

## **AUTOSYSTEM: Software para Gestão de Autoescolas**

**Autor: Sérgio Felipe Cristo Santiago**

**Orientador: Ezequias Ferreira de Souza**

**Curso: Tecnologia em Análise e Desenvolvimento de Sistemas**

**Período: 6º Período**

**Área de Pesquisa: Ciências Exatas e da Terra**

**Resumo:** Os sistemas de informação exercem uma enorme importância nos dias atuais, proporcionando as empresas maior controle das informações, segurança e consistência no fluxo de dados. O presente artigo tem como objetivo demonstrar o *software* AutoSystem e suas funcionalidades. O *software* foi desenvolvido com base em algumas experiências pessoais, e algumas conversas informais com alunos e ex-alunos de autoescolas, onde foi comprovado que a maioria das autoescolas ainda não contam com um sistema de automação, tornando o processo de habilitação muitas vezes inflexível tanto para os alunos, quanto para as autoescolas. O AutoSystem é um *software* de gestão que visa agilizar os processos de cadastro do aluno, manutenção da sua agenda de aulas práticas e também proporcionar um controle de aulas mais prático aos alunos e instrutores. O *software* foi desenvolvido utilizando a linguagem *JavaScript* como base e conta com um sistema *Web* para gestão da autoescola e um aplicativo *mobile* que funciona como uma agenda de aulas para controle dos alunos e instrutores. O artigo também apresenta as ferramentas que foram utilizadas no desenvolvimento do *software* e como ele foi desenvolvido com base nos diagramas UML.

**Palavras-chave:** gestão de aulas, sistema de informação, *software* para gestão de autoescolas.

### **1. INTRODUÇÃO**

Nos últimos anos o mundo vem passando por uma transformação digital e uma automatização de processos nos mais diversos tipos de serviços, processos que antes eram manuais e armazenados em planilhas de papel, hoje podem ser realizados pelo computador, *tablet* ou celular e serem armazenados de forma mais segura e eficiente.

Apesar da importância e da popularização dos sistemas de informação, ainda estamos longe de termos todos os problemas solucionados através da tecnologia, muitas empresas ainda não fazem o uso de um sistema o que torna seus processos lentos, gera inconsistência nos dados e causa dificuldades nas tomadas de decisões estratégicas.

As autoescolas da cidade de Manhuaçu, Minas Gerais, enfrentam esse problema, pelo fato de não possuírem um sistema de informação para

automatizar seus processos, acabam apresentando diversos problemas na manutenção dos mesmos, e sobrecarregando a equipe de atendimento na busca de informações sobre os alunos e sobre sua respectiva agenda de aulas, bem como a inserção e manutenção dessas informações de forma manual.

Visto que o setor de autoescolas ainda carece de atenção em relação a automatização, os sistemas de informação poderiam apresentar soluções para agilizar os processos, dar apoio a equipe de atendimento e aos alunos a fim de melhorar o fluxo de dados e proporcionar um controle mais eficiente. O objetivo do sistema é permitir um controle mais eficiente tanto para os atendentes quanto para os alunos, permitindo que o aluno tenha acesso direto a sua agenda de aulas e suas informações pessoais, de forma automatizada, assim como a instituição tenha acesso e possa manter de forma mais eficiente as informações de seus alunos, instrutores e frota de veículos.

Diante dos problemas e necessidades apresentados, uma pergunta norteia este trabalho: A implantação de um sistema de informação facilitará os processos e aumentará a precisão das informações por parte das autoescolas?

Com o objetivo de resolver esses problemas, foi desenvolvido o *Software* AutoSystem que conta com um sistema *Web* que visa eliminar o uso de planilhas de papel, permitindo ao CFC (Centro de Formação de Condutores) realizar o cadastro e a manutenção dos dados de alunos, instrutores e veículos, assim como agendamento de aulas práticas e exames de forma mais rápida e eficiente. O sistema conta também com um aplicativo *mobile* direcionado aos alunos e instrutores onde ambos podem ter acesso a sua agenda de aulas, proporcionando assim um controle melhor das informações.

## **2. REFERENCIAL TEÓRICO**

### **2.1 TECNOLOGIA NA GESTÃO DA INFORMAÇÃO**

A Tecnologia da Informação (TI) é uma área ou grupo no qual se desenvolve a grande maioria de atividades e serviços desenvolvidos pela área da computação. (OLIVEIRA, 2017). A gestão da tecnologia da informação nada mais é que a administração dos recursos tecnológicos de uma empresa, instituição ou organização, alinhado ao seu planejamento e à sua estratégia.

O'Brien (2004, p.3) diz que:

A tecnologia da informação pode ajudar todos os tipos de empresas a melhorarem sua eficiência e eficácia de seus processos de negócios, tomada de decisões gerenciais e colaboração de grupos de trabalho e com isso pode fortalecer suas posições competitivas em mercado em rápida transformação.

Sendo assim, a tecnologia da informação é uma mescla de todas as atividades e soluções fornecidas por recursos de computação. A TI está enriquecendo todo o processo organizacional, auxiliando na otimização das atividades, facilitando a

comunicação e melhorando o processo decisório, pois as informações são mais eficientes e eficazes, chegam ao gestor com mais velocidade e precisão (BEAL, 2009).

## **2.2 AUTOMAÇÃO DE PROCESSOS**

Os processos empresariais surgiram como forma de organizar e padronizar a forma de se trabalhar. Segundo (OLIVEIRA, 2014), um processo empresarial se trata de um trabalho contínuo e repetitivo e que segue uma série de passos sequenciais que são adotados e padronizados por uma organização para alcançar um objetivo desejado.

A automação de processos veio para gerar mais resultados. De acordo com o autor (RIBEIRO, 2016), “a automação de processos é uma técnica que consiste em usar a tecnologia e a integração de sistemas e dados para aprimorar o controle e o andamento do fluxo de trabalho, por meio do monitoramento em tempo real e, quando possível da substituição de atividades manuais por atividades automatizadas”.

A automação de processos proporciona atividades mais organizadas e confiáveis. De acordo com a autora (CASTRO, 2016), a automação de processos diminui consideravelmente os riscos de atrasos em projetos grandes pelo fato de facilitar a identificação dos responsáveis por cada etapa do projeto. Sendo assim é possível obter processos mais inteligentes e autônomos, sem que haja preocupações desnecessárias.

## **2.5 TECNOLOGIAS USADAS NO DESENVOLVIMENTO DO SOFTWARE**

As seguintes tecnologias foram utilizadas para o desenvolvimento do *Software AutoSystem*.

### **2.5.1 JavaScript**

Para o projeto em questão, foi escolhido a linguagem de programação *JavaScript* que será utilizada tanto na parte visual (*front-end*) quanto nas regras de negócio e processos internos (*back-end*). O *JavaScript* é uma linguagem de programação interpretada, fracamente tipada, criada inicialmente para rodar do lado do cliente, utilizada em milhões de páginas *web* para validação, detectar objetos e adicionar uma série de funcionalidades (SILVA, 2010).

O *JavaScript* possui interação direta com várias bibliotecas e permite desenvolver um código mais simples e legível, utiliza também do JSX que permite que o HTML seja utilizado no formato de *JavaScript*.

