|  |  |
| --- | --- |
| F:\unitri.png | **CENTRO UNIVERSITÁRIO DO TRIÂNGULO PÓS-GRADUAÇÃO EM DESENVOLVIMENTO JAVA** |

**Balder Education**

**Desenvolvimento de Sistema de Gerenciamento de Alfabetização para a Plataforma em Nuvem com Android e Java Web, utilizando Webservice RESTful e internet das coisas.**

**Rogério Fontes Tomaz**

**Marcos Fernando Barbosa**

**Luís Eduardo Marques**

**Fábio Santos Alves**

**Marcelo Rossi**

**Lidiana Vieira Silva**

**Uberlândia, dezembro de 2015**

|  |  |
| --- | --- |
| **F:\unitri.png** | **CENTRO UNIVERSITÁRIO DO TRIÂNGULO PÓS-GRADUAÇÃO EM DESENVOLVIMENTO JAVA** |

**Balder Education**

**Desenvolvimento de Sistema de Gerenciamento de Alfabetização para a Plataforma em Nuvem com Android e Java Web, utilizando Webservice RESTful e internet das coisas.**

**Rogério Fontes Tomaz**

**Marcos Fernando Barbosa**

**Luís Eduardo Marques**

**Fábio Santos Alves**

**Marcelo Rossi**

**Lidiane Vieira Silva**

Trabalho de conclusão de curso apresentado ao Curso de Pós-Graduação *lato sensu* em Desenvolvimento Java do Centro Universitário do Triângulo - UNITRI, como requisito básico à obtenção do título de Especialista em Desenvolvimento Java, sob a orientação dos professores Dr. Marcos A Lopes da Silva e Msc. Sônia A Santana.

**Uberlândia, dezembro de 2015.**

**Desenvolvimento de Sistema de Gerenciamento de Alfabetização para a Plataforma em Nuvem com Android e Java Web, utilizando Webservice RESTful e internet das coisas.**

**Rogério Fontes Tomaz**

**Marcos Fernando Barbosa**

**Luís Eduardo Marques**

**Fábio Santos Alves**

**Marcelo Rossi**

**Lidiana Vieira Silva**

Trabalho de conclusão de curso apresentado ao Curso de Pós-Graduação *lato sensu* em Desenvolvimento Java do Centro Universitário do Triângulo - UNITRI, como requisito básico à obtenção do título de Especialista em Desenvolvimento Java.

|  |
| --- |
| Sônia Aparecida Santana, Msc.  (Orientador)  Marco Alberto Lopes da Silva, Dr.  (Orientador) |
| Marcos Alberto Lopes da Silva, Dr.  (Coordenador de Curso) |

**Uberlândia, dezembro de 2015**

*“Educação não transforma o mundo. Educação muda as pessoas. Pessoas transformam o mundo”* ***Paulo Freire***

**RESUMO**

Um dos grandes problemas da sociedade atual está relacionado com a educação. De acordo com uma pesquisa realizada pela UNESCO em 2015, existe por volta de 781 milhões de pessoas analfabetas no mundo todo. Neste cenário amplia-se a necessidade de alfabetizar estes cidadãos a fim de incluí-los socialmente e auxiliar na prática de uma profissão. A ideia da organização do projeto *Balder-Education* é apresentar uma solução para ajudar a sociedade que não tem condições para estudos. Os adultos irão ter condições de se alfabetizar pela plataforma de ensino e métrica *Balder-Education* que usa o método criado por Paulo Freire[1]. O software funciona tanto em ambiente web como em dispositivos móveis e tem o objetivo de ser uma ferramenta de apoio e suporte ao aprendizado, além de gerar relatórios e estatísticas do desempenho do aluno. A tecnologia utilizada envolve a utilização Java EE para o desenvolvimento da plataforma *web*, bem como *webservice* e arquitetura *RestFul*, o que evita duplicação de código. O *webservice* recebe requisições dos clientes via protocolo http e retorna a resposta usando o formato de dados *JSON.* A interface obedece aos requisitos de ergonomia e, no caso do dispositivo *mobile*, a responsividade. O sistema foi disponibilizado de forma gratuita utilizando a estrutura de *cloud s*endo que, para a persistência dos dados, optou-se pelo banco de dados relacional *MySQL*. Este fica fisicamente localizado em um *host* virtual de baixo custo mensal proporcionando alta disponibilidade e segurança. Além disso, ressalta-se a utilização de IoT[2] de forma que, com a utilização de um leitor digital, o aluno acessará, automaticamente, o sistema. De uma forma geral, o sistema mostrou-se satisfatório ao apresentar conteúdo a pessoas semi-analfabeta buscando palavras baseadas em seu cotidiano. Desde a adoção de um grupo fechado de teste, conseguimos avaliar, com ajuda de professores especializados a evolução do aluno, através de estatísticas baseadas no seu progresso e relatórios gerados pelo sistema de estatistica.

**SUM**Á**RIO**

* **Resumo**
* **Introdução**
* **A escolha do tema**
* **Delimitação do Tema**
* **Problematização do Tema**
* **Objetivos**
* **Justificativas**
* **Revisão da Literatura**
* **Google Android**
* **Java-Server Faces**
* **PrimeFaces**
* **PrimeFaces e outras bibliotecas**
  + **Containers**
  + **WebServices**
  + **WebServices SOAP-WSDL vs RESTFul**
  + **Framework para RESTFul em Java (Jersey)**
  + **Json e Gson**
* **Metodologia**
* **Definição da área ou população-alvo**
* **Análise e discursão da proposta**
* **Conclusões**
* **Referência Bibliográfica**

1. **INTRODUÇÃO**

No cenário atual de força de conhecimento e falta de alfabetização, esse trabalho tem como objetivo, demostrar através da educação e utilização do método do Paulo Freire de ensino, é apresentado um ambiente didático para progressão do aprendizado do conteúdo.

Diante deste cenário é importante que as organizações avaliem se a mobilidade poderá colaborar para que seus processos sejam mais ágeis, visando resultados com projeção de crescimento.

O problema atual que o suporte de aprendizado para pessoas adultas é quase nulo em termos de democratização tecnológica, sendo de difícil acesso a professores para lugares inclusive remotas.

Usando mobilidade, a ideia que o aluno sempre estará evoluindo e está cada vez mais progredindo no seu conhecimento diário.

Esse projeto visa mostrar, que é possível a democratização do aprendizado, usando um método consolidado mundialmente, através de tecnologia e satisfação pessoal do aluno.

A seção 2 apresenta a escolha do tema com os seus objetivos e justificativas. Na seção 3 temos uma revisão da literatura acerca das metodologias e ferramentas utilizadas no desenvolvimento deste trabalho. Por fim na seção 4 é apresentada a metodologia de trabalho, descrevendo passo a passo a construção do software e testes realizados. Por fim na seção 5, as conclusões do trabalho.

**2. A ESCOLHA DO TEMA**

O tema deste trabalho foi escolhido considerando no conteúdo estudado nas disciplinas cursadas e conhecimentos adquiridos no decorrer do curso de Pós-Graduação em Desenvolvimento Java, assim como a existência de uma circunstância real, ou seja, a necessidade de se resolver um problema mundial de falta de leitura e escrita por pessoas analfabetas e analfabetas funcionais sem alcance a estudos.

Como o tema e as práticas envolvidas são de interesse de todos da equipe, o tema se torna relevante e até mesmo indicado, afinal também será possível contar com o auxílio de profissionais do curso, capacitados no assunto abordado, permitindo dessa forma, que novas experiências sejam adquiras e conhecimentos agregados e a empresa terá um software de qualidade a um custo irrisório.

**2.1. Delimitação do Tema**

Na parte prática do projeto, o sistema a ser construído contemplará um aplicativo web contendo vídeos de ambientalização do tema para ser escolhido pelo aluno, ainda na parte web será apresentada a progressão dos alunos em cada módulo, gerando estatísticas de aprendizado para suporte aos professores gerando relatórios dessas estatísticas geradas. Ainda para dispositivos móveis será apresentado conteúdos menores para suporte diário de aprendizado, dependendo do nível de dificuldade apresentado pelas estáticas geradas nos relatórios.

Além disso, para atender todos os aplicativos e unificar o sistema, ambas as aplicações consumirão um WebService RESTful.

**2.2. Problematização do Tema**

O objeto desse estudo são pessoas com carência de estudos e ajudar a sociedade que não tem condições de alfabetização e conhecimento de leitura e escrita. A idéia é automatizar o método Paulo feire, através de uma plataforma de ensino e métrica.

Há, portanto a necessidade de um software específico que atenda a estas exigências e que, funcione em dispositivos móveis, pois os alunos devem ter suporte a aprendizagem diária, mesmo sem a presença de um tutor/professor. Além disso, deve gerar relatórios sobre a estatística de aprendizagem. As figuras 1 e 2 apresentam a lista completa dos requisitos do software.

**2.3. Objetivos**

O objetivo geral do projeto consiste em construir um software composto por uma aplicação web e móvel que permita realizar cadastros de conteúdo e progressão de aulas, e também visualizar relatórios de progressão de evolução de aprendizado dos alunos.

**Objetivo (s) específico (s):**

* Analisar e progressão de aprendizagem de pessoas analfabetas ou com alfabetismo funcional.
* Construir vários webservices de integração baseado no conceito REST, para transmitir e obter dados. Desenvolvido na linguagem Java, utilizando o framework da Spring e RestEasy da Jboss e ainda as bibliotecas Jackson e Gson da Google, ela centralizará o consumo de dados evitando duplicação de código. Este WebService será executado sob o servidor Apache e Jboss WildFly 8.2, acessando banco de dados relacional MySQL.
* Construir uma aplicação para dispositivos móveis na plataforma *Google Android*, utilizando a linguagem Java e biblioteca *Gson* da Google. Esta aplicação deve possibilitar aos alunos receber drops de conteúdo para ajudar na sua progressão de aula;
* Desenvolver uma aplicação web, em *JSF2.2,* usando a biblioteca PrimeFaces 5.2 para interface e biblioteca RestEasy da Jboss, para comunicação com o w*ebservice* onde o usuário administrador autenticado visualize relatórios de progressão dos alunos, e um usuário que seja aluno autenticado visualize as aulas e suas progressões de estudo e apoio de dificuldades.

**2.4. Justificativas**

Em nosso cotidiano a necessidade do conhecimento tecnológico básico está cada vez mais amplo, nas mínimas tarefas do dia a dia como ir ao banco pagar uma conta ou sacar dinheiro no caixa eletrônico, estamos envolvidos em sistemas e tecnologias cada vez mais avançados. Analisamos que uma pessoa alfabetizada já passa algumas dificuldades para acompanhar esta evolução, imaginamos em particular os analfabetos quanto à dificuldade neste exemplo.

Visto esta afirmação resolvemos buscar uma solução na qual ajudaremos analfabetos a aprender a ler e escrever assim como incluí-los neste avançado mundo tecnológico.

A didática da nossa inovação é baseada nos estudos e métodos desenvolvidos por Paulo Freire, abrangendo o ensino prático à adultos analfabetos, fazendo assim com que o usuário além de obter ensino de leitura e escrita há também um contato mais próximo com a tecnologia através da nossa aplicação web e mobile.

Conforme uma pesquisa realizada pela UNESCO em 2015, existe por volta de 781 milhões de pessoas analfabetas no mundo todo. Neste cenário amplia-se a necessidade de alfabetizar estes cidadãos à fim de prepará-los para o meio em que vive com a sociedade e na prática de sua profissão[3].

Ao entrar em contato com a tecnologia digital, o adulto enriquecerá seus conhecimentos interagindo com uma variedade de cenários e objetos que estão envolvidos em seu cotidiano, facilitando assim o seu aprendizado com a nossa ferramenta de ensino e diminuindo a estatística de analfabetismo, atingindo assim os objetivos da educação pedagógica para adultos.

**3. REVISÃO DA LITERATURA**

Tanto para o desenvolvimento do sistema, quanto para a compreensão do trabalho como um todo, é importante conhecer as tecnologias utilizadas, permitindo dessa forma, relacionar a parte textual com a prática.

Para isto, foram realizadas pesquisas com o intuito de descrever as tecnologias escolhidas para o projeto, justificar as escolhas e expor o cenário atual em que se encontram.

**3.1. Google Android**

Em 05 de novembro de 2007, a OHA (*Open Handset Alliance*) anunciou a plataforma *Android*. A *Open Handset Alliance* é um grupo formado atualmente por 84 empresas de tecnologia, liderado pela *Google*, e que se uniram para acelerar a inovação em dispositivos móveis e oferecer uma solução mais barata, melhor e com mais rica experiência ao usuário [RABELLO, 2014].

O *Android* é uma poderosa, ousada e flexível plataforma de desenvolvimento para aplicativos móveis e ao mesmo tempo um sistema operacional baseado no Linux. [LECHETA, 2010].

Para desenvolver aplicações *Android* a linguagem utilizada é o Java, assim como todos os seus recursos, no entanto, não existe uma máquina virtual Java (JVM), mas uma específica chamada *Dalvik*. Sendo assim, depois de compilado o *bytecode* (.*class*) é convertido para o formato .dex (*Dalvik Executable*) que representa a aplicação *Android* compilada [PEREIRA e SILVA, 2009]. Atualmente a plataforma *Android* é a forma mais utilizada para a criação de clientes interativos. A instalação e distribuição do aplicativo é mais simples e menos onerosa que seu concorrente direto, o iOS da *Apple* e, ao contrário deste, pode-se utilizar outros canais de venda, além da *Google Play,* como a Verizon *apps* da Verizon e a *appstore* da Amazon.com [MEDNIEKS et all, 2012].

Em 25 de Junho de 2014 na conferência *Google* I/O, foi anunciada a chegada da mais nova versão do sistema operacional *Android*, a qual foi intitulada inicialmente pelo termo provisório “*Android* L”. Até a finalização deste trabalho a data para o lançamento da versão final para os consumidores ainda não havia sido definida [MOTOGBRASIL, 2014]. Entretanto, o *kit* de desenvolvimento e a versão *preview* do sistema para desenvolvedores, já se encontra disponível, possibilitando que sejam testados os novos recursos e desenvolvidas ou adaptadas as aplicações existentes para o novo sistema.

De acordo com o site oficial do desenvolvedor *Android*, serão várias mudanças visuais e de desempenho nesta nova versão. Dentre as modificações são destacadas algumas como [ANDROID DEVELOPER, 2014].

* **Um novo *design* da interface do usuário** - uma experiência consistente em móveis e na web com *Material Design,* o novo padrão visual;
* **Um novo *Runtime*** – o ART (*Android Runtime*) será o novo tempo de execução padrão na próxima versão;
* **Notificações melhoradas** - melhor controle sobre as notificações, simplificando o uso e a forma como são sincronizadas com dispositivos não portáteis;
* **Maior eficiência –** objetivo em tornar a energia da plataforma eficiente com o *Project* Volta, dando mais controle sobre o uso de recursos.

Além dessas, várias outras mudanças foram anunciadas e algumas até já se encontram documentadas no site do desenvolvedor *Android,* tornando possíveis os testes com o kit de desenvolvimento. E até que esteja disponível a versão final do sistema, muitas outras modificações poderão ocorrer.

**3.2. Java-Server Faces**

Segundo LUCKOW e MELO [2010], *Java-Server Faces* (JSF) é a especificação para um framework de componentes para desenvolvimento web em Java. Este framework foi projetado com o objetivo de facilitar o desenvolvimento e a manutenção de aplicações executadas em um servidor de aplicações Java e processar as interfaces gráficas de volta a um cliente de destino [NETBEANS, 2014].

A versão 2 do JSF foi criada para facilitar o seu uso com funções e desempenhos melhorados. Comparando o JSF com outros dois frameworks: Struts e Vraptor, chega-se à conclusão que uma das vantagens do JSF é por ele ser uma especificação, pois o Struts é um framework que auxilia a criação de aplicação para a *Web*, o seu original parou na versão 1.3.9 o que foi substituído pelo framework WebWork apenas por questões de marketing por ter mudado totalmente o conceito na versão 2.0 e *o Vraptor* é um framework (MVC) - Modelo Visão e Controlador - que trabalha de maneira RESTful, que além de ser pouco utilizado no mercado também tem uma dependência muito grande de HTML, CSS e JS.

