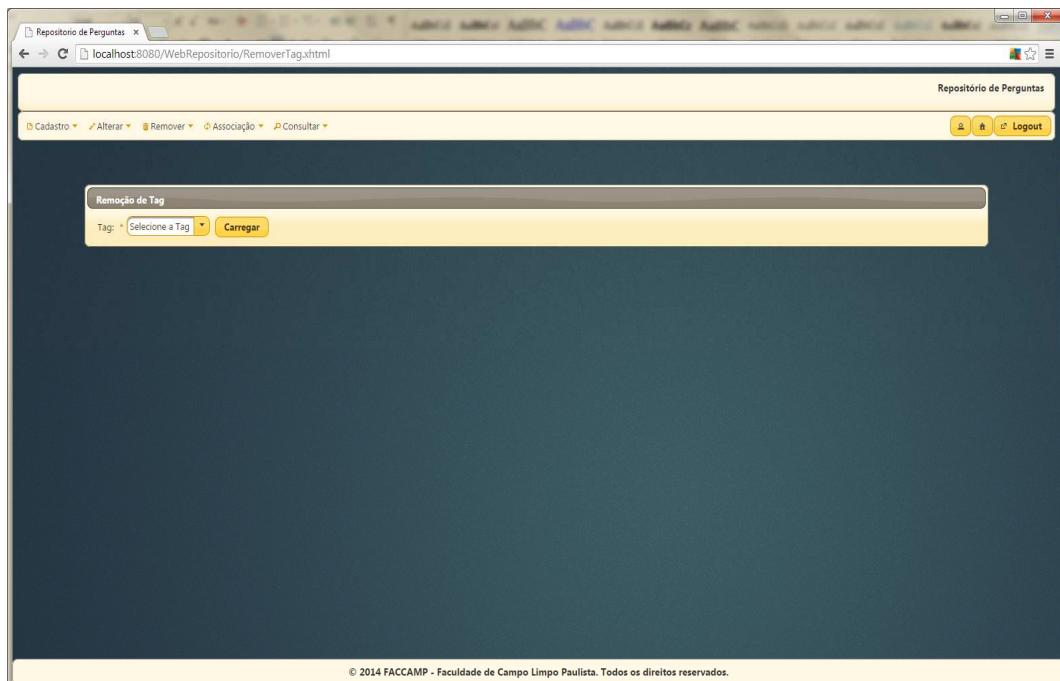


Figura A28: Tela de remoção de Tag's

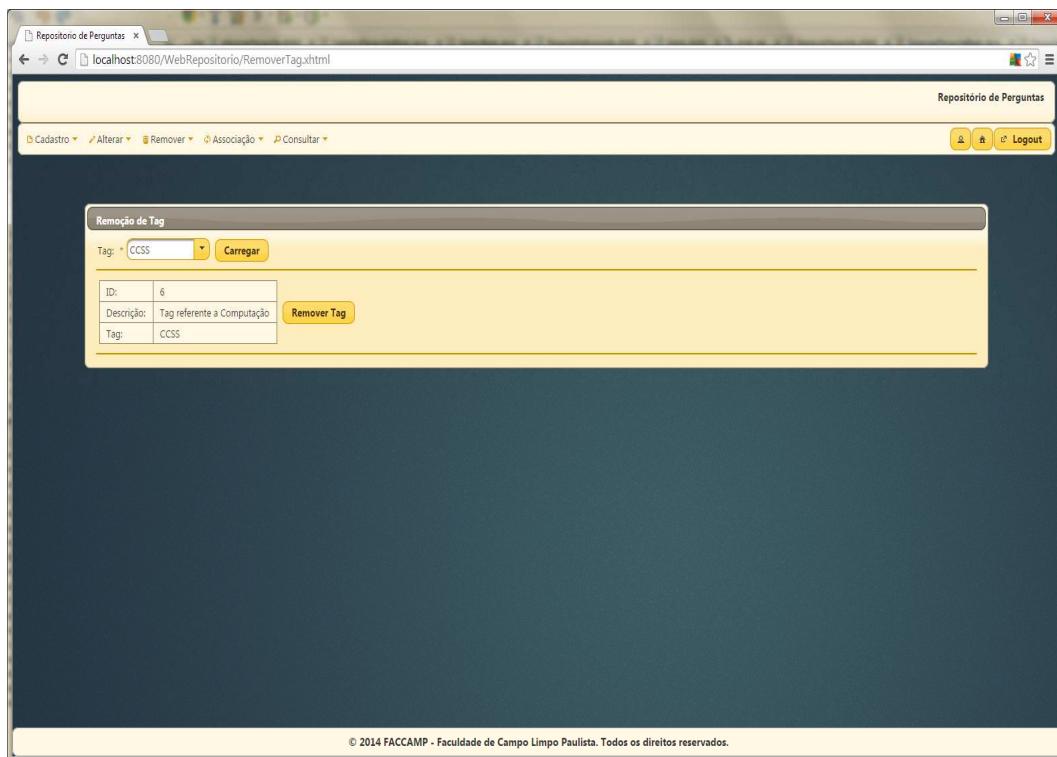


Figura A29: Demonstração remoção de Tag's

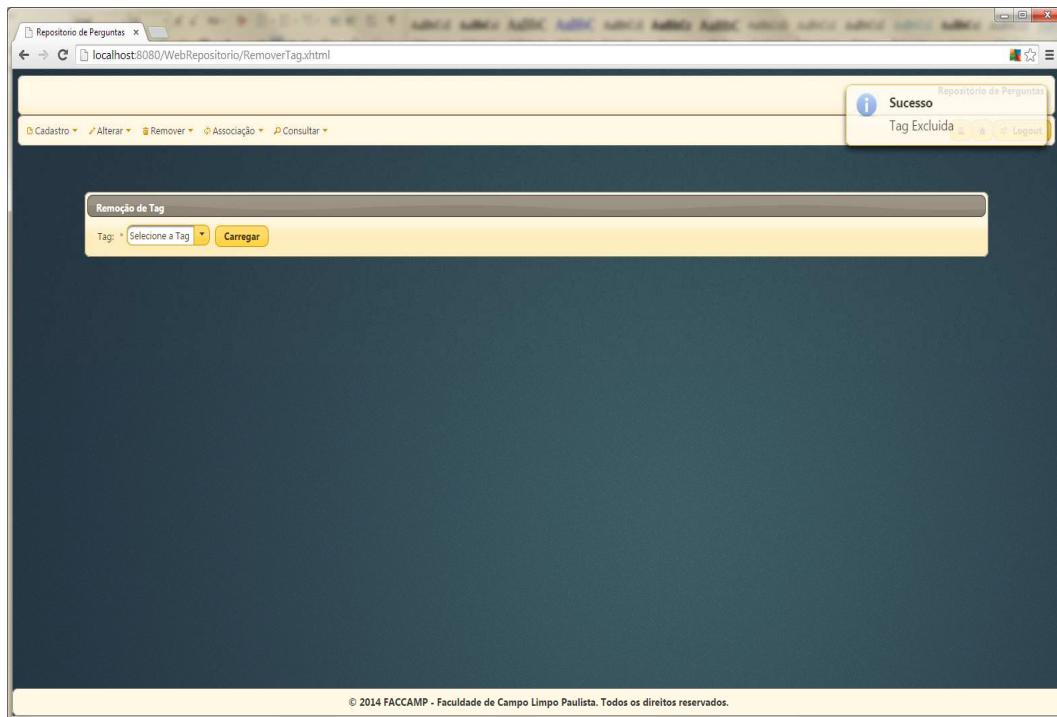


Figura A30: Mensagem de sucesso de remoção de Tag's

Associação de Tag's

Através desta funcionalidade, é possível fazer associação de Tag's a diversas perguntas previamente cadastradas no sistema, conforme demonstrado nas Figuras A31, A32 e A33

