Desenvolvimento do *Software* LocalAnalitica: Auxiliar na análise e gerenciamento de sistemas de locações de automóveis

Amauri de Souza Teixeira, André Constantino da Silva

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP)
Campus Hortolândia – SP – Brasil

amauri.souza@aluno.ifsp.edu.br, andre.constantino@ifsp.edu.br

***Abstract**: With the evolution of the car rental systems and the great generation of data due to the growth and demands for rentals, difficulties are encountered by managers to analyze and manage this information. Motivated by the importance of a system to help in this aspect, this work created a software for displaying graphics based on data from a database of a rental company. Using Information Visualization concepts and HTML5, CSS3, React.js, APEXCHARTS.JS technologies, a software for the Web platform was built so it can be more easily changed to get data from any rental system database.*

***Resumo**: Com a evolução dos sistemas de locações de automóveis e a grande geração de dados com o crescimento e as demandas por locações, dificuldades são encontradas por gerentes para análise e gerenciamento dessas informações. Motivado pela importância de um sistema para auxiliar nesse aspecto, este trabalho criou um software para exibição de gráficos baseado em dados oriundos de uma base de dados de uma locadora. Empregando conceitos de Visualização de Informações e usando as tecnologias HTML5, CSS3, React.js, APEXCHARTS.JS foi construído um software para plataforma Web, de modo que possa ser mais facilmente alterado para obter dados de um banco de dados MySQL de qualquer sistema de locação.*

1. Introdução

De acordo com a Associação Brasileira das Locadoras de Automóveis (ABLA, 2020), a partir dos dados coletados do Cadastro Geral de Empregados e Desempregados (CAGED) e do Relatório Anual de Informações Sociais (RAIS), as empresas de aluguel de veículos empregam em torno de 80.000 mil pessoas no país. Na Figura 1, podem ser observados os números de empregados direto por estado.

Segundo Levi Fonseca (2017 *apud* Figo, 2017), "As pessoas precisam do carro para uma série de coisas, mas não necessariamente necessitam ter a posse do bem. Há um custo alto envolvido na compra e na manutenção de um veículo, por isso elas estão dispostas a pagar uma mensalidade pela comodidade". Assim, muitas pessoas optam por alugar um carro, ao invés de adquirir um. Além disso, pessoas optam por alugar um carro onde ele deseja ter uma melhor locomoção, como em casos de viagens. Ressalta-se que, ao alugar um carro, também é possível escolher serviços e equipamentos opcionais, como cadeiras para crianças.

Com as mudanças e escolhas dos consumidores, isso gera uma grande quantidade de dados que causam dificuldades no gerenciamento por parte do administrador. Podemos apontar, como exemplo, os modelos de automóveis mais locados; quais são os públicos e suas características como a idade e o gênero; locações realizadas em determinado período.

Posição	Estado	Número de empregos
1°	São Paulo	21.692
2°	Minas Gerais	7.657
3°	Rio de Janeiro	7.025
4°	Bahia	6.239
5°	Pernambuco	5.537
6°	Ceará	4.121
7°	Paraná	3.502
8°	Goiás	2.485
9°	Rio Grande do Sul	2.460
10°	Pará	2.277
11°	Distrito Federal	2.076
12°	Rio Grande do Norte	2.064
13°	Amazonas	1.625
14°	Maranhão	1.595
15°	Sergipe	1.370
16°	Mato Grosso do Sul	1.309
17°	Espírito Santo	1.273
18°	Santa Catarina	1.192
19°	Alagoas	1.176
20°	Piauí	974
21°	Mato Grosso	848
22°	Paraíba	658
23°	Rondônia	441
24°	Tocantins	367
25°	Amapá	227
26°	Roraima	110
27°	Acre	78
TOTAL	*BRASIL*	*80.378*

Fontes: ABLA, CAGED, RAIS.