Alta compatibilidade com plataformas, sistemas e navegadores web; É mais leve e rápida que outras linguagens de programação; Faz com que as páginas na internet sejam mais dinâmicas e interativas, características essenciais do UX; Os navegadores interpretam a linguagem por conta própria, tirando a necessidade de usar um compilador; Erros de programação

são mais fáceis de encontrar e de corrigir; Entre as mais populares, é a linguagem de programação mais fácil de aprender; Executa comportamentos específicos em uma página, como cliques e efeitos personalizados. (ANDREI, 2019)

### 2.5.2 NodeJS

Como o *JavaScript* foi uma linguagem que cresceu muito na *web* nos últimos anos, hoje é possível utilizá-lo do lado do servidor (*server-side*), para trabalhar com banco de dados por exemplo. Para isso podemos utilizar o *NodeJS*. *NodeJS* é uma plataforma de desenvolvimento do lado do servidor, de código aberto, que executa *JavaScript*, assim é possível acessar bancos de dados, fornecer dados através de uma API ou realizar qualquer outra operação de linguagens *back-end*. Dentre as vantagens que motivaram a utilização do *NodeJS* para o desenvolvimento deste projeto podemos citar principalmente a simplicidade que a plataforma oferece ao desenvolvedor, a ferramenta não possui dependências, sendo fácil e de rápida instalação, além da vantagem de diminuir a curva de aprendizado uma vez que usando a linguagem *JavaScript* é possível utilizar a mesma lógica do *front-end* para o *back-end* (FERNANDES, 2017).

### 2.5.3 ReactJS

O *JavaScript* pode ser utilizado literalmente em todas as pontas de um projeto, pensando nisso algumas bibliotecas de *front-end* foram desenvolvidas, entre elas está o poderoso *ReactJS*.

O *ReactJS* é uma biblioteca desenvolvida pelo *Facebook* que tem como foco a componentização de uma interface, e que permite ao desenvolvedor escrever código HTML dentro do *JavaScript* utilizando o JSX.

O *ReactJS* possui código aberto para interfaces gráficas que tem como foco tornar a experiência do usuário mais eficiente e agradável. A interface gerada pelo *ReactJS* pode ser categorizada como o “V” no padrão MVC (*Model-View-Controller*) (ANDREI,2019).

### 2.5.4 React Native

Assim como o *ReactJS*, o *React Native* é uma biblioteca desenvolvida pelo *Facebook*, focada no desenvolvimento de interfaces UI (*User Interface*), que são componentes que o usuário consegue ver e interagir (Botões, formulários, menus) utilizando o *JavaScript*, porém o *React Native* é voltado para o desenvolvimento *Mobile*.

O *React Native* surgiu para suprir algumas necessidades no desenvolvimento, como os dispositivos móveis interpretam apenas código nativo (*Java*, *Kotlin*, *Objective-C* e *Swift*), para que um aplicativo rodasse tanto em

dispositivos *Android* quanto em dispositivos *IOS (Apple)*, era necessário desenvolver duas bases de código, cada uma direcionada para um sistema operacional específico. O *React Native* é totalmente desenvolvido em *JavaScript*, e nos dias atuais os dispositivos móveis são capazes de entender essa linguagem, então com apenas uma base de código é possível desenvolver *apps* para *Android* e *IOS (Apple)*.

O *React Native* possui diversas funcionalidades muito interessantes que aumentam a praticidade e a produtividade do desenvolvimento. Uma delas é o *Hot Reloading*, que faz com que o programa fique rodando em desenvolvimento, e a cada atualização no código uma versão nova do arquivo modificado é injetado na aplicação, levando menos de 1 segundo para atualizar (mesmo com aplicações grandes e complexas). Para o desenvolvimento *mobile* isso é um grande passo, pois em outros *frameworks* nativos, a cada mudança no código, a aplicação precisava ser recompilada por completo, levando muito mais tempo (BECKER,2020).

### **2.5.5 Banco de dados MySQL**

Para Date (2004, p.6), podemos definir um banco de dados como “... um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem informações quando solicitado”, o autor ainda faz uma analogia comparando um banco de dados com o equivalente a um armário eletrônico de arquivamento, ou seja, um repositório para arquivos de dados eletrônicos.

O *SGBD* (Servidor e Gerenciador de Banco de Dados) escolhido para o projeto do artigo em questão foi o *MySQL*, devido a vários fatores, entre eles a vantagem de sua licença de uso ser de código aberto, sua confiabilidade e sua capacidade de processamento sendo bem similar às de versões de *SGBDs* pagos. “Por possuir essas características, a ferramenta é indicada para uso em aplicações em todas a áreas de negócio, independentemente do tamanho da sua aplicação” (MILANI, 2007, p.24).

## **3. METODOLOGIA**

### **3.1 Unidade de Análise**

As instituições escolhidas para o desenvolvimento do *Software*, *AutoSystem* foram as autoescolas da cidade de Manhuaçu. Nenhuma das autoescolas conta com um sistema apropriado para a execução de suas atividades básicas e seu atendimento ao público. Todos os cadastros, consultas e alterações são feitas manualmente, o que gera um grande atraso no atendimento, e em algumas ocasiões gera falhas durante os processos, o que

pode acarretar a perda de informações ou até mesmo a inconsistência das mesmas.

### **3.2 Tipo de Pesquisa**

O tipo de pesquisa utilizado foi a pesquisa exploratória, que de acordo com GIL (2008, p.27), é caracterizado por ter como objetivo principal “[...] desenvolver, esclarecer e modificar conceitos e ideias, tendo em vista a formulação de problemas mais precisos ou hipóteses pesquisáveis para estudos posteriores.” O presente trabalho tem como objetivo o desenvolvimento de um sistema que contém funcionalidades que tornarão a rotina do seguimento de autoescolas mais simples e eficiente.

Para isso, a técnica aplicada foi a de estudo de campo, que para GIL (2008, p.57)

[...] os estudos de campo procuram muito mais o aprofundamento das questões propostas do que a distribuição das características da população segundo determinadas variáveis. Como consequência, o planejamento do estudo de campo apresenta muito maior flexibilidade, podendo ocorrer mesmo que seus objetivos sejam reformulados ao longo do processo de pesquisa. [...] no estudo de campo estuda-se um único grupo ou comunidade em termos de sua estrutura social, ou seja, ressaltando a interação de seus componentes. Assim, o estudo de campo tende a utilizar muito mais técnicas de observação do que de interrogação.

Após o levantamento dos dados foi decidido desenvolver um sistema que inicialmente deixasse as atividades mais ágeis para de certa forma melhorar o atendimento das autoescolas.

### **3.3 Caracterização da amostra**

A partir dos dados coletados, foi possível identificar que a grande maioria das pessoas que foram questionadas sentem falta de mais organização dos dados por parte da autoescola, além de preferirem a ideia de utilizar um aplicativo como agenda de aulas, fugindo da forma com que é feito atualmente.

### **3.4 Coleta de Dados**

Os dados foram coletados através de algumas conversas informais com alguns colaboradores de autoescolas, bem como a análise dos formulários de cadastros de alunos e também através de conversas informais com alguns alunos e ex-alunos das autoescolas locais.

## **4. RESULTADOS E DISCUSSÕES**

### **4.1 Diagramas da UML**

Após a análise dos dados e dos requisitos, foram desenvolvidos os diagramas da UML, para compor a modelagem de dados.