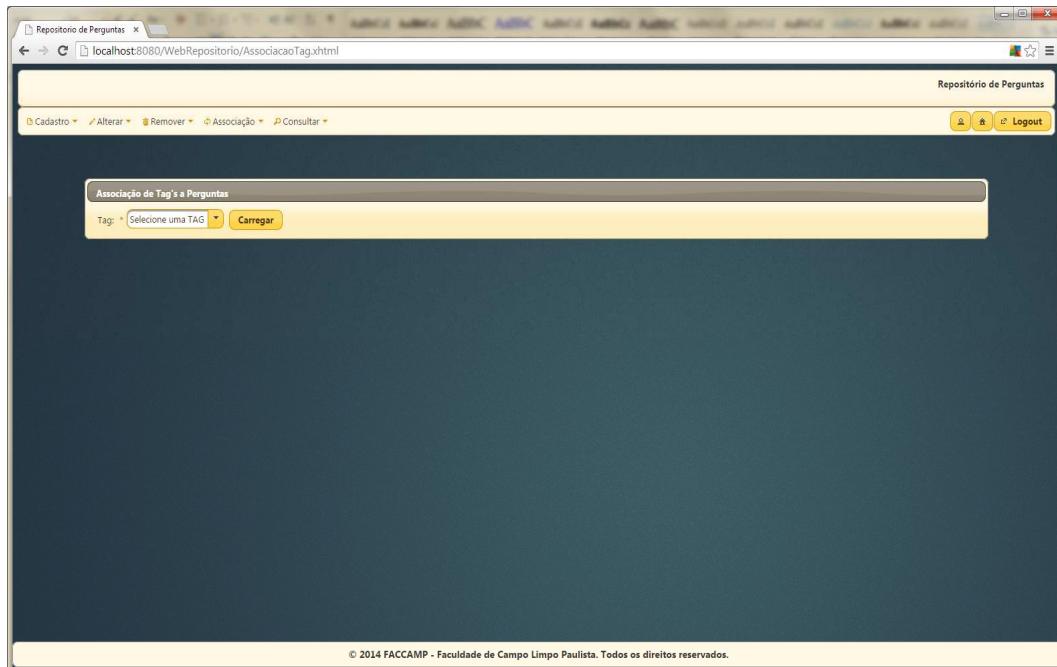


Figura A31: Tela associação de Tag's

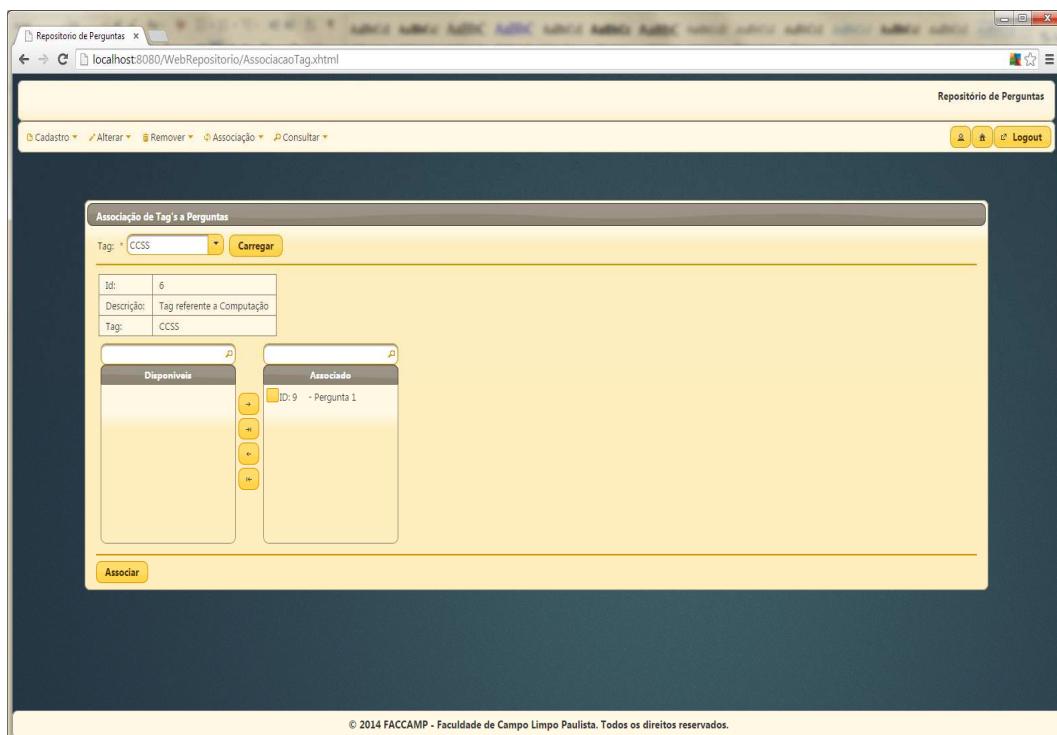


Figura A32: Demonstração associação de Tag's

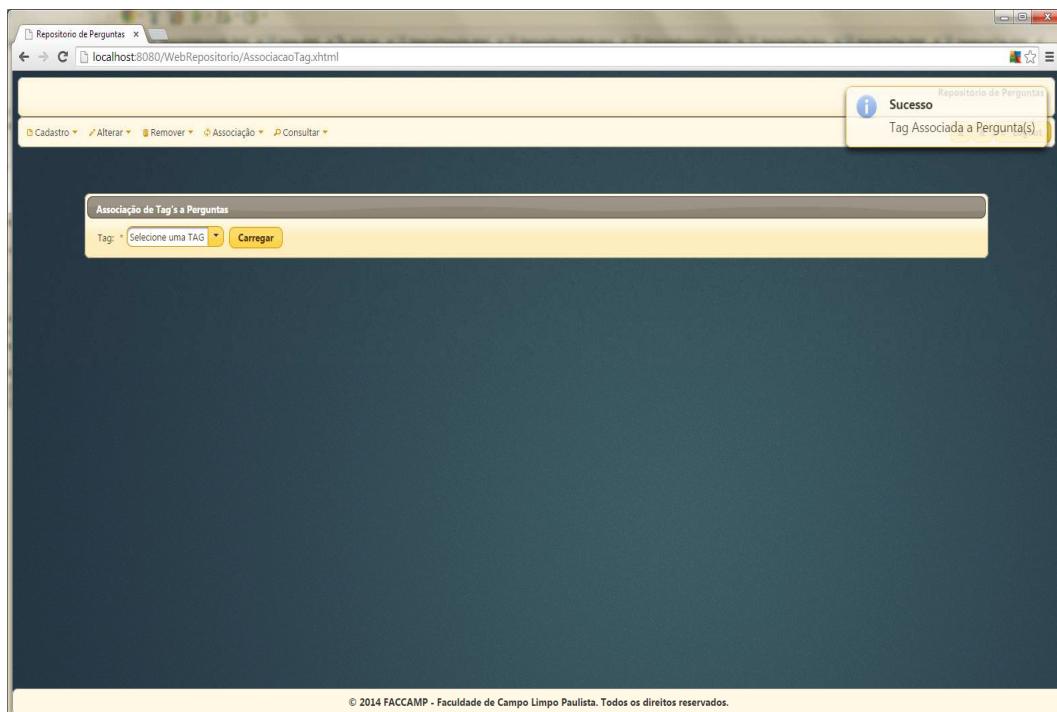


Figura A33: Mensagem de sucesso da associação de Tag's

Desassociação Tag's de Perguntas

Através desta funcionalidade, é possível efetuar a desassociação de pergunta(s) anteriormente associadas a determinada Tag's, conforme demonstrado nas Figuras A34, A35 e A36.