Figura 1. Ranking de Empregos Diretos - Ano base 2017. Fonte: ABLA (2017)

Por conseguinte, o mercado de locações de automóveis, e dentre outros relacionados a disponibilização de produtos e serviços, exige um planejamento estratégico da empresa, onde fica evidente a importância da utilização de um sistema que disponibiliza uma visão analítica das informações, métricas, contadores e gráficos, que indicam ao administrador do estabelecimento o estado atual do seu produto/serviço, o seu desenvolvimento e crescimento.

O gerenciamento das informações que se refere ao contexto analítico tem como objetivo auxiliar o entendimento do administrador em relação ao seu produto ou serviço, visto como os consumidores os enxergam e o como ele se comporta no mercado. Tendo em vista problemas no gerenciamento de informações, de como elas são dispostas para os responsáveis e a falta de aplicação de técnicas da área de Visualização de Informações, é visto como um problema exigindo soluções para ele.

Segundo (NASCIMENTO; FERREIRA, 2011) a construção de uma visualização que possibilite uma observação de informações pode ser alcançada organizando os dados segundo algum critério e apresentando-os de modo visual. E, como afirma Mozzato e Grzybovski (2011), “percebe-se que a análise de conteúdo é um conjunto de técnicas de análise de comunicações, que tem como objetivo ultrapassar as incertezas e enriquecer a leitura dos dados coletados”.

O objetivo deste trabalho é o desenvolvimento de uma aplicação para proporcionar uma visão analítica das informações e uma melhor análise realizada pelo administrador do estabelecimento de locação de automóveis. Os dados de entrada do sistema são obtidos da aplicação utilizada pelo estabelecimento de locação e, após processo de tratamento, exposto ao usuário, como propõe a área de Visualização de Informações que, de acordo com (NASCIMENTO; FERREIRA, 2011), “As técnicas de visualização de informações também podem ser utilizadas para filtrar, analisar e gerenciar grandes quantidades de dados, como as informações que são geradas e disponibilizadas diariamente na Internet.”. Onde se é feito um estudo para entender as melhores técnicas para facilitar o compreendimento das informações por forma de representações visuais de dados (NASCIMENTO; FERREIRA, 2011).

Assim, acredita-se que a aplicação proposta permite uma maior facilidade e qualidade de análise com a utilização de gráficos sobre locações, valores, impressões, públicos,

interações, visitas e dentre outras funcionalidades. Desta forma, o gerente terá disposto para ele uma aplicação Web que proporciona uma experiência de monitoramento e análise para melhores estratégias e planos de ações que deseja executar.

O restante deste artigo está dividido da seguinte forma: fundamentação teórica, apresentada na Seção 2; metodologia, apresentada na Seção 3; o desenvolvimento da aplicação na Seção 4; os resultados obtidos na Seção 5; e as conclusões na Seção 6.

2. Fundamentação Teórica

A fundamentação teórica deste trabalho se inicia na explicação de Visualizações de Informações, o que é, utilização, contribuições; e, por fim, sobre desenvolvimento do sistema.

2.1 Visualização de Informações

Com grandes quantidades de informações reunidas e que tenham a necessidade de serem analisadas por algum indivíduo, sendo exibidas de diversas maneiras (como listas, gráficos, *labels,* tabelas e relatórios) e ainda sujeitas a difícil (e consequentemente, pode ocorrer problemas de interpretação), por consequência não deixando claro o que realmente deve ser exibido para aquele determinado contexto. Nesse contexto surge a Visualização de Informações que, segundo (NASCIMENTO; FERREIRA, 2011), tem como objetivo final "auxiliar no entendimento de determinado assunto, o qual, sem uma visualização, exigiria maior esforço para ser compreendido".

Desta forma, as técnicas de visualização podem beneficiar de modo que facilite o reconhecimento e interpretação de informações (NASCIMENTO; FERREIRA, 2011):

> "Resumidamente, as técnicas de visualização de informações procuram representar graficamente dados de um determinado domínio de aplicação de modo que a representação visual gerada explore a capacidade de percepção do homem e este, a partir das relações espaciais exibidas, interprete e compreenda as informações apresentadas e, finalmente, deduza novos conhecimentos." (NASCIMENTO; FERREIRA, 2011).