#### **4.1.1 Diagrama de Caso de Uso**

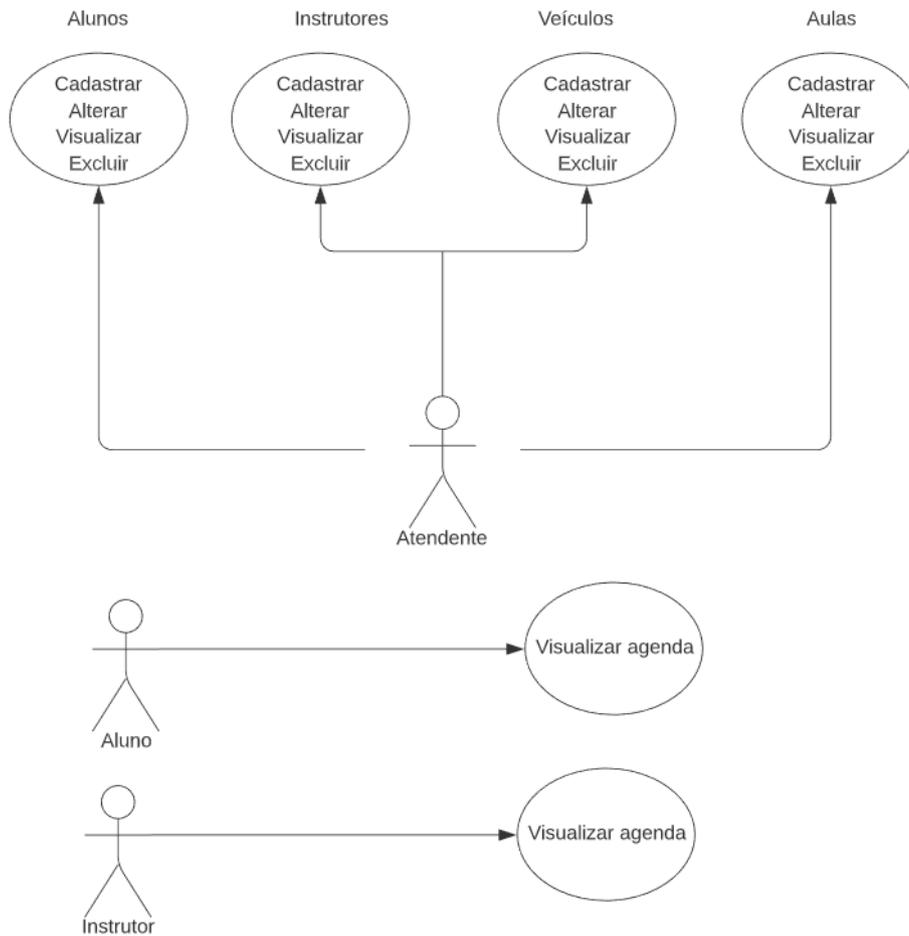
Esse diagrama documenta as ações que são executadas no sistema pelo ponto de vista do usuário. Ou seja, ele descreve as funcionalidades do sistema e a forma com que os usuários interagem com essas funcionalidades.

Neste diagrama descreveremos o ponto de vista dos atores (atendente da autoescola e aluno), e apontaremos a parte funcional do sistema.

O ator atendente poderá cadastrar um aluno e manter seu respectivo cadastro, cadastrar e manter os veículos presentes na autoescola, cadastrar os instrutores que são colaboradores da autoescola e também realizar o agendamento e a manutenção de aulas práticas respectivas a cada aluno.

Os atores aluno e instrutor poderão acessar suas respectivas agendas de aulas através do *app mobile*, não podendo adicionar, alterar ou deletar nenhuma informação, como exemplificado na figura 1.

FIGURA 1: Diagrama de Caso de Uso



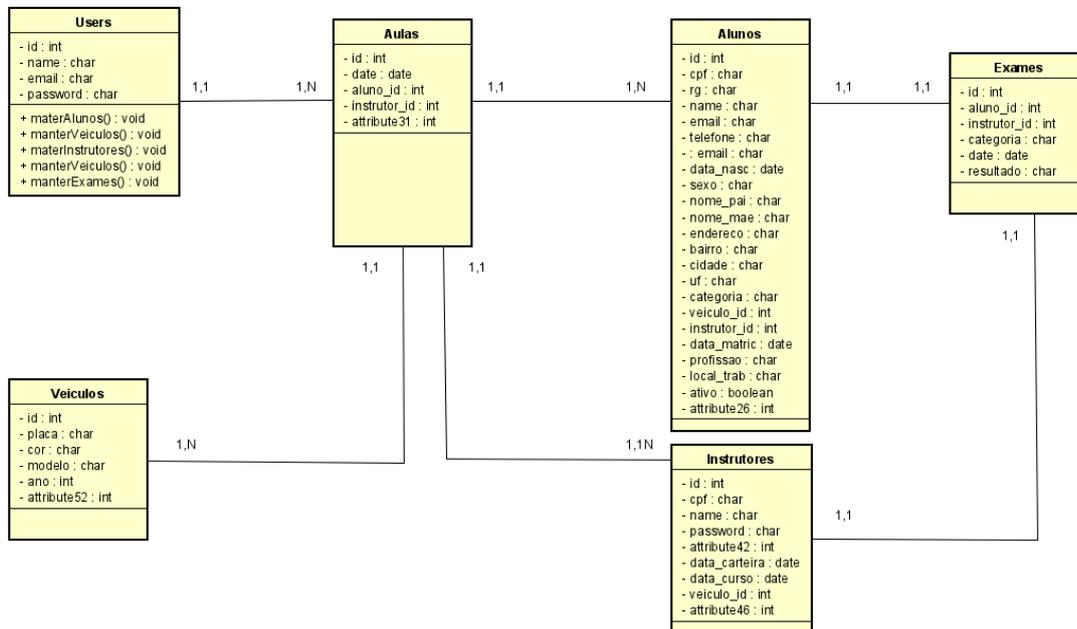
Fonte: Acervo Pessoal

#### 4.1.2 Diagrama de Classes

Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes (TYBEL, 2016).

Nesse diagrama apresentamos a relação entre as classes e os atributos de cada uma dessas classes, como exemplificado na figura 2.

FIGURA 2: Diagrama de Classes



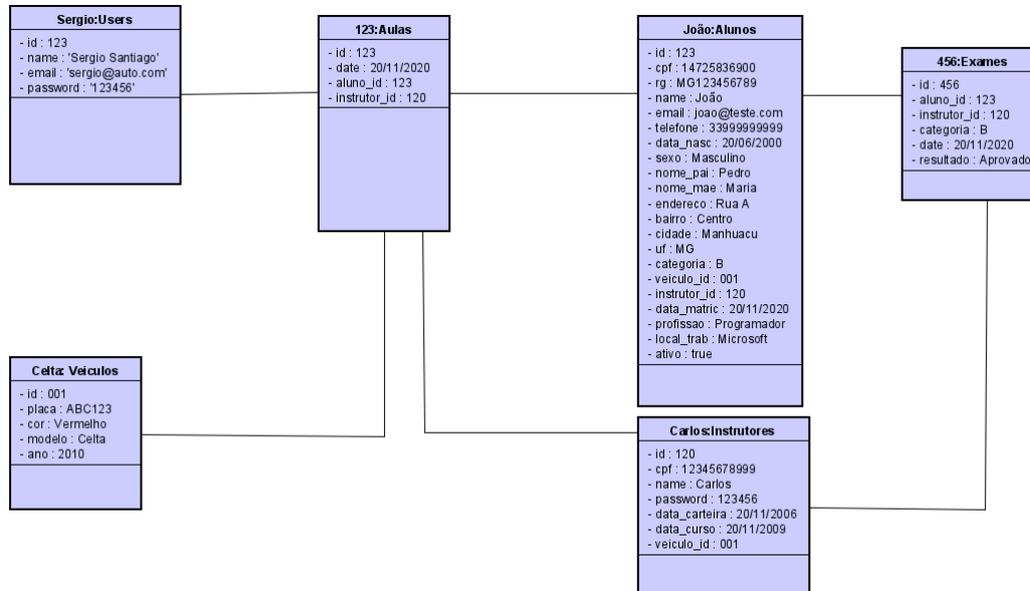
Fonte: Acervo pessoal

### 4.1.3 Diagrama de Objetos

O diagrama de objetos pode ser considerado como uma variação do diagrama de classes, porém o diagrama de objetos mostra as classes já instanciadas. O diagrama de objetos atua semelhantemente a um perfil de usuário que atua dentro do sistema e se relaciona com os demais objetos.