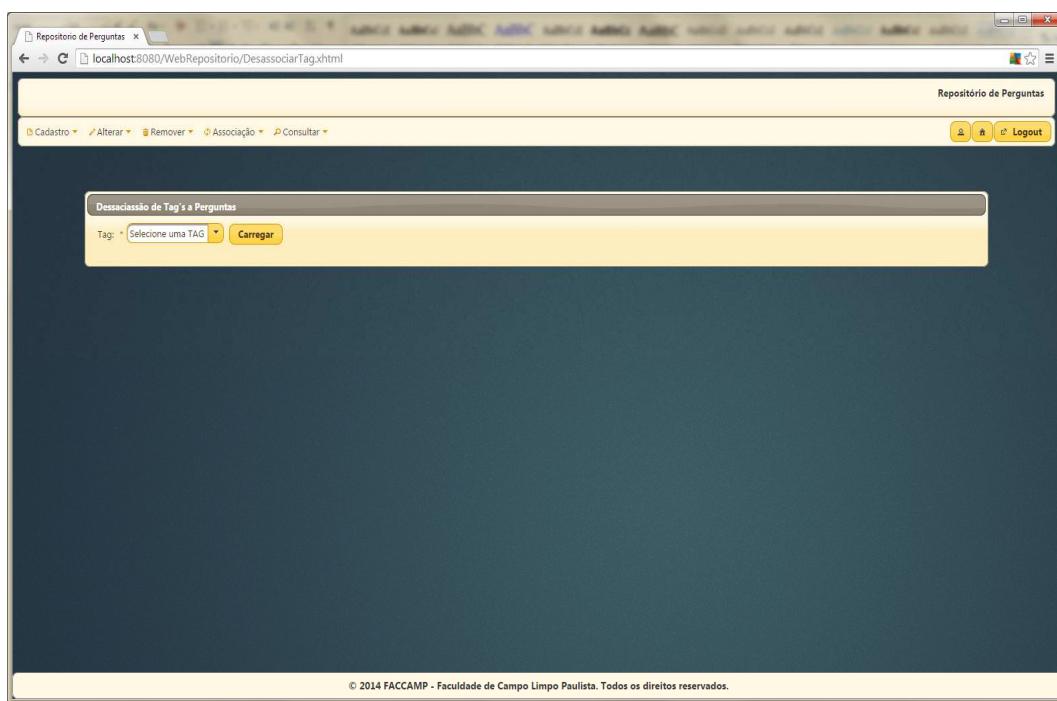


Figura A34: Tela desassociação de Tag's

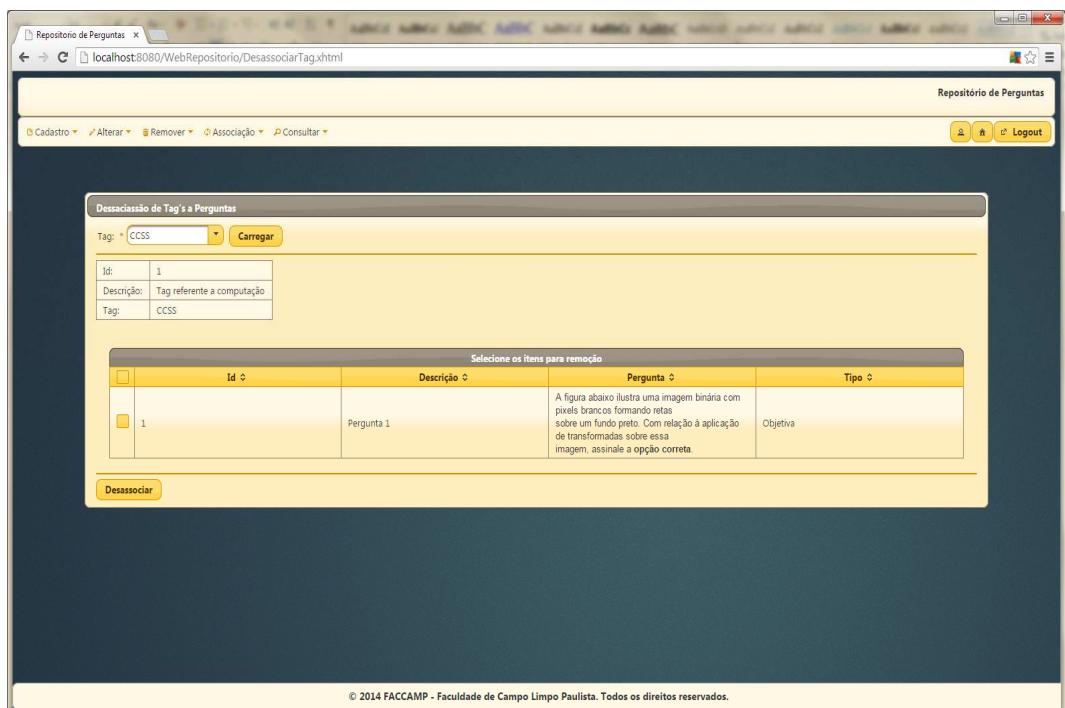


Figura A35: Demonstração de desassociação de Tag's

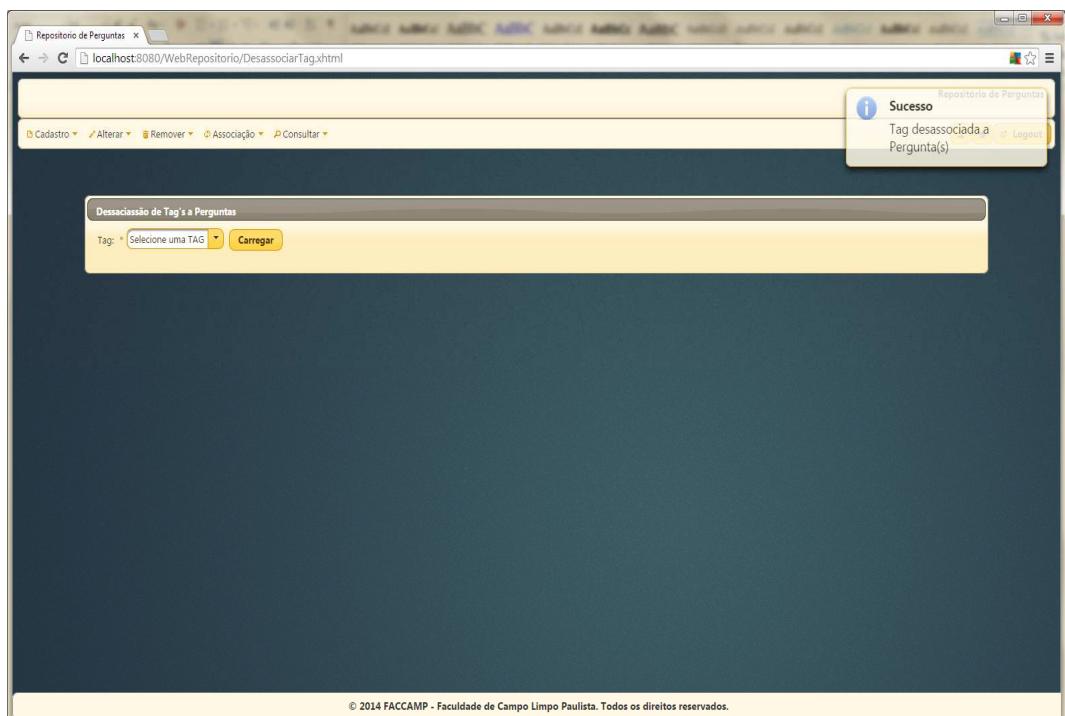


Figura A36: Mensagem de Sucesso de desassociação de Tag's

Consulta Gêneros

Através desta funcionalidade, é possível consultar todas os gêneros cadastrados no sistema, no qual possui filtros para seleção de determinado(s) gênero(s) conforme demonstrado na Figura A37.

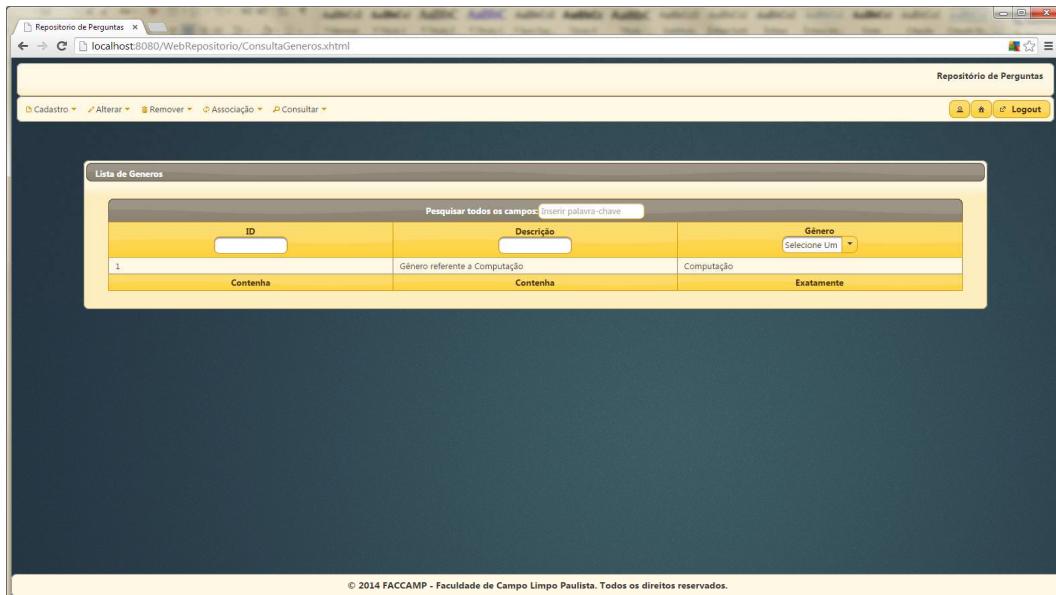


Figura A37: Tela consulta de Gêneros

Consulta Pergunta

Através desta funcionalidade é possível consultar perguntas cadastradas no sistema, no qual possui filtros para seleção de determinada(s) pergunta(s), conforme demonstrado nas Figuras A38 e A39.

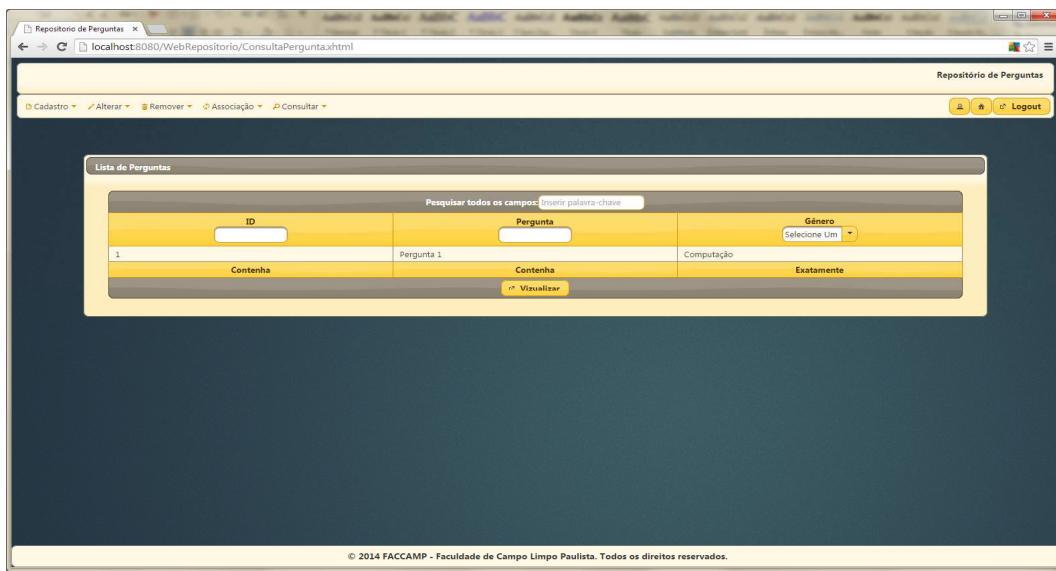


Figura A38: Tela consulta de Pergunta

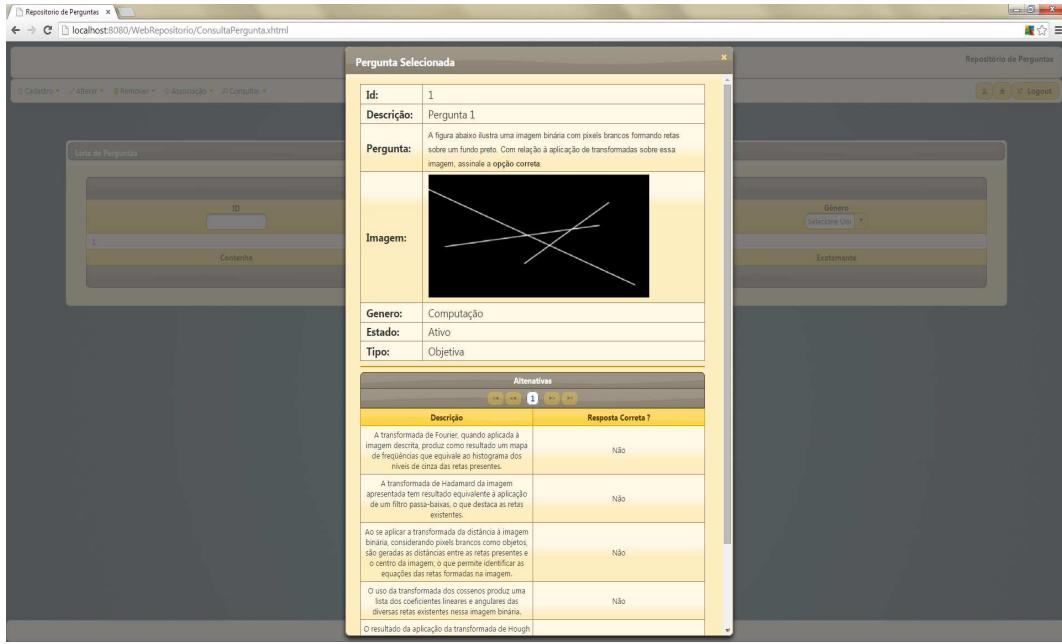


Figura A39: Visualização da Pergunta Selecionada

Consulta Tag's

Através desta funcionalidade é possível consultar *Tag's* cadastradas no sistema, no qual possui filtros para seleção de determinada(s) *Tag's* conforme demonstrado na Figura A40.

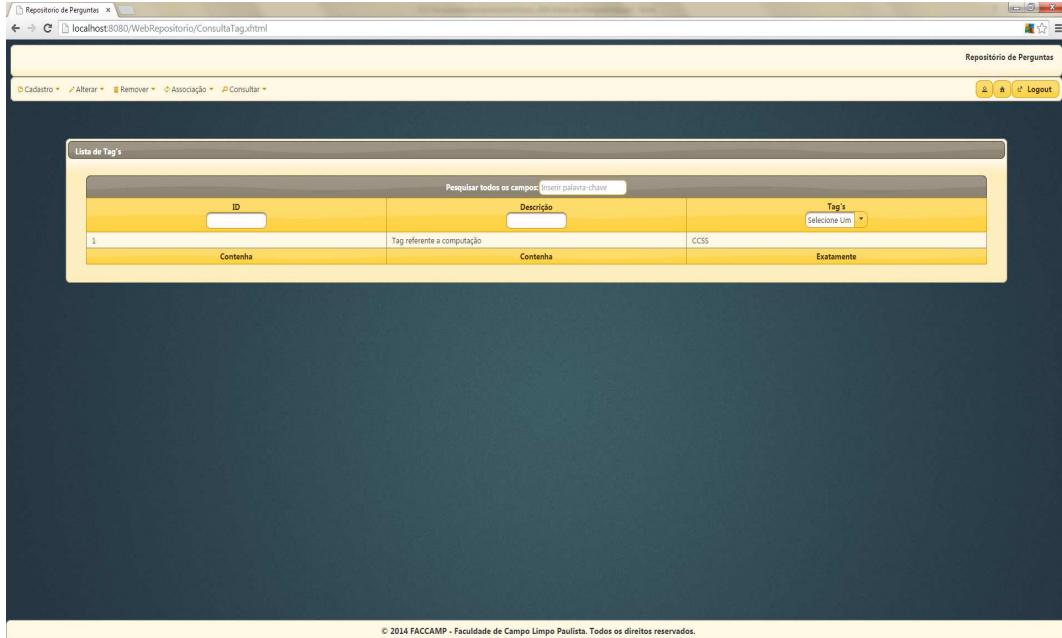


Figura A40: Tela consulta de Tag's

Consulta de Tag's X Perguntas

Através desta funcionalidade é possível visualizar as *Tag's* e suas respectivas perguntas associadas, no qual também possui filtros para seleção de determinada(s) associação(ões) conforme demonstrado na Figura A41.