Outras vantagens são a facilidade e flexibilidade de seu uso e a utilização na disciplina Frameworks para Desenvolvimento Web do curso.

**3.3. PrimeFaces**

O PrimeFaces é uma biblioteca de componentes para o Java-Server Faces. Possui código aberto e uma grande comunidade dedicada e ativa, que inclui diversos desenvolvedores. Seu principal objetivo é fazer com que o desenvolvedor crie uma camada de visualização, com uma aparência agradável para o usuário final, sem necessitar de dependências e configurações adicionais. Dentre alguns componentes disponíveis podem ser citados: gráficos, tabelas, caixas de mensagens, painéis, menus e templates [PRIMEFACES, 2014].



**Figura 1 - Exemplo de componente DataTable do Primefaces[PRIMEFACES, 2014].**

3.3.1. *PrimeFaces* e outras bibliotecas

O PrimeFaces foi um dos primeiros a estar totalmente convertido para o JSF 2.0 consequentemente possui um número maior de componentes. A documentação do PrimeFaces é mais clara proporcionando um material completo através de um guia de usuário que mostra as definições de todos os componentes e exemplos de códigos utilizando os mesmos.

Outros componentes tais como: RichFaces, IceFaces, OpenFaces e EasyFaces podem ser utilizados para a implementação JSF também.

**3.4. Containers**

O Apache Tomcat é um contêiner Java e um servidor ao mesmo tempo. Ele suporta a execução das tecnologias Java Servlet e Java-Server Pages (JSP), o que permite que o Java funcione dentro de um ambiente web.

O GlassFish é um servidor de aplicação, suportando todas as especificações da API Java EE e linguagens dinâmicas, além de permitir aplicações corporativas portáveis, escaláveis e fácies de integrar. Ele utiliza uma variante do Apache Tomcat (Grizzly) para manter essa eficiência [ORACLE, 2014].

O servidor de aplicações JBoss é uma plataforma baseada em Java EE 5 utilizado para desenvolvimento e implantação de aplicativos empresariais. Ele suporta tanto APIs tradicionais quanto APIs do Java. Ele integra o Apache Tomcat como seu contêiner Web e inclui recursos para cache de dados, clustering, mensagens, transações e um serviço Web integrado [ORACLE, 2014].

Como podemos observar acima os servidores de aplicação GlassFish e JBoss são muito mais poderosos e como há a necessidade de execução em ambiente web atendendo várias requisições ao mesmo tempo utilizaremos o Jboss na sua versão 8.2 com o Tomcat direcionando as requisições para as aplicações web.

**3.5. *WebServices***

*Service-Oriented Architecture* (SOA) ou Arquitetura Orientada a Serviços estabelece a relação entre consumidores e prestadores de serviços através de uma lista disponibilizada pelo prestador que pode ser descoberta pelo cliente. Os serviços descritos na lista através de um contrato podem ser usados e reutilizados mantendo uma relação mínima de dependência (acoplamento fraco) e, além disso, podem ser compostos, ou seja, um serviço pode utilizar outros serviços.

FIGURA 2 ESTÁ SOLTA.

**Figura 2. Arquitetura de Webservices. [IBM 2014].**

WebServices são softwares constituintes da Arquitetura Orientada a Serviços (SOA) com grande interoperabilidade. Podem interligar sistemas de fornecedores distintos usando diferentes sistemas operacionais e escritos em linguagens de programação diferentes [RAMOS, 2012].

A pilha de protocolos utilizados nos webservices pode ser dividida em 4 camadas [CERAMI, 2002]:

Busca e Descoberta – utilizando protocolo UDDI.

WSDL – Responsável pela descrição dos serviços.

Troca de mensagens – usando protocolos XML-RPC, SOAP e XML

Transporte – usando protocolos |HTTP, SMTP, FTP, BEEP.

***3.6. Webservices* *SOAP-WSDL vs RESTFul***

Ricardo Ramos de Oliveira [RAMOS, 2012] realizou uma comparação entre a manutenibilidade de webservices confeccionados usando protocolo SOAP e linguagem de definição WSDL e webservices confeccionados usando a arquitetura RESTFul. Segundo sua conclusão webservices SOAP-WSDL são mais manuteníveis do lado cliente, enquanto webservices RESTFul são mais manuteníveis do lado servidor.

**3.7. Framework para RESTful em Java (Jersey)**

O JAX-RS é a especificação Java para o uso de Web Services Rest. O Jersey implementa e expande a especificação JAX-RS. Sua versão mais atual é a 2.7 [JERSEY, 2014].

**3.8. JSON e Gson**

A biblioteca Gson é uma biblioteca que pode ser usada para converter objetos Java em suas representações em JSON [GSON, 2014].

**4. Metodologia**

O trabalho será desenvolvido com base um uma pesquisa exploratória e consulta a material de livros o método do Paulo Freire, através do desenvolvimento de um protótipo de descoberta de conhecimento a partir de bases textuais. A metodologia de desenvolvimento deste trabalho é dividida em três etapas:

Etapa 1: análise da literatura focando nas seguintes áreas: descoberta de conhecimento e computação distribuída de conteúdo.

Etapa 2: proposição da arquitetura lógica e física com suporte à execução distribuída de conteúdo.

Etapa 3: modelagem da camada de persistência voltada ao processo descoberta de conhecimento com bases nos conteúdos disponibilizados.

Etapa 4: desenvolvimento da camada de persistência e do protótipo de descoberta de conhecimento com base na arquitetura de conteúdo, proposta anteriormente.

Etapa 5: testes do protótipo considerando um cenário de uso.

Etapa 5: avaliação dos resultados obtidos através da utilização da arquitetura proposta.

**4.1. Definição da área ou população-alvo**

O B*alder-Education*, foi desenvolvido através dos métodos do Paulo Freire tendo um público alvo que é bastante vasto, pensando nas dimensões continentais do nosso país ou até mesmo mundiais. Através das *“palavras geradoras*” e meios tecnológicos, será levado para adultos analfabetos condições de se alfabetizarem e assim enriquecendo seus conhecimentos, terá mais oportunidades no mercado de trabalho.

**4.2. Arquitetura**

O sistema é baseado na ideia de micro-serviços **[MICRO-SERVIÇOS]**, onde cada componente principal é separado do outro para dividir a responsabilidade entre as aplicações. A ideia deste projeto é que a aplicação principal seja responsável pela visualização do conteúdo (balder-site), e com ele tem um backend para consumir e integrar serviços, com as outras aplicações e seu banco (balder-site-core), essa integração pode ser feita com o sistema de geração de estatísticas (balder-statistics) ou com o sistema de administração (balder-admin), onde faremos todo o controle de criação de conteúdo e manipulação desses pelos professores.

Todas as aplicações serão integradas via RestFul **[MATURIDADE]**, no nível de maturidade 2, sendo que no futuro atingiremos a maturidade 3.

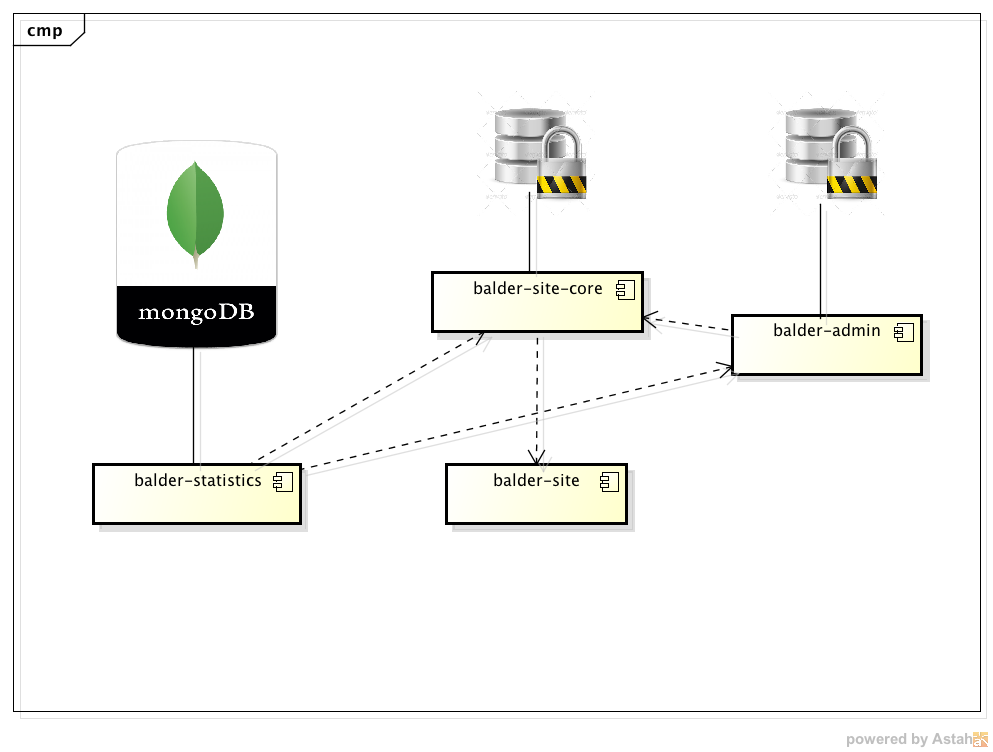
****

Figura 3 - Componentes da Arquitetura

**Balder admin:**

Aplicação criada em Java EE 7, usando os frameworks JSF 2.2, CDI 1.1, EJB 3.2, JPA 2.1 e Validation 1.1, rodando em Maven 3 e usando o servidor de aplicação JBoss WildFly 8.2, com sistema gerenciador de base de dados MySql.

**Balder Site:**

Aplicação feita em AngularJS, usando bawer, grunt.

**Balder site core**

Aplicação feita em Spring boot, utilizando os fameworks, Spring security, Spring MVC, Spring Data-JPA, e servidor apache Tomcat embarcado e banco mysql.

**Balder statistics**

Aplicação feita em Spring boot, utilizando os fameworks, Spring security, Spring MVC, Spring Data-JPA, e servidor apache Tomcat embarcado e banco Mongo-DB.

**4.3. Requisitos Funcionais**

A plataforma de ensino deve disponibilizar:

* O envio de sugestões para melhoria do mesmo.
* A autenticação do usuário.
* A alteração do usuário.
* A autenticação no sistema, adquirindo assim um acesso personalizado.
* A pesquisa de recursos disponíveis no Balder.

Para o administrador:

* Manter perfil para professores.
* Manter perfil para alunos.

Para o professor:

* Manter o conteúdo do sistema.
* Acompanhamento à evolução e dificuldades do aluno sobre o conteúdo estudado.

Para os administradores ou professores

* Definição do que é preciso para o aluno alcançar os níveis de seu desenvolvimento,

Ainda:

* A plataforma deverá ser intuitiva de modo que a pessoa que utilize o serviço não fique presa a mesma.
* A plataforma deverá ter uma inteligência própria para que o usuário que utilize a mesma possa ir avançando os níveis de educacionais.
* A plataforma deverá fornecer requisitos ao (s) usuário (s), administradores, professores e alunos para que o mesmo possa saber como está evoluindo a sua alfabetização.
* A plataforma de ensino deve ser disponibilizada pelo administrador o cadastro do professor na respectiva aula.
* A plataforma de ensino permitir que o administrador e o professor incluam aulas por níveis.
* A plataforma de ensino deverá ter inteligência própria para que a palavra lançada pelo professor ou administrador seja dividida em várias silabas;
* A plataforma de ensino analisa se o exercício foi realizado “CORRETO” ou “ERRADO”, caso esteja errado, a palavra virá destacada de vermelho, caso esteja correto, virá destacado de verde.
* A plataforma exibe um som ao escolher a sílaba para que o aluno possa ouvir como é a pronuncia da mesma.

As figuras abaixo de 1 até 20 mostram, esquematicamente os requisitos implementados através dos respectivos diagramas de caso de uso.

[RF 01] Enviar sugestões

Figura 1 - Requisito que possibilita que usuários enviem sugestões ou críticas relacionadas ao Balder.

**Ator:** Aluno

**Prioridade:** Desejável

*Entrada:* ***Sugestões/críticas***

**Saída Principal:** Confirmação do envio da informação

**Fluxo dos eventos principais:**

* O usuário acessa a tela de *Contato*
* O usuário insere nome, e-mail, comentário e confirma
* Sistema confirma o envio das informações

[RF 02] Efetuar cadastro no sistema

Figura 2 – Requisito quepossibilita que usuários não autenticados efetuem um cadastro no sistema.

**Ator:** Usuário não autenticado

**Prioridade:** Importante

*Entrada:* ***Informações cadastrais***

**Saída Principal:** Confirmação de cadastro

**Saída Alternativa:** Cadastro não realizado

**Fluxo dos eventos principais:**

* 1. O usuário não autenticado seleciona a opção *Cadastre-se*
* 2. O usuário não autenticado insere o número de matrícula pelo SIAC e confirma
* 3. O usuário não autenticado insere e-mail, login, senha, dica de senha e confirma
* 4. Sistema confirma cadastro

[RF 03] Alterar cadastro no sistema

Figura 3 – Requisito quepossibilita que usuários autenticados alterem informações relativas ao seu cadastro.

**Ator:** Usuário autenticado

**Prioridade:** Desejável

*Entrada:* ***Dados cadastrais modificados***

**Pré-condições:** Estar autenticado no sistema

**Saída Principal:** Confirmação da alteração do cadastro

**Saída Alternativa:** Alteração de cadastro não realizada

**Fluxo dos eventos principais:**

* 1. Usuário seleciona a opção *Alterar cadastro*
* 2. Usuário altera informações cadastrais e confirma
* 3. Sistema confirma a alteração do cadastro

[RF 04] Autenticar no sistema

Figura 4 – Requisito que valida o acesso de usuários cadastrados no sistema, possibilitando acesso personalizado.

**Ator:** Usuário

**Prioridade:** Importante

*Entrada:* ***login e senha***

**Pré-condições:** Estar cadastrado no sistema

**Saída Principal:** Usuário é direcionado para página pessoal

**Saída Alternativa:** Mensagem de acesso negado

**Fluxo dos eventos principais:**

* 1. Usuário seleciona a opção *Entrar*
* 2. Usuário insere login e senha
* 3. Sistema direciona usuário autenticado para a página pessoal

[RF 05] Realizar busca no sistema

Figura 5 – Requisito que permite que usuários localizem recursos disponíveis no Balder, através de filtros de busca.

**Ator:** Usuário

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

1. Usuário acessa a tela de *Biblioteca*

* 2. Usuário insere palavras-chave, seleciona tipo de recurso pesquisado, quantidade de registros por página e confirma.
* 3. Sistema retorna o resultado da busca.

RF 06] Realizar cadastro professor

Figura 6 – Requisito quepermite que administradores faça cadastro no sistema.

**Ator:** Administrador

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

* 1. Usuário acessa a tela de *cadastro de Professor.*
* 2. Sistema retorna o resultado da busca

RF 07] Realizar Edição do professor

Figura 7 – Este requisito permite que administradores ou próprio faça alteração do perfil do professor no sistema.

**Ator:** Professor

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

3. Usuário acessa a tela de perfil para alteração *do Professor.*

4. Sistema retorna o resultado da busca

RF 08] Realizar Exclusão do professor

Figura 8 –Este requisito permite que administradores ou próprio faça exclusão do perfil do professor no sistema.

**Ator:** Professor

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

5. Usuário acessa a tela de perfil para exclusão *do Professor.*

6. Sistema retorna o resultado da busca

RF 09] Realizar cadastro aluno

Figura 9 – Este requisito permite que administradores ou próprio aluno faça cadastro no sistema.

**Ator:** Aluno

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

7. Usuário acessa a tela de *cadastro de* Aluno*.*

8. Sistema retorna o resultado da busca

RF 10] Realizar Edição do aluno

Figura 10 – Este requisito permite que administradores ou próprio aluno, faça alteração do perfil do professor no sistema.

