

CemHelp

Raphael Dutra Gonçalves

Ludmila Breder Furtado Campos

Curso: Tecnologia em Análise e Desenvolvimento de Sistemas Período: 6º

Área de Pesquisa: Ciências Exatas e da Terra

Resumo: O presente artigo tem como objetivo expor a importância de como é necessário a utilização de sistemas que integram e facilitam a vida dos usuários de certas empresas. Como método foi usado o embasamento em necessidades visto do que eu andei pesquisando adotando a metodologia de pesquisa descritiva e qualitativa, com instrumento a análise de requisitos, com a função de colher os dados necessários para o processo descrito, principalmente com informações e necessidades dos usuários que iriam usar. O artigo irá apresentar as ferramentas utilizadas no desenvolvimento do sistema com um breve estudo a respeito da Linguagem e de Modelagem e seus diagramas. O trabalho demonstra também o uso do *framework Ruby on Rails* apresentando algumas de suas definições e a sua utilização para fazer a conciliação com o banco de dados *postgreSQL*.

Palavras-chave: Empresas, *framework*, Usuários, *Ruby on Rails*

1. INTRODUÇÃO

É crescente o uso de sistemas *web* ou *softwares* para facilitar o armazenamento de informações, tendo em vista isso foi criado um sistema para se ter controle desses dados, um serviço de '*HelpDesk*', que vai servir como controle necessário das informações, melhorando a visualização desses dados, podendo apresentar até um atendimento mais ágil e sem duplicidade de notícias a respeito do seu atendimento. Dentre os vários aspectos que caracterizam uma boa prestação de serviço, um dos principais é o atendimento. Desta maneira, um sistema de atendimento a chamados de suporte deve apresentar versatilidade, tendo como procedimento de abertura de incidentes, o uso de ferramentas *web*. É o que faz a ferramenta *CemHelp*.

Esse tipo de *software(web)* tem a função de auxiliar a equipe de suporte ou quem for atender os clientes para coordenar e solucionar os incidentes que ocorrem com os clientes, assegurando que os chamados gerados por estes incidentes não sejam perdidos, esquecidos ou negligenciados. Constitui em um mecanismo computacional facilitador da informação.

O desenvolvimento de uma interface adequada entre o usuário e o computador é uma tarefa complexa, as características envolvidas em tal interação exigem que o usuário entenda bem a sua funcionalidade, às vezes tendo a necessidade de interligar a outros conhecimentos que não sejam só da computação, mas para isso a ideia é simplificar e deixar claro e prático para a funcionalidade não ser prejudicada e nem o desempenho no trabalho.



Basicamente, o programa *CemHelp* é um sistema de registro e gestão de ocorrências operacionais e gerenciais relacionadas a área financeira e de atendimento ao cliente.

O usuário irá estar encaminhando as informações de seus atendimentos para dentro do sistema, com objetivo de registrar seus clientes, atendimentos e ações que foram ocasionadas durante o tempo de atendimento.

O presente artigo está organizado da seguinte forma: no capítulo 1, apresenta uma introdução ao trabalho desenvolvido bem como os objetivos a serem alcançados; o capítulo 2 apresenta fundamentos teóricos (Referencial teórico), onde são apresentados conceitos relacionados, caracterizados como essenciais para uma melhor compreensão do conteúdo deste trabalho. Em seguida; o capítulo 3, apresenta os procedimentos metodológicos adotados na pesquisa. O capítulo 4 finalizando com as referências bibliográficas.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta algumas informações sobre como será esse sistema designado ao atendimento a clientes, sendo apresentados também o *software web* que irá auxiliar nesse objetivo.

2.1. Sistema de informação

Um Sistema de Informação pode ser definido tecnicamente como um conjunto de componentes que coletam (ou recuperam), processam, armazenam e distribuem informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma organização. (LAUDON, 2007).

O estudo dos sistemas de informação surgiu como uma subdisciplina das ciências da computação, com o objetivo de racionalizar a administração da tecnologia no seio das organizações. O campo de estudo foi-se desenvolvendo até vir mesmo a fazer parte dos estudos superiores dentro da administração. Existem, por exemplo, sistemas de informação operacional ou de processamento de transações (que gerem a informação referente às transações que têm lugar numa empresa), sistemas de informação de gestão (para solucionar problemas empresariais em geral), sistemas de informação estratégicos ou de apoio à decisão (analisam as distintas variáveis de negócio para a tomada de decisões), sistemas de informação executiva (para diretores, gerentes e administradores), sistemas de automatização de escritórios (aplicações que ajudam no trabalho administrativo) e sistemas especializados (que emulam o comportamento de um especialista numa área concreta).

Para nosso estudo o objetivo do nosso sistema irá ser utilizado como um *software web* de apoio, sendo um veículo para os funcionários que forem abrir seus atendimentos, tenha então um caminho para percorrer inserindo informações de forma prática e sendo um sistema de fácil utilização.



2.2. HelpDesk

Atualmente o gerenciamento do setor de TI das empresas vem tornando-se complexo, sendo um desafio para seus gestores. Os clientes hoje, particularmente com o surgimento da *internet*, esperam e exigem um serviço de qualidade e dentro do prazo. As empresas que estão olhando para o futuro, estão investindo em operações de serviços ao cliente, para atingirem uma vantagem competitiva (UNIPRESS, 2001, p. 4). Para melhorar o nível do serviço de atendimento e manter a qualidade e a eficiência do mesmo, torna-se obrigatório uso de um sistema de *Help Desk*. Este tipo de sistema está no centro das atividades de TI. É a primeira linha de defesa para resolver os problemas relacionados a seus clientes. Em muitas empresas ele vem sendo utilizado para resolver as questões comerciais, solicitações de clientes, gerenciar produtos e liberar serviços. Segundo COELHO(2003, p.2), sistemas de *Help Desk* compõem cada vez mais o crescente mercado de 15 tecnologias computacionais, assim como *Data Warehouse*, *Groupware*, *Data Mining* etc. Conforme Silva (2004, P.12) “uma ferramenta de *Help Desk*, sem dúvida proporciona um diferencial que além da documentação e sistematização oferece possibilidade de análise dos processos e fluxo de informações”. Conforme UNIPRESS (2001, p.4) os desafios de um sistema de *Help Desk* corporativo atualizado devem incluir: a) o volume de chamados, que vem aumentando devido a complexidade e expansão dos negócios; b) os métodos de comunicação: telefone, *e-mail* e computação móvel; c) as iniciativas da empresa para fazer o *Help Desk* uma parte integral das estratégias de seus investimentos. A demanda e o volume de trabalho das equipes de suporte estão aumentando, fazendo com que ela se torne um componente crítico. “Para ser capaz de fornecer os serviços de TI de acordo com a funcionalidade, níveis de serviço e custo acordados, a forma de trabalho orientada a processos é a melhor forma de estruturação da área de TI” (MAGALHÃES; PINHEIRO, 2007, p.23).