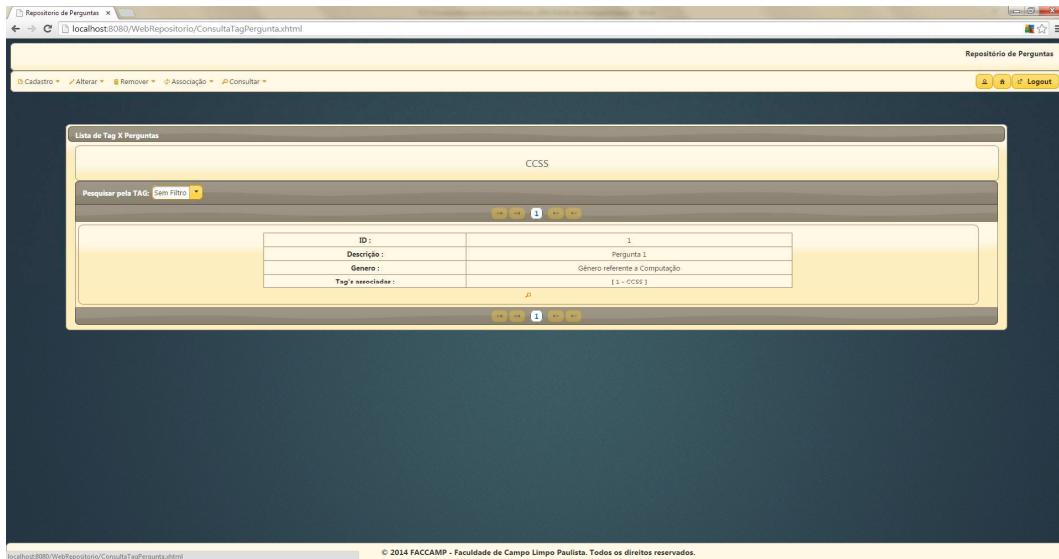


Figura A41: Tela consulta de Tag's associadas a Pergunta(s)

Usuários

Através desta funcionalidade é possível acessar os usuários cadastrados no sistema; permitindo inserção de novos usuários, alteração de usuários e remoção de usuários, conforme demonstrado na Figura A42 e A43.

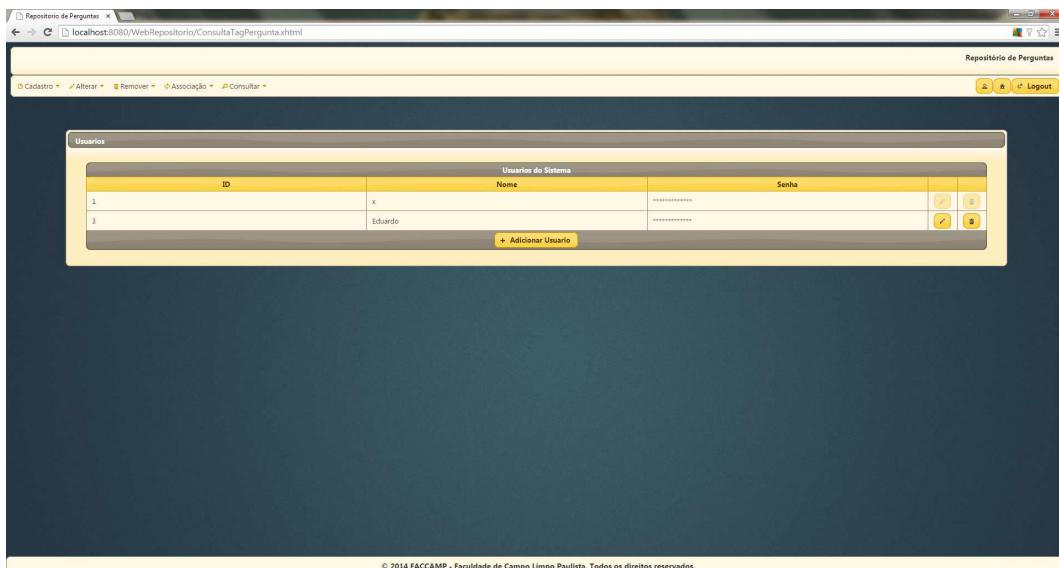


Figura A41: Tela Usuário

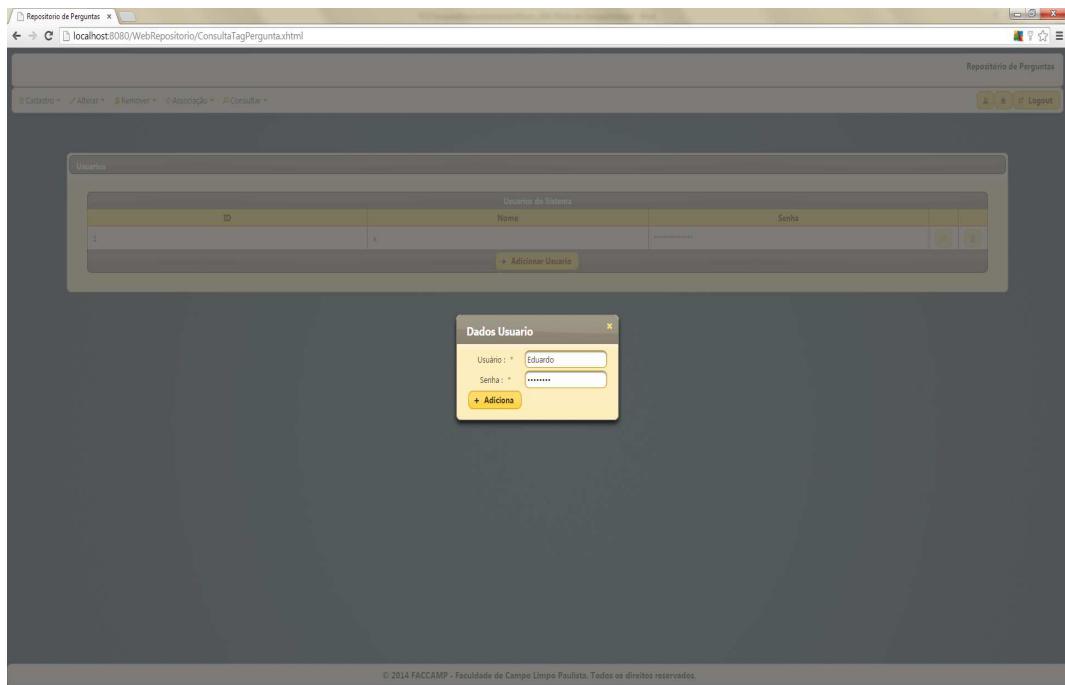


Figura A42: Alteração de Usuário

FACULDADE CAMPO LIMPO PAULISTA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Desenvolvimento Web Ágil Utilizando Framework

ANEXO B – PROJETO DE SOFTWARE

Este anexo tem como objetivo demonstrar o projeto elaborado para o desenvolvimento do sistema Repositório de Perguntas. A Figura B1 demonstra o diagrama DER do Banco de Dados, as tabelas e seus relacionamentos entre si. As tabelas demonstradas abaixo são criadas automaticamente através do framework *Hibernate*, que as cria no instante que é invocado o método *createEntityManager* ().

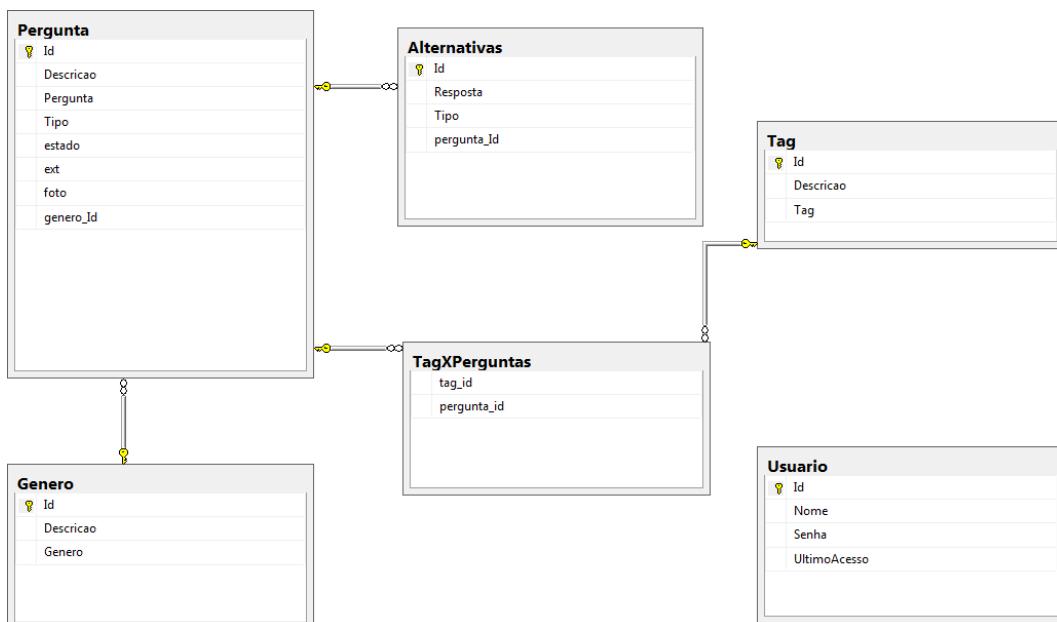


Figura B1 – Banco de Dados

A Figura B2 representa as principais funcionalidades para os casos de usos disponíveis para o usuário.