**Ator:** Aluno

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

9. Usuário acessa a tela de perfil para alteração *do* Aluno*.*

10. Sistema retorna o resultado da busca

RF 11] Realizar Exclusão do aluno

Figura 11 – Este requisito permite que administradores ou próprio aluno faça exclusão do perfil do professor no sistema.

**Ator:** Aluno

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

11. Usuário acessa a tela de perfil para exclusão *do* Aluno*.*

12. Sistema retorna o resultado da busca

RF 12] Realizar Cadastro de Professor na Aula

Figura 12– Este requisito permite que administradores cadastre o professor na respectiva aula.

**Ator:** Administrador.

**Prioridade:** Importante

*Entrada:* ***Dados do Professor***

**Saída Principal:** Confirmação do cadastro do Professor na aula

**Saída Alternativa:** Cadastro não realizado

**Fluxo dos eventos principais:**

13. Administrador acessa o conteúdo do professor*.*

14. Administrador cadastra o conteúdo referente a aula.

RF 13] Realizar Adição de conteúdo no sistema

Figura 13 – Este requisito permite que administradores ou professores adicione conteúdo para ser visualizado pelos alunos no sistema.

**Ator:** Aluno, Professor, Administrador.

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

14. Usuário acessa a tela de cadastro de conteúdo, para inserir exercícios iniciais, médios, avançados ou media-types*.*

15. Sistema retorna o resultado da busca

RF 14] Realizar Alteração de conteúdo no sistema

Figura 14 – Este requisito permite que administradores ou professores alterar o conteúdo para ser visualizado pelos alunos no sistema.

**Ator:** Aluno, Professor, Administrador.

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

16. Usuário acessa a tela de cadastro de conteúdo, para alterar, conteúdo de texto ou media-types*.*

17. Sistema retorna o resultado da busca

RF 15] Realizar Exclusão de conteúdo no sistema

Figura 15 – Este requisito permite que administradores ou professores excluir o conteúdo para não ser mais, visualizado pelos alunos no sistema.

**Ator:** Aluno, Professor, Administrador.

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

18. Usuário acessa a tela de cadastro de conteúdo, para excluir, conteúdo de texto ou media-types*.*

19. Sistema retorna o resultado da busca

RF 16] Relacionar conteúdo com níveis { básico, intermediário, avançado }

Figura 16 – Este requisito permite que sistema avance os níveis a partir do desempenho do aluno.

**Ator:** Sistema

**Prioridade:** Importante

*Entrada:* ***Exercício***

**Saída Principal:** Exercício encontrado com sucesso

**Saída Alternativa:** Nenhum exercício encontrado

**Fluxo dos eventos principais:**

20. Usuário acessa a tela de cadastro de qual nível esta atualmente*.*

21. Sistema retorna o exercício a ser realizado

22. O sistema analisa se o aluno conseguiu atingir o nível do exercício e muda de nível

RF 17] Dividir as palavras em várias sílabas

Figura 17 – Esse requisito permite o administrador ou professores ao lançar as palavras, o sistema separe as sílabas e embaralhe.

**Ator:** Sistema

**Prioridade:** Importante

*Entrada:* ***Palavras***

**Saída Principal:** Palavra encontrada com sucesso

**Saída Alternativa:** Nenhuma palavra encontrada

**Fluxo dos eventos principais:**

23. Professor lança as palavras dependendo do nível

24. Sistema separa a sílabas das palavras lançadas.

RF 18] Valida o exercício feito pelo aluno com “CORRETO” ou “ERRADO”

Figura 18 – Esse requisito permite que o sistema valide se o exercício feito pelo aluno está de forma correta ou incorreta

**Ator:** Sistema

**Prioridade:** Importante

*Entrada:* Exercício

**Saída Principal:** Exercício encontrado

**Saída Alternativa:** Nenhum exercício encontrado

**Fluxo dos eventos principais:**

25. Usuário acessa a tela do exercício

26. Sistema retorna o resultado da busca

27. O sistema valida se o exercício está correto ou errado, mostrando as palavras “CORRETO” ou “ERRADO”

RF 19] Criar relatórios de métricas de evolução do aluno.

Figura 19 – Esse requisito permite que se o aluno juntar as sílabas corretamente, o sistema exibe a pronuncia para o aluno.

**Ator:** Sistema.

**Prioridade:** Importante

*Entrada:* Pronuncia

**Saída Principal:** Pronuncia da palavra executada

**Saída Alternativa:** Nenhuma pronuncia foi encontrada

**Fluxo dos eventos principais:**

28. Aluno junta as sílabas, formando as palavras

29. Sistema valida a palavra formada

30. Se a palavra estiver correta, sistema exibe a pronuncia.

RF 20] Criar relatórios de métricas de evolução do aluno.

Figura 20 – Esse requisito permite ao administrador e professores, visualizar a progressão do aluno no sistema. O relatório deverá conter a evolução e dificuldades do aluno sobre o conteúdo estudado.

**Ator:** Professor, Administrador.

**Prioridade:** Importante

*Entrada:* ***Palavras-chave***

**Saída Principal:** Resultado da busca

**Saída Alternativa:** Nenhum resultado encontrado com as palavras-chave indicadas

**Fluxo dos eventos principais:**

31. Usuário acessa a tela de nível para visualizar sua progressão atual*.*

32. Sistema retorna o resultado da busca

33. O Administrador ou professor define os quais os objetivos que deverá ser atingidos pelos alunos.

**4.4- Requisitos não funcionais**

A plataforma disponibilizada deverá ser gratuita e possuir uma estrutura mínima para implantação. Será toda desenvolvida na linguagem de desenvolvimento Java, com uso de software livre sobre a licença GPL. A tabela 1 mostra, detalhadamente, os requisitos a serem implementados.

**Tabela 1 – Requisitos não Funcionais do Balder Education**

|  |  |
| --- | --- |
| **Requisitos não funcionais** |  |
| **Requisito** | **Detalhamento** |
| Banco de dados | Deve-se utilizar SQL padrão, para se ter independência de banco de dados utilizaremos o Framework hibernate na camada de persistência, afim de facilitar transações no banco de dados. |
| Interface Web | A camada de apresentação utilizará o Framework Prime Faces para facilitar o tratamento das ações dos usuários. |
| Linguagem de Desenvolvimento (servidor) | O sistema deverá ser desenvolvido em Java, que é uma linguagem multi-plataforma. |
| Linguagem de Desenvolvimento | O sistema será desenvolvido em Java como linguagem de servidor e Java script como linguagem Cliente. Java Script será utilizado nos casos em que se precisar validar formulários HTML. |
| Banco de dados (Distribuição) | O banco de dados que suportará o ambiente será o MySQL; |
| Servidor de Aplicação | O sistema deverá ser implantado em um servidor Web JBoss |
| Browsers Cliente | As estações clientes deverão executar o sistema através do browser Microsoft Internet Explorer 10 ou maior, Firefox 35 ou superior, *google chrome* 40 ou superior |

Colocar aqui o diagrama completo.

Figura 2 – Diagrama de caso de uso contendo os requisitos funcionais do sistema.

Análise e Discussão da Proposta

**Conclusão**

A arquitetura monolítica de sistemas vem perdendo espaço por arquiteturas mais leves como micro-serviços e integrações via serviços RestFul, tornando hoje algo mais viável, rápido e escalável. Ao contrário das arquiteturas monolíticas que tudo é juntado em um único pacote e um único deploy. Devido essa a grande vantagem do micro serviço, onde tudo é separado por aplicações menores baseadas em regras de negócios comuns. Com isso todas as regras de negócios ficam separadas em pequenos serviços, facilitando assim o a manutenção e deploys continuados e tirando de um único sistema a responsabilidade por todas as regras de negócio.

O projeto proposto é para um bem social, ondem qualquer professor poderá cadastrar aulas e conteúdos e acompanhar seus alunos para visualizar a progressão de aprendizagem do mesmo, sendo de grande potencial para os ensinos de analfabetos e analfabeto-funcionais, principalmente na data adulta.

É evidente que os maiores resultados serão obtidos em longo prazo, pois o sistema ainda não está pronto, e estará em atualização em curto prazo para o termino, no entanto, com o potencial que demonstra, quando estiver em produção será uma ferramenta de grande porte para análise e acompanhamento de pessoas com essas dificuldade de aprendizados, já que ele é baseado no método XXX, utilizado por larga escala por professores especializados por ser criação do Dr. Paulo Feire, renomado e aceito no mundo todo.

A aplicação móvel e web e de estatísticas tendem a evoluir, sendo que, a intenção é que posteriormente ocorram melhorias e seja desenvolvido um módulo de acompanhamento para dispositivos móveis, o qual permitirá que este visualize os relatórios e aulas de reforço de aprendizagem e realize pedidos das suas dificuldades para seus professores

Com o modelo de WebService baseado em RESTful construído é possível que se trabalhe de forma incremental, sempre adicionando novos recursos e ainda com o benefício de atender a qualquer outro tipo de aplicação futura. Sendo que nosso nível de maturidade de RestFul hoje, só atinge o nível 2.

No processo de desenvolvimento algumas dificuldades foram encontradas, relativas ao conhecimento das tecnologias ou pela restrição de tempo do projeto. Contudo, estes imprevistos foram superados buscando ajuda de profissionais da área, pesquisando soluções, trabalhando em grupo e realizando decisões de projeto, como a remoção de algumas funções extras almejadas na concepção do projeto, colaborando dessa forma para que os requisitos definidos fossem cumpridos.

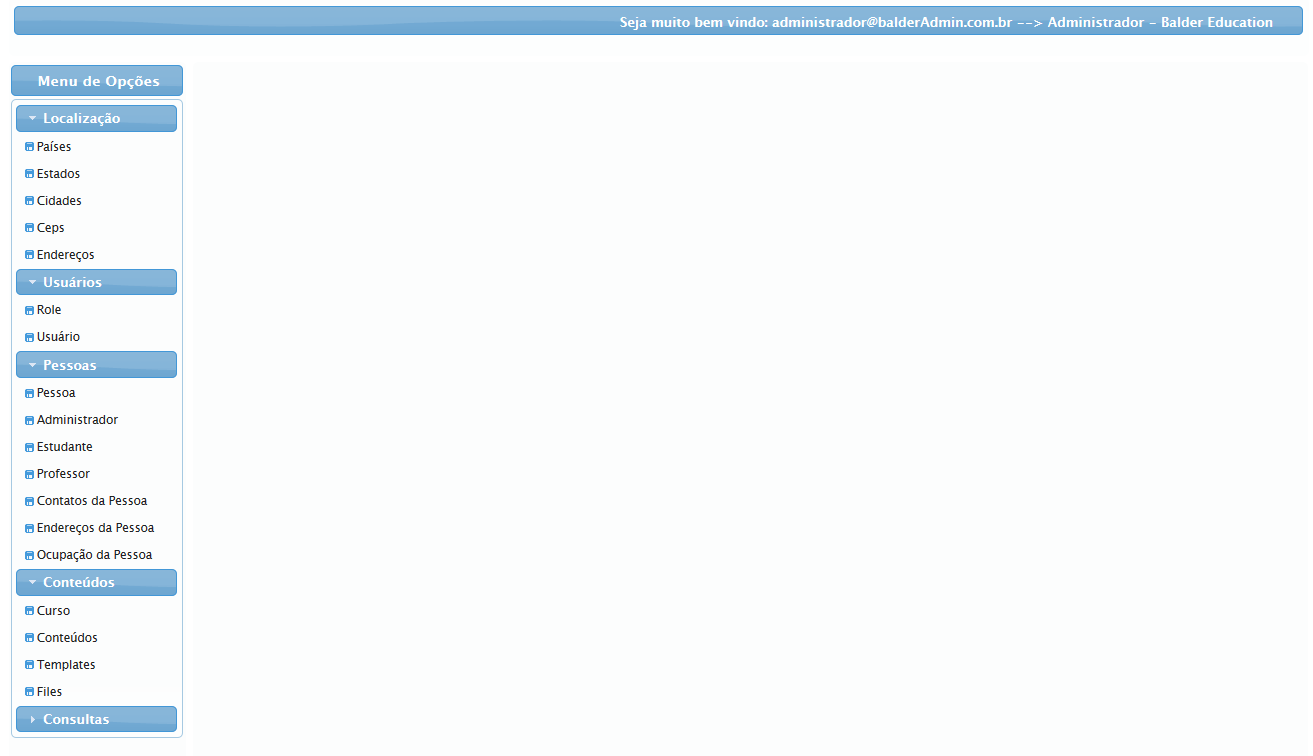
Por fim, as tecnologias escolhidas para o projeto contribuíram para que o trabalho resultasse em vasto aprendizado, experiências foram adquiridas por todos os integrantes do grupo, e ainda, os requisitos do projeto puderam ser alcançados de forma satisfatória. A grande aprendizagem do grupo além de métodos educacionais baseados no Dr. Paulo Freire, foi de arquitetura de projetos de micro serviços em Java, como integrar via RestFul, hospedar no cloud da Amazon, e trabalhar em equipe para aprender processos tanto técnicos quanto dissertativos.

**ANEXO 1 – IMAGENS DO SISTEMA DO ADMINISTRADOR.**

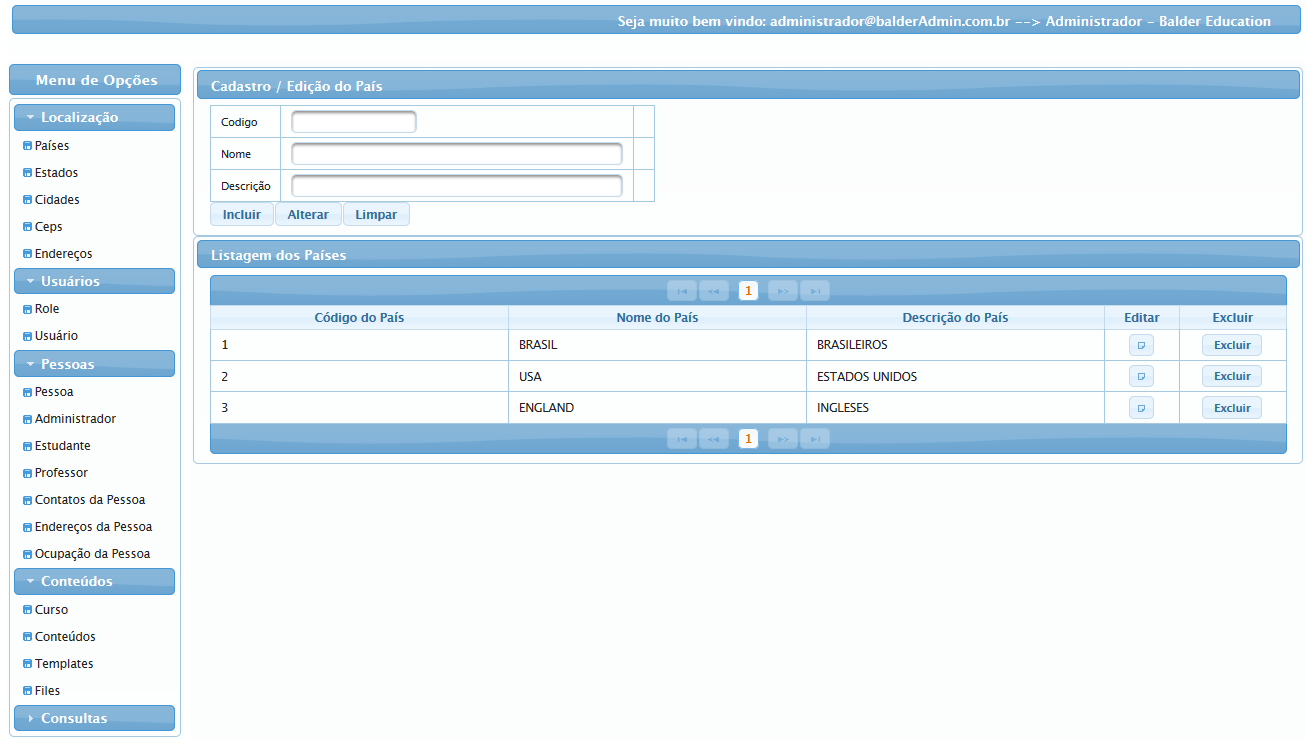
**Figura 1 – Tela de Login do Administrador**