2.3. FERRAMENTAS

Este capítulo apresenta algumas informações sobre as ferramentas utilizadas no processo de desenvolvimento do sistema de *software*.

2.3.1. Linguagem do desenvolvimento: *Ruby on Rails*

O artigo porque utilizar o *Ruby on Rails* em seu projeto do site Codificar explica que o ***Ruby on Rails*** ou simplesmente *Rails* que é um *framework* (é uma coleção de bibliotecas e ferramentas capazes de facilitar o desenvolvimento) *web* de código aberto (*open source*) escrito na linguagem de programação *Ruby* sob a licença MIT.



A linguagem promete aumentar velocidade e facilidade no desenvolvimento de sites orientados a banco de dados, uma vez que é possível criar aplicações com base em estruturas pré-definidas. As aplicações criadas utilizando o *framework Rails* são desenvolvidas com base no padrão de arquitetura *MVC (Model-View-Controller)*.

Ruby on Rails incentiva e facilita o uso de padrões *web*, tais como *JSON* ou *XML* para transferência de dados e *HTML*, *CSS* e *JavaScript* para exibição e interface do usuário.

2.3.2. Banco de Dados

Nesse sistema o banco de dados será feito pelo *software (Postgresql)*, o *PostgreSQL*. O artigo *PostgreSQL Tutorial* do site *DevMedia* explica que normalmente chamado de *Postgres*, é um sistema de gerenciamento de banco de dados do tipo objeto-relacional (*ORDBMS*) com ênfase em extensibilidade e em padrões de conformidade. Como um servidor de banco de dados, sua principal função é armazenar dados de forma segura, apoiando as melhores práticas, permitindo a recuperação dos dados a pedido de outras aplicações de *software*. Ele pode lidar com cargas de trabalho que vão desde pequenas aplicações *single-machine* a aplicações de grande porte voltadas para a *Internet*, onde será utilizada de forma simultânea por vários usuários. Ele consiste em um processo de servidor que lê e grava os arquivos de banco de dados reais, e um conjunto de programas cliente que se comunicam com o servidor. O mais comumente utilizado é o comando *psql*, que permite ao usuário executar consultas SQL e visualizar os seus resultados. Nenhum dos clientes acessa os arquivos do banco de dados diretamente, o que é deixado inteiramente para o servidor. Todos os arquivos de banco de dados são armazenados em um diretório como, por exemplo, */var/lib/pgsql* ou */usr/local/pgsql*.

3. METODOLOGIA

O desenvolvimento do sistema foi através de pesquisa bibliográfica e estudo de caso realizada em uma empresa escolar, onde havia essa necessidade pendente.

Foi usado a pesquisa descritiva que realiza o estudo, a análise, o registro e a interpretação dos fatos do mundo físico sem a interferência do pesquisador. São exemplos de pesquisa descritiva as pesquisas mercadológicas e de opinião (BARROS e LEHFELD, 2007).

3.1. Etapas de desenvolvimento do *software*

Trata-se da produção de um *Software* de controle dos atendimentos que ocorrem na empresa, onde serão executadas 4 etapas, que são: Análise de Requisitos e Modelagem dos Dados; Desenvolvimento do *Software*; Implementação do *Software* e Testes. Todas essas etapas são descritas a seguir.



3.1.1. Análise de Requisitos e Modelagem dos Dados

Neste primeiro passo foram identificadas as necessidades de um *software* para *web* que realizasse o controle dos atendimentos aos clientes da empresa, onde ele vai registrar os clientes que entram para se cadastrar na empresa e então registrar as informações dentro dos atendimentos que são criados, após isso realizado o fluxo do funcionamento e sequência das telas e por último desenvolvido um protótipo do sistema, utilizando o próprio *Ruby On Rails*. Para desenvolver um sistema em *web* que irá fazer esse controle dos atendimentos, como está sendo a comunicação com os usuários decorrentes desses atendimentos, e quem está praticando cada ação e abrindo seus atendimentos.

Além desse registro de cursos, alunos para cadastrar os atendimentos, na parte principal iremos mostrar os atendimentos que estão pendentes e atrasados, além de todas essas comunicações serem registradas pelos usuários que estão praticando a ação e criando esses atendimentos, temos também a autorização de que só o administrador poderá registrar um usuário e editá-lo, dando assim autonomia para a coordenadora, evitando que os funcionários não modifiquem ou atrapalhem o serviço de alguma outra pessoa, e caso o usuário comum se sinta mais confortável em ter sua própria senha ou achar que precisa modificá-la também terá essa opção. A modelagem dos dados foi realizada a partir da *UML (UNIFIED MODELING LANGUAGE)*, O artigo Introdução à construção de Diagrama de Classes Um Estudo de Caso do site DevMedia explica que a *UML* tem origem na compilação das “melhores práticas de engenharia” que provaram ter sucesso na modelagem de sistemas grandes e complexos. Sucedeu aos conceitos de *Booch*, *OMT (Rumbaugh)* e *OOSE (Jacobson)* fundindo-os numa única linguagem de modelagem comum e largamente utilizada. A *UML* pretende ser a linguagem de modelagem padrão para modelar sistemas concorrentes e distribuídos. (De acordo com o Artigo no site DevMedia). A *UML* é uma linguagem visual para modelar sistemas orientados a objetos. Isso quer dizer que a *UML* é a linguagem constituída de elementos gráficos (visuais) utilizados na modelagem que permitem representar os conceitos do paradigma da orientação a objetos.

3.1.2. Desenvolvimento do *Software*

Para o desenvolvimento do *software* foram levantados alguns requisitos: Missão: criar um sistema para os funcionários com a intenção de auxiliar em seus atendimentos. Problema: Como problema tem-se a necessidade de ter um sistema para atendimentos, gerando um histórico para a organização das informações. Partindo deste ponto foi realizado o levantamento e a solução foi então o uso de uma ferramenta que auxiliou no desenvolvimento *web*, sendo criado um *software* de fácil utilização e com *design* intuitivo, que além de incluir os atendimentos que são



feitos dentro da empresa, o sistema também tem a capacidade de registrar dentro desses atendimentos, as ações que são desenvolvidas, gerando um histórico de como foi o atendimento, e caso o cliente for procurar, saberá o real motivo de como foi a resolução do mesmo. Feito o levantamento dos requisitos necessários para funcionamento do software, fluxo de dados e sequência das telas do sistema.