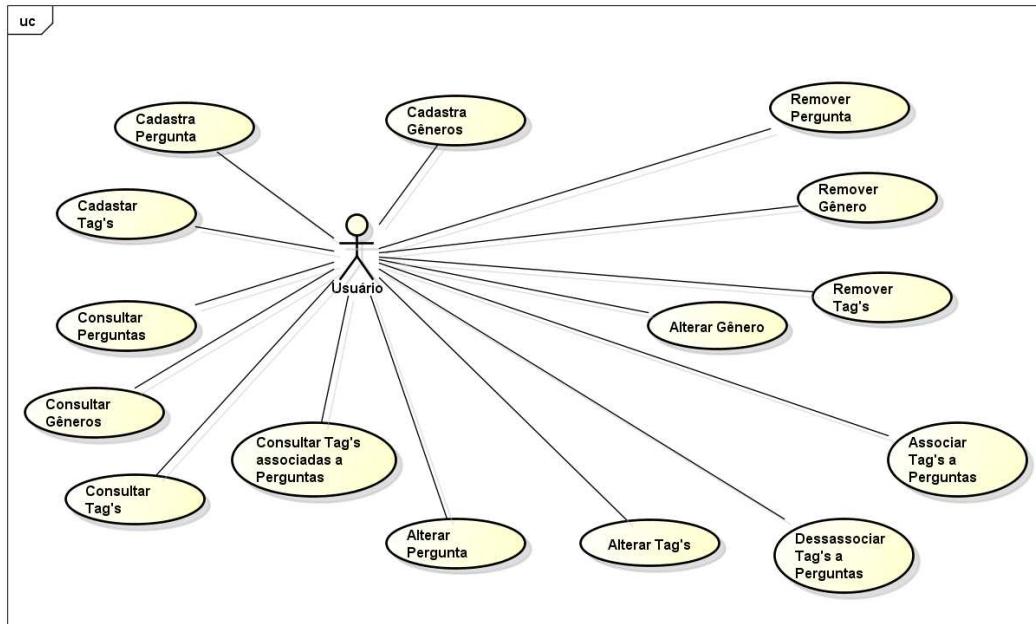


Figura B2 – Diagrama de Casos de Usos

As entidades das classes (Figura B3, B4, B5, B6 e B7 respectivamente) representam a estrutura fundamental do sistema. A Figura B3 e a representação do modelo das entidades que serão persistidas automaticamente no Banco de Dados.

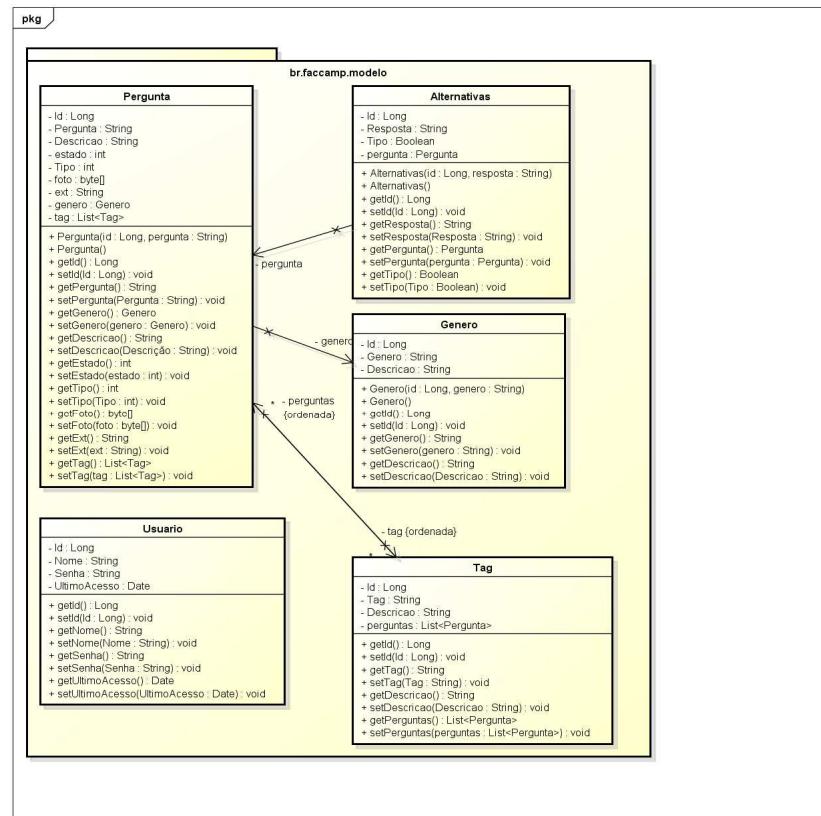


Figura B3 – Diagrama de Classes Modelo

A Figura B4 e B5 representas as classes responsáveis pela camada de controle, que iram gerenciar todo o processamento entre as páginas *JSF (Java Server Faces)*.

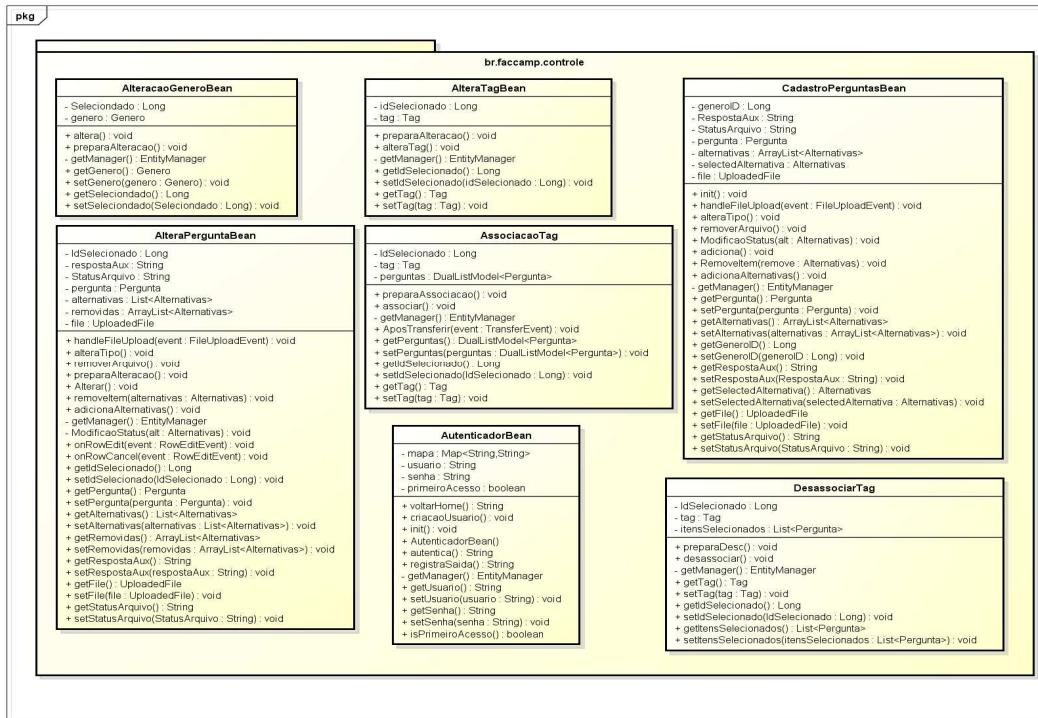


Figura B4 – Diagrama de Classes Controle

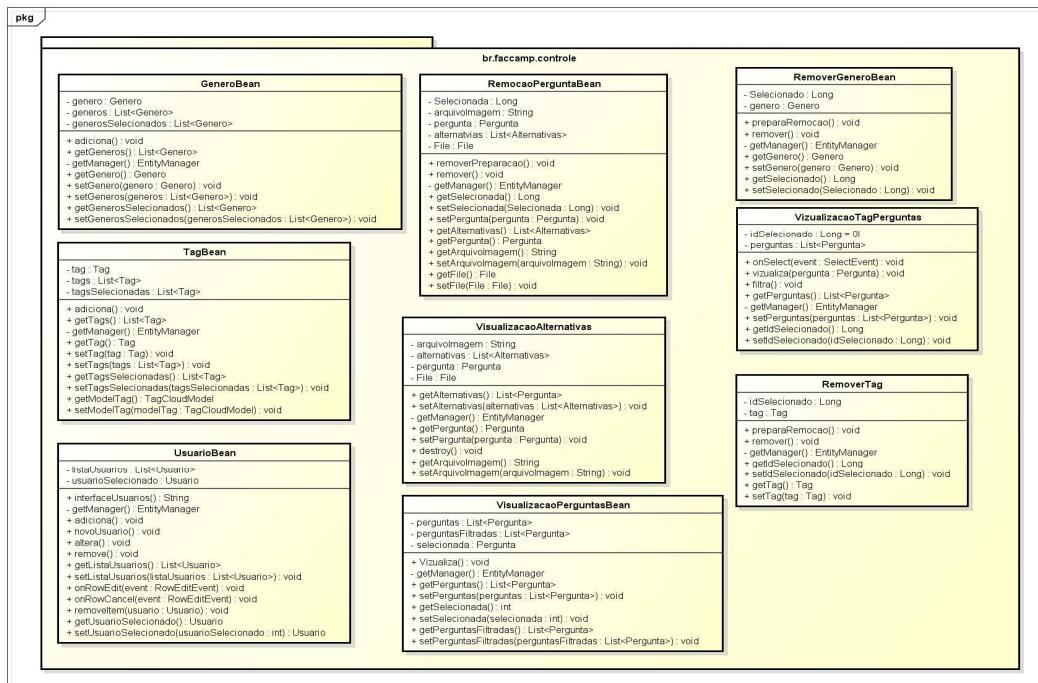


Figura B5 – Diagrama de Classes Controle

A Figura B6 representa as classes responsáveis pela manipulação dos dados com o Banco de Dados, denominado *CRUD* (*Create, Read, Update, Delete*), sendo apenas a representação para utilização dos métodos do framework *Hibernate*.