O diagrama de objetos nos permite ter uma visão geral do conjunto de instâncias que existem em um determinado momento da execução de um sistema, ou seja, o diagrama de objetos pode ser considerado como uma cópia de instâncias das classes. (GUERRA, 2015)

FIGURA 3: Diagrama de Objetos



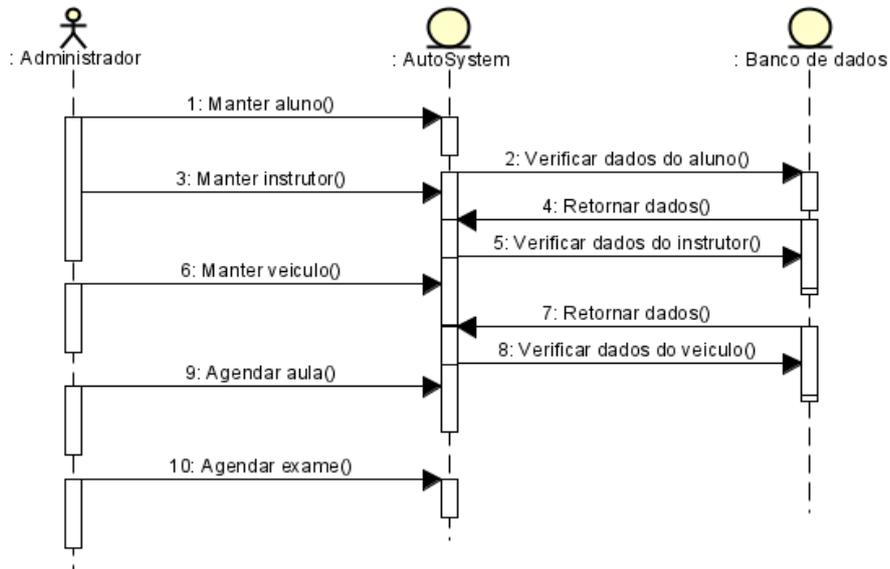
Fonte: Acervo pessoal

#### 4.1.4 Diagrama de Sequência

O diagrama de sequência preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em determinado processo, ou seja, quais condições devem ser satisfeitas e quais métodos devem ser disparados entre os objetos envolvidos e em que ordem durante um processo (SILVA, 2009).

O diagrama de sequência normalmente se identifica com um determinado caso de uso, porque um caso de uso, em geral, refere-se a uma operação específica disparada por um usuário. Desta forma, o diagrama de sequência também permite documentar um caso de uso (SILVA, 2009).

FIGURA 4: *Diagrama de Sequência*



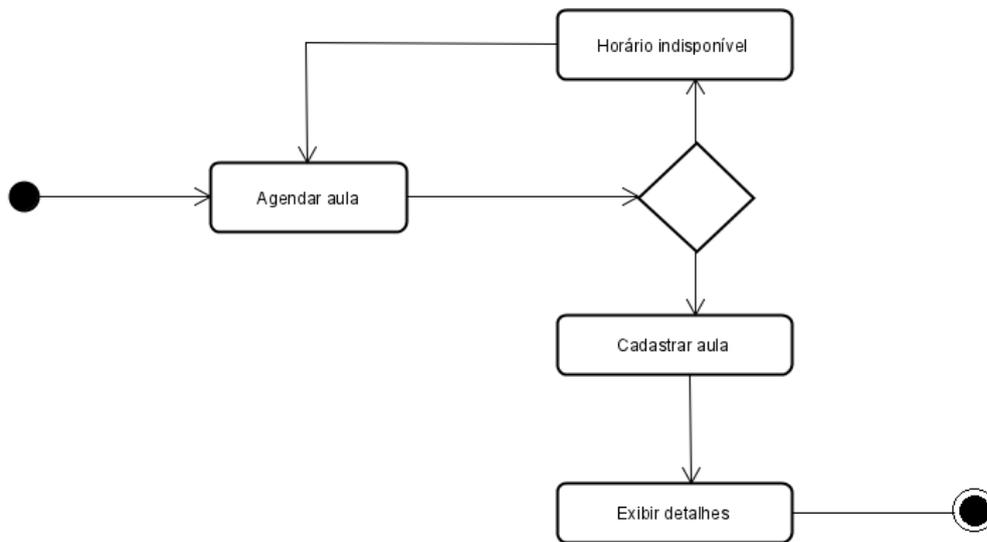
Fonte: Acervo pessoal

#### 4.1.5 Diagrama de Atividades

O diagrama de atividades é considerado um diagrama de comportamento, porque ele é capaz de descrever quais ações devem acontecer no sistema, agindo de forma semelhante aos diagramas de caso de uso e de máquina de estado.

O Diagrama de atividades mostra as atividades do usuário com acesso ao sistema. O usuário administrador pode cadastrar, alterar, excluir alunos do sistema, além de poder também, cadastrar, alterar e excluir instrutores e veículos, podendo agendar aulas e tem acesso a agenda de aulas práticas dos alunos podendo alterar a data ou excluir registros de aulas. Como é mostrado na figura 5, o usuário administrador pode tentar realizar o agendamento de uma aula, caso horário esteja disponível a aula é agendada, caso contrário ele deve remarcar em um horário válido.

FIGURA 5: *Diagrama de Atividades*



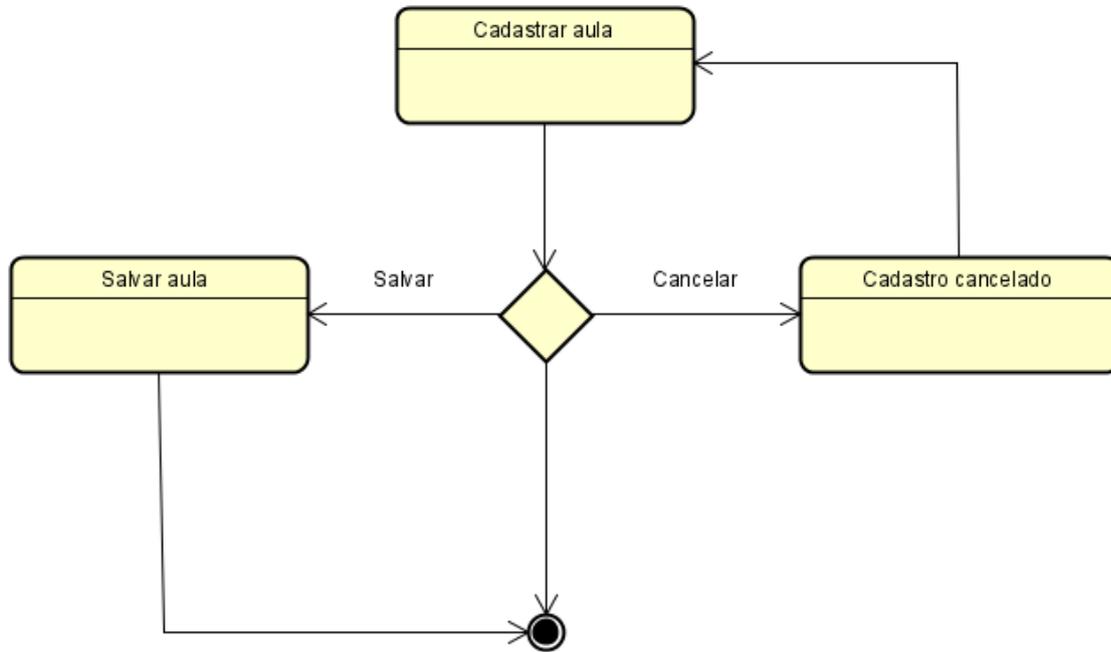
Fonte: Acervo pessoal

#### 4.1.6 Diagrama de Máquina de Estado

Através de sua simbologia gráfica, ele procura demonstrar o comportamento de um elemento por meio de transições de estado. O elemento modelado muitas vezes é uma instância de uma classe, no entanto, pode se usar esse diagrama para modelar o comportamento de um caso de uso, o comportamento de um dado durante uma transação ou mesmo o comportamento de um sistema completo – neste caso estaremos considerando o caso de uso ou o sistema como objetos (SILVA, 2009).

Diagramas de estados retratam principalmente estados e transições. Estados são representados por retângulos com cantos arredondados e rotulados com o nome do estado. As transições são marcadas com setas que fluem de um estado para outro, mostrando como os estados mudam. (Luci Software, 2019).