Para determinar qual técnica a ser utilizada, devemos utilizar um modelo de referência de visualização que, segundo (FREITAS *et al.*, 2001), permitiria a "identificação dos componentes essenciais a serem considerados na utilização de uma determinada técnica ou no desenvolvimento de uma nova".

Na Figura 2 podem ser visualizadas as etapas de um modelo clássico proposto por Haber e McNabb (1990 *apud* FREITAS *et al.,* 2001). O modelo se inicia com a etapa de "Filtragem", onde os dados são filtrados, logo em seguida é realizado um "Mapeamento" para ser gerado uma imagem no passo de "*Rendering*", onde se obtém uma representação visual dos dados/informações.

Figura 2. Representação do modelo clássico de visualização. FONTE: HABER e MCNABB (1990 apud FREITAS *et al.*, 2001)

Na Figura 3, podemos visualizar o modelo sugerido por Card *et al.* (1999), que se tem um modelo com mais elementos e a interações humana em relação a cada etapa.

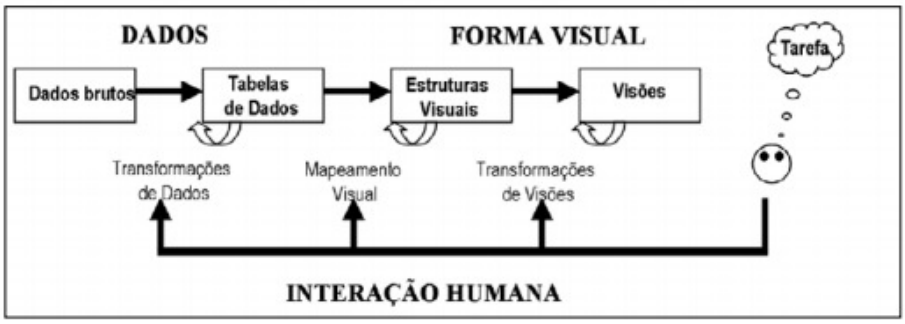

Figura 3. Representação do modelo sugerido de visualização de Card *et al.* (1999)
FONTE: CARD *et al.* (1999 *apud* FREITAS *et al.*, 2001)

O modelo se inicia com a etapa de "Transformações de Dados", que conforme dito por (NASCIMENTO; FERREIRA, 2011), "Um conjunto de dados brutos é processado e organizado em uma representação lógica mais estruturada, geralmente na forma de uma ou mais tabelas". Deste modo, essa etapa pode trabalhar a informação para que ela seja mais clara e menos redundante para ser melhor apresentada.

A segunda etapa do modelo é o "Mapeamento Visual" que, segundo (NASCIMENTO; FERREIRA, 2011), é o momento em que é "construída uma estrutura visual que represente visualmente os dados da tabela".

A terceira, e última etapa, é a de "Transformações de Visões" que, segundo (NASCIMENTO; FERREIRA, 2011), "consiste em associar os itens de dados a marcas visuais em um substrato visual. Cada atributo dos dados pode ser associado a propriedades gráficas das marcas".

Complementando, conforme dito por (VAZ; CARVALHO, 2004), "As informações devem estar claras e simplificadas, não prejudicando ou falhando em uma interpretação, sendo que, em alguns casos uma visualização mal feita pode levar a grandes prejuízos".

2.2 Desenvolvimento de *Software*

O processo para o desenvolvimento de *software* é baseado em modelos do desenvolvimento do produto. Segundo SOMMERVILLE (2019, p. 31), modelo de desenvolvimento de *software* "é uma representação simplificada de um processo de *software*. Cada modelo representa um processo a partir de uma perspectiva particular e, desse modo, fornece apenas informações parciais sobre o processo".

Atualmente, de acordo Sommerville (2019, p. 31), existem três tipos de modelos de desenvolvimento de *software* genéricos, são eles cascata; desenvolvimento incremental; integração e configuração. A escolha de um modelo ideal não existe, isso irá depender do contexto com um todo, para que a aplicação possa ser desenvolvida.