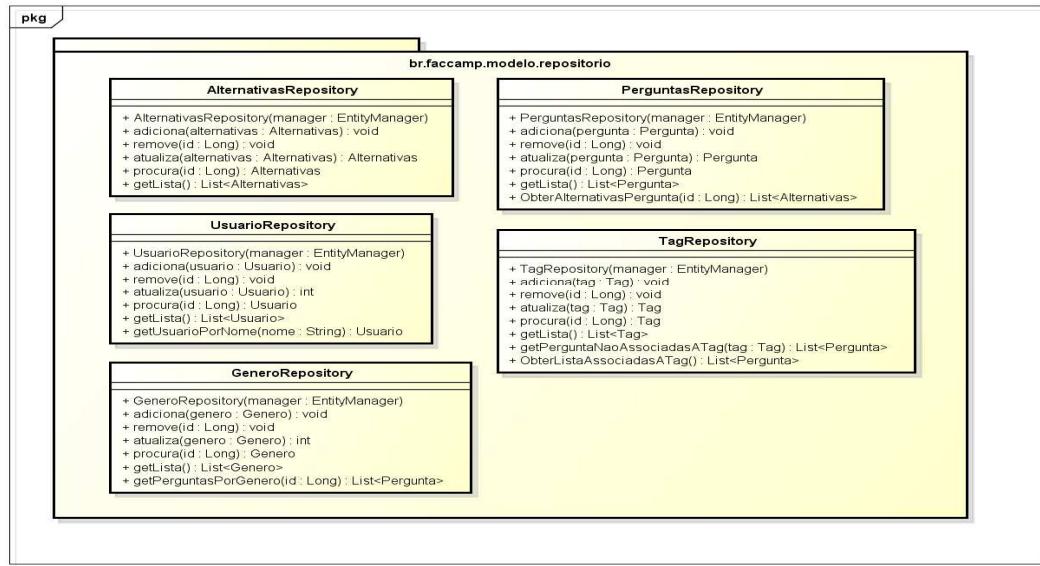


Figura B6 – Diagrama de Classes Repositório

A Figura B7 representa as classes úteis, responsável pela conversão de arquivo em uma cadeia de *Bytes* e vice-versa e a classe responsável pela implementação de conversão para o componente *pickList* do framework *PrimeFaces* e as classes filtros, responsável pelo controle a nível de acesso e o filtro responsável pela geração da classe *EntityManager* do framework *Hibernate* que representa a *conexão com o Banco de Dados*.

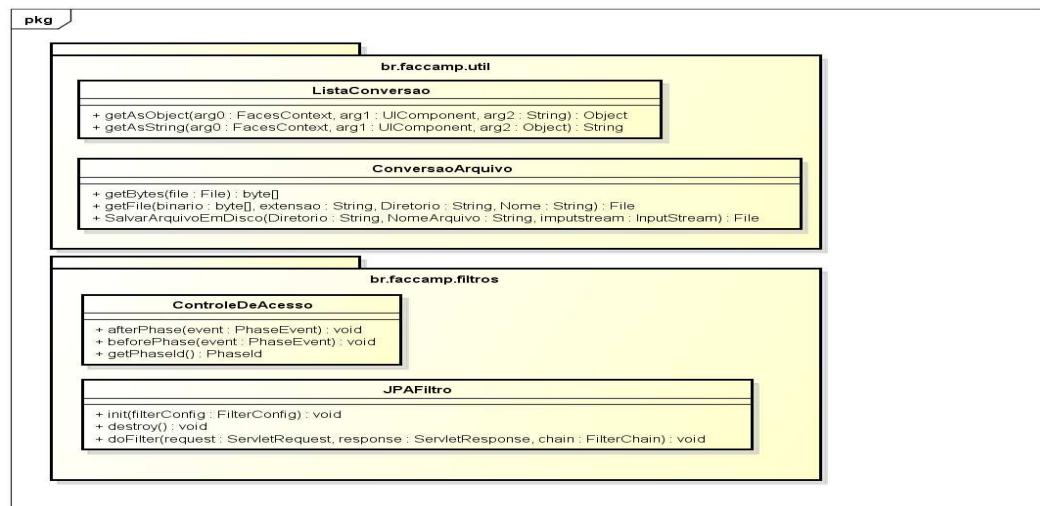


Figura B7 – Diagrama de Classes Útil

**FACULDADE CAMPO LIMPO PAULISTA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

Desenvolvimento Web Ágil Utilizando Framework

ANEXO C – MANUAL DE OPERAÇÃO E IMPLANTAÇÃO

Para a implantação do sistema e seu devido funcionamento, os seguintes passos devem ser executados:

- Instalação e Configuração do Servidor de Aplicação (*GlassFish*);
- Executar o *Deploy* da aplicação *web* no Servidor de Aplicação;
- Configuração do XML de Persistência para conexão com o SGDB (Banco de Dados).

Instalação GlassFish

Primeiramente será necessário obter o instalador do servidor de Aplicação GlassFish, que pode ser obtido gratuitamente a partir do seguinte endereço “<https://glassfish.java.net/download.html>” no link *glassfish-4.1.zip*.

Inicie o instalador, caso a seguinte mensagem aparece, conforme ilustrado na Figura C1, proceda com os seguintes passos:

1. Abra *Prompt de Comando*, conforme Figura C2;
2. Navegue até o diretório onde se encontra o instalador e digite o seguinte comando e o execute:
 - `glassfish-4.0-windows-ml.exe -j "C:\Program Files\Java\jdk1.8.0_11"`, conforme Figura C3”.

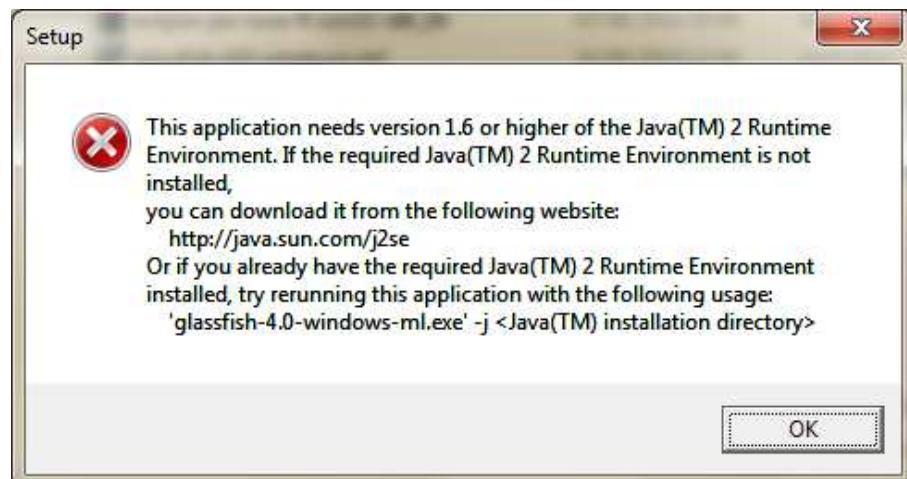


Figura C1: Mensagem de Erro do Instalador

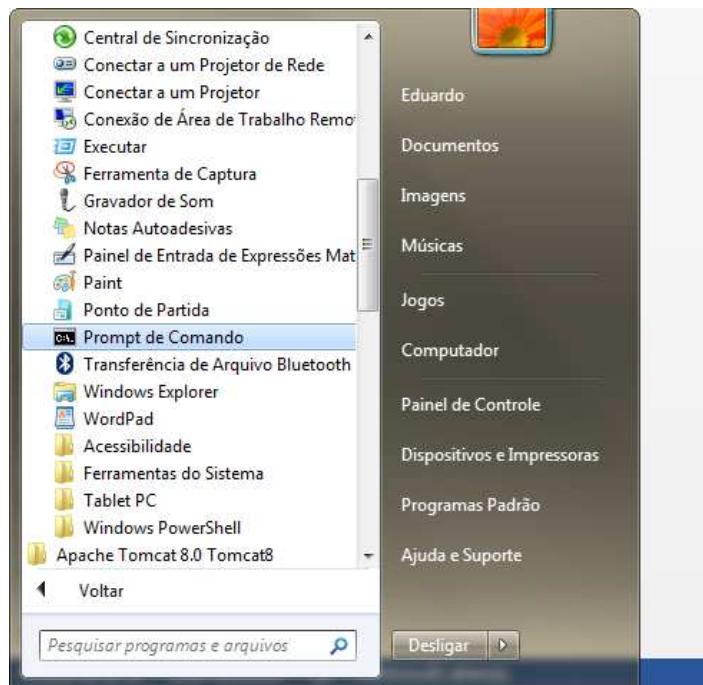


Figura C2: Prompt de Comando

A screenshot of a Windows Command Prompt window titled 'cmd C:\Windows\system32\cmd.exe'. The window displays the following text:

```
Microsoft Windows [versão 6.1.7601]
Copyright © 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Eduardo>j:
J:>cd Programação 3
J:\Programação 3>glassfish-4.0-windows-ml.exe -j "C:\Program Files\Java\jdk1.8.0_11"
```

The command entered was 'glassfish-4.0-windows-ml.exe -j "C:\Program Files\Java\jdk1.8.0_11"', which is intended to run the GlassFish 4.0 Java EE application server.

Figura C3: Comando de execução

Na tela de introdução do instalador, Figura C4, clique em próximo até a tela pronto para instalar, Figura C5