3.1.3. Implementação do Software

Foi utilizado primeiramente o *Ubuntu 18.04*, que era um sistema operacional necessário para o uso da ferramenta utilizada que é o *Ruby on Rails*, sendo um *framework* de fácil utilização, e como editor de texto no desenvolvimento foi usado o *Atom*, com a base de dados feita pelo *Postgresql* e de *Gems* que funcionam como bibliotecas dentro do *Ruby* e que também auxiliou na criação dos relatórios que identificam todos os alunos registrados no banco de dados e um gráfico, apresentando uma quantidade de usuários que criaram os chamados com *status* “atendimento”.

3.1.4. Testes

A fase de testes é dividida em 4 etapas: Unidade, Integração, Sistema e Aceitação.

Na etapa de Unidade o objetivo será detectar erros de lógica ou de implementação em pequenas partes do sistema.

Na etapa de Integração o objetivo será a detecção de falhas provenientes da integração interna dos componentes do sistema

Na etapa de Sistema o *software* já vai estar integrado completamente, no entanto os testes visam identificar falhas em relação aos requisitos do sistema, no que se diz respeito à comunicação com outros sistemas.

Na etapa de Aceitação terá como objetivo verificar se o *software* está pronto e pode ser usado pelo usuário final. Para isso é verificado se o sistema realiza as funções para o qual ele foi criado, satisfazendo as necessidades do Cliente.

3.1.4.1. Detalhamento do Sistema

O sistema *CemHelp* é criado para uma aplicação onde iremos ter a possibilidade de solucionar um problema decorrente da falta de auxílios diante das informações que são necessárias para ser guardadas durante um atendimento ao cliente, e suas ações.

Na tela de *Login* são validados usuário e senha, em seguida o sistema direciona para a um outro menu onde o usuário irá ver uma tela principal, com alguns campos, primeiro vem o “Atendimento”, “Alunos”, “Curso”, “Alterar Senha”, “Usuário”, e a opção de “Sair”, Iremos já ter algumas informações já predefinidas para o banco de dados na opção de cursos, e então o usuário precisará registrar os



alunos, e após isso já pode iniciar os atendimentos, abrindo eles e completando as ações decorrentes desses chamados com os clientes.

3.1.4.2. Descrição dos Componentes

3.1.4.2.1. Tela de Login

O sistema inicia-se pela tela de *login* com o pedido de usuário e senha.

3.1.4.2.2. NavBar

Após o *login* a tela se redireciona para um menu principal, que possibilita a escolha das atividades disponíveis no sistema.

3.1.4.2.3. Cadastro Usuário

Você irá poder registrar informações criando um usuário novo, e podendo editar.

3.1.4.2.4. Registrar informações

Registra informações que serão extremamente úteis para a funcionalidade no sistema, como a inserção de dados para Aluno e Curso e após isso você pode iniciar a encaixar dados dentro de seus atendimentos.

3.1.4.2.5. Inserir Alunos

Nessa opção é para adicionar os alunos e seus respectivos dados necessários para seu atendimento.

3.1.4.2.6. Inserir Cursos

Nessa opção será registrado os cursos de acordo com a necessidade de cada unidade da empresa.

3.1.4.2.7. Alterar Senha

Nessa opção o usuário poderá alterar sua senha como uma validação da sua senha atual.

3.1.4.2.8. Cadastrar Atendimento

Essa opção é para iniciar o registro dos chamados de necessidades dos clientes da empresa e inserir ações dentro desses atendimentos criados.



3.1.4.2.9. Exportar para *pdf* dados de Atendimentos

Essa opção você irá conseguir visualizar em gráficos os atendimentos que ainda estão em abertos.

3.1.4.2.10. Exportar para *pdf* dados de alunos cadastrados

Essa opção você irá conseguir visualizar um relatório dos clientes que foram registrados.

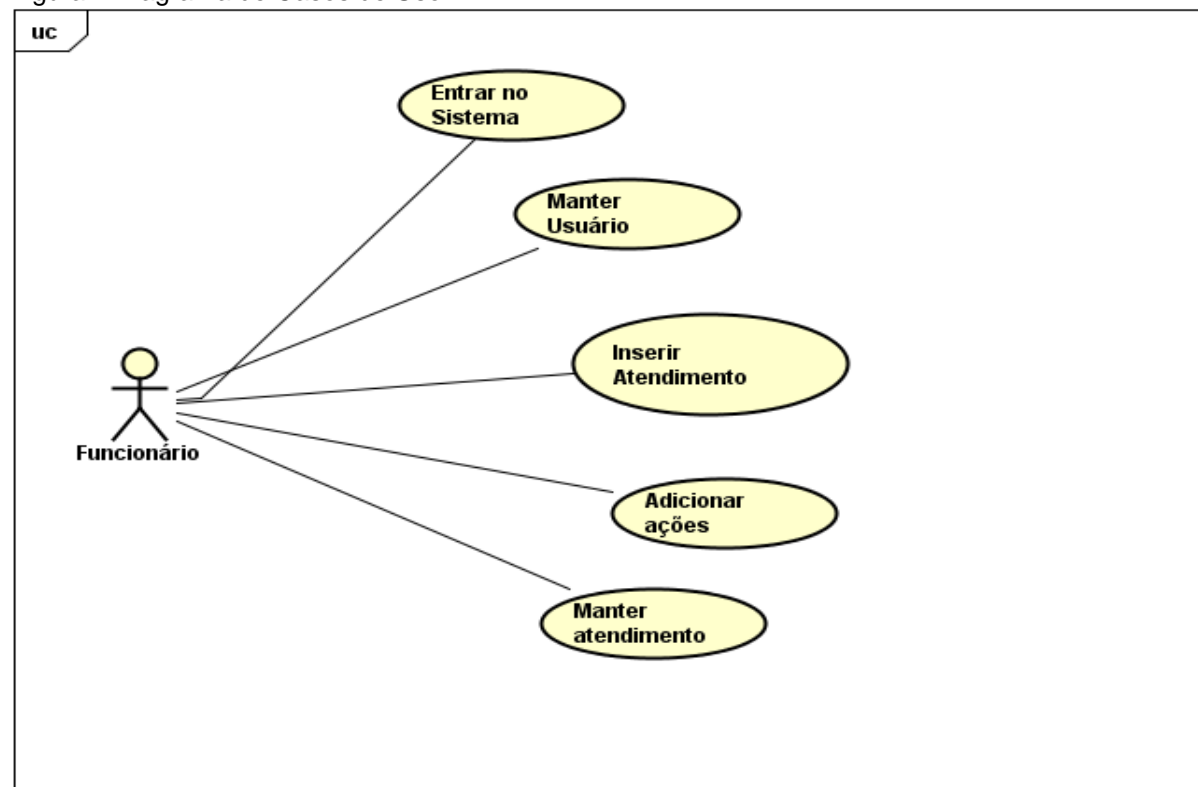
3.2. Diagramas

3.2.1. Diagramas de Caso de Uso

O diagrama de caso de uso descreve a proposta para o sistema, permitindo o levantamento de requisitos funcionais do sistema, mostrando as funcionalidades do sistema do ponto de vista do usuário.