Para este trabalho, foi escolhido o desenvolvimento incremental; atualmente os processos de *software* se baseia no mesmo e o que adapta para proposta da aplicação. De acordo, com o Sommerville (2019, p. 35), "O desenvolvimento incremental se baseia na ideia de desenvolver uma implementação inicial, obter *feedback* dos usuários ou terceiros e fazer o *software* evoluir através de várias versões, até alcançar o sistema necessário". Na Figura 4, é apresentado o diagrama de Desenvolvimento Incremental.

Um desenvolvimento incremental se inicia na "Descrição geral", onde se obtém a ideia geral do *software,* do que se trata a aplicação e, após, este entendimento três tarefas são iniciadas. A etapa seguinte é a "Especificação", onde são tratados todos os detalhes do desenvolvimento. Em seguida, se realiza o "Desenvolvimento", onde é o desenvolvimento da

atividade planejada. A “Validação” trata exclusivamente no sentido de validar as tarefas realizadas e se estão de acordo com a especificação. Conforme, os incrementos são obtidos, versões são geradas para o cliente avaliar as atividades feitas durante o incremento, e a cada incremento podendo conter aprimorações e melhorias para o *software.*

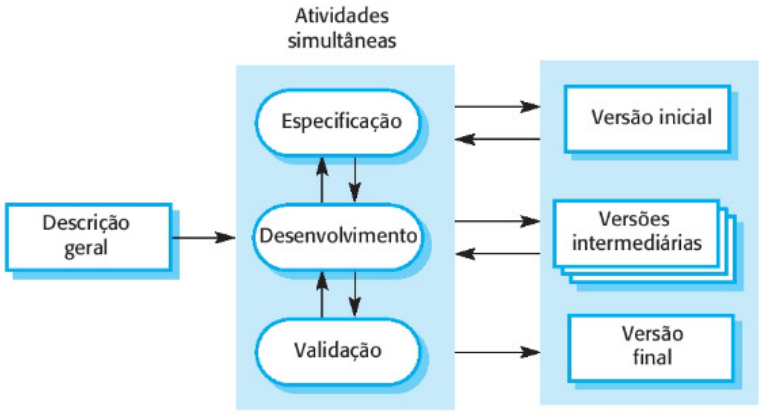

Figura 4. Representação das etapas do desenvolvimento incremental. FONTE: SOMMERVILLE (2019, p. 35)

Metodologias ágeis seguem uma abordagem de desenvolvimento incremental. Como exemplo tem-se o SCRUM que, de acordo CRUZ (2013), “é um *framework* para gerenciamento de projetos ágeis que, apesar de muito utilizado na área de desenvolvimento de *software*, pode ser utilizado para o planejamento, gerenciamento e Desenvolvimento de qualquer produto, principalmente por ser um *framework* iterativo e incremental”.

3. Metodologia

Para obter os resultados esperados relacionados à Visualização de Informações, foram estudados trabalhos acadêmicos e artigos sobre o tema, tendo como ponto de partida a definição encontrada no dicionário sobre visualização: “ato de transformar em imagem mental conceitos abstratos” (VISUALIZAÇÃO, 2020).

Para a implementação da solução dividiu-se o projeto em 3 partes; onde cada incremento foi validado pelo orientador deste trabalho. A primeira parte foi a criação de serviços em Node.js com o objetivo de consumir o banco de dados MySQL do *software* de locação de veículos do administrador. Neste caso, foram construídos 5 serviços que disponibilizam informações sobre automóveis, locações, clientes e faturamento. Um exemplo de serviço é a disponibilização dos modelos de automóveis mais locados.

A segunda parte foi um *middleware* desenvolvido em Node.js com objetivo de consumir os serviços disponibilizados pela primeira parte da solução. O *middleware* foi construído considerando a existência de diversas aplicações para o controle de locações de automóveis. Assim, caso alguém deseje utilizar este trabalho em um sistema real, deverá apenas desenvolver a primeira parte da solução e disponibilizando informações necessárias para