FIGURA 6: Diagrama de Máquina de estado



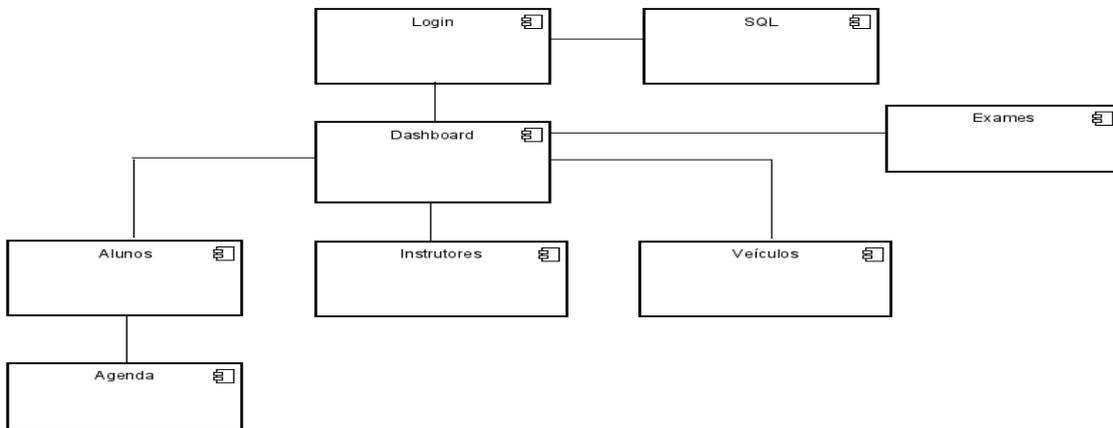
Fonte: Acervo pessoal

#### 4.1.7 Diagrama de Componentes

Na UML, os diagramas de componentes mostram a estrutura do sistema de *software*, que descreve os componentes do *software*, suas interfaces e suas dependências. É possível utilizar diagramas de componentes para modelar sistemas de *software* em um alto nível ou para mostrar componentes em um nível de pacote mais baixo.

Esse tipo de diagrama suporta o desenvolvimento com base em componentes no qual um sistema de *software* é dividido em componentes e interfaces que são reutilizáveis e substituíveis (IBM, 2019).

FIGURA 7: Diagrama de Componentes



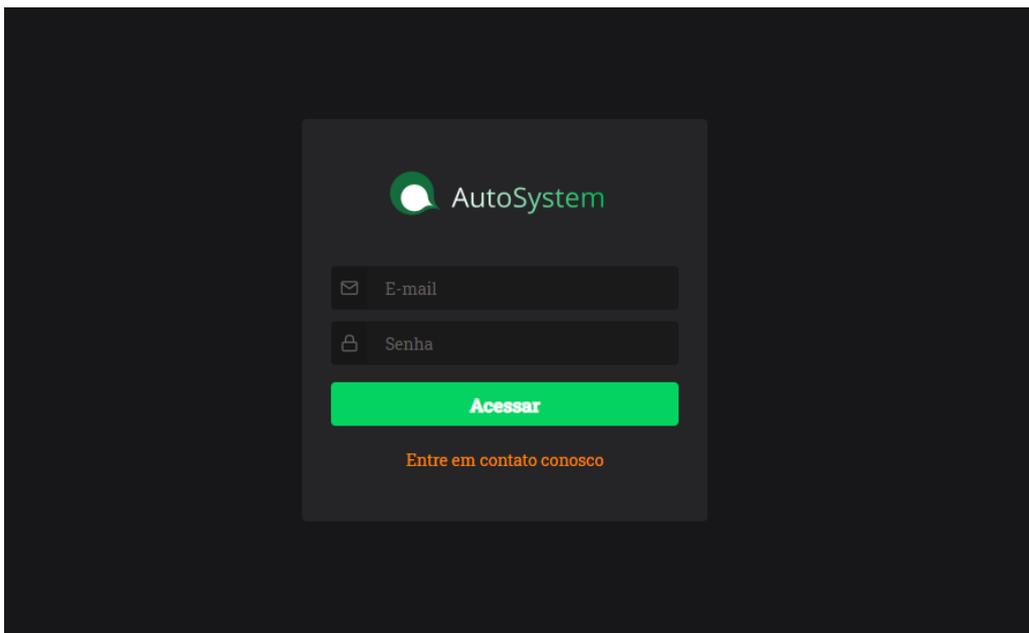
Fonte: Acervo pessoal

## 4.2 Visão Geral do AutoSystem

### 4.2.1 Mini Mundo

Os colaboradores responsáveis pelas atividades de cadastro e manutenção dos registros, deverão realizar *login* no sistema utilizando suas credenciais de autenticação. Caso as credenciais estejam incorretas, o usuário receberá uma mensagem com o erro ocorrido, caso as credenciais estejam corretas, o usuário terá acesso a *Dashboard* do sistema.

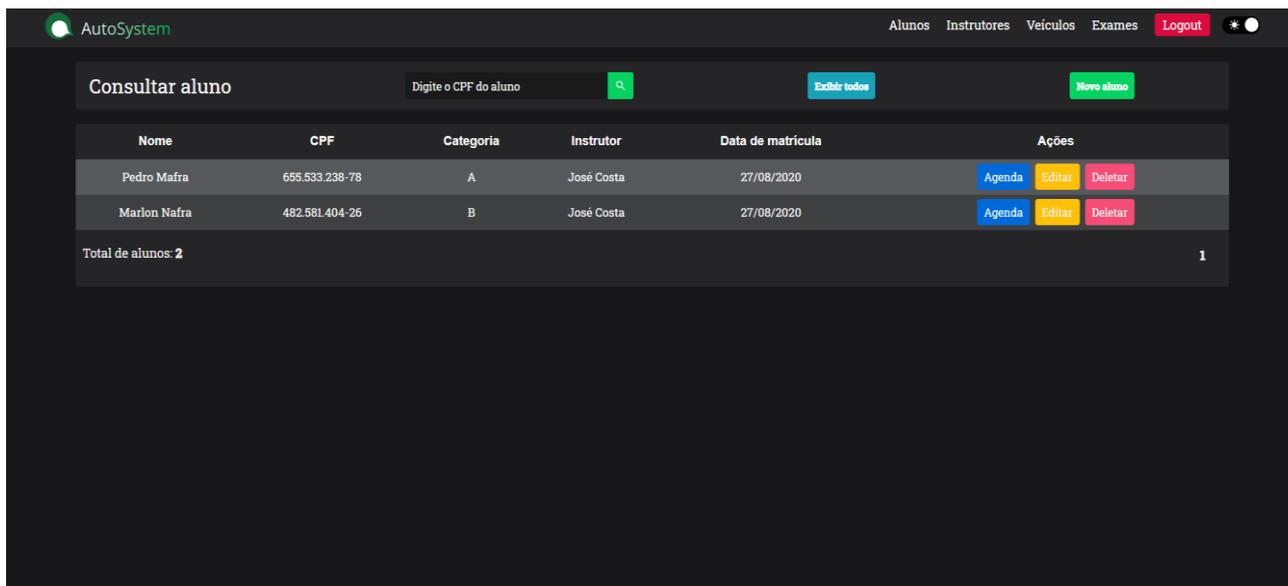
FIGURA 8: Tela de login



Fonte: Acervo Pessoal

Após a realização do *login*, o usuário terá acesso ao *dashboard* do sistema, onde a partir dessa tela, terá acesso a todas as outras telas do sistema, Como as telas de cadastro de alunos, instrutores, veículos, cadastro de aulas e agendamento de exames.

FIGURA 9: Tela de Dashboard



The screenshot shows the 'AutoSystem' dashboard. At the top, there are navigation links for 'Alunos', 'Instrutores', 'Veículos', 'Exames', and 'Logout'. Below this is a search bar for 'Consultar aluno' with a search icon and a 'Novo aluno' button. The main content is a table with the following data:

Nome	CPF	Categoria	Instrutor	Data de matrícula	Ações
Pedro Mafra	655.533.238-78	A	José Costa	27/08/2020	Agenda Editar Deletar
Marlon Nafra	482.581.404-26	B	José Costa	27/08/2020	Agenda Editar Deletar
Total de alunos: 2					1

Fonte: Acervo Pessoal

Todos os aluno e instrutores cadastrados no sistema *Web*, tem acesso a um aplicativo para que possa acompanhar de maneira mais prática sua agenda de aulas, sendo possível apenas a visualização das informações, ou seja tanto alunos quanto instrutores não podem alterar os dados que são exibidos no aplicativo.