A figura 1 (um) mostra o diagrama de casos de uso e a baixo uma representação do mesmo.

Figura 1:Diagrama de Casos de Uso



Fonte: Dados do sistema

Especificação funcional do caso de uso



Nome: Inserir Atendimento

Objetivo: Cadastrar Atendimento e poder inserir informações relevante do mesmo

Ator: Funcionário da Empresa

Cenário Principal

1. O sistema exibe uma tela principal onde a principal funcionalidade é a criação de seus atendimentos, que precisam se alguns requisitos para serem preenchidos:

- 1.1. Aluno
- 1.2. Curso
- 1.3. Problema
- 1.4. Status

Cenário Alternativo 1

1. O sistema exibe as telas onde também precisamos adicionar os alunos, com as seguintes informações:

- 1.1. Digite seu nome
- 1.2. Digite seu email
- 1.3. Digitar o telefone(1)
- 1.4. Digitar o telefone(2)
- 1.5. Sistema salva alterações
2. Mostra informações geradas
3. Editar informações criadas

Cenário Alternativo 2

1. O sistema dá a opção de pesquisa dos atendimentos:
2. O sistema retorna a busca

Cenário Alternativo 3

1. O sistema exibe campo de mostrar ações
2. O Usuário irá adicionar ações

Cenário Alternativo 4

1. O sistema exibe um formulário de atendimentos pendentes, onde contém os atendimentos com status: "Atendimento", e os campos a serem vistos são:

- 1.1. Status
- 1.2. Usuário
- 1.3. Data/ Hora
- 1.4. Aluno
- 1.5. Curso
- 1.6. Problema

No apêndice A, seguem os diagramas de Classes, de Objetos, de Sequência, de Máquina de Estado, de Componentes e de Atividades.

3.3. Telas

As telas foram registradas no Apêndice B, a partir da página 16 já encontramos.



3.4. Requisitos Mínimos para o Sistema

O sistema *CemHelp* é uma aplicação construída com o sistema operacional *Ubuntu 18.04* sendo necessário para a utilização da ferramenta *Ruby on Rails*, tendo em vista que só utilizando *Ubuntu 18.4* para fazer a conexão local do sistema e o banco de dados utilizado foi o *postgres*.

Requisitos de Hardware

O *CemHelp* necessita de alguns requisitos mínimos para que o usuário o tenha em pleno funcionamento, *internet* necessária para carregar um navegador da *internet*, fora isso não há necessidade específica de um hardware para ser usado

4. CONCLUSÃO

Pela observação dos aspectos analisados hoje em dia as empresas estão buscando uma facilidade para auxiliar seus funcionários, trazendo sistemas com visuais mais modernos e também sendo mais claros para funcionários que podem ter dificuldades. Nesse caso o sistema a ser criado será uma chave para lidar com isso, será criado algo com um visual robusto, fácil de entendimento e com possíveis retoques caso necessário pelo contratante, que caminhando para as necessidades da empresa em específico.

No trabalho descrito foi avaliado alguns pontos, primeiro como o problema, por que a necessidade de um *software* para a empresa? Nessa situação foi então feito uma avaliação com as pessoas da empresa, de certa forma uma pesquisa para um melhor entendimento, além das exigências do coordenador geral, e os problemas eram vários, utilização de *softwares* que não se adequavam tanto as necessidades dos usuários, além de ter muita funcionalidade desnecessária, complicando a vida de quem usava. A solução para o problema foi criar algo que demandasse exclusivamente dos funcionários e coordenação, customizado para adequar as necessidades que aquelas pessoas ali exigiam, e então foi feito um *software web* que faz controle dos atendimentos que serão registrados, e não só isso, com níveis de autorizações, para que o usuário não modifique a área do registro de novos funcionários, o controle de todos os clientes que vão precisar ser atendidos pela empresa, os cursos, e as ações que são construídas diante de cada atendimento que é registrado, o funcionário que será o usuário mais frequente dentro do sistema precisará para iniciar o registro dos cursos e alunos para iniciar seu atendimento, é algo simples e prático, além de podermos pesquisar alguns atendimentos que podem estar pendentes, finalizados, buscar por alunos, na página principal foi criado também alguns formulários com certas condições para os usuários avaliarem se o atendimento está sendo executado no período correto, e também dois relatórios para acompanhar a quantidade de atendimentos abertos por



usuário e um formulário em *pdf* que mostra o registro dos alunos que foram cadastrados.

Para o desenvolvimento do sistema foi necessário o uso do *Linux(Ubuntu 18.04)* para o uso da ferramenta de desenvolvimento *Ruby On Rails* que é um *framework* onde foi realizado o desenvolvimento, implantação e manutenção de uma aplicação *web*, e é utilizado da linguagem orientada à objeto, e a plataforma que foi utilizado para criação da base de dados é o *postgresql*.

Pode ser encontrada a dificuldade pelo fato dele haver a necessidade de estar acessado a uma *internet*, e não há como logar local dentro da empresa, só por meio do *link* na *web*.

Como vantagem será um *software* de clara e fácil utilização, com linguagem em português, além das medidas já serem de conversão para nossos padrões. O *software* foi baseado em uma demanda que existe, vinculando a uma empresa tornando o personalizado para as suas necessidades.

Para o futuro do sistema, ainda será dividido em unidades, já que são 7(sete), e cada uma terá acesso apenas as informações definidas pela empresa em questão, além das necessidades que vão surgir de vários relatórios complexos para avaliar bem como está sendo os atendimentos e ajudar a relatar certas coisas, avaliando melhor o atendimento dos funcionários.

5. REFERÊNCIAS

Portalgsti – **Sobre Ruby on Rails.** Disponível em: <<https://www.portalgsti.com.br/ruby-on-rails/sobre/>> Acesso em 03/12/2018.

Nathália – **Ruby on Rails Tutorial.** Disponível em: <<https://www.devmedia.com.br/ruby-on-rails-tutorial/31285>> Acesso em 03/12/2018

Santos, Angelica. **Domingues Desenvolvimento de um sistema HelpDesk.** Disponível em: <<http://tcconline.utp.br/media/tcc/2014/08/DESENVOLVIMENTO-DE-UM-SISTEMA-DE-HELP-DESK.pdf>> Acesso em 03/12/2018

Feltrin, Wesley Eli. **Reestruturação de atendimento de HelpDesk.** Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/897/1/CT_COTEL_2011_2_01.pdf> Acesso em 02/12/2018

Edson escrito em: 2015. **Tutorial Postgresql.** Disponível em: <<https://www.devmedia.com.br/postgresql-tutorial/33025>> Acesso em 02/12/2018

Paulo, Cristian, escrito em: 2007. **Sistema de Help Desk e Controle de Chamados baseado em WROKFLOW.** Disponível em:



<<http://dsc.inf.furb.br/arquivos/tccs/monografias/2007-1cristianpauloprigo1vf.pdf>>
Acesso em 02/12/2018

DevMedia - **Introdução à Construção de Diagrama de Classes um Estudo de Caso.** Disponível em: <<https://www.devmedia.com.br/artigo-sql-magazine-57-introducao-a-construcao-de-diagrama-de-classes-um-estudo-de-caso/1086>> Acesso em 25/01/2019

Codificar - **Porque utilizar o *Ruby on Rails* em seu projeto.** Disponível em: <<https://codificar.com.br/blog/porque-utilizar-o-ruby-on-rails-em-seu-projeto/>>
Acesso em 25/01/2019

GUEDES, Gilleanes T. A. **UML 2 – Guia Prático.** 2. Ed. Novatec Editora, 2014

DevMedia - **Testes de Software - Parte 2.** Disponível em: <<https://www.devmedia.com.br/testes-de-software-parte-02/9419>> Acesso em 25/01/2019

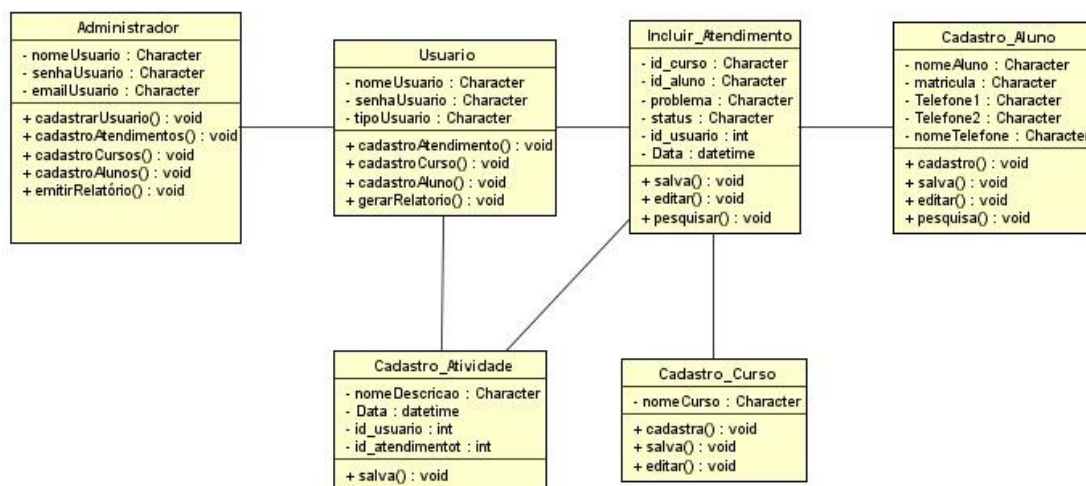


APÊNDICE A

No apêndice A é apresentado os diagramas da UML.

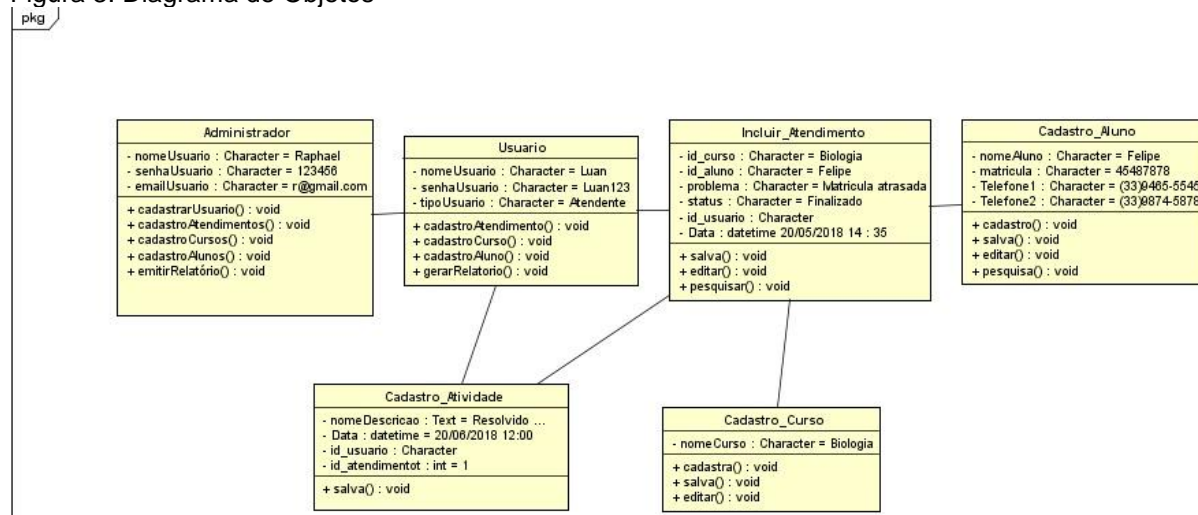
A figura 2 (dois) mostra o diagrama da classes que é uma representação do banco de dados do sistema.

Figura 2:Diagrama de Classes



A figura 3(três) apresenta o diagrama de objetos que tem o objetivo de demonstrar os dados que tabela irá receber.

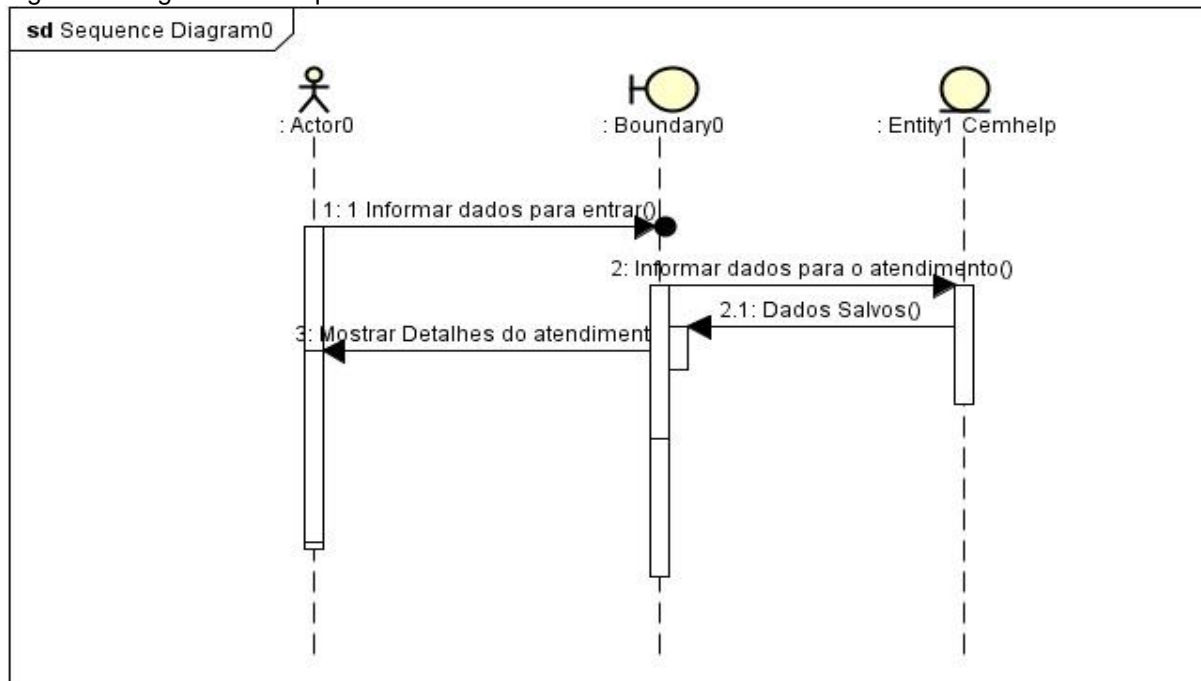
Figura 3: Diagrama de Objetos





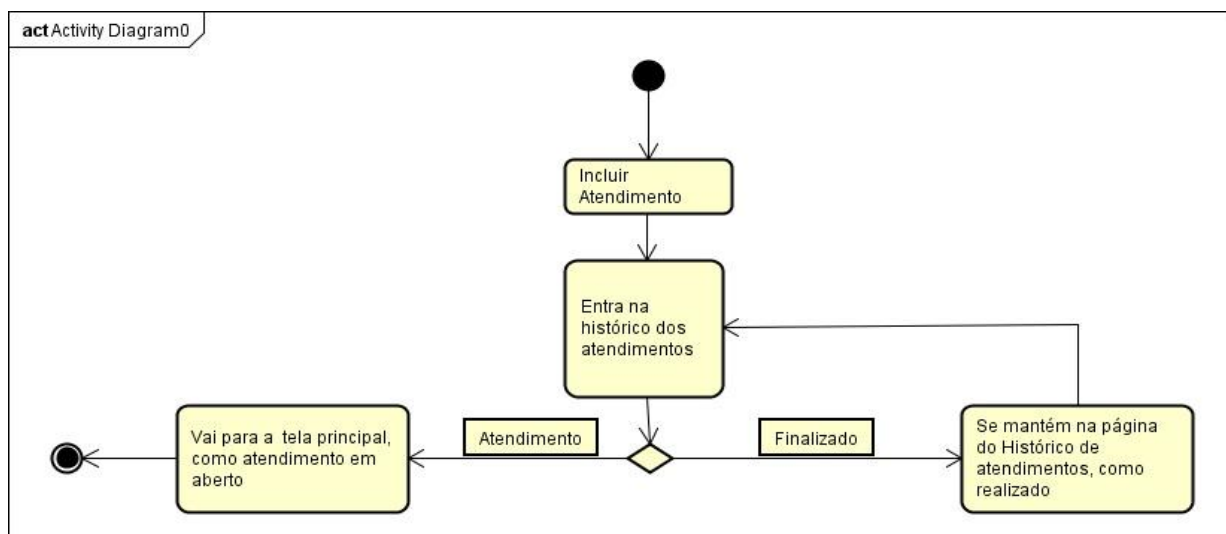
A figura 4(quatro) é o Diagrama de Sequência apresenta a parte do sistema de cadastro de requisições.

Figura 4: Diagrama de sequência



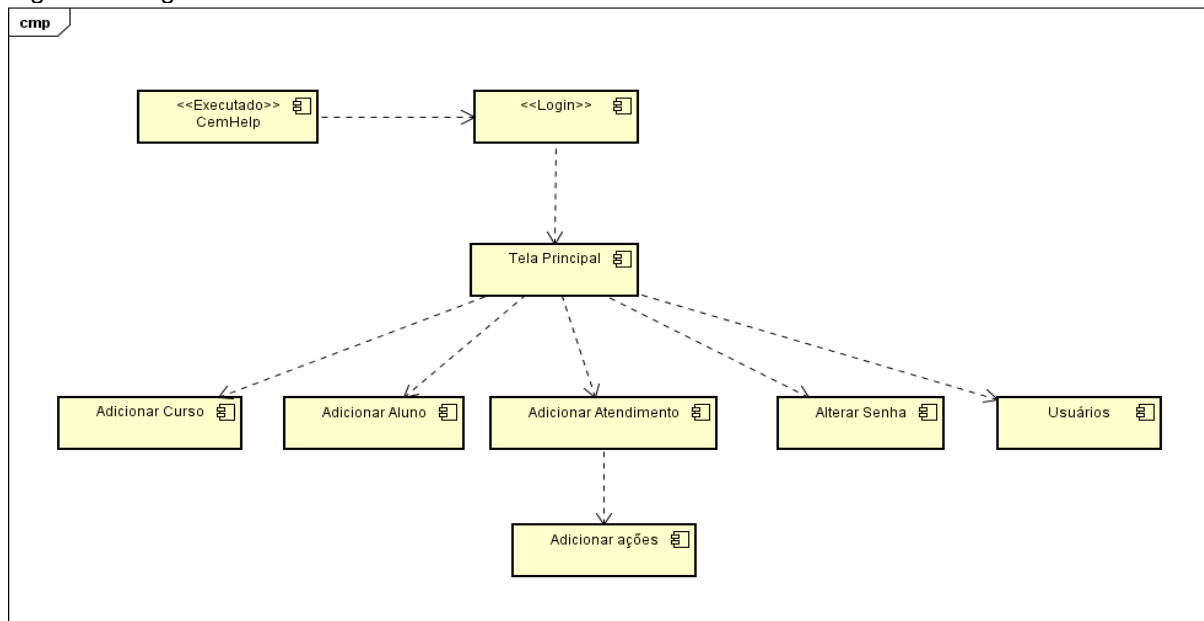
A figura 5(cinco) demonstra como funciona a inserção das ações e que é representado pelo Diagrama de Máquina de Estado.

Figura 5: Diagrama de Máquina de Estado



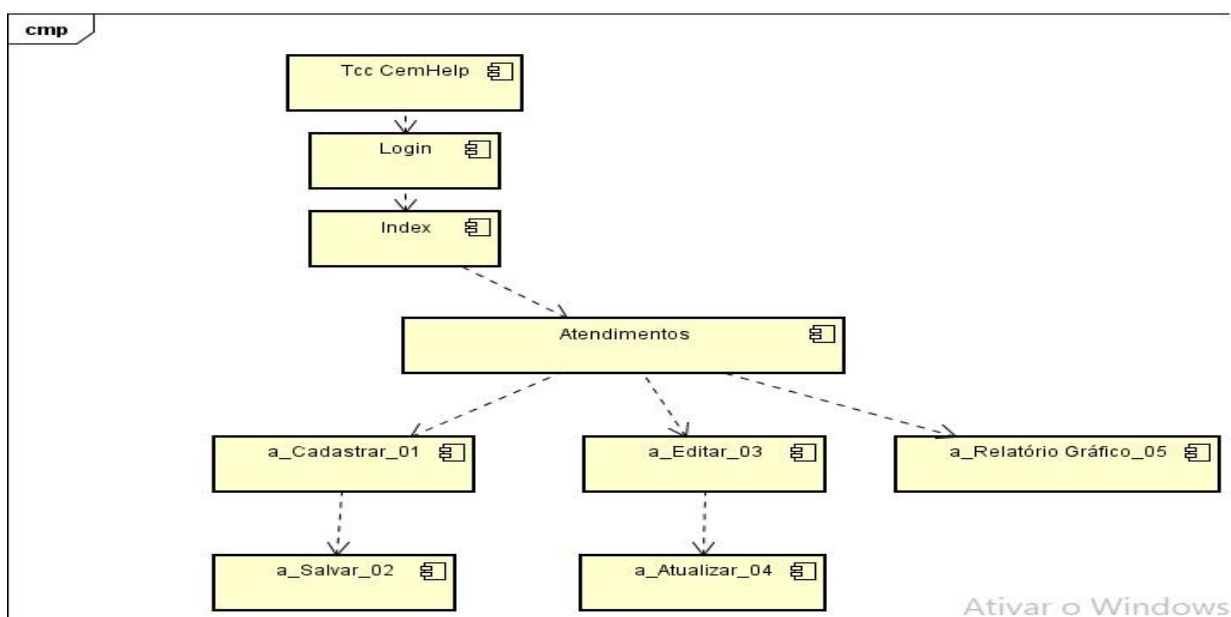
A figura 6(seis) apresenta uma das atividades do sistema, que incluir atendimento com o que for necessário para mostrar a inserção das ações.

Figura 6: Diagrama de Atividades



A figura 7(sete) apresenta as telas que foram criadas no sistema para que ele funcionasse, sendo apresentado pelo Diagrama de Componentes.

Figura 7: Diagrama de componentes





APÊNDICE B

O apêndice B trata-se das telas do sistema.

A figura 8 (oito) apresenta a tela inicial do sistema, solicitando *login* e senha.

A Figura 9 (nove) apresenta o menu do sistema, que possibilita encaminhar para as demais atividades do sistema.

Status	Aluno	Problema	Curso	Usuário	Data/hora
Atendimento	Cléber Ribeiro Nascimento	wddw	Arquitetura e Urbanismo	Vanda Lopes	04/12/2018 23:32
Atendimento	Ana Clara Aguiar Pereira Guimarães	dssds	Agronomia BCH	Raphael	05/12/2018 14:09
Atendimento	Ana Clara Aguiar Pereira Guimarães	test4	Arquitetura e Urbanismo	Raphael	04/12/2018 16:19
Atendimento	Cléber Ribeiro Nascimento	teste3	Arquitetura e Urbanismo	Raphael	04/12/2018 15:48
Atendimento	Ewerton Afonso de Oliveira	teste2	Ciências Cont. BCH	Raphael	04/12/2018 15:47



A Figura 10 (dez) apresenta a área de atendimentos do sistema, que possibilita adicionar um atendimento e gerar gráficos.

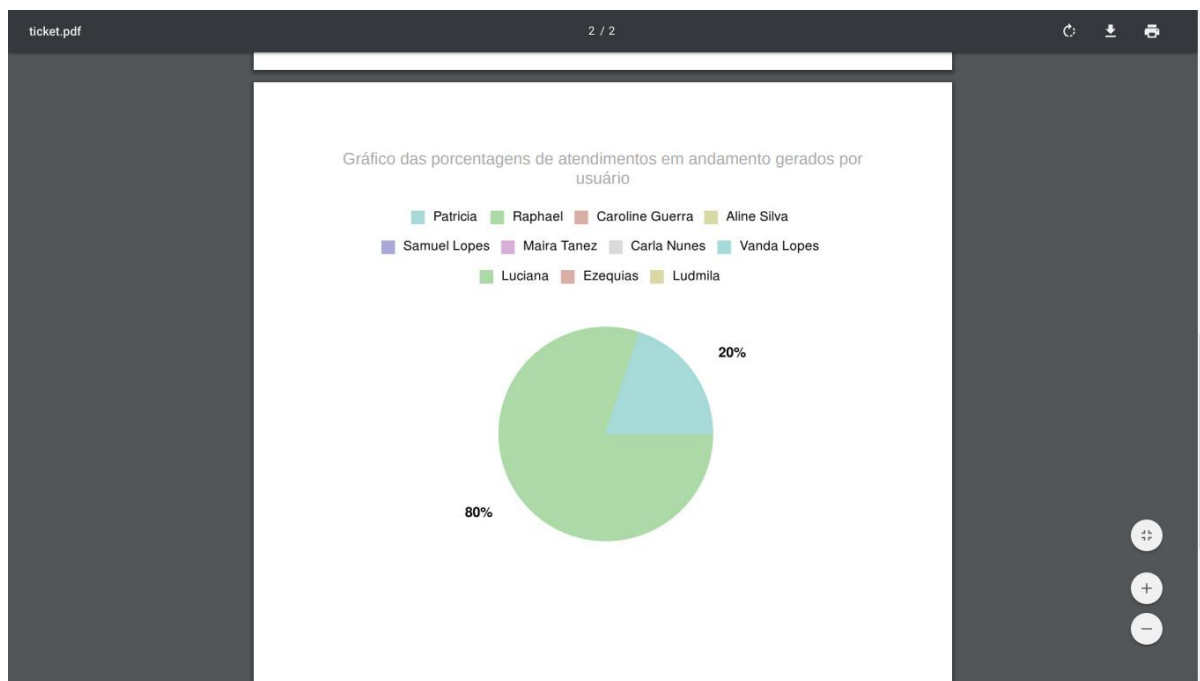
Status	Aluno	Problema	Curso	Usuário	Data/Hora		
Atendimento	Clêber Ribeiro Nascimento	wddw	Arquitetura e Urbanismo	Vanda Lopes	04/12/2018 23:32	Mostrar Ações	Editar status
Finalizado	Ana Clara Aguiar Pereira Guimarães	test4	Artes Visuais LICENC	Vanda Lopes	04/12/2018 23:32	Mostrar Ações	Editar status
Atendimento	Ana Clara Aguiar Pereira Guimarães	dssds	Agronomia BCH	Raphael	04/12/2018 18:23	Mostrar Ações	Editar status
Atendimento	Ana Clara Aguiar Pereira Guimarães	test4	Arquitetura e Urbanismo	Raphael	04/12/2018 16:19	Mostrar Ações	Editar status

A Figura 11 (onze) apresenta um gráfico de atendimentos do sistema, que é gerado na parte de atendimentos.





A Figura 12 (doze) apresenta um gráfico de atendimentos do sistema, que é gerado na parte de atendimentos.



A Figura 13 (treze) apresenta uma área para adicionar atendimento.

Novo Atendimento

* Aluno(a)
 Seleccione um Aluno

* Curso
 Seleccione um Curso

* Problema
 Escreva o problema

* Status
 Atendimento

Cem - Centro Educacional de Manhauçu
 Aguardando cemhelp.herokuapp.com



A Figura 14 (quatorze) apresenta uma área para editar atendimento.

Editar

Status

Atendimento

ATUALIZAR VOLTAR

Cem - Centro Educacional de Manhuaçu

A Figura 15 (quinze) apresenta uma área que mostra o atendimento criado.

Status: Atendimento

Aluno(a): Cléber Ribeiro Nascimento

Problema: wddw

Cursos: Arquitetura e Urbanismo

Criou Atendimento: Vanda Lopes

Ações:

Descrição: dws

Data/Hora: 04/12/2018 23:32

Criou ação: Vanda Lopes

CRIAR

Editar Status | Voltar

Cem - Centro Educacional de Manhuaçu

Aguardando cemhelp.herokuapp.com...



A figura 16 (dezesseis) apresenta a área do Alunos.

Buscar por Nome

Alunos

NOVO ALUNO

Nome	Email	Matricula	Telefone(1)	Telefone(2)	
Leiliane Mônica de Oliveira Santos	LEILIANEMONICA@YAHOO.COM.BR	20182700	(33)8851-7510	(34)3434-4343	Editar
Elaine Rosa Santana	nanasantanrosa@gmail.com	11464865	(31)8650-1707		Editar
Jamile Pires Rodrigues	jamile-jpr@hotmail.com	7425022	(33)8402-6975		Editar
Ewerton Afonso de Oliveira	ewertoneoliveira@hotmail.com	5712641	(28)9997-9979		Editar
Ana Clara Aguiar Pereira Guimarães	UNOPAREAD@HOTMAIL.COM	7348019	(33)8890-2397		Editar
Adriana Aparecida Ferreira de Oliveira	adrianaprado0701@hotmail.com	34434343	(33)8876-3325		Editar
Gilvane Batista Hott	gilvanehott@yahoo.com.br	13638724	(33)8816-1331		Editar
Leandro Martins de Oliveira	leandro2011martinsoliveira@hotmail.com	9461155	(31)9832-3963		Editar

A figura 17 (dezesete) apresenta a área de Adicionar Alunos.

Nome

Escreva o nome

Email

Escreva o E-mail do Cliente

Matricula

Digite a Matricula

Telefone 1

(00)0000-0000

Telefone 2

(00)0000-0000

[CRIAR](#) [VOLTAR](#)

Cem - Centro Educacional de Manhauçu



A figura 18 (dezoito) apresenta a área de Editar Alunos.

Editar Aluno

* Nome
Leiliane Mônica de Oliveira Santos

Email
LEILIANEMONICA@YAHOO.COM.BR

Matricula
20182700

Telefone 1
(33)8851-7510

Telefone 2
(34)3434-4343

[CRIAR](#) [VOLTAR](#)

Cem - Centro Educacional de Manhuaçu

A figura 19 (dezenove) apresenta a área de Cursos.

Cursos

[NOVO CURSO](#)

Nome	
Administração BCH	Editar
Agronomia BCH	Editar
Artes Visuais LICENC	Editar
Arquitetura e Urbanismo	Editar
Biologia LICENC	Editar
Design de interiores	Editar
Ciências Cont. BCH	Editar
Economia BCH	Editar
Educação física LICENC	Editar
Educação física BACH	Editar



A figura 20 (vinte) apresenta a criação de um novo curso.

A figura 20 (vinte) apresenta a a edição de um curso.



A figura 21 (vinte e um) apresenta a opção de alterar senha

Trocar de senha

Senha atual

Nós precisamos da sua senha atual para confirmar a mudança.

Senha

Confirme sua senha

ALTERAR VOLTAR

Cem - Centro Educacional de Manhauçu

A figura 22 (vinte e dois) apresenta a opção de usuários.

Usuários

NOVO USUÁRIO

Email	Nome	Empresa	Nivel	
patricia@admin.com	Patricia	Unopar	manager	Editar
manager@admin.com	Raphael	Unopar	manager	Editar
caroline.guerra@escolacem.com.br	Caroline Guerra	Unopar	salesman	Editar
aline.silva@escolacem.com.br	Aline Silva	Unopar	salesman	Editar
samuel.lopes@escolacem.com.br	Samuel Lopes	Unopar	salesman	Editar
maira.tanez@escolacem.com.br	Maira Tanez	Unopar	salesman	Editar
carla.nunes@escolacem.com.br	Carla Nunes	Unopar	salesman	Editar
financeiro.unopar@escolacem.com.br	Vanda Lopes	Unopar	salesman	Editar
luciana@teste.com	Luciana		manager	Editar
ezequias@teste.com	Ezequias		manager	Editar

https://cemhelp.herokuapp.com/users/new



A figura 23 (vinte e três) apresenta a opção de um novo usuário.

A figura 24 (vinte e quatro) apresenta a opção de edição dos usuários.



A figura 25 (vinte e cinco) apresenta uma tela a após a criação de um curso.



A figura 26 (vinte e seis) apresenta uma tela a após a criação de um Aluno.