****

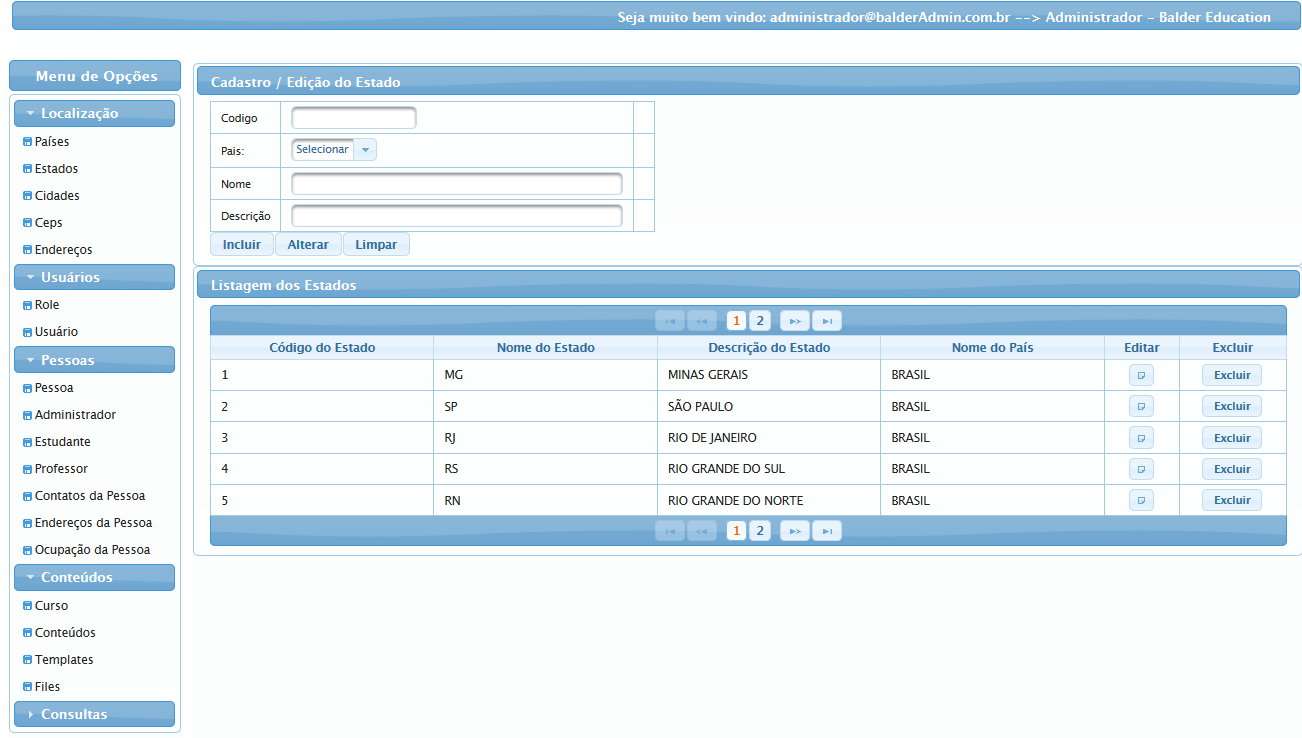
**Figura 2 – Menu de opções do administrador**

****

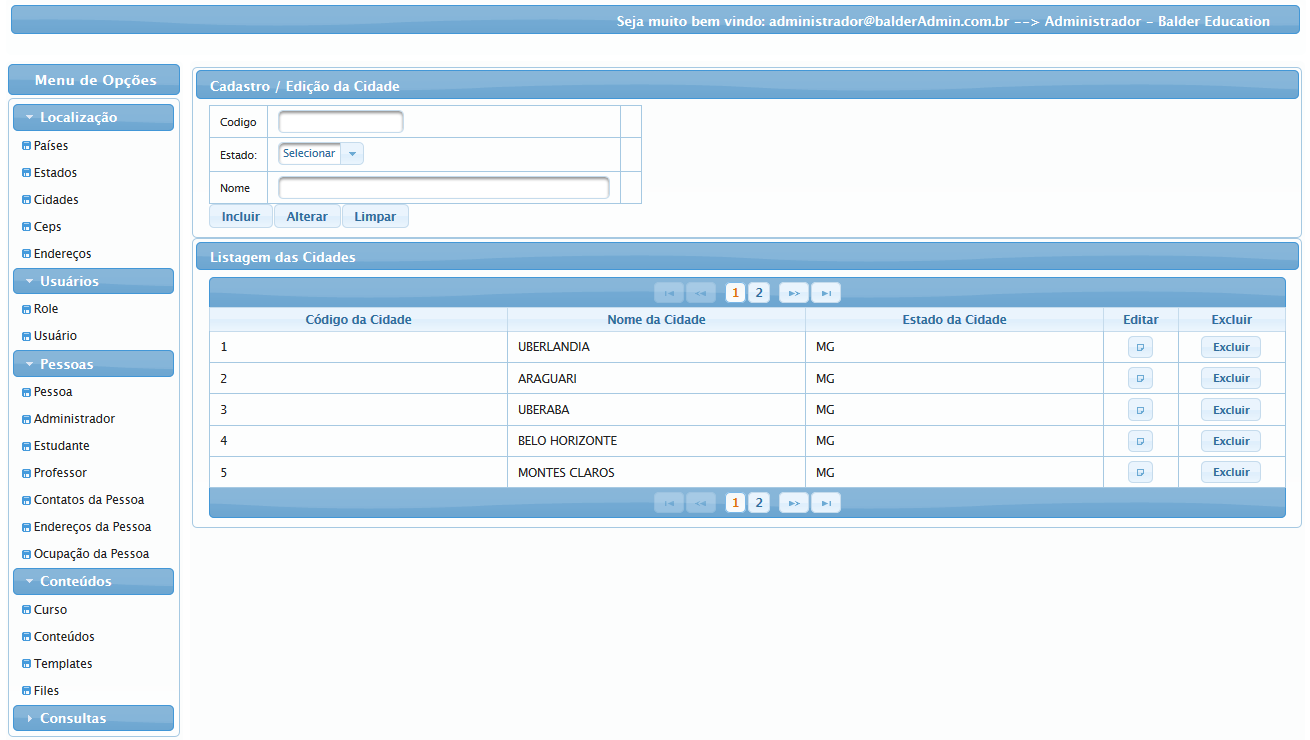
**Figura 3 – Inclusão / Edição do País**

****

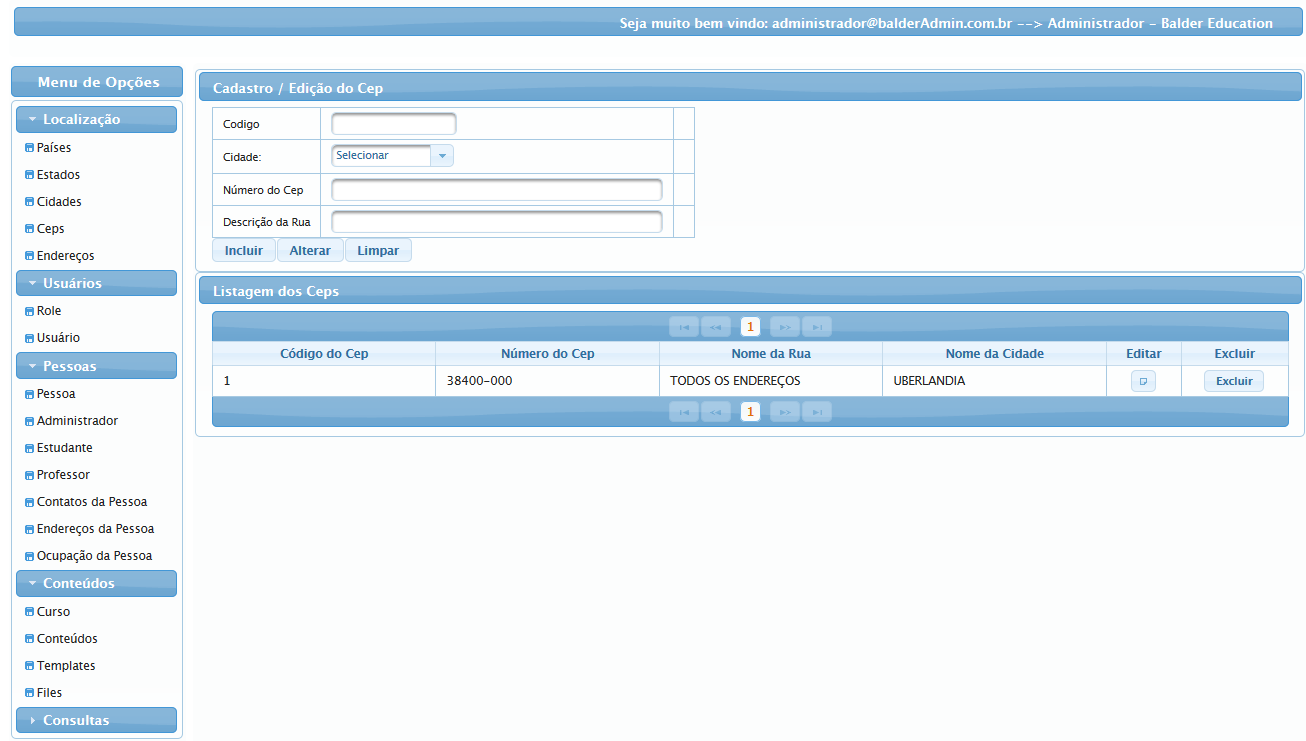
**Figura 4 – Inclusão / Edição do Estado**

****

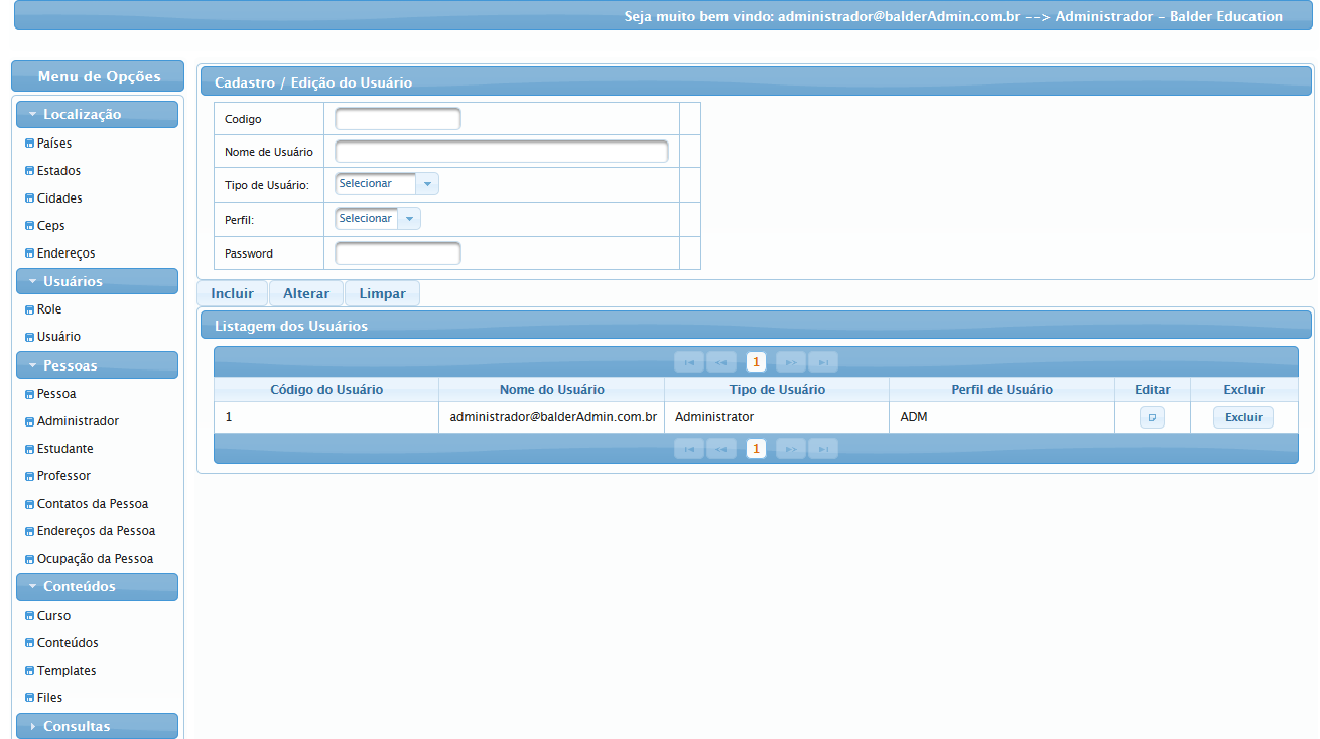
**Figura 5 – Inclusão / Edição da Cidade**