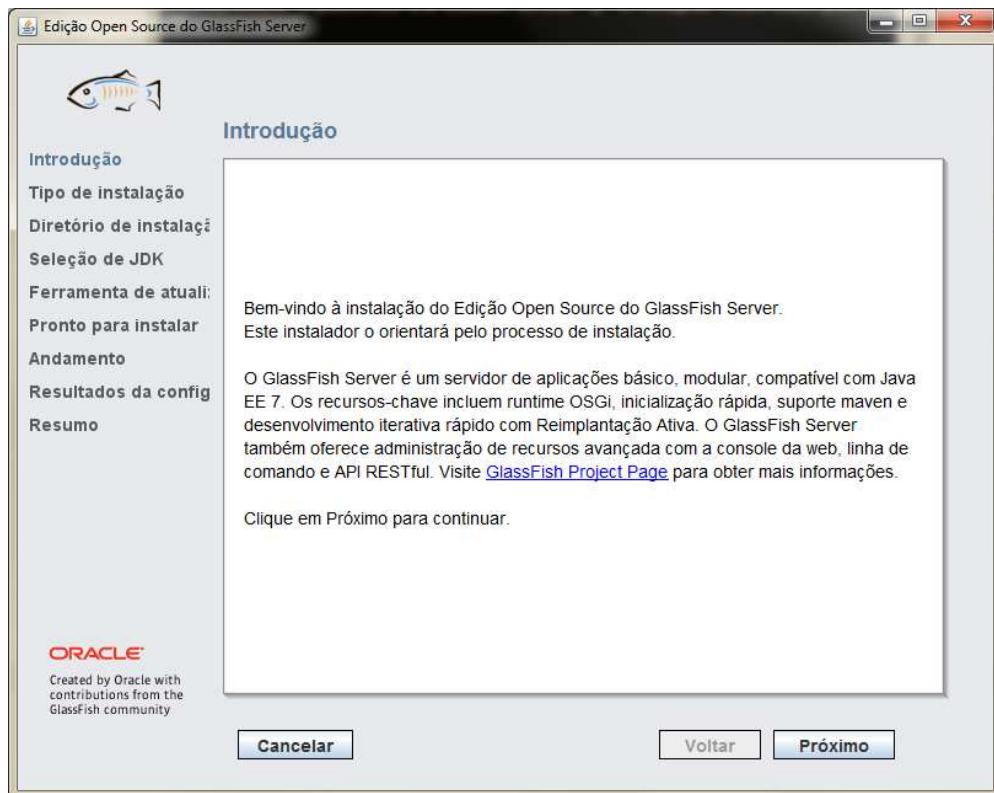


Figura C4: Tela de Introdução do Instalador

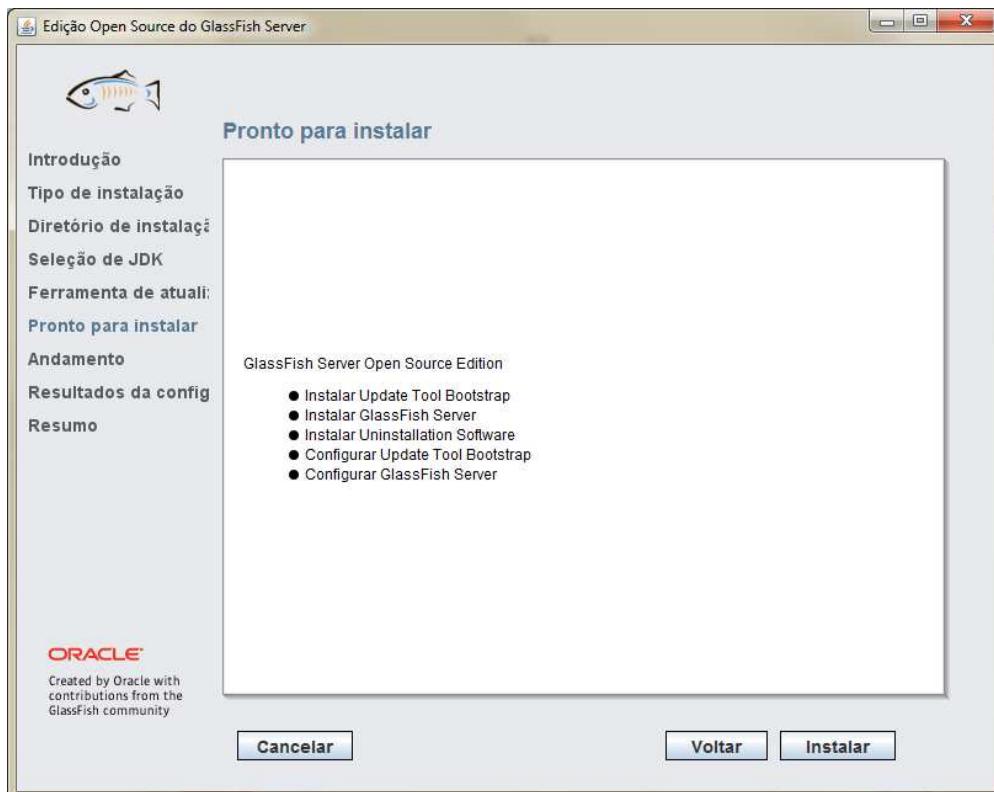


Figura C5: Tela Pronto para Instalar

Ao término da instalação a tela de resumo será mostrada, conforme Figura C6.

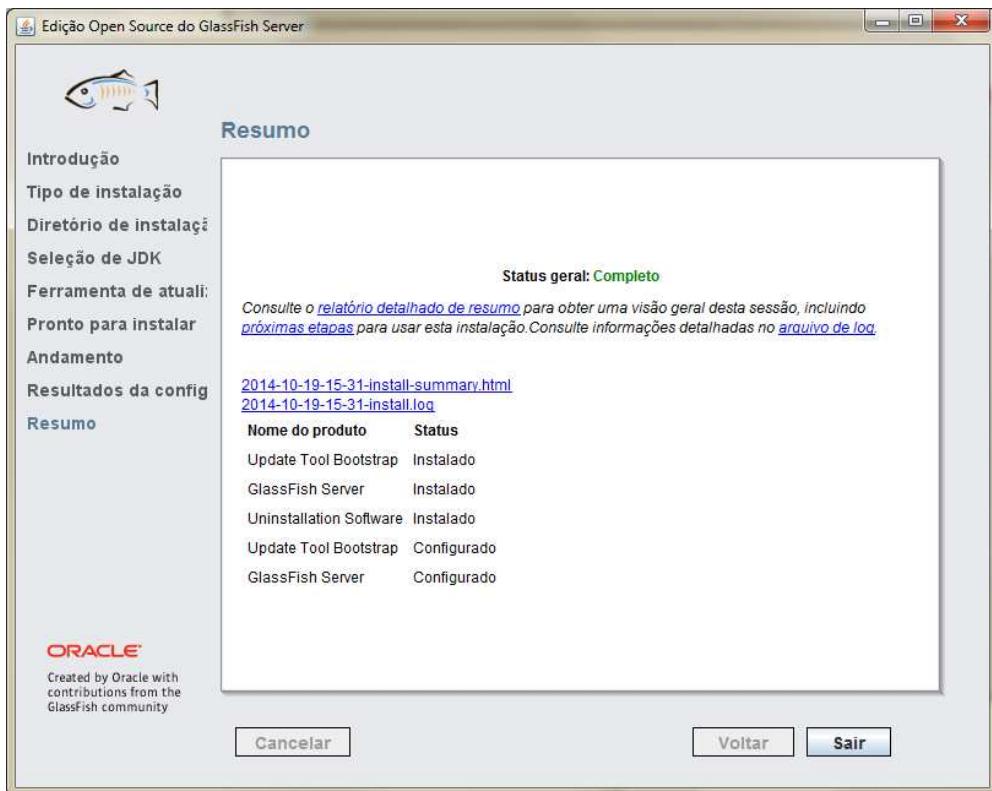


Figura C6: Tela de conclusão da instalação

Configuração e *Deploy* aplicação no servidor de Aplicação

Com o servidor de aplicação instalado (*GlassFish*), procederemos com a configuração e o *Deploy* da aplicação. Primeiramente devemos iniciar o servidor de aplicação, para isto bastar ir para o seguinte caminho: Iniciar, Todos os programas, Edição *Open Source* do *GlassFish Server* e Iniciar servidor de aplicativos; conforme demonstrado na Figura C7.

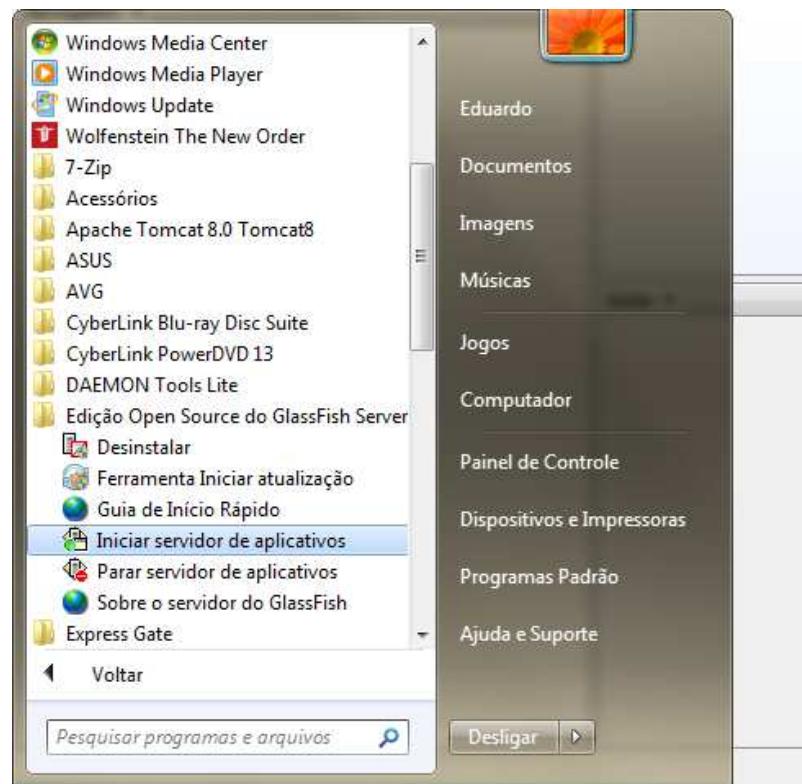


Figura C7: Inicialização do Servidor

Acesse através do navegador a seguinte endereço “localhost:4848”, para ter acesso ao sistema principal do servidor de aplicação, conforme Figura C8.

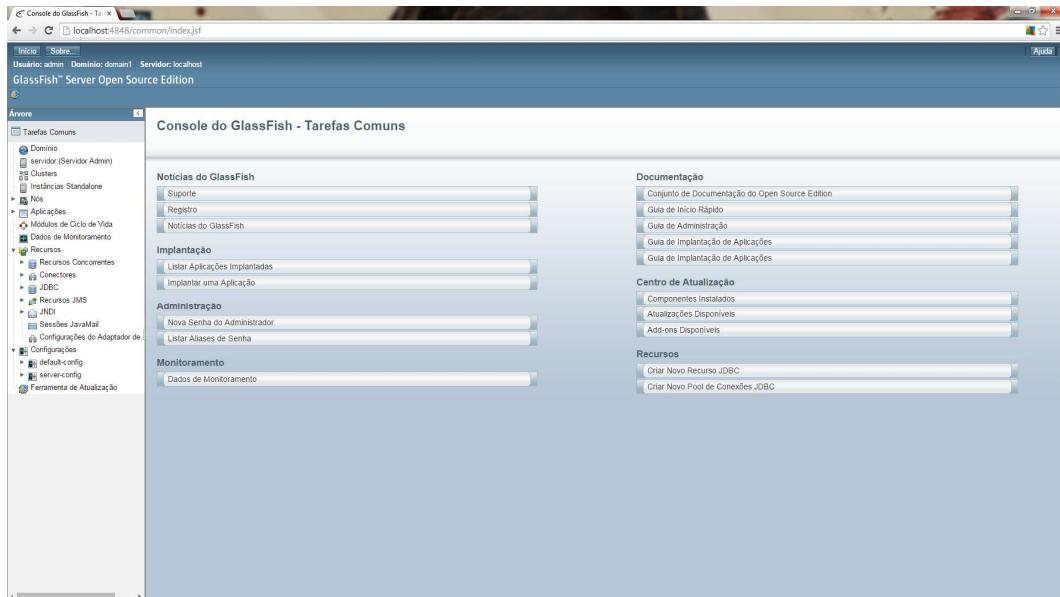


Figura C8: Home do GlassFish

Dentro do servidor de aplicação procederemos com o *Deploy* da aplicação. Para isso devemos acessar no menu lateral a esquerda a opção “Aplicações”

Na tela aplicações, conforme a Figura C9, deve-se clicar no botão implantar.

