

Uma Aplicação Web para Verificação Automática e em Tempo Real da Situação da Matrícula de Estudantes

Vicente Heitor Terra Silva

¹Departamento de Ciências da Computação
Universidade Federal de Lavras (UFLA) - Lavras, MG - Brasil

Abstract. *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

Resumo. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

Sumário

1	Introdução	3
1.1	Contextualização e Motivação	3
1.2	Objetivos	4
2	Base Teórica	5
2.1	AngularJS	5
2.1.1	Diretivas (<i>Angular directives</i>)	5
2.1.2	Controladoras (<i>controllers</i>)	7
2.2	Webservices e REST	8
2.3	Play Framework	9
3	EasyHalf: Uma aplicação web para verificação automática e em tempo real da situação de matrícula de estudantes	12
3.1	Pesquisa de viabilidade EasyHalf e levantamento de requisitos	12
3.1.1	Pesquisa com os alunos	13
3.1.2	Entrevista com estabelecimentos	18

4	Descrição das funcionalidades implementadas	21
4.1	Funcionalidade exclusivas para os estabelecimentos	21
4.1.1	Verificação de matrícula	21
4.1.2	Status do cadastro	22

Lista de Tabelas

1	Tabela de requisitos básicos do sistema	13
2	Principais problemas enfrentados pelos alunos	16

1. Introdução

1.1. Contextualização e Motivação

Segundo a Lei nº 12.993 [do Brasil 2003], conhecida como "Lei da Meia-Entrada", é assegurado aos estudantes o direito de acesso a alguns estabelecimentos/eventos, tais como cinemas, teatros, espetáculos musicais, mediante o pagamento de metade do valor cobrado para o público geral. Para que esse direito seja garantido, os estabelecimentos nos quais os estudantes requerem o direito à meia-entrada exigem algum documento que comprove que o indivíduo realmente se encontra matriculado em alguma instituição de ensino. Os certificados aceitos como comprovação pela maioria dos estabelecimentos, especificados no texto da lei, são os seguintes:

Carteira de Identificação Estudantil (CIE), emitida pela Associação Nacional de Pós-Graduandos (ANPG), pela União Nacional dos Estudantes (UNE), pela União Brasileira dos Estudantes Secundaristas (UBES), pelas entidades estaduais e municipais filiadas àquelas, pelos Diretórios Centrais dos Estudantes (DCEs) e pelos Centros e Diretórios Acadêmicos, com prazo de validade renovável a cada ano, conforme modelo único nacionalmente padronizado e publicamente disponibilizado pelas entidades nacionais antes referidas e pelo Instituto Nacional de Tecnologia da Informação (ITI), com certificação digital deste, podendo a carteira de identificação estudantil ter 50% (cinquenta por cento) de características locais [do Brasil 2003].

Entretanto, há algumas instituições de ensino em que os DCEs e Centros Acadêmicos não emitem Carteiras de Identificação Estudantil (CIE). Assim sendo, os estudantes precisam recorrer a UNE para adquiri-la, mediante o pagamento de uma taxa de R\$ 25,00 anuais. Os estudantes que, por não terem conhecimento desse serviço ou por não estarem dispostos (ou não terem condições financeiras) a arcar com custos necessários, utilizam a forma mais comum de se ter acesso ao desconto, que é por meio da aquisição manual ou eletrônica de um atestado de matrícula. Cabe ressaltar que esses atestados geralmente são emitidos gratuitamente pelos Departamentos de Controle Acadêmico das instituições de ensino. Visto que muitos estabelecimentos exigem que os atestados de matrícula apresentados no momento da solicitação do desconto sejam recentes, essa estratégia possui algumas limitações, tais como:

1. É necessário que os estudantes retirem atestados de matrícula sempre que tiverem a necessidade de fazer uso do direito à meia-entrada, o que pode trazer transtornos e dificuldades, principalmente quando o processo para emissão do atestado de matrícula é moroso. Em algumas instituições de ensino, o aluno deve imprimir o atestado de matrícula e levá-lo à secretaria do seu respectivo curso para que o mesmo seja validado; e
2. Gera-se demanda para impressão de diversas cópias de atestados de matrícula ao longo do ano, o que, para instituições de médio e grande porte, pode representar impacto ambiental significativo.

Há estabelecimentos que aceitam atestados de matrícula mais antigos, porém essa estratégia garante pouca confiabilidade ao processo de comprovação da situação da matrícula do estudante perante a instituição de ensino. Por exemplo, é possível que um

estudante apresente um atestado de matrícula recente, mesmo não tendo mais vínculo com a instituição de ensino (devido a trancamentos, desligamentos, entre outros motivos) no momento em que está solicitando o desconto. O mesmo problema ocorre com relação às CIE emitidas pela UNE (ou por outras entidades autorizadas, conforme apresentado anteriormente neste texto), uma vez que a validação da matrícula do estudante é feita uma única vez, no momento de emissão da carteirinha, e a mesma tem validade de 12 (doze) meses, em geral [UNE 2016].

1.2. Objetivos

Esse trabalho tem o objetivo de aprimorar o processo de verificação da situação da matrícula dos estudantes em suas respectivas instituições de ensino, garantindo maior agilidade e confiabilidade a esse processo. Para isso, propõe-se o desenvolvimento de uma aplicação *web* capaz de permitir que um determinado estabelecimento, devidamente cadastrado na instituição de ensino, verifique, em tempo real, a situação da matrícula (por exemplo, ativa, trancada, excluída, entre outras) de um estudante, apenas informando seu número de matrícula. A princípio, os dados a serem enviados ao estabelecimento para verificação são: foto do estudante, nome completo, número de um documento de identificação (por exemplo, RG ou CPF) e situação da matrícula. Como objetivos específicos, tem-se:

1. Redução da emissão de cópias de atestados e, conseqüentemente, do impacto ambiental;
2. Melhoria na agilidade e na confiabilidade do processo de verificação de matrículas; e
3. Garantia de maior facilidade de acesso ao direito da meia-entrada por parte dos estudantes.

Considerando que as informações necessárias para se realizar tal verificação são privadas, foi feita uma consulta à Pró-Reitoria de Graduação (PRG) da Universidade Federal de Lavras (UFLA) e a mesma informou que a Procuradoria Geral (PG) da UFLA define uma série de condições para que esses dados possam ser disponibilizados de forma legal. O parecer completo da PG/UFLA pode ser encontrado nas referências do trabalho [UFLA 2015], contudo, algumas dessas condições são resumidas a seguir:

1. O estabelecimento que terá acesso aos dados dos estudantes deverá, por meio de seu representante legal, estar devidamente cadastrado junto à instituição de ensino e terá que assinar um termo de responsabilidade com respeito ao uso desses dados; e
2. O estudante deverá autorizar o acesso e a disponibilização de seus dados aos estabelecimentos de seu interesse e também terá que assinar um termo de responsabilidade com respeito às suas decisões.