****

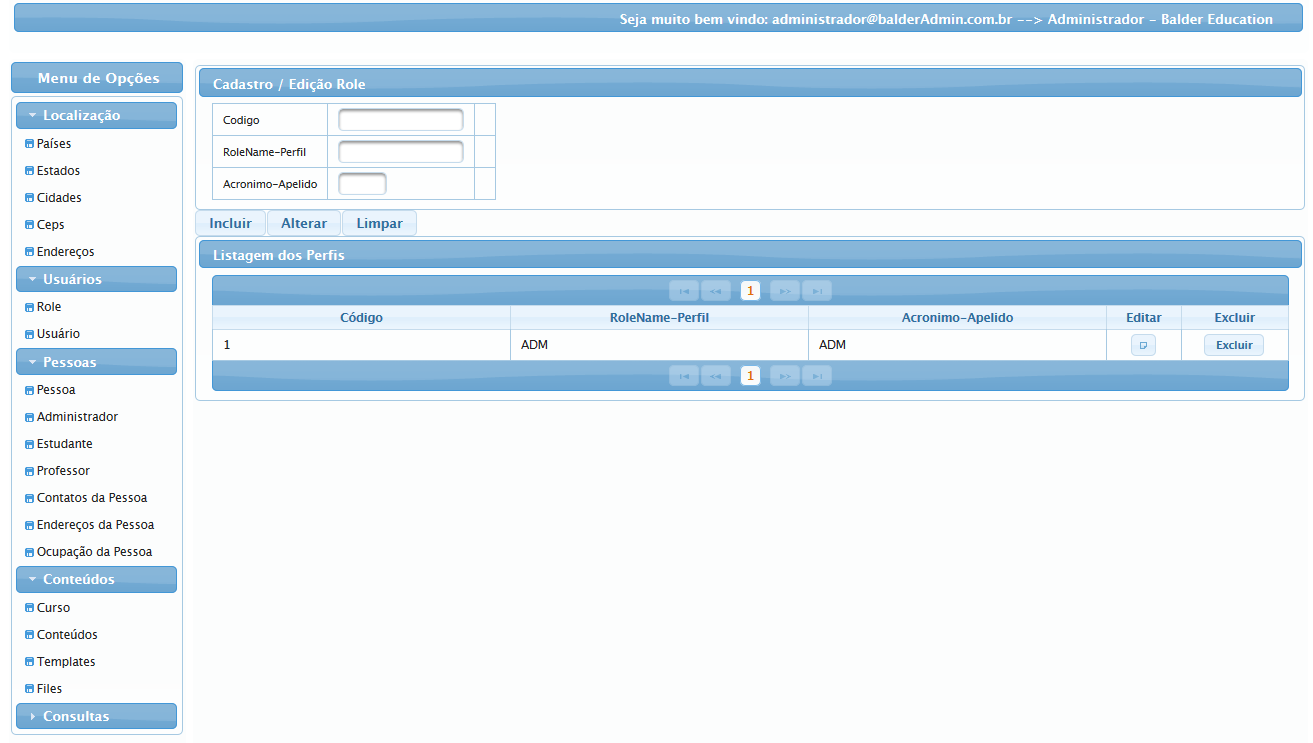
**Figura 6 – Inclusão/Edição do CEP**

****

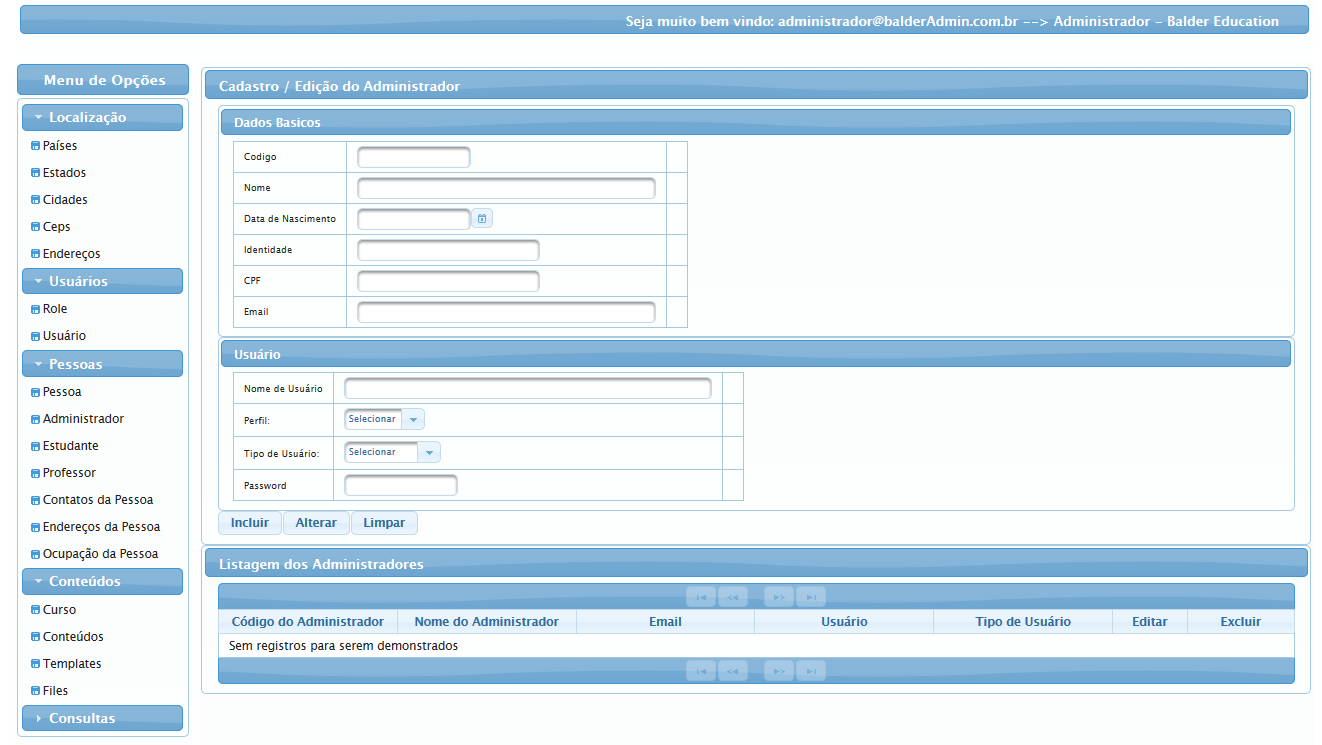
**Figura 7 –Inclusão/Edição do Usuário**

****

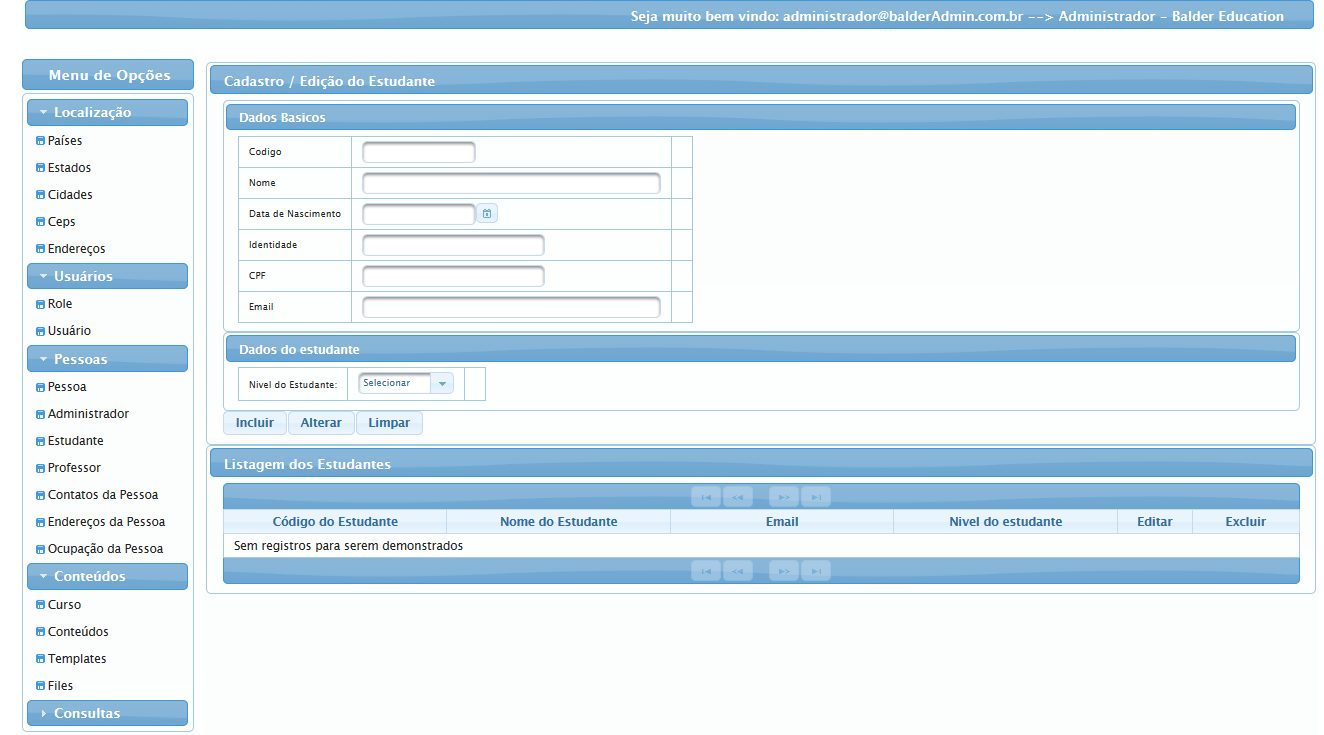
**Figura 8 –Inclusão/Edição de Role**

****

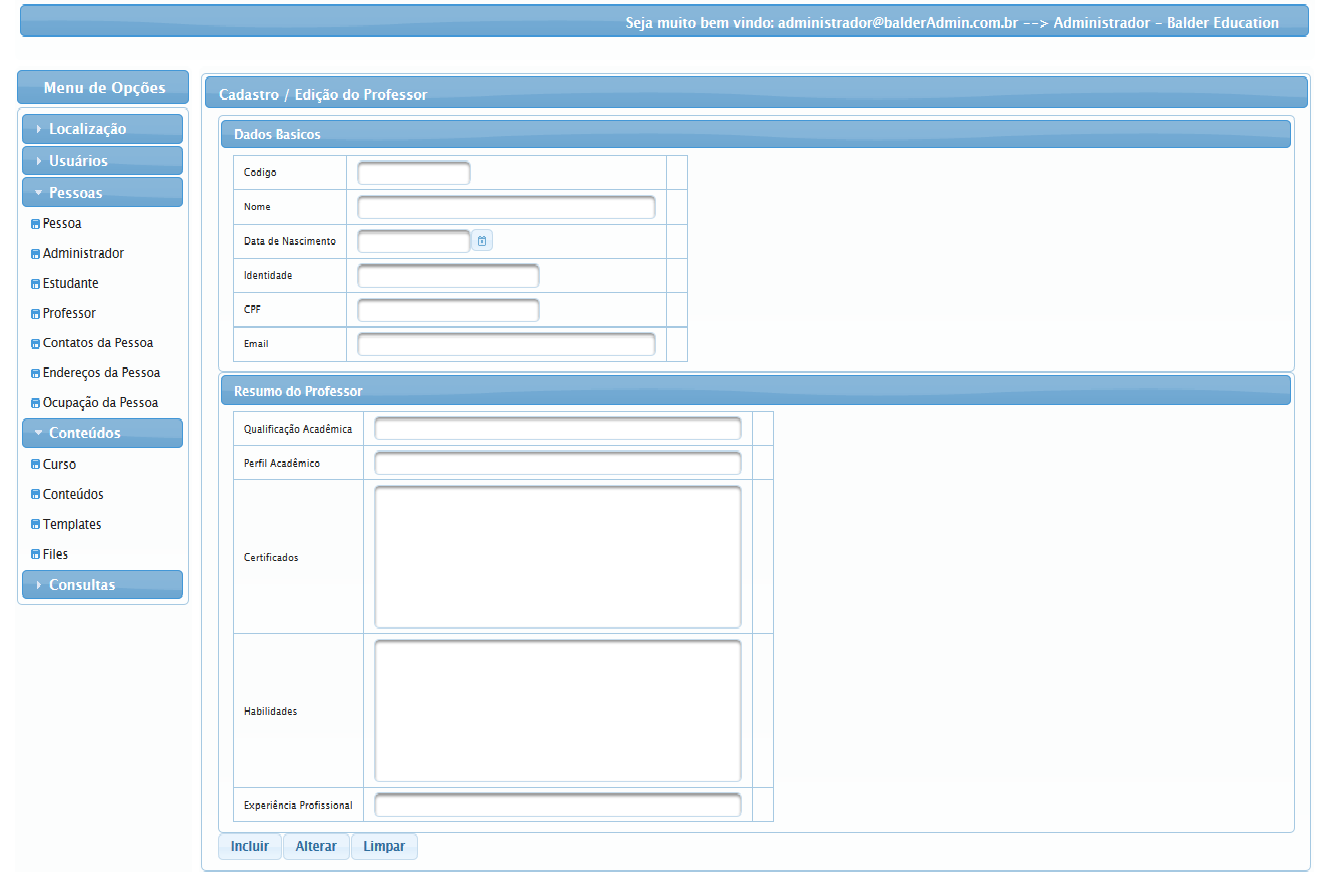
**Figura 10 – Inclusão / Edição do administrador**

****

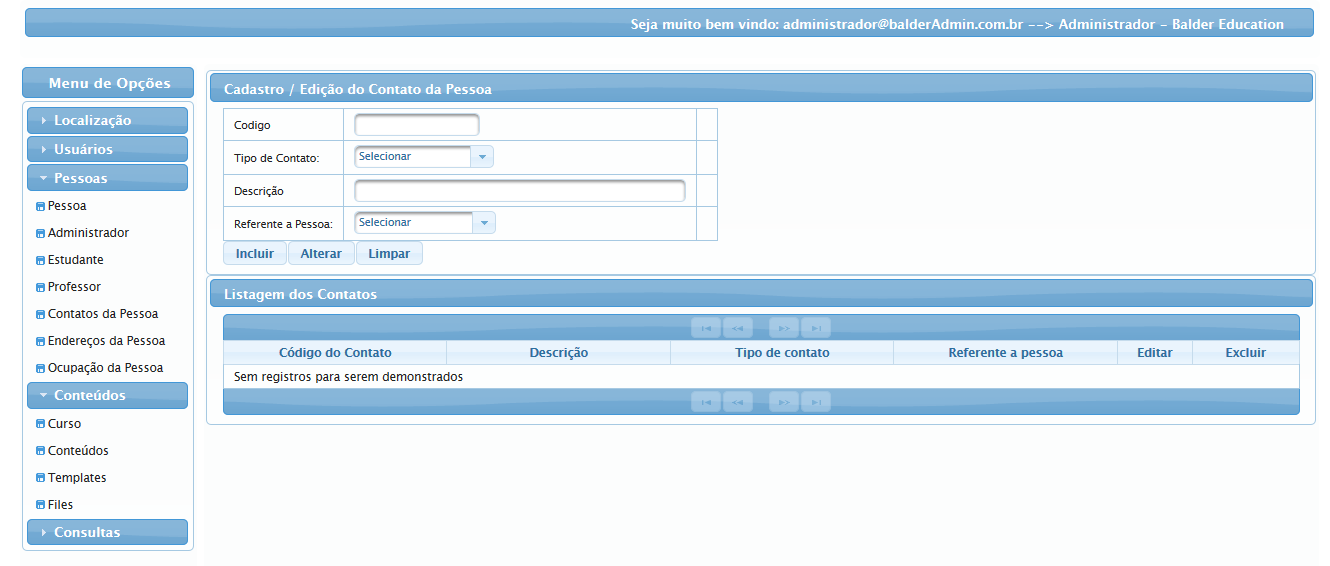
**Figura 11 – Inclusão / Edição do Estudante**

****

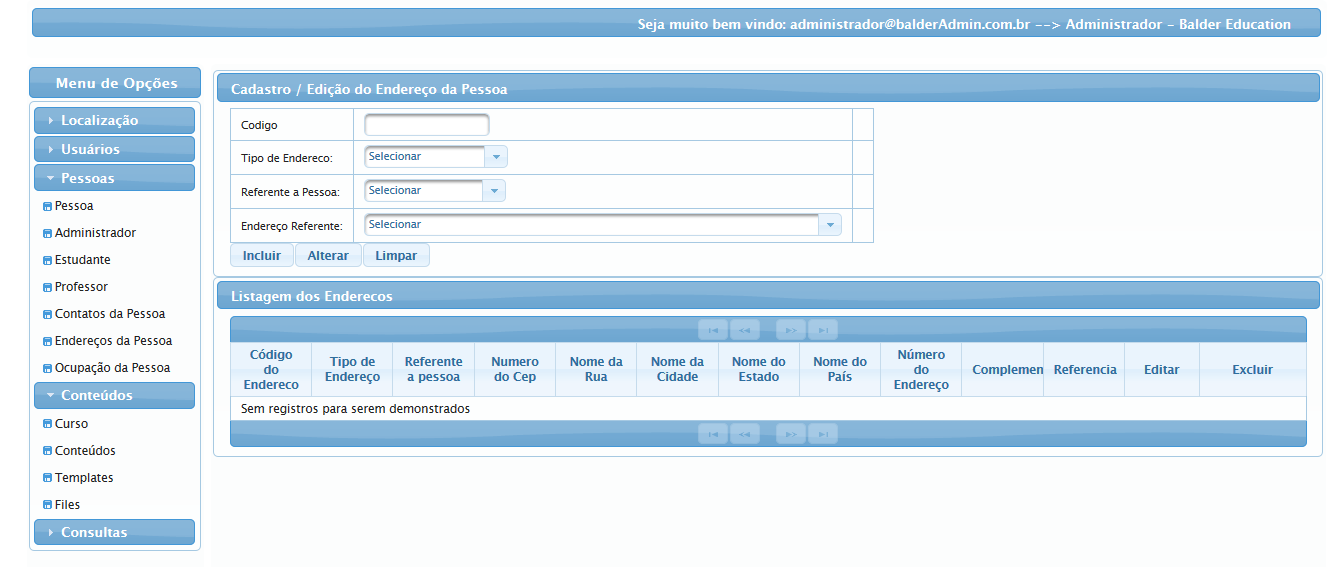
**Figura 12 – Inclusão / Edição do Professor**