Para ter acesso ao aplicativo existe uma tela de *login* onde o usuário escolhe se seu perfil é aluno ou instrutor, caso o perfil do usuário seja aluno, ele irá visualizar um campo onde digitará seu CPF, após isso caso esteja cadastrado no aplicativo ele será encaminhado para a *dashboard*, onde terá acesso direto a sua agenda de aulas práticas. Já o usuário com o perfil de instrutor, deverá digitar seu CPF e além disso por questões de segurança deverá inserir sua senha de acesso, considerando que sua agenda possui dados sensíveis referente aos alunos. após a autenticação ele será direcionado para a *dashboard*, onde terá acesso a sua agenda de aulas, do dia, da semana e do mês de forma completa. As Figuras 10 e 11 demonstram as telas de *login* em ambos os perfis de usuário.

Figura 10: Tela de login do aluno



Fonte: Acervo Pessoal

Figura 11: Tela de login do instrutor



Fonte: Acervo Pessoal

## 5. Conclusão

De acordo com os resultados das pesquisas podemos concluir que o sistema proposto será muito válido para agilizar os atendimentos, evitar falhas e proporcionando uma experiência mais agradável tanto para os atendentes de autoescolas que poderão realizar seu trabalho de uma forma mais simples e eficaz, quanto para os alunos e instrutores que terão acesso contínuo a sua agenda de aulas, o que torna a comunicação entre as partes mais eficiente.

O *software* foi desenvolvido para funcionar como um *WebApp* para a administração dos dados e um aplicativo mobile para a visualização dos dados. O objetivo principal do *software* é resolver os problemas referentes ao armazenamento, manutenção dos dados, e a comunicação entre as autoescolas e seus alunos. Através do *Software* AutoSystem será muito mais simples realizar

os processos internos, gerando mais agilidade nos atendimentos e consequentemente gerando mais valor para ambas as partes.

## 6. Referências

ANDREI. **O que é React e Como Funciona.** Disponível em: < <https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript/>>. Acesso em: 25 nov 2020.

BACKER, Lauro. **O que é React Native.** Disponível em: < <https://www.organicadigital.com/blog/o-que-e-react-native/>>. Acesso em: 26 nov 2020.

BEAL, ADRIANA. **O sistema de informação como estratégia empresarial.** São Paulo: Atlas, 2009.

CASTRO, Bruna Amaral. **Automatização de processos ou automação de processos.** Rio dos sinos, 2016. Disponível em: < <https://blog.smlbrasil.com.br/automatizacao-de-processos/>> Acesso em: 22 nov 2020.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social.** 6. ed. São Paulo. Atlas, 2008.

GUERRA, Patrick. **Diagramas estruturais da UML: Diagrama de objetos.** Disponível em: <https://micreiros.com/diagrama-de-objetos/>. Acesso em: 30/11/2020.

FERNANDES, Diego. **NodeJS: Vale a pena? Vantagens, vagas e salário.** Rocketseat. 2017. Disponível em: < <https://blog.rocketseat.com.br/nodejs-vale-a-pena-vantagens/>>. Acesso em: 25 nov 2020.

MILANI, André. **MySQL-guia do programador.** Novatec Editora, 2007.

O'BRIEN, J. A. **Sistemas de informação: e as decisões gerenciais na era da Internet.** 2. ed. São Paulo: Saraiva 2004.

OHIRA, Lilian Marques. **Identificação de requisitos para usabilidade de software assistivo.** 2009.

OLIVEIRA, Aline Carrijo de, **Dicionário escolar: língua portuguesa,** Blumenau, Vales das Letras, 2017

OLIVEIRA, Wallace. **O que é processo empresarial.** Venki, 2014. Disponível em: < <https://www.venki.com.br/blog/o-que-e-processo-empresarial/> > Acesso em: 20 nov 2020.

O que é diagrama de atividades UML? **Lucidchart**, c2019. Disponível em: Acesso em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-atividades-uml>>. 30 nov 2020.

O que é um diagrama de máquina de estados? **Lucidchart**, c2019. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-maquina-de-estados-uml>> Acesso em 28 nov 2020.

RIBEIRO, Marcus. **Automação de processos sem complicação**. Disponível em: <<https://pluga.co/blog/gestao-empresarial/automacao-de-processos/>> Acesso em: 20 nov 2020.

SILVA, Maurício Samy. **JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript**. 2010.

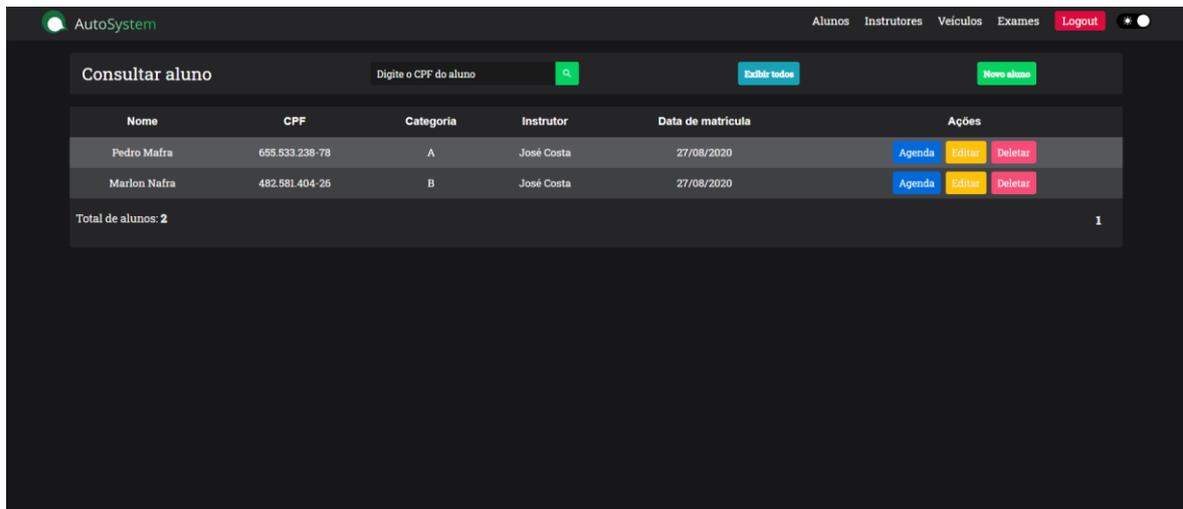
SILVA, Paulo César Barreto da. Diagrama de sequência - Utilizando UML. **DevMedia**, 2009. Disponível em: <<https://www.devmedia.com.br/artigo-sql-magazine-64-utilizando-uml/12665>> Acesso em 28 nov 2020.

TYBEL, Douglas. Orientações básicas na elaboração de um diagrama de classes. **DevMedia**, 2016. Disponível em: <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>> Acesso em 28 nov 2020

## Apêndice A:

Algumas telas que compõe o sistema AutoSystem.