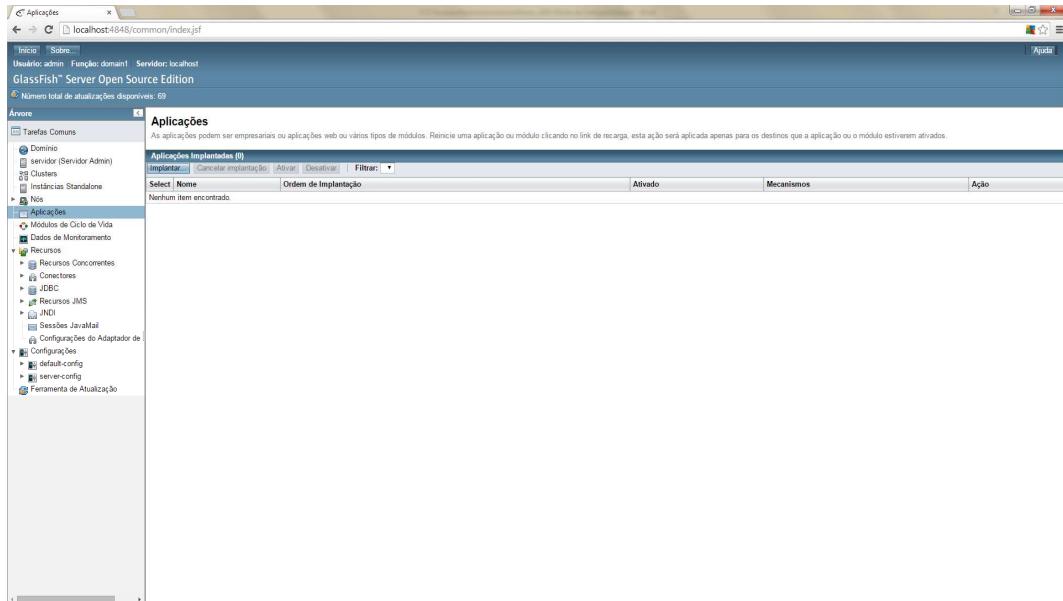


Figura C9: Tela Aplicações do GlassFish

No botão escolher arquivo, deve ser informado a localização do *WAR* (*Web application Archive*), denominado com o nome “WebRepositorio.war”. Após o carregamento do arquivo no servidor *GlassFish*, automaticamente os restantes dos dados abaixo obrigatórios, conforme Figura C10, serão preenchidos. Basta-se clicar OK para proceder com o *Deploy* da aplicação.

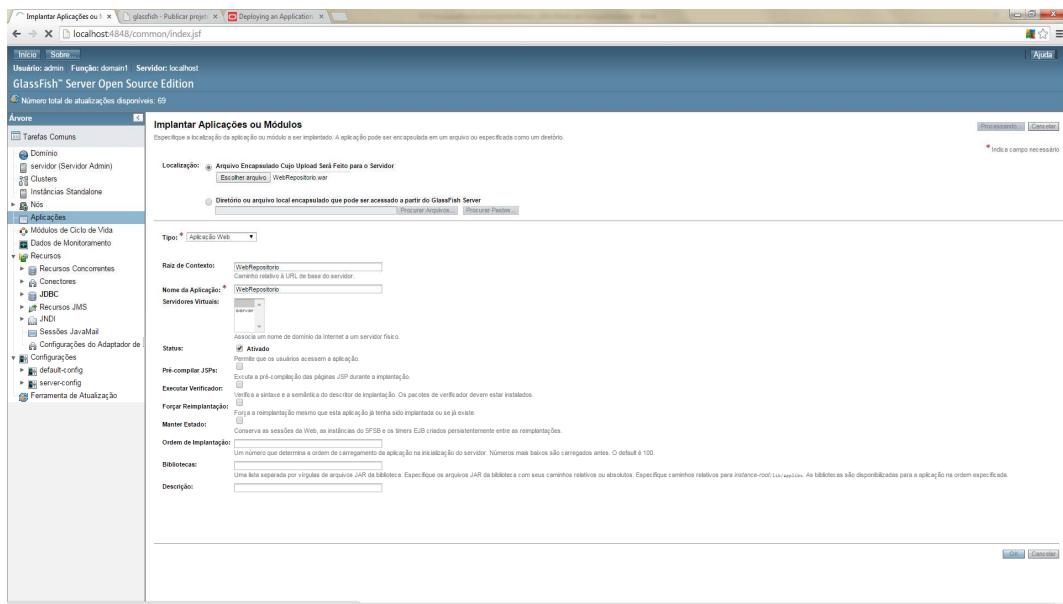
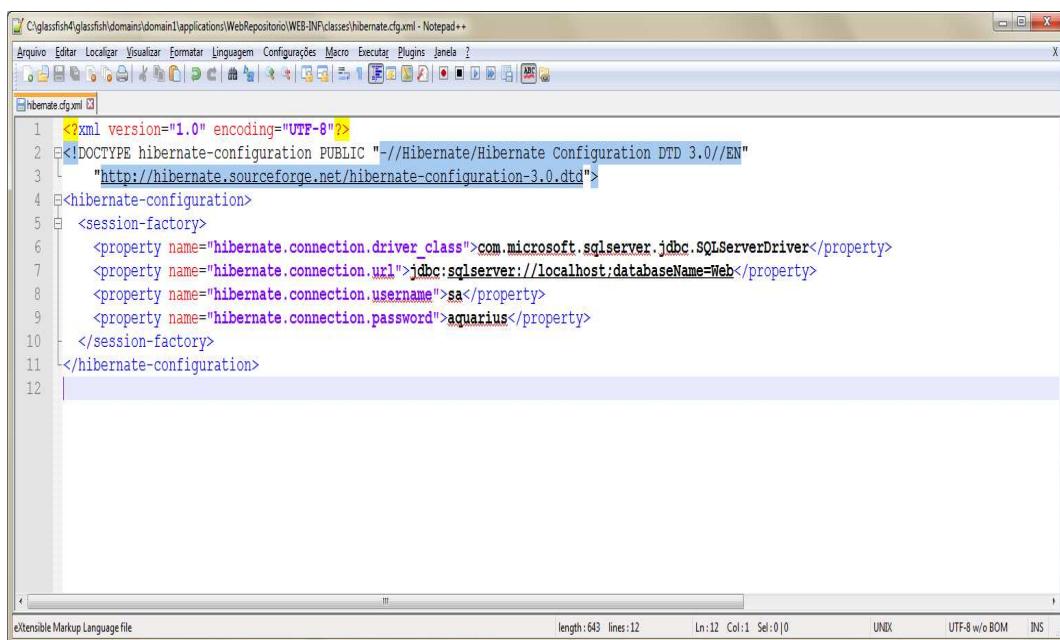


Figura C10: Tela de Deploy (implantação)

Configuração XML Banco de Dados

Com o servidor de aplicação iniciado e aplicação implantada, procederemos com a configuração do XML de Persistência do Banco de Dados. Para isto deve-se abrir o arquivo hibernate.cfg.xml, Figura C11, da aplicação que se encontra no seguinte caminho:

“C:\glassfish4\glassfish\domains\domain1\applications\WebRepositorio\WEB-INF\classes”.



The screenshot shows a Notepad++ window with the file 'hibernate.cfg.xml' open. The code is an XML configuration for a database connection:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="hibernate.connection.driver_class">com.microsoft.sqlserver.jdbc.SQLServerDriver</property>
<property name="hibernate.connection.url">jdbc:sqlserver://localhost:databaseName=Web</property>
<property name="hibernate.connection.username">sa</property>
<property name="hibernate.connection.password">aquarius</property>
</session-factory>
</hibernate-configuration>
```

Figura C11: Arquivo hibernate.cfg.xml

Conforme o banco de dados escolhido, as seguintes configuração abaixo das propriedades deve ser informada nas Tag's Property.

Na Tag Property hibernate.connection.password deve ser informado conforme a tabela:

Tabela 1 – Banco de Dados

Banco de Dados (SGBD)	Propriedade
DB2	org.hibernate.dialect.DB2Dialect
DB2 AS/400	org.hibernate.dialect.DB2400Dialect
DB2 OS390	org.hibernate.dialect.DB2390Dialect
PostgreSQL	org.hibernate.dialect.PostgreSQLDialect
MySQL	org.hibernate.dialect.MySQLDialect
MySQL with InnoDB	org.hibernate.dialect.MySQLInnoDBDialect

MySQL with MyISAM	org.hibernate.dialect.MySQLMyISAMDialect
Oracle (any version)	org.hibernate.dialect.OracleDialect
Oracle 9i	org.hibernate.dialect.Oracle9iDialect
Oracle 10g	org.hibernate.dialect.Oracle10gDialect
Sybase	org.hibernate.dialect.SybaseDialect
Sybase Anywhere	org.hibernate.dialect.SybaseAnywhereDialect
Microsoft SQL Server	org.hibernate.dialect.SQLServerDialect
SAP DB	org.hibernate.dialect.SAPDBDialect
Informix	org.hibernate.dialect.InformixDialect
HypersonicSQL	org.hibernate.dialect.HSQLDialect
Ingres	org.hibernate.dialect.IngresDialect
Progress	org.hibernate.dialect.ProgressDialect
Mckoi SQL	org.hibernate.dialect.MckoiDialect
Interbase	org.hibernate.dialect.InterbaseDialect
Pointbase	org.hibernate.dialect.PointbaseDialect
FrontBase	org.hibernate.dialect.FrontbaseDialect
Firebird	org.hibernate.dialect.FirebirdDialect

Na *Tag Property* hibernate.connection.url deve ser informado a *URL* do banco de dados.

Na *Tag Property* hibernate.connection.driver_class deve ser informado o usuário do Banco de Dados.

Na *Tag Property* hibernate.connection.password deve ser informado a senha do usuário do Banco de Dados.