****

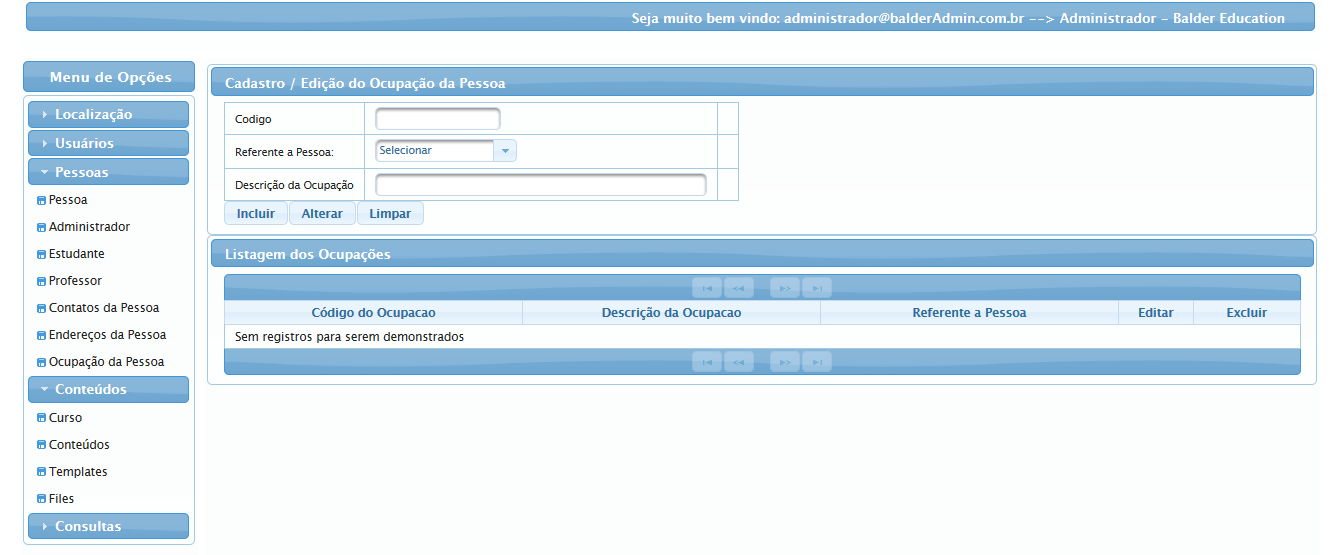
**Figura 13 – Inclusão / Edição da Pessoa**

****

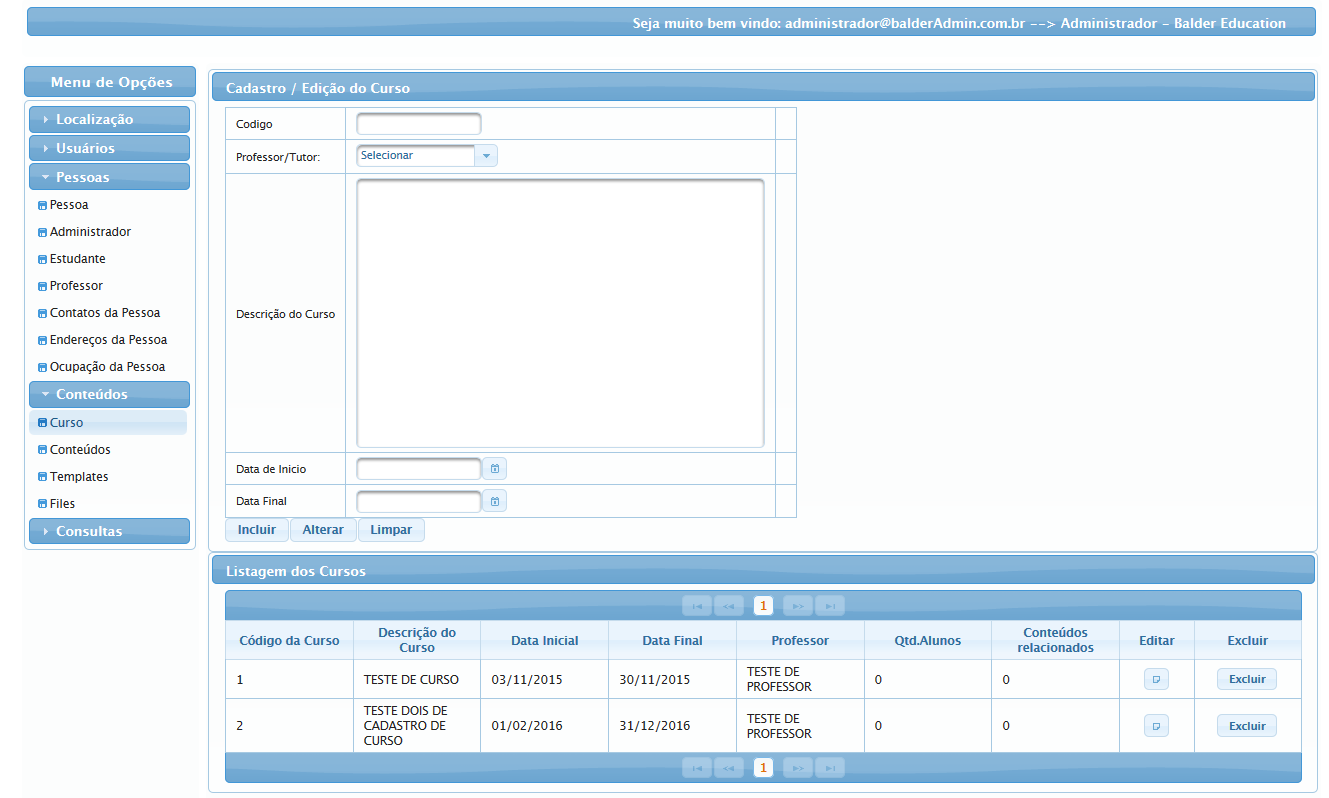
**Figura 14 –Inclusão / Edição do Endereço da Pessoa**

****

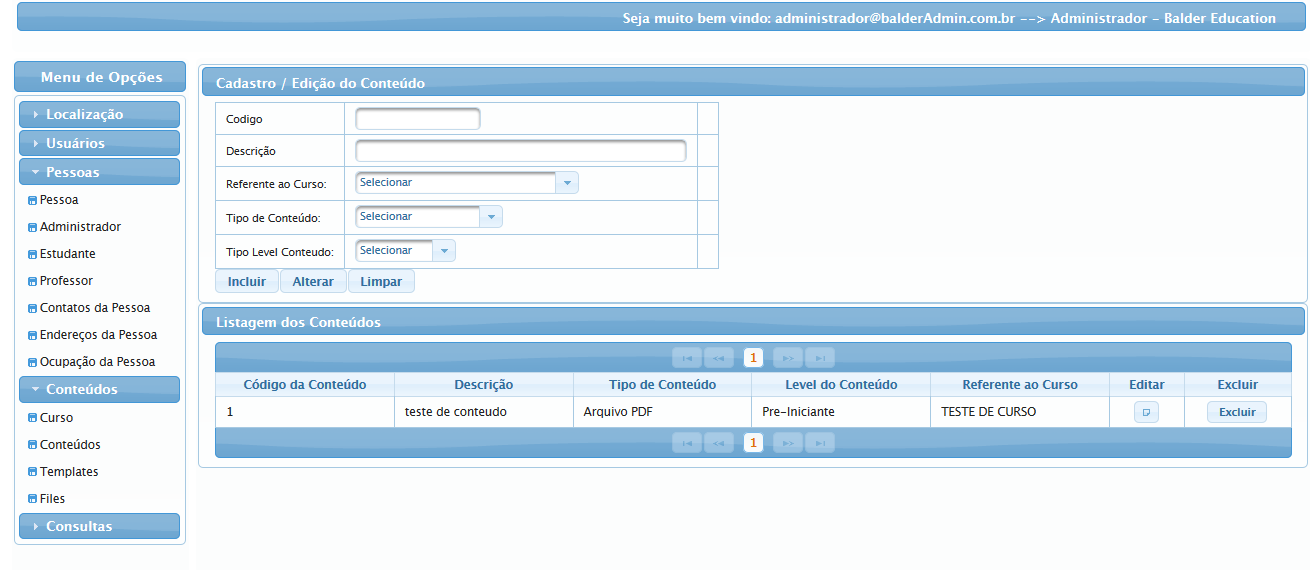
**Figura 15 – Inclusão / Edição da Ocupação da Pessoa**

****

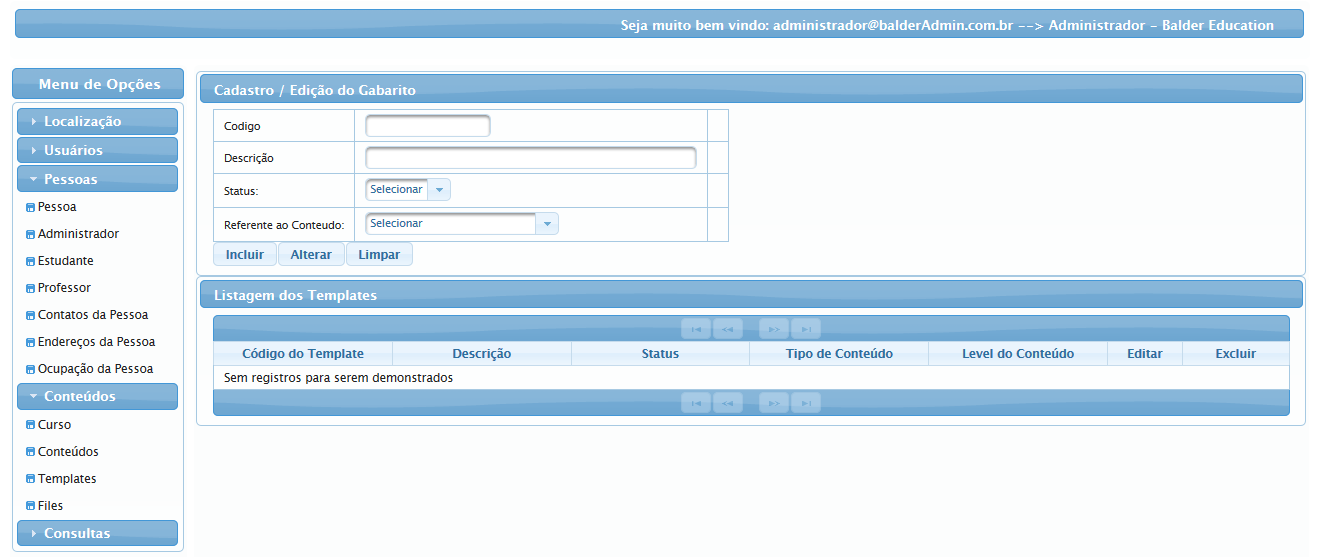
**Figura 16 – Inclusão / Edição do Curso**

****

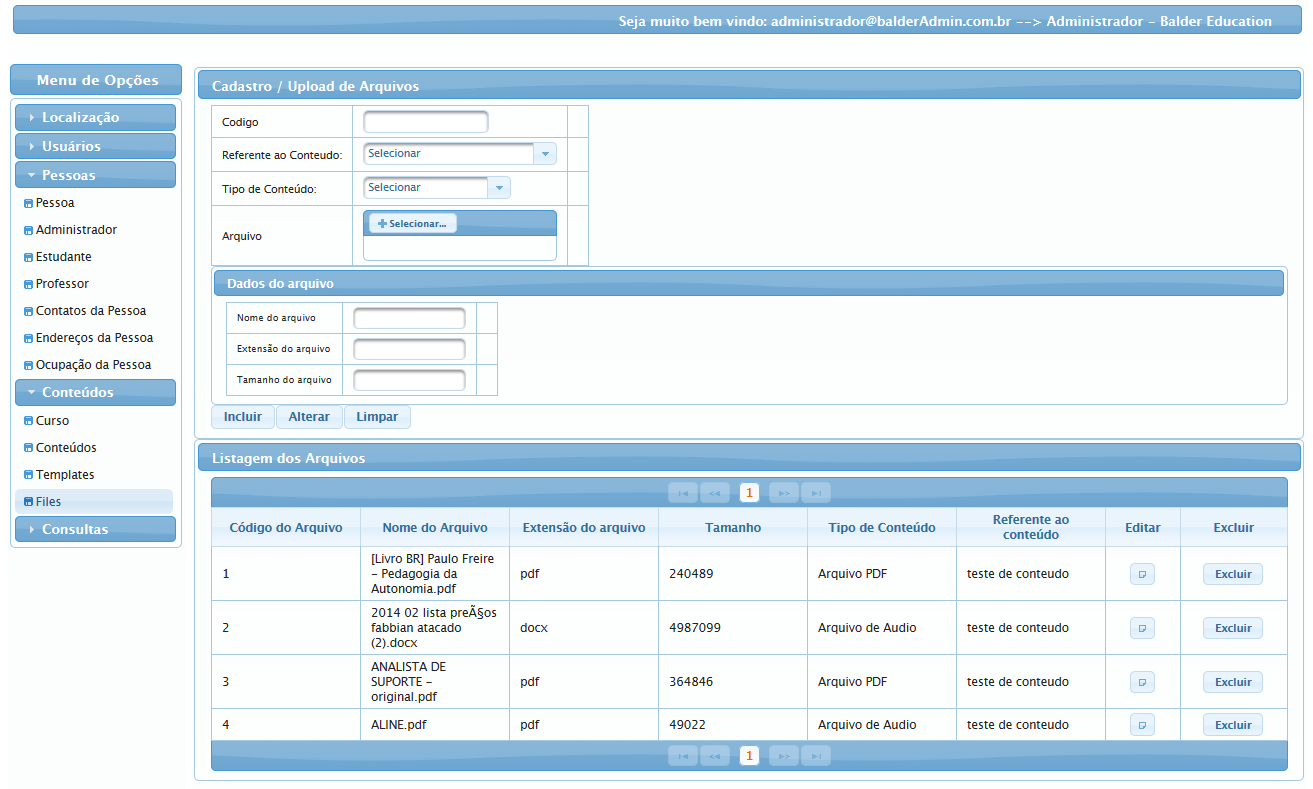
**Figura 17 – Inclusão / Edição de Conteúdos**

****

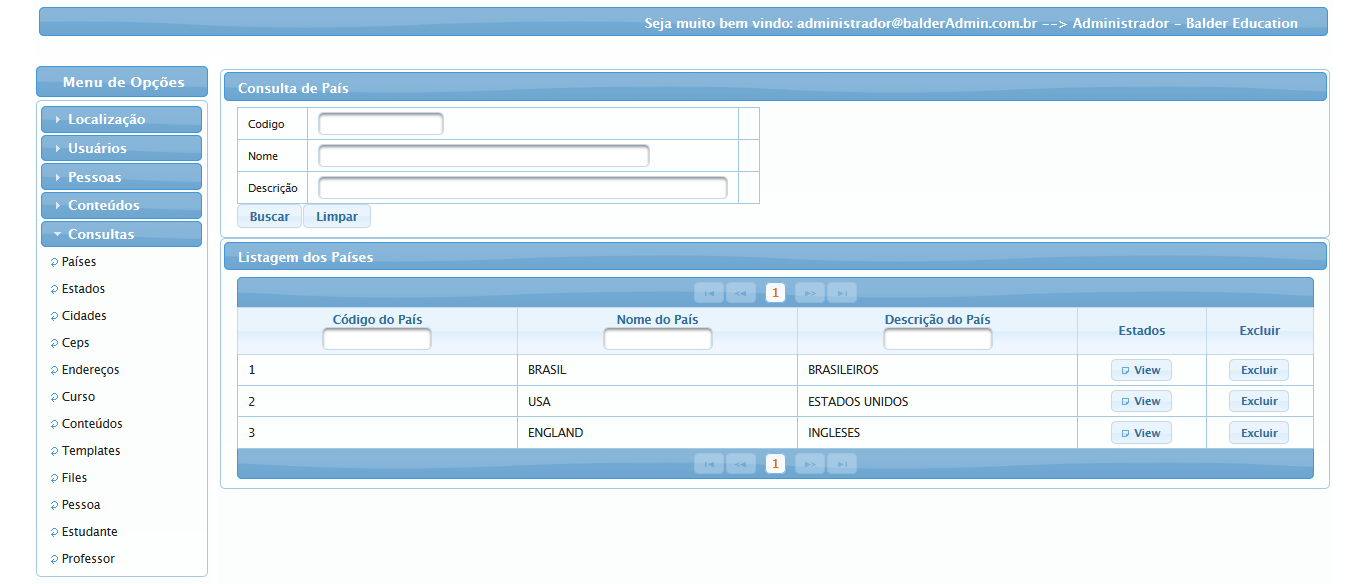
**Figura 18 – Inclusão / Edição do Gabarito**