FIGURA 12: *Dashboard*



AutoSystem

Alunos Instrutores Veículos Exames Logout

Consultar aluno

Digite o CPF do aluno

Exibir todos Novo aluno

Nome	CPF	Categoria	Instrutor	Data de matricula	Ações
Pedro Mafra	655.533.238-78	A	José Costa	27/08/2020	Agenda Editar Deletar
Marlon Nafra	482.581.404-26	B	José Costa	27/08/2020	Agenda Editar Deletar

Total de alunos: 2

1

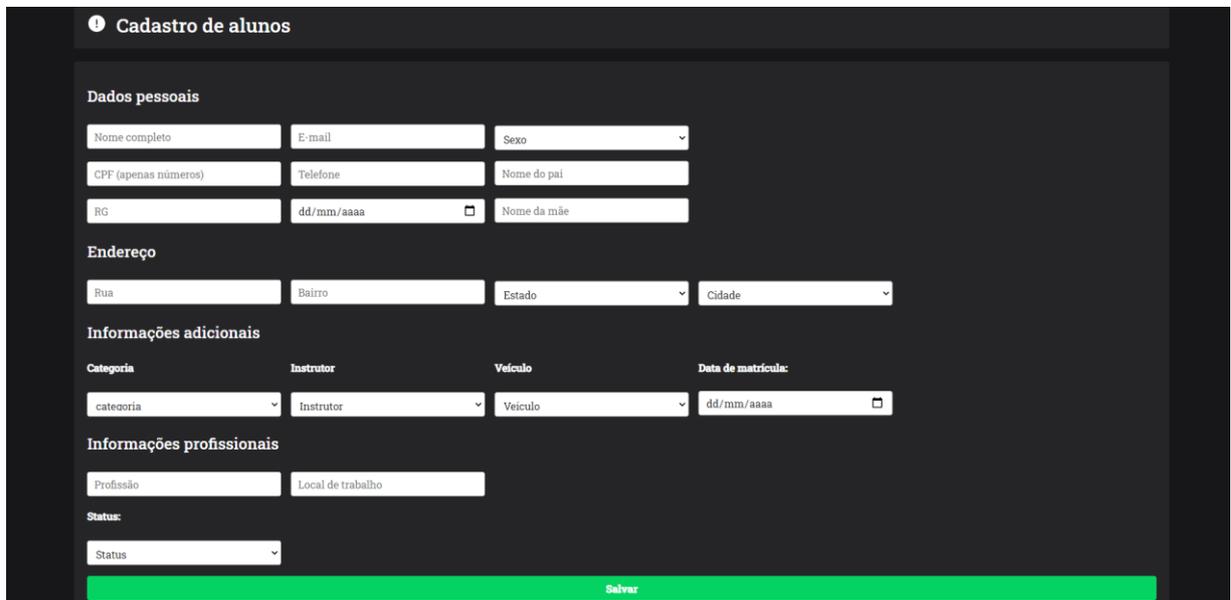
Fonte: Acervo Pessoal

Na tela “*Dashboard*” (figura 12) será mostrada uma tabela com o registro de todos os alunos, em cada cadastro é possível realizar edição do aluno, deleção do cadastro e acessar a agenda de aulas que é mostrada na figura 16.

Na “*Dashboard*” também é possível pesquisar por um aluno específico através do seu CPF, e acessar a tela de cadastro de alunos (figura 13).

Todas as telas do sistema têm acesso a um *header* onde é possível navegar entre as telas, realizar *logout* da aplicação e alternar o tema entre claro ou escuro.

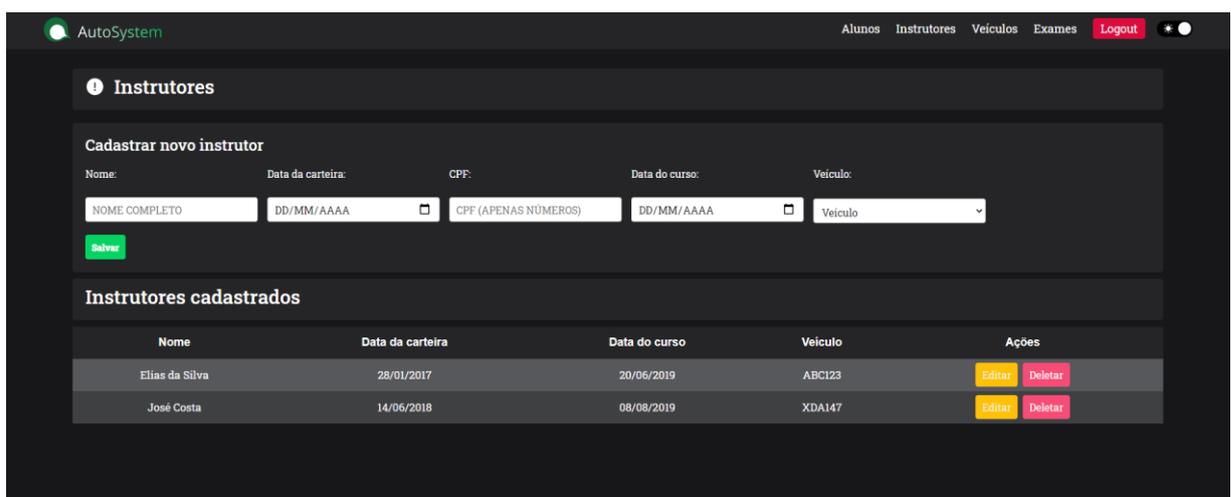
FIGURA 13: Cadastro de alunos



Fonte: Acervo Pessoal

Na figura 13 é demonstrada a tela de cadastro de aluno, onde o usuário administrador deverá preencher os campos com as informações do aluno e posteriormente clicar no botão de salvar.

FIGURA 14: Tela de instrutores



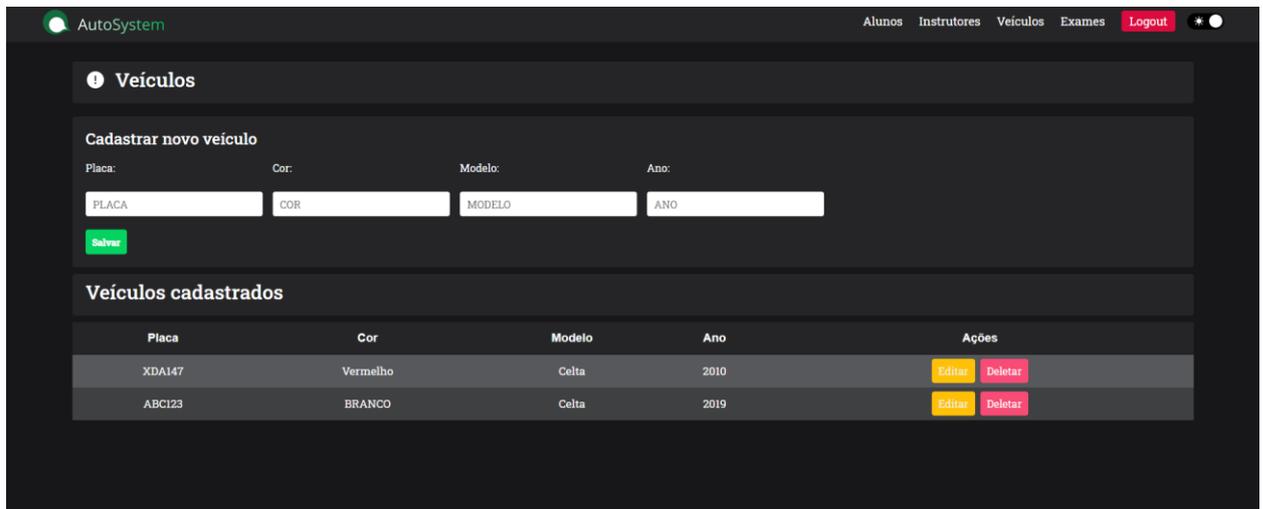
Nome	Data da carteira	Data do curso	Veículo	Ações
Elias da Silva	28/01/2017	20/06/2019	ABC123	<a href="#">Editar</a> <a href="#">Deletar</a>
José Costa	14/06/2018	08/08/2019	XDA147	<a href="#">Editar</a> <a href="#">Deletar</a>

Fonte: Acervo Pessoal

Na tela de Instrutores, o usuário terá acesso a uma tabela com todos os instrutores cadastrados no sistema, podendo editar e deletar algum registro

específico, nessa tela também é possível realizar um novo cadastro, preenchendo os campos e clicando em salvar.