O restante deste trabalho está organizado da seguinte forma: o Capítulo 2 apresenta as informações básicas sobre as ferramentas e tecnologias que serão utilizadas no desenvolvimento deste trabalho; e no Capítulo 3 são discutidos os procedimentos metodológicos necessários para se alcançar os objetos traçados para este trabalho e o cronograma de desenvolvimento e entregas das etapas do trabalho.

2. Base Teórica

2.1. AngularJS

O AngularJS é um *framework* de desenvolvimento para aplicações *web* dinâmicas. No AngularJS a linguagem HTML é usado como modelo e o angular permite que a sintaxe do HTML seja estendida para criação de novos componentes de forma simples e clara. Segundo [AngularJS.org],

”O AngularJS é o que o HTML teria sido se tivesse sido projetado para aplicações”.

O HTML é uma linguagem onde se insere anotações em um texto para que tal texto seja sintaticamente distinguível, o que o torna ideal para documentos estáticos [AngularJS.org], porém, em se tratando de aplicações dinâmicas o HTML traz limitações. Para contornar tais limitações é necessário que os desenvolvedores de aplicações *web* usem de estratégias para ”forçar”o navegador a fazer o que o desenvolvedor precisa.

Os problemas de compatibilidade entre documentos estáticos e aplicações dinâmicas normalmente são contornados com o uso de bibliotecas como o JQuery, o qual traz ferramentas para a manipulação do HTML DOM (*Document Object Model*), e também com o uso de *frameworks*.

O AngularJS tem uma abordagem diferente, nele tenta-se minimizar os problemas já mencionados de compatibilidade entre documentos estáticos e aplicações criando novas estruturas HTML. O Angular permite alteração das estruturas padrões do HTML por meio das diretivas(*directives*), na Listagem 1 é exibido um exemplo básico de um código HTML utilizando o AngularJS, o funcionamento deste código básicos será explicado na Sessão 2.1.1.

Listing 1. Código HTML básico utilizando Angular

```
1 <html ng-app="myApp">
2 <head>
3 <script src="angular.min.js"></script>
4 </head>
5 <body>
6 <div ng-controller="myController">
7 <label>Nome:</label>
8 <input type="text" ng-model="seuNome" placeholder="Digite seu
   nome:">
9 <hr>
10 <h1>Ola {{seuNome}}!</h1>
11 </div>
12 </body>
13 </html>
```

2.1.1. Diretivas (*Angular directives*)

Primeiramente pode ser observado no código da Listagem 1, os atributos *ng-app* e *ng-controller*.

Com esses atributos dizemos ao Angular o escopo da aplicação, por meio da diretiva *ng-app*, ou seja, todo o código dentro da *tag* com o atributo *ng-app* será controlado pelas controladoras atribuídas ao módulo chamado de "myApp". Cada parte do código pode ser atribuído a uma controladora diferente, portanto para definir qual controladora é responsável por certa parte do código adiciona-se o atributo *ng-controller* especificando o nome da respectiva controladora.

No código da Listagem 1 podemos ver um exemplo de *data binding*. Através do atributo *ng-model* o AngularJS nos permite ligar o valor do *input* a uma variável chamada "seuNome". Com *Data Binding* é possível alterar a *view* sempre que o modelo for alterado. Assim como alterar o modelo quando a *view* for alterada, o que nos permite eliminar o trabalho com manipulação de DOM (*Document Object Model*). Com isso é possível exibir o valor digitado no *input* em tempo real, para isso basta colocar o nome da variável atribuída ao *ng-model* entre chaves duplas.

1. Ligação bidirecional de dados (*two-way data binding*):

Listing 2. Diretiva ng-model

```
1 -><input type="text" ng-model="ctrl.seuNome"
2   placeholder="Digite seu nome:">
```

Seguindo a mesma ideia do primeiro exemplo, no código da Listagem 2 um campo *input* é ligado a uma variável, contudo, agora essa variável já vem com um valor atribuído a ela na controladora responsável por aquela parte do código, dessa forma ao se carregar a página o campo que possui o atributo *ng-model* já virá preenchido com o valor da variável "seuNome". Isso é chamado de *two-way data binding* por permitir que o modelo seja alterado quando a *view* é alterada e vice-versa, propriedade que é aplicada em inúmeras situações;

2. Escuta de eventos (*Event listeners*):

Listing 3. Diretiva ng-click

```
1 -><button ng-click="ctrl.addNome()">
2   Adicionar a lista</button>
```

Com os *event listeners* é possível detectar eventos como clique, passagem de mouse por uma elemento, tecla acionada, etc.

Uma diretiva muito usada é chamada de *ng-click*, mostrada na Listagem 3, com ela é possível invocar ações quando o elemento é clicado. No exemplo foi atribuído essa diretiva a um botão que ao ser clicado chamará uma função chamada "addNome", a qual foi definida na controladora. Essa função adiciona um novo elemento a uma lista chamada de "listaNomes". O valor adicionado está salvo na variável "seuNome", graças ao *data binding* feito anteriormente com a diretiva *ng-model*;

3. Repetição de elementos:

Listing 4. Diretiva ng-repeat

```
1 -><ul><li ng-repeat="nome in ctrl.listaNomes track by
   $index">
2   {{ nome }}</li></ul>
```

O Angular também permite iterar em uma lista de elementos diretamente no HTML, por meio do código mostrado na Listagem 4. Isso é alcançado com uso da

diretiva *ng-repeat*, cada elemento da lista chamada de "listaNomes" criada na controladora é escrito na tela e graças ao *data binding* a medida que essa lista recebe novos elementos a tela é atualizada automaticamente.

2.1.2. Controladoras (*controllers*)

O Angular também permite definir o comportamento dos elementos do DOM de forma simples e de fácil leitura. Para isso são criadas controladoras (*controllers*) onde serão inseridos os códigos *Javascript* que definirão as ações ou estado dos elementos do DOM. Na Listagem 5 temos um exemplo de uma *controller*:

Listing 5. Código da controladora 'myController'

```
1 ->angular.module('myApp', [])
2   .controller('myController', function() {
3     var ctrl = this;
4     this.seuNome = "";
5     this.listaNomes = [];
6
7     this.addNome = function() {
8       this.listaNome.push(this.seuNome);
9     }
10  });
```

Com isso, o código HTML usado na Listagem 1 pode ser incrementado dando condições para o usuário adicionar os nomes digitados a uma lista e exibir o estado, em tempo real, desta lista. Os incrementos resultam no código exibido na Listagem 6

Listing 6. Código utilizando todos os exemplos citados

```
1 -><html ng-app="myApp">
2 <head>
3 <script src="https://ajax.googleapis.com/ajax/libs/angularjs
4   /1.6.6/angular.min.js"></script>
5 </head>
6 <body>
7   <div ng-controller="myController as ctrl">
8     <label>Nome:</label>
9     <input type="text" ng-model="ctrl.seuNome"
10       placeholder="Digite seu nome:">
11     <button ng-click="ctrl.addNome()">Adiciona a lista</b
12       utton>
13     <div>
14       <ul>
15         <li ng-repeat="nome in ctrl.listaNomes track by
16           $index">{{ nome }}</li>
17       </ul>
18     </div>
19   </div>
20 </body>
```

```

19 <script type="text/javascript">
20 angular.module('myApp', [])
21   .controller('myController', function() {
22     var ctrl = this;
23     this.seuNome = "Joao";
24     this.listaNomes = [];
25
26     this.addNome = function() {
27       this.listaNomes.push(this.seuNome);
28     }
29   });
30 </script>
31 </html>

```

2.2. Webservices e REST

A ideia é que a aplicação proposta neste trabalho possa ser implantada em diversos tipos de instituições de ensino. Também é interessante, no ambiente atual, que essa aplicação possa ser utilizada em diferentes plataformas como dispositivos móveis e plataformas *Web* por exemplo. Para tornar a aplicação proposta independente da infraestrutura dos sistemas das instituições de ensino, mas ao mesmo tempo interoperante com as mesmas, propõe-se o uso de *webservices*.

O uso de *webservices* possibilita o acesso de informações e interação entre componentes de um sistema, ignorando os detalhes de implementação das aplicações. Com isso, os dados podem ser obtidos de diferentes plataformas sem a necessidade de mudar o modelo em que aplicação foi desenvolvida.

Será usado no desenvolvimento da aplicação proposta neste trabalho, um estilo arquitetural para desenvolvimento de *webservices*, conhecido como REST (*Representation State Transfer*). Segundo [Burke 2013], REST é definido por um conjunto de princípios arquiteturais, sendo eles:

1. Recursos Endereçáveis (*Addressable Resources*):

No REST, deve-se levar em consideração um novo conceito conhecido como "recurso". Segundo [Fielding 2000], "qualquer informação que pode ser nomeada pode ser um recurso". Basicamente, qualquer informação que pode ser obtida, inserida, alterada ou removida em uma aplicação, como as entidades por exemplo, pode ser um recurso. O primeiro princípio do REST indica que todo recurso de uma aplicação deve ser endereçável, normalmente por meio de URLs, como por exemplo: `/app/usuario`. No caso "usuario" é um recurso da aplicação "app" e está referenciado pela URL acima. Isso é necessário para que se tenha acesso a esses recursos posteriormente.

2. Uma Interface Uniforme e Contida (*Uniform, Constrained Interface*):

É usado um conjunto restrito e bem definido de métodos para manipular os recursos da aplicação. Normalmente, utilizam-se os métodos pré-definidos no protocolo HTTP, a saber, GET, POST, PUT, DELETE, etc. Quando se deseja buscar determinado recurso ou uma lista de recursos, pode-se usar do método GET para fazer uma requisição a esse recurso utilizando sua URL, como no

exemplo: GET /app/usuario. Essa requisição permite que sejam retornados os usuários da aplicação. Se necessário, pode-se obter um usuário específico, acrescentando à URL o valor de uma chave do recurso (usuário) que o identifique unicamente. O exemplo a seguir retorna um usuário com código igual a '1': GET /app/usuario/1

3. Orientado a Representação (*Representation-Oriented*):

Um recurso requisitado por meio de uma URL pode ser retornado em vários formatos diferentes como HTML, JSON, XML, entre outros. Por isso é necessário que seja descrito na resposta, também utilizando o protocolo HTTP, o formato do recurso que será enviado no corpo da resposta. Isso permite que sejam conhecidos quais formatos de representação estão sendo utilizados entre o cliente e o servidor. É possível também que o cliente do serviço informe, na requisição, qual tipo de representação ele tem preferência em receber.

4. Comunicação Independente (*Communicate Statelessly*):

Essa característica explicita que na arquitetura REST, não se guarda o estado da comunicação entre cliente e servidor. O servidor recebe e manipula os dados vindos da requisição e depois os destaca. Essa característica garante maior poder de escalabilidade às aplicações, uma vez que é possível escalonar requisições entre diversas máquinas de forma mais simples [Fielding 2000].

5. HATEOAS (em ingles *Hypermedia as The Engine of the Application State*):

É uma abordagem que permite a incorporação de *links* para outros serviços e informações junto com os dados do recurso. Por exemplo, junto com os dados de um recurso 'filmes', pode vir um *link* para o serviço de reserva ou locação do filme em uma locadora *online*.

2.3. Play Framework

Para o desenvolvimento do *back-end* da aplicação será utilizado Play Framework. O Play vem sendo utilizado para desenvolvimento de aplicações web com Java de forma simples e eficiente desde 2007 além uma comunidade ativa com mais dez mil pessoas [PlayFramework 2017]. Nativamente permite o desenvolvimento nos padrões REST(*Representation State Transfer*), oferecendo suporte às requisições HTTP de forma simples, além de ser fácil de ser construída e hospedada, facilitando o desenvolvimento da aplicação nos padrões pretendidos.

O Play possibilita a vinculação de métodos escritos nas classes controladoras do sistema á URLs(Em ingles *Uniform Resource Locator*). Essa vinculação é feita de forma simples e organizada através de um arquivo de configuração padrão, chamado de "Routes"que é criado em qualquer projeto Play básico.

O arquivo de configuração de rotas do Play Framework é um aquivo de texto comum onde serão especificadas uma URL(Em ingles *Uniform Resource Locator*) para cada método ou função do sistema que necessita ser consumido pelo *front-end* da aplicação.

Algumas características essenciais de uma controladora utilizada no Play pode ser vista na Listagem 7 .

Listing 7. Exemplo de uma controladora em um projeto Play

```
1 package controllers;
2
3 import models.Usuario;
4 import play.mvc.Controller;
5 import play.mvc.Result;
6 import play.libs.Json;
7
8 public class UsuarioController extends Controller {
9
10     public Result new() {
11         Usuario newUser = new Usuario();
12         newUser.save();
13
14         return ok();
15     }
16
17     public Result getUser(Integer id) {
18         Usuario user = new Usuario();
19         user = Usuario.find(id);
20
21         return ok(Json.toJson(user));
22     }
23 }
```

A Listagem 7 mostra uma controladora que contém lógica de negócios relacionada a uma entidade chamada "Usuario". Uma controladora no Play é um arquivo ".java" comum que deve ser estendido da classe "Controller" definida pelo próprio Play. Como os métodos da controladora serão chamados em requisições HTTP, o retorno dos mesmos devem ser em formatos genéricos, que consigam ser lidos pela aplicação *front-end*. Os formatos mais comuns são o JSON (em inglês *JavaScript Object Notation*) e XML (em inglês *Extensible Markup Language*).

No caso do Play o retorno dos métodos devem ser no formato de uma classe pré definida pelo Play chamada de "Result" retorna uma resposta de uma requisição HTTP e acoplado a essa resposta poder ser retornado, se necessário, o resultado do método no formato JSON.

Listing 8. Arquivo de rotas em um projeto Play

```
1 #USER ROUTES
2
3 POST /user/new      controllers.UsuarioController.new()
4 GET /user/:id      controllers.UsuarioController.getUser(
                        id:Integer)
```

Na Listagem 8 tem-se duas rotas que seriam criadas no arquivos de rotas do Play, na linha 3 cria-se uma rota para um método de criação de um novo usuário e a segunda rota vista na linha 4 tem-se um método que busca um usuário pelo seu número de identificação. A estrutura do arquivo é simples, primeiramente deve-se especificar o tipo de requisição HTTP (em inglês *Hypertext Transfer Protocol*), ou seja, se é um método

GET,POST,PUT ou DELETE. Em seguida se define a URL em si, a qual poder ser da forma que o desenvolvedor preferir. Por ultimo deve-se dar o caminho, dentro do projeto, onde se encontra o código fonte do método a ser executado quando a URL for invocada.

3. EasyHalf: Uma aplicação web para verificação automática e em tempo real da situação de matrícula de estudantes

Nessa sessão do trabalho são descritos os procedimentos tomados para entender a viabilidade das ideias definidas para a aplicação a ser desenvolvida. E também para conhecer as necessidades dos potenciais usuários da aplicação, com o intuito de desenvolver uma aplicação útil e interessante para uso. Para atingir os objetivos propostos neste trabalho, os seguinte procedimentos metodológicos foram realizados:

1. Pesquisa de viabilidade do software e definição de requisitos: A pesquisa de viabilidade e elicitação dos requisitos da aplicação proposta neste trabalho foi feita com base em entrevistas e questionários aplicados a um ou mais indivíduos representantes dos estudantes, das instituições de ensino e dos estabelecimentos que oferecem desconto de meia-entrada esses indivíduos. Nesses questionários foram abordadas questões que ajudaram a entender quais são as principais necessidades dos usuários e quais seriam as melhores formas de suprir tais necessidades.
2. Avaliação da Aplicação: a avaliação da aplicação foi feita com testes de usuários seguidos de respostas a um questionário de avaliação cujos resultados estão descritos na Sessão 3.1.1.

3.1. Pesquisa de viabilidade EasyHalf e levantamento de requisitos

A primeira etapa do desenvolvimento do trabalho foi a pesquisa de viabilidade do *software*. O levantamento foi feito com os principais tipos de potenciais usuários do EasyHalf, sendo eles os alunos que possuem o direito ao desconto de meia-entrada e os estabelecimentos que por lei devem conceder o desconto em seus produtos ou serviços.

Para a obtenção das informações relevantes na definição dos requisitos do sistema e sua viabilidade foi utilizado um questionário para os estudantes e outro para os estabelecimentos interessados.

O questionário aplicado aos estudantes era composto de seis perguntas, sendo quatro perguntas objetivas e duas de múltipla escolha. O questionário foi divulgado através de uma rede social em grupos de alunos dos cursos de Sistemas de Informação e Ciência da Computação da Universidade Federal de Lavras. E também com o envio de e-mails aos alunos do departamento de Ciência da Computação, e-mails enviados pelo orientador deste trabalho e por outros professores do departamento. O questionário obteve cento e trinta e três respostas e ficou disponível para preenchimento de outubro de 2016 á março de 2017.

Já o questionário aplicado para os estabelecimento era composto de sete perguntas todas discursivas. O questionário foi aplicado á três estabelecimentos, os quais possuem a maior frequência de solicitações do descontos de meia entrada, sendo eles o cinema local, a empresa de transporte público da cidade e uma agência organizadora de eventos local, responsável pelo maior evento feito na cidade e por outros menores.

Com base nos resultados dos questionários foram definidos os seguintes requisitos básicos da aplicação:

Tabela 1. Tabela de requisitos básicos do sistema

Requisito	Descrição
1	O sistema deve permitir que usuários se cadastrem.
2	O sistema deve exigir que um usuário se autentique para que o mesmo possa utilizar suas funcionalidades.
3	O sistema deve exibir um ambiente para os usuários cadastrados do tipo estudante e um para os usuários cadastrados do tipo estabelecimentos.
4	O ambiente para os estabelecimentos deve permitir a consulta da situação de um determinado estudante, que por ventura esteja solicitando o desconto.
5	A consulta deve retornar a situação atual, atualizada, da matrícula consultada. Assim como informações do estudante detentor da matrícula.

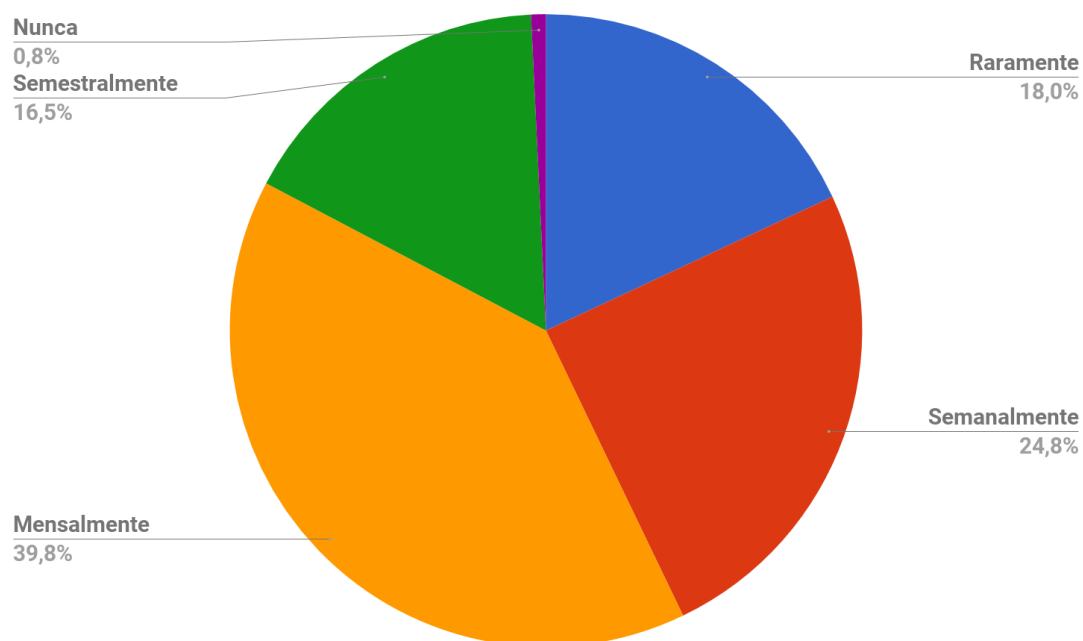
3.1.1. Pesquisa com os alunos

Para a compreensão das necessidades dos alunos ao se solicitar o desconto em diferentes tipos de estabelecimentos, foi aplicado um questionário para se entender importantes aspectos de como os estudantes fazem uso do direito a meia-entrada que lhes é garantido.

O primeiro ponto abordado no questionário foi a frequência com que os estudantes solicitam o desconto de meia-entrada em qualquer que seja o tipo de estabelecimento.

Na Figura 1 são exibidos os resultados das respostas de 133 alunos, quando perguntados sobre frequência de uso do desconto de meia-entrada em estabelecimentos como cinemas, empresas de transporte público, entre outros. Os resultados obtidos dessa pergunta mostraram que mais de 60% dos alunos com frequência mensal ou maior. Essa frequência elevada mostrou a importância de que o processo de concessão de desconto seja ágil e o mais simples possível.

Figura 1. Frequência de uso do desconto de meia-entrada

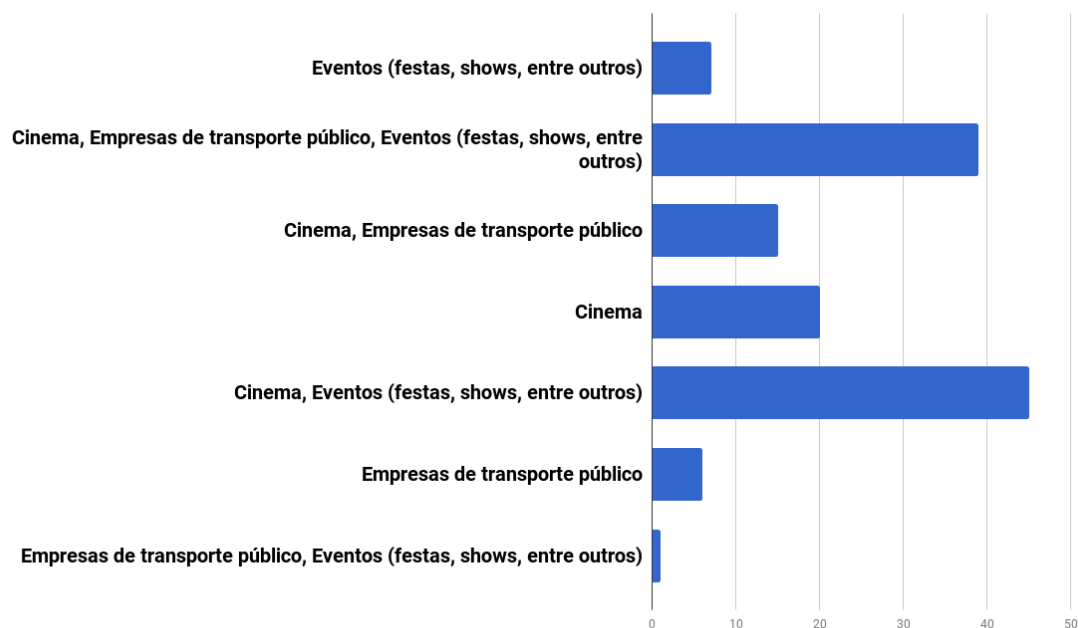


A segunda pergunta feita no questionário visou esclarecer em quais tipos de estabelecimentos os alunos mais fazem o requerimento de descontos.

Os resultados exibidos na Figura 2 mostraram que a maior partes dos alunos utilizam o desconto em cinemas. Esses mesmos alunos que utilizam os descontos em cinemas também o fazem com frequência em eventos(shows,festas) e também com no transporte público.

As informações obtidas nessa pergunta permitiram o planejamento do levantamento de requisitos com os estabelecimentos. Conhecendo os estabelecimentos onde se tem maior volume de requisições por descontos se tornou mais fácil de planejar o conteúdo e com quais estabelecimentos deveriam ser feitas as entrevistas, para que os requisitos pudessem ser os mais precisos possíveis.

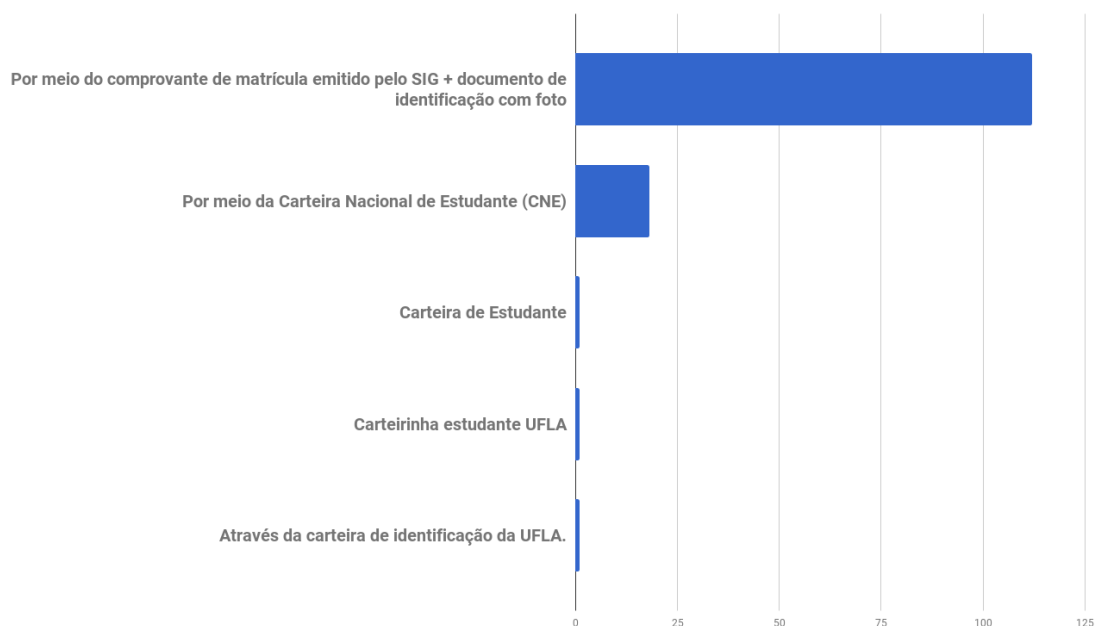
Figura 2. Lugares onde os alunos tem costume de solicitar o desconto de meia-entrada



A terceira pergunta do questionário teve o intuito de descobrir qual a forma de comprovação de matrícula mais usada pelos alunos quando solicitam os descontos.

A Figura 3 mostrou que a grande parte dos alunos comprovam seu vínculo estudantil por meio comprovante de matrícula impresso, emitido pela instituição de ensino. Já foi mencionado na introdução deste trabalho os problemas com essa forma de comprovação de matrícula, portanto os resultados dessa pergunta reforçam o fato de que é necessário um método de verificação de matrícula menos moroso.

Figura 3. Documentos utilizados pelos alunos para comprovação de vínculo estudantil



Na quarta pergunta foi feito o levantamento dos principais problemas encontrados pelos estudantes ao tentarem o acesso ao desconto de meia-entrada.

Tabela 2. Principais problemas enfrentados pelos alunos

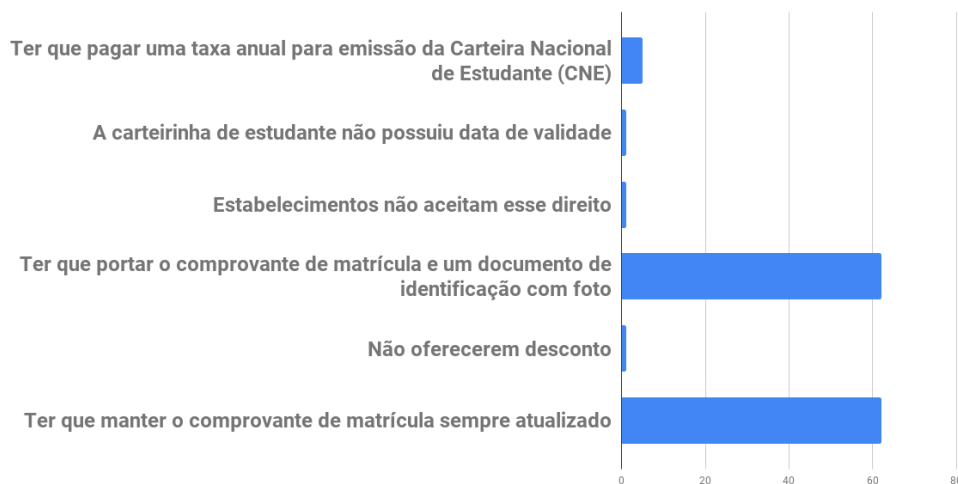
Problemas	Números de relatos	Porcentagem
O estabelecimento não aceitou o comprovante de matrícula, pois estava desatualizado	47	35,3 %
O estabelecimento não aceitou o comprovante de matrícula, pois o mesmo não apresentava data de vencimento	80	60,2 %
O estabelecimento não aceitou o comprovante de matrícula, pois eu não portava um documento de identificação com foto	30	22,6 %

A Tabela 2 mostra os resultados mais comuns referenciados pelos alunos no questionário. A maior parte dos problemas foi gerado pela data de validade ou a ausência de tal data no documento utilizado para comprovação de vínculo estudantil. O EasyHalf é idealizado exatamente para contornar essa situação, com uma verificação em tempo real não é necessário que o documento contenha data atualizada, ou que nem mesmo contenha data de validade.

Na quinta pergunta do questionário foi levantado o que os alunos achavam trabalho nos processos já existentes para comprovação de matrícula. Gerando assim bases para se definir o que exatamente o EasyHalf deveria aprimorar no processo de verificação, de forma que sanasse as dificuldades enfrentadas pelos alunos.

Os resultados dessa pergunta, como pode ser visto na Figura 4, reafirmaram os problemas levantados antes da consulta com os alunos. A grande parte dos alunos responderam que a necessidade de sempre carregar um comprovante de matrícula, atualizado, é o que mais dificulta o acesso ao desconto de forma ágil e eficaz.

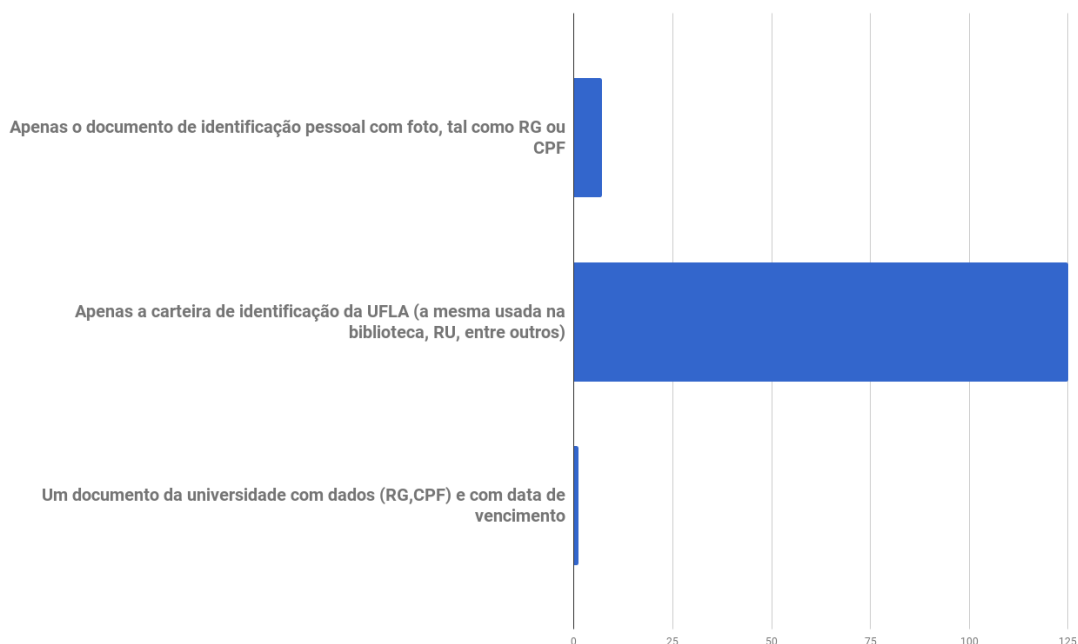
Figura 4. Dificuldades enfrentadas pelos alunos para conseguirem o desconto de meia-entrada



Na sexta pergunta foi perguntado aos alunos qual documento, ou tipo de documentos, eles achavam ideal para portarem no momento da solicitação dos descontos.

A maioria esmagadora dos alunos responderam que o documento ideal seria a carteira de estudante emitida pela universidade, como pode ser visto na Figura 5. Documento que já utilizado pelos estudantes em diversos outros lugar na própria universidade, lugares como a biblioteca e o restaurante universitário.

Figura 5. Documento considerado ideal para portar ao solicitar o desconto de meia entrada



3.1.2. Entrevista com estabelecimentos

Com os resultados da consulta aos alunos foi possível definir alguns estabelecimentos que seriam considerados de prioridade no levantamento de requisitos. Os resultados obtidos com os alunos mostraram que os estabelecimentos onde se tem a maior frequência de solicitações do desconto de meia-entrada são os cinemas, empresas de transportes e em eventos como shows, festas, etc.

Tendo em vista tais resultados foram feitas consultas a alguns estabelecimentos, da cidade de Lavras-MG, que atuam nas áreas com alta frequência de solicitações do desconto. As perguntas feitas nas entrevistas foram as seguintes:

1 - "Com qual frequência, alunos procuram seu estabelecimento solicitando o desconto de meia-entrada (p. ex. semanalmente, mensalmente, semestralmente, raramente, nunca)?"

A primeira pergunta foi sobre a frequência com que os alunos solicitam o desconto de meia-entrada. Para ambos o cinema local e a empresa de transporte público a frequência de solicitação é diária. Já a empresa de eventos entrevistada respondeu que é solicitado descontos em todos os eventos realizados pela empresa.

2 - "Qual o volume de solicitações desse tipo de desconto ao mês, em seu estabelecimento?"

Na segunda pergunta tentou-se descobrir se os estabelecimentos conseguiam dizer um valor exato ou estimado de número de solicitações por descontos de meia-entrada. O cinema não soube dar um valor exato de clientes, mas garantiu que de 75% a 80% das

vendas são de meia-entrada. A organizadora de eventos também não soube estimar a quantidade de solicitações dentre todas as vendas.

3 - "Qual é o tipo de documento exigido pelo estabelecimento para comprovação da situação da matrícula dos alunos?"

Os documentos aceitos pelos estabelecimentos são bem diversos, são aceitos atestados de matrícula das instituições, recibos de mensalidades, ou qualquer documento que contenha os dados do estudante solicitante e da instituição, no mínimo data de emissão e que mostre o tipo de vínculo do solicitante com a instituição.

Porém, esses atestados em sua maior parte não possuem data de validade, nesse caso os estabelecimentos limitam sua aceitabilidade pela data de emissão do documento. O tempo em que o documento é aceito após a data de emissão varia de um estabelecimento para outro, dentre os entrevistados o prazo foi de uma semana para a empresa de transporte e de um mês para o cinema e a organizadora de eventos.

4 - "Vocês acham satisfatório, em termos de eficácia e confiabilidade, o método que o estabelecimento adota para comprovação da situação da matrícula dos alunos? Se não, descreva o que não te satisfaz no método atual."

Ambos o cinema e organizadora de eventos mostraram algum tipo de insatisfação com o estado atual do processo de verificação de matrícula. O principal problema com o método atual é a confiabilidade, problema que foi apontado por ambas as empresas diversas vezes na entrevista e na resposta à essa pergunta quanto a satisfação com o método atual.

Os problemas de verificar a matrícula por meios dos documentos de costume sempre giram em torno da sua falta de data de validade. Com a ausência da data de validade não é possível prevenir que o desconto seja dado a quem não possui o direito, o que traz prejuízos aos estabelecimentos, principalmente os entrevistados que grande parte do seu volume de vendas são com desconto de meia-entrada.

A empresa de transporte não disponibilizou nenhum funcionário do setor gerencial capaz de responder a essa pergunta, por isso não há o *feedback* da empresa de transporte para esse tópico.

5 - "Você acharia interessante um software que permitisse a verificação, em tempo real, da situação da matrícula dos alunos, tendo como contrapartida, a necessidade de o estabelecimento realizar e manter atualizado um cadastro junto à instituição de ensino?"

Ambos o cinema e organizadora de eventos mostraram interesse na ideia e deixaram claro que o EasyHalf poderia aprimorar seus processos de vendas com desconto de meia-entrada. Mesmo desmostrando interesse os estabelecimentos levantaram pontos que foram úteis no planejamento do desenvolvimento do EasyHalf e também no amadurecimento da ideia.

Ambos os estabelecimentos citaram que o desconto também é concedido a outras pessoas além de alunos, pessoas como jovens de baixa renda, deficientes, idosos, entre outros. Esse fato mostrou que a abrangência do EasyHalf pode ser aumentada e aprimorada, aspecto que deve ser abordado no trabalho desenvolvido até aqui mas que deverá ser lidado no futuro.

A organizadora de eventos citou o problema de um solicitante não aceitar o resultado mostrado pelo EasyHalf no momento da verificação. Esse fato foi abordado no desenvolvimento e buscou-se contorná-lo com termos e condições de devem ser aceitos ao se cadastrar no sistema. Neles os usuários concordam que se solicitado a verificação por meio do EasyHalf, os mesmos aceitam o resultado da consulta.

A empresa de transporte não disponibilizou nenhum funcionário do setor gerencial capaz de responder a essa pergunta, por isso não há o *feedback* da empresa de transporte para esse tópico.

6 - "Quais seriam os dados necessários para uma verificação eficaz e confiável da matrícula de um aluno?"

Algumas das informações sobre o solicitante listadas como importantes pelos entrevistados foram:

- Data da matrícula
- Data de vigência de matrícula
- Nome completo
- Situação da matrícula no momento da consulta
- Documento de identificação com foto

4. Descrição das funcionalidades implementadas

4.1. Funcionalidade exclusivas para os estabelecimentos

4.1.1. Verificação de matrícula

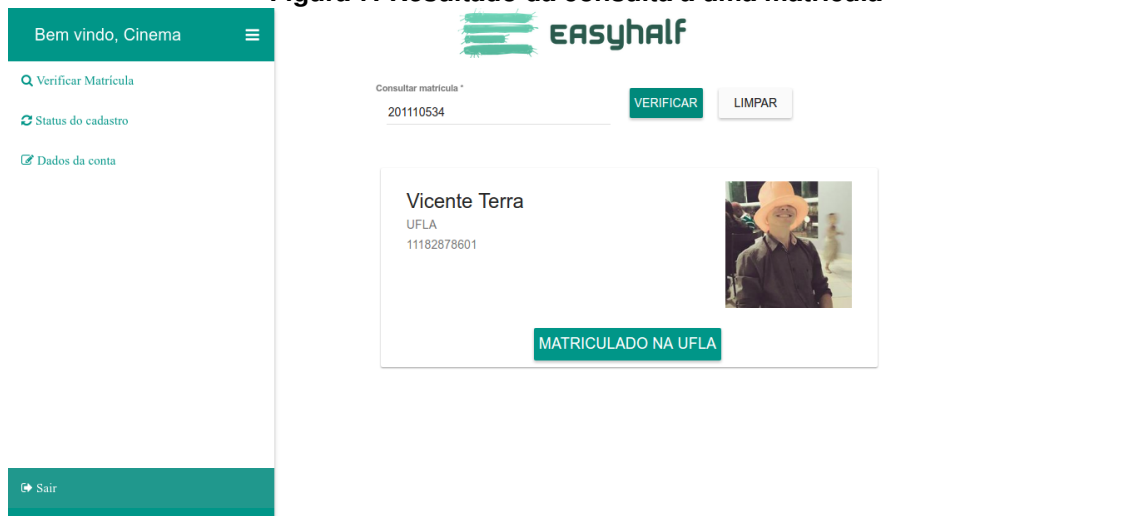
A principal funcionalidade do sistema é a verificação de matrícula dos solicitantes do desconto. O sistema permite que um estabelecimento devidamente cadastrado e autenticado faça a consulta por meio do número de matrícula do usuário.

Figura 6. Tela de verificação de matrícula

A imagem mostra a interface de usuário para a verificação de matrícula. No topo, há uma barra de navegação verde com o texto "Bem vindo, CineA" e um ícone de menu. Abaixo, no mesmo lado, há uma lista de opções: "Verificar Matrícula" (destacada com um ícone de lupa), "Status do cadastro" (com um ícone de seta circular) e "Dados da conta" (com um ícone de checklist). No canto inferior esquerdo da barra, há um botão "Sair" com um ícone de porta. No centro da tela, há o logotipo "EASYhalf" com uma barra decorativa verde. Abaixo do logotipo, há um campo de texto rotulado "Consultar matricula *" e dois botões: "VERIFICAR" e "LIMPAR".

Na Figura 6 é exibida a tela da aplicação onde o estabelecimento fará a consulta pela matrícula de um aluno solicitante do desconto de meia entrada. Basta que o operador do sistema digite o número de matrícula informado pelo solicitante e clique em "verificar", assim, o sistema retornará a resposta da consulta como pode ser visto na Figura 7.

Figura 7. Resultado da consulta á uma matrícula



O sistema retornará informações básicas do usuário dono da matrícula consultada, informações como seu nome, universidade em que está matriculado e foto. Tais dados serão exibidos para confirmação da identidade do solicitante, com eles o operador do estabelecimento pode checar se o solicitante é mesmo o aluno dono da matrícula informada. Além das informações pessoais para verificação da identidade do solicitante o sistema exibirá a situação da matrícula no momento da consulta.

4.1.2. Status do cadastro

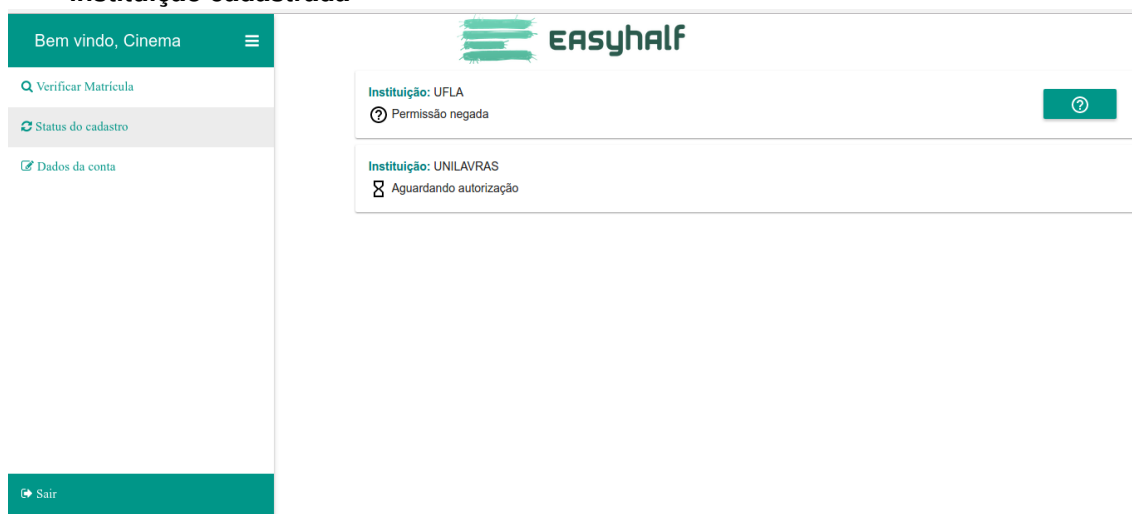
Mesmo com o cadastro no sistema, para o estabelecimento consultar as informações de um aluno é necessário que a instituição de ensino em que o aluno está matriculado autorize tal consulta. Medida que foi implementada para garantir que as informações pessoais do aluno sejam acessadas apenas por estabelecimentos que tenham disponibilizados todos os documentos solicitados no cadastro, documentos que atestarão a autenticidade do estabelecimento e seu compromisso em usar as informações do usuário de forma correta.

Após o cadastro no sistema o estabelecimento estará aguardando a liberação, ou não, para consultar os alunos de cada uma das instituições cadastradas no sistema. Por isso é possível que o estabelecimento consulte a sua situação com cada uma das instituições. Na tela de status do cadastro que pode ser vista na Figura 8 é exibida a lista com todas as instituições cadastradas junto com a situação do estabelecimento.

A situação do estabelecimento pode ser de três tipos:

- Aguardando autorização : Esse status quer dizer que a instituição ainda não avaliou o estabelecimento, ou seja, ainda não deu um parecer favorável nem desfavorável.

Figura 8. Status da permissão para consultar dados de alunos de cada instituição cadastrada



- Permissão concedida: Sinaliza que a instituição avaliou todos os dados do estabelecimento e julgou que o mesmo pode fazer consulta aos seus alunos. Com esse status o estabelecimento conseguirá consultar matrículas de qualquer aluno daquela instituição específica.
- Permissão negada: Esse status sinaliza que a instituição verificou os dados e documentos do estabelecimento mas encontrou algum problema ou impedimento que levou á negação do direito de consultar os dados de seus alunos. Quando uma instituição nega o acesso a um estabelecimento ela precisa das uma justificativa para tal ação. Nessa justificativa a instituição informará o motivo da negação, podendo ser por falta de documentos obrigatórios por exemplo, ou por inconsistências nas informações de cadastro do estabelecimento, dentre outros problemas. A exibição da justifica das instituições podem ser vistas na tela da Figura 9.

Figura 9. Exibição da justificativa em caso de permissão negada



Referências

AngularJS.org. Angularjs - developer guide.

Burke, B. (2013). *RESTfull Java with JAX-RS 2.0*. O'Reilly, 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2 edition.

do Brasil, R. F. (2003). Lei nº 12.993 de dezembro de 2013 seção 1.

Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures.

PlayFramework (2017). Play framework web documentation.

UFLA, P. G. (2015). Parecer n.00111/2015/gab/pfufla/pgf/agu.

UNE (2016). União nacional dos estudantes.