****

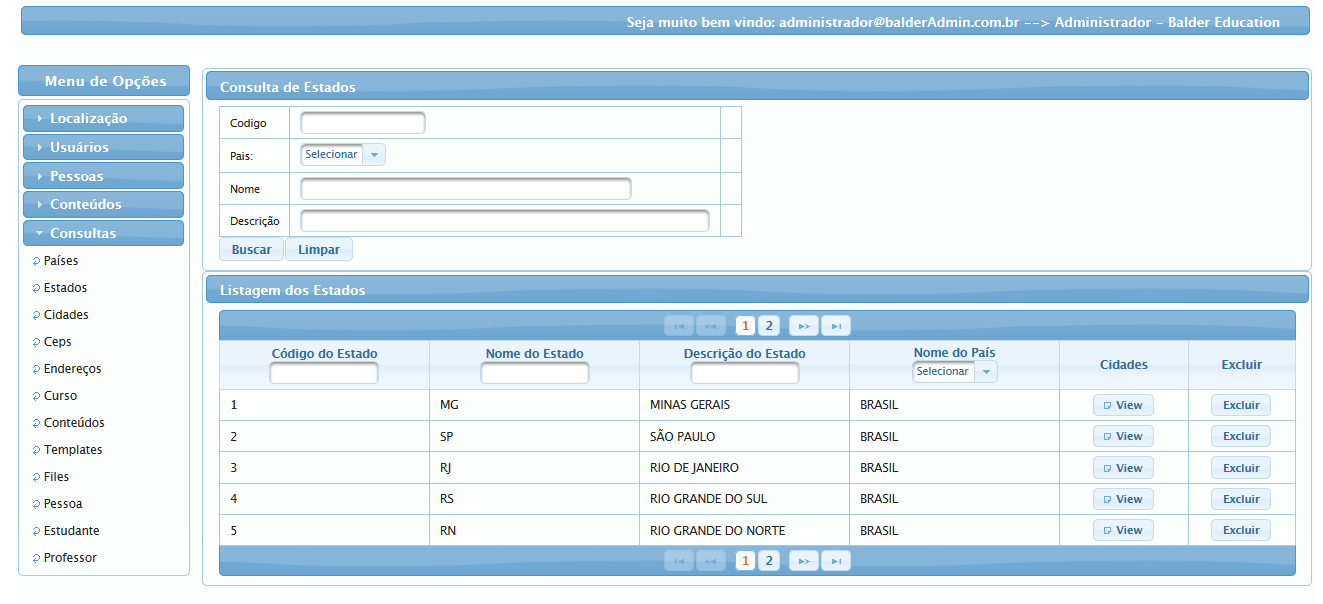
**Figura 19 – Inclusão de Arquivos**

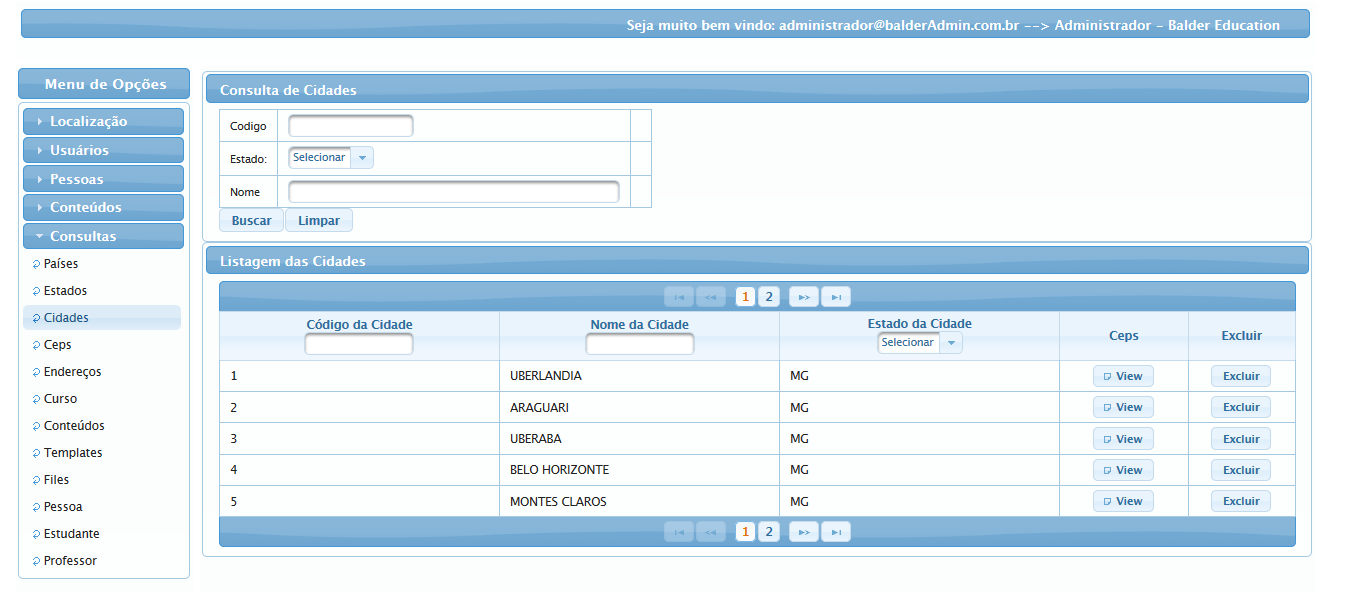
****

**Figura 20 – Consulta de País**

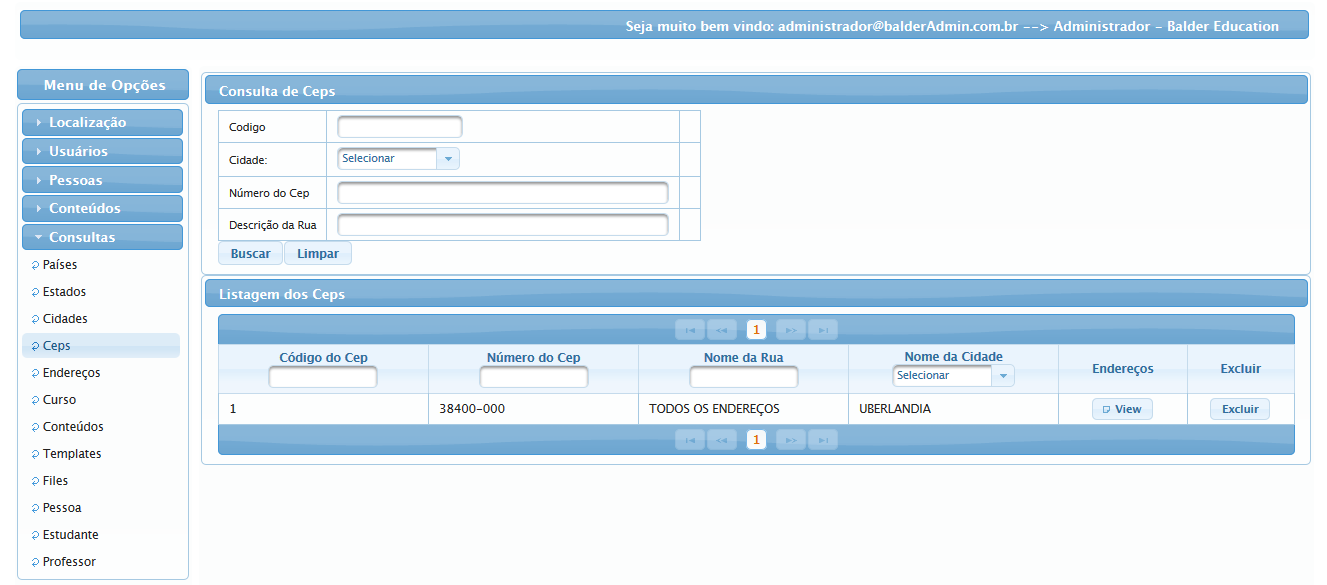
****

**Figura 21 – Consulta de Estados**

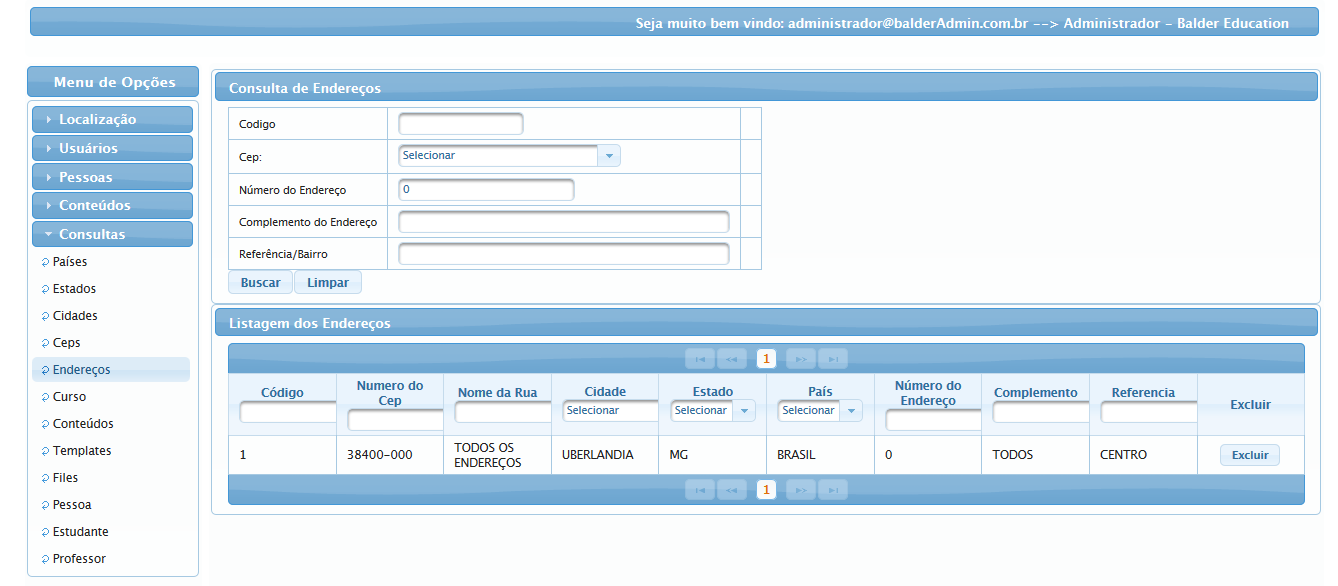
****

**Figura 22 –Consulta de Cidades**

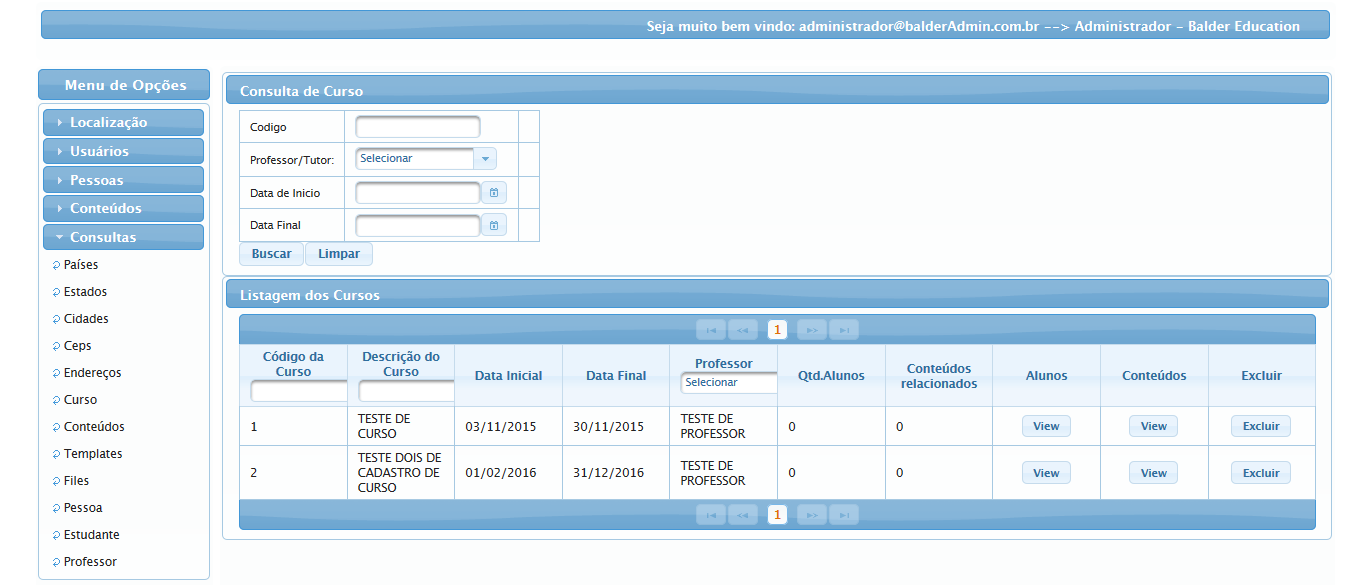
**Figura 23 – Consulta de Ceps**

****

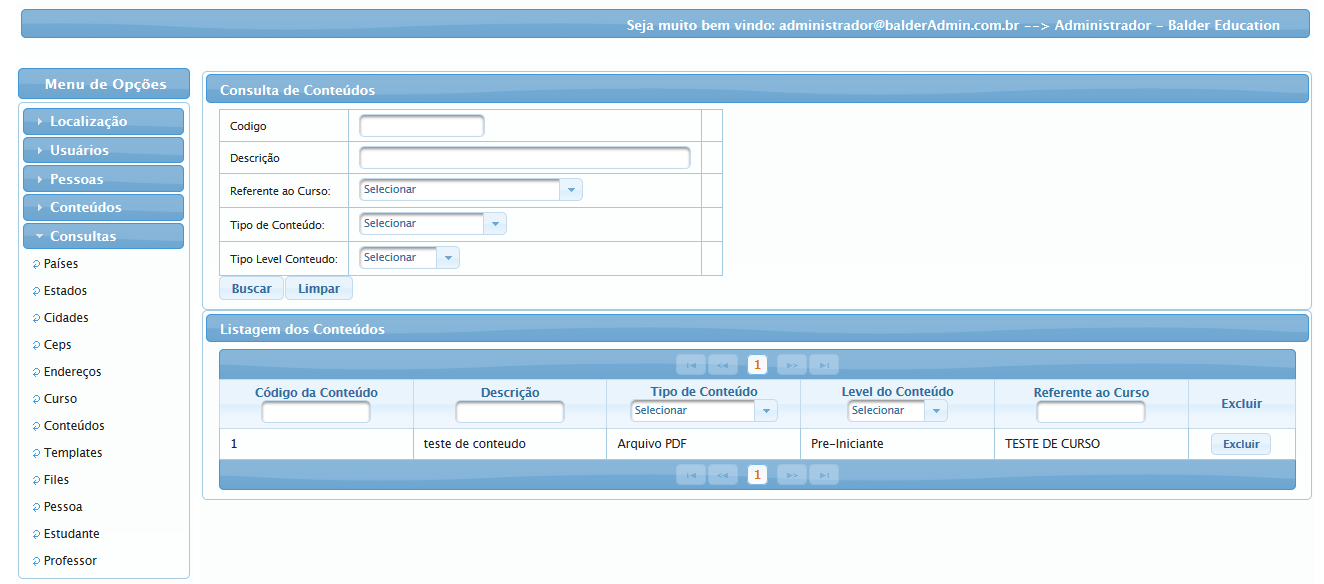
**Figura 24 – Consulta de Endereços**

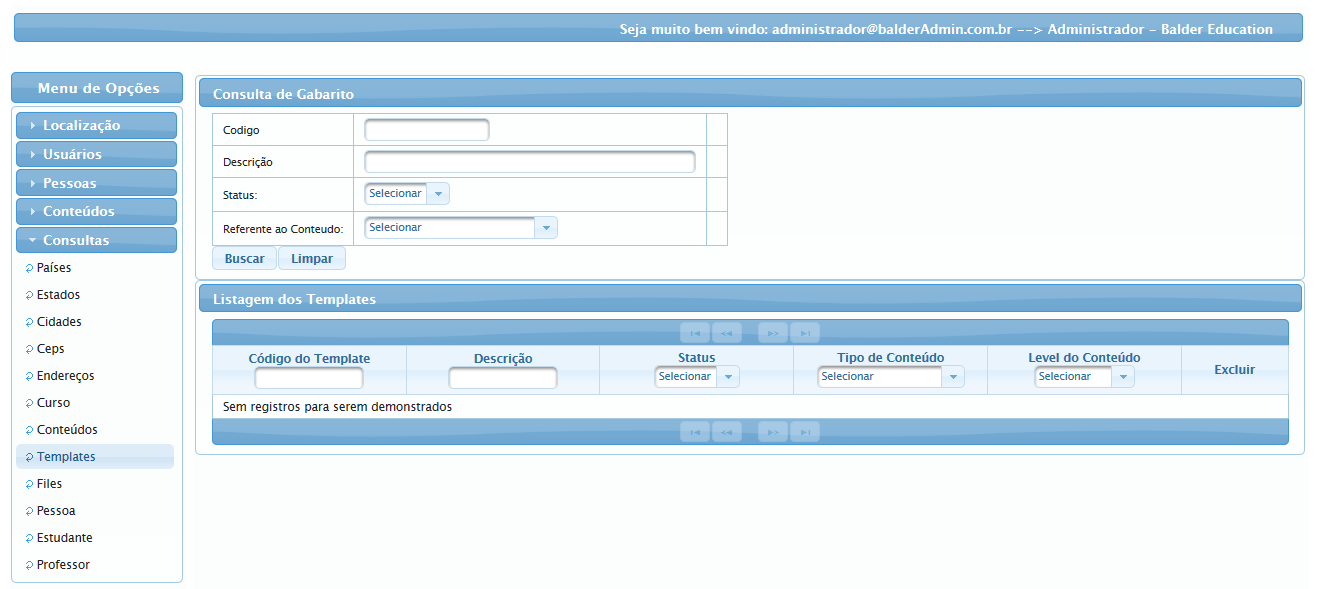
****

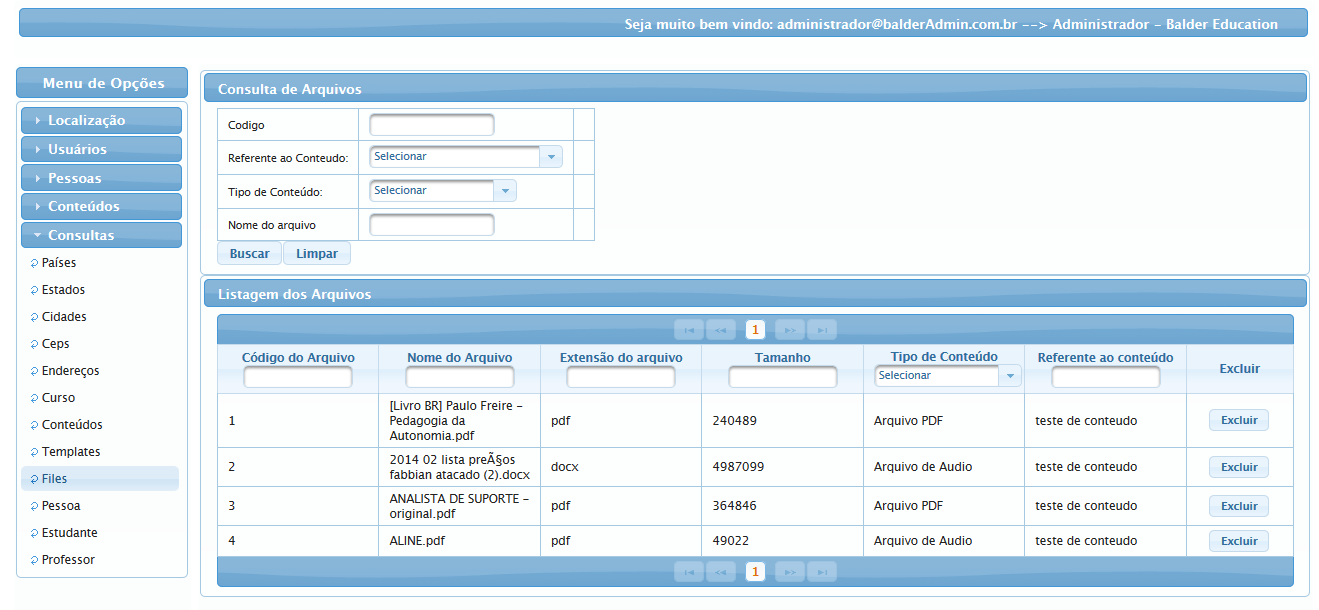
**Figura 25 – Consulta de Curso**

****

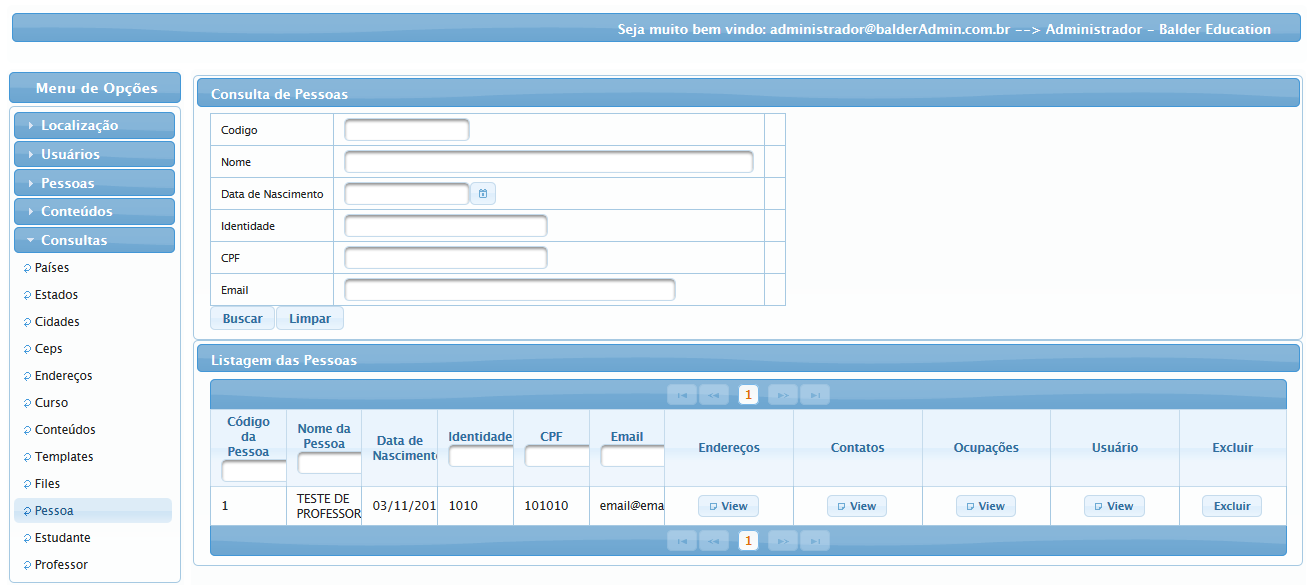
**Figura 26 – Consulta de Conteúdos**

****

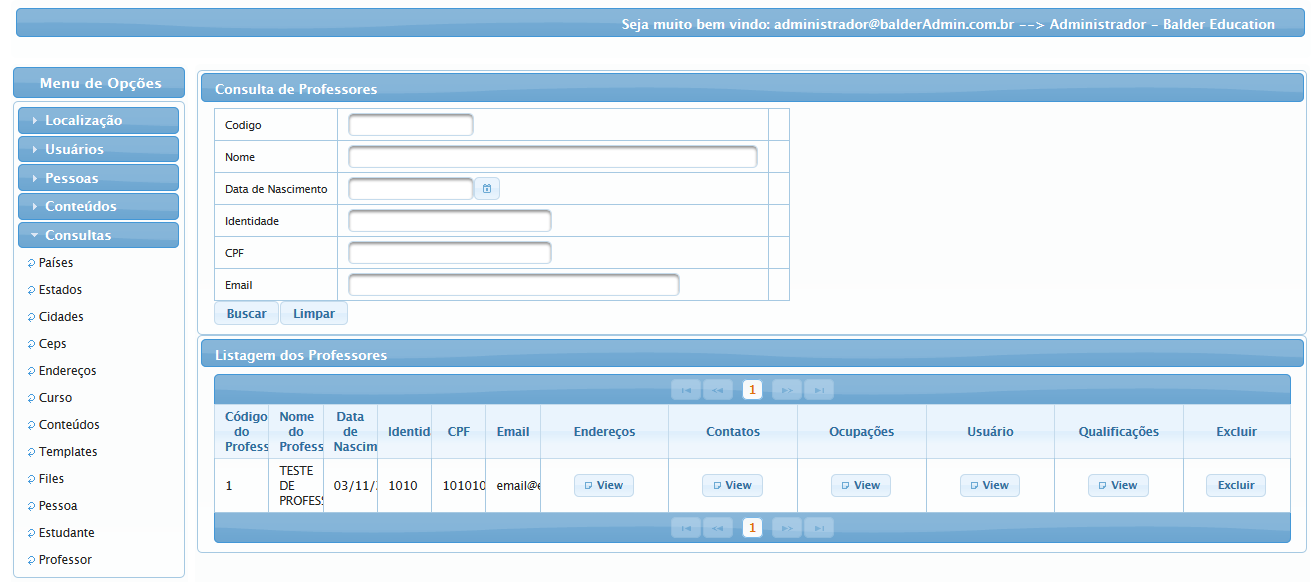
**Figura 27 – Consulta de Gabarito**

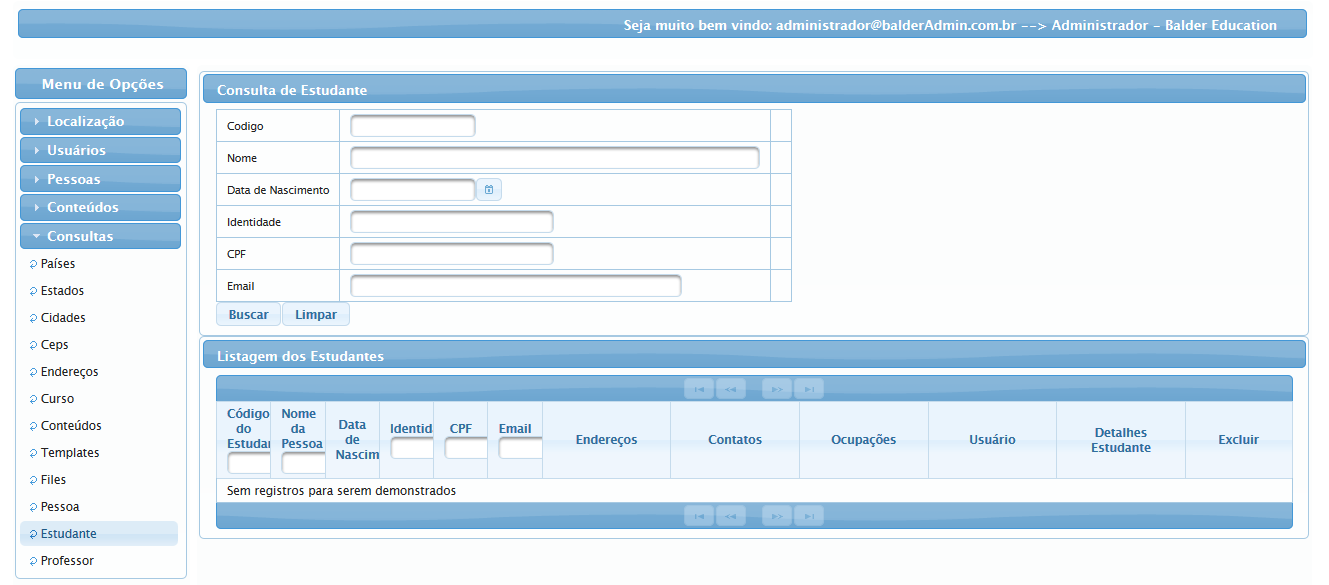
**Figura 28 – Consulta de Arquivos**

**Figura 29 – Consulta de Pessoas**

****

**Figura 30 – Consulta de Professores**

****

**Figura 31 – Consulta de Estudantes**

**REFERÊNCIA BIBLIOGRÁFICA**

RAMOS, Ricardo Ramos de. **Avaliação de manutenibilidade entre as abordagens de web services RESTful e SOAP-WSDL**, 2012. Disponível na internet. http://www.teses.usp.br/teses/disponiveis/55/55134/tde-24072012-164751/pt-br.php. 19 abril 2014.

LUCKOW, Décio Heinzelmann; MELO, Alexandre Altair de. **Programação Java para a Web.** 1. ed. São Paulo: Novatec Editora, 2010.

MEDNIEKS, Zigurd; DORNIN, Laird; MEIKE, G. Blake; NAKAMURA, Masumi. **Programando o Android.** 2. ed. São Paulo: Novatec Editora, 2012.

LECHETA, Ricardo. Google Android – Aprenda a criar aplicações para dispositivos móveis com o Android SDK. 2. ed. rev. e ampl. São Paulo: Novatec Editora, 2010.

NETBEANS. **Introdução ao JavaServer Faces 2.x**. Disponível na internet. https://netbeans.org/kb/docs/web/jsf20-intro\_pt\_BR.html. 17 abril 2014.

JERSEY. **RESTful Web Services in Java**. Disponível na internet. https://jersey.java.net/. 19 abril 2014.

ORACLE. **Comparing Oracle GlassFish Server and JBoss: Which Application Server Is Right for You?** Disponível na internet. http://www.oracle.com/us/products/middleware/application-server/oracle-glassfish-server/comparing-glassfish-jboss-wp-117118.pdf. 17 abril 2014.

GSON. **Gson User Guide**. Disponível na internet. https://sites.google.com/site/gson/gson-user-guide. 23 abril 2014.

PRIMEFACES. **PrimeFaces User´s Guide 4.0**. Disponível na internet. http://primefaces.googlecode.com/files/primefaces\_users\_guide\_4\_0\_edtn2.pdf. 22 de abril de 2014.

LEATHEM, Brian; FRYC, LUCAS; ROGERS, Sean. **Developer Guide -** **Develop applications using RichFaces 5.** Disponível na internet. http://primefaces.googlecode.com/files/primefaces\_users\_guide\_4\_0\_edtn2.pdf. 22 de abril de 2014.

ICEFACES. **ICEfaces Documentation.** Disponível na internet. http://www.icesoft.org/wiki/display/ICE/ICEfaces+Documentation. 22 de abril de 2014.

FRANCO, Rebeca Such Tobias. **Estudo comparativo entre frameworks Java para desenvolvimento de aplicações web: JSF 2.0, grails e spring web MVC**, 2012. Disponível na internet. http://repositorio.roca.utfpr.edu.br/jspui/handle/1/492. 23 abril 2014.

IBM DEVELOPER WORKS. **Standards and web services**. Disponível na internet. http://www.ibm.com/developerworks/webservices/standards/. 19 abril 2014.

CERAMI, Ethan. **Web Services Essentials**. 1.ed. O'Reilly Media, 2002.

MICRO-SERVIÇOS. [**http://martinfowler.com/articles/microservices.html**](http://martinfowler.com/articles/microservices.html)

**MATURIDADE** http://martinfowler.com/articles/richardsonMaturityModel.html