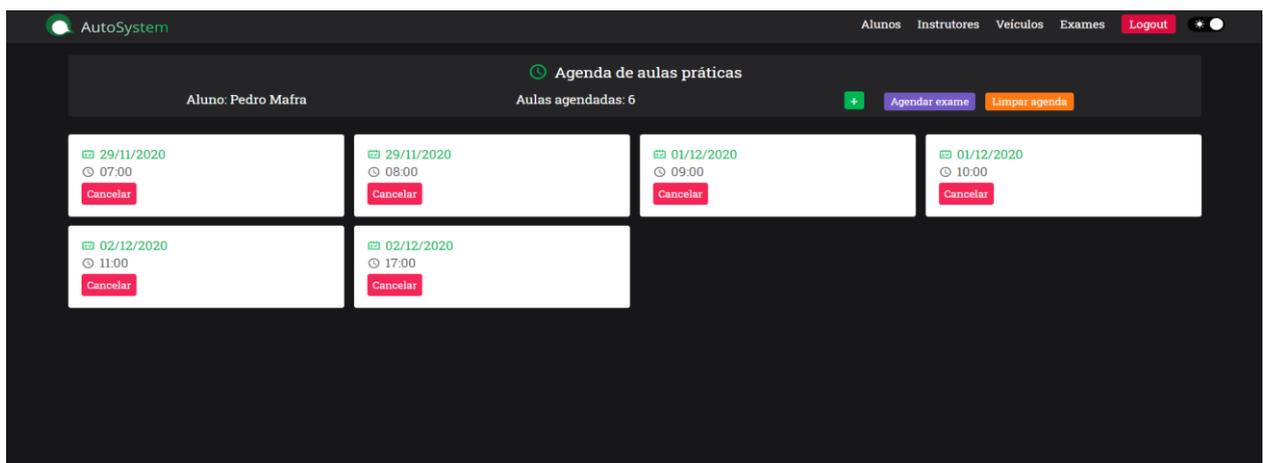
FIGURA 15: Tela de veículos



Fonte: Acervo Pessoal

Na tela de veículos, é apresentada uma tabela que exibe todos os veículos cadastrados no sistema, permitindo ao usuário administrador editar ou deletar esses registros, a tela de veículos também permite que um novo cadastro seja criado, preenchendo os campos e clicando em salvar.

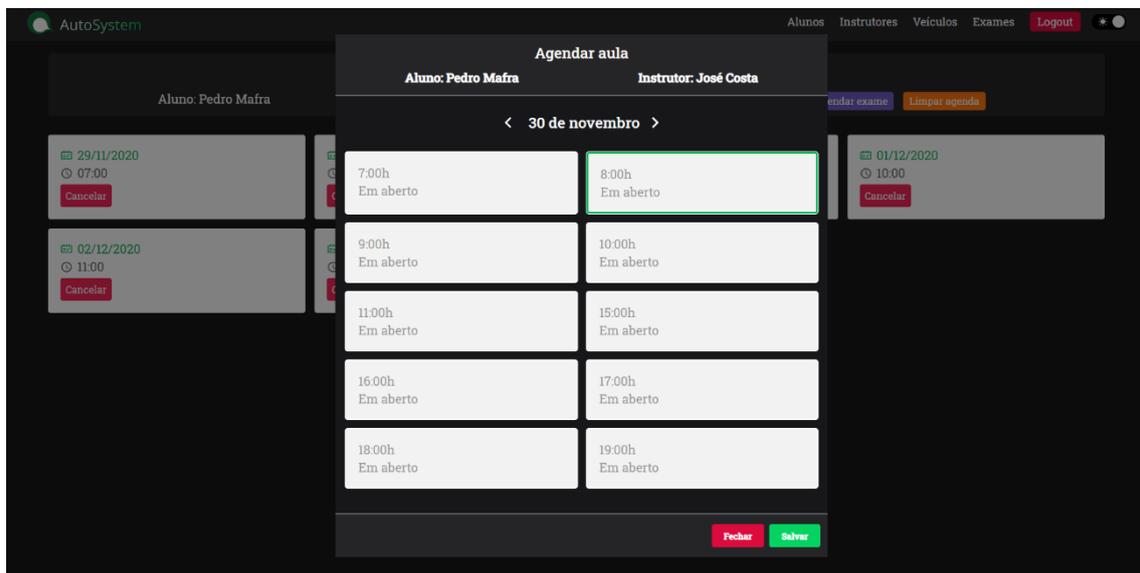
FIGURA 16: Agenda



Fonte: Acervo Pessoal

Através da tela “Dashboard” o administrador tem acesso a agenda de aulas práticas dos alunos, na tela de Agenda é possível cancelar uma aula agendada, realizar o agendamento de novas aulas e agendar um exame caso o aluno tenha 20 aulas práticas concluídas.

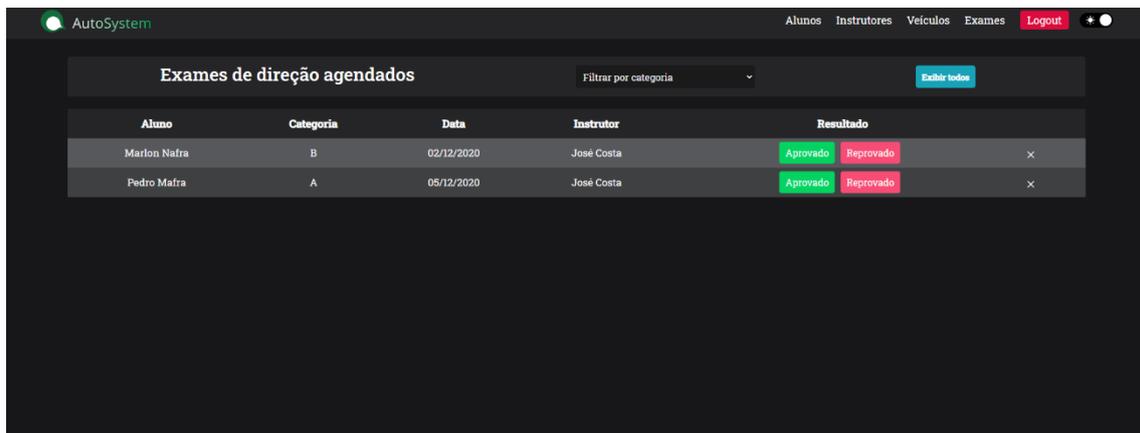
FIGURA 17: *Agendar aula prática*



Fonte: Acervo Pessoal

Na tela de Agenda, é possível agendar novas aulas práticas para o aluno clicando no botão “+”, após o clique uma nova janela será aberta com um calendário que mostra quais horários e dias estão disponíveis, permitindo ao administrador agendar aulas apenas em horários que estejam livres.

FIGURA 18: *Tela de exames*



Fonte: Acervo Pessoal

A tela de exames, permite ao administrador visualizar quais alunos estão com exames em aberto. A tela permite que os exames sejam filtrados por categoria e permite que após o exame ser realizado o resultado seja registrado no sistema.

FIGURA 19: Tela de Dashboard do aluno no aplicativo mobile



Fonte: Acervo Pessoal

A Figura 19 mostra a tela de *Dashboard* do aplicativo *mobile* do usuário que utiliza o perfil de aluno, é possível visualizar a quantidade de aulas agendadas, finalizadas e quantas restam para serem realizadas, o usuário também pode ver sua agenda com os dias e horários de suas respectivas aulas.

FIGURA 20 e FIGURA 21: Tela de *Dashboard* do instrutor no aplicativo mobile



Fonte: Acervo Pessoal



Fonte: Acervo Pessoal

As figuras 20 e 21 representam a tela de *Dashboard* do usuário no aplicativo *mobile* com o perfil de instrutor, onde é possível visualizar a agenda completa de aulas do dia, da semana ou do mês. O instrutor poderá ver a data, horário e o aluno que terá aula no respectivo dia, além de automaticamente ter o *status* de suas aulas alterado conforme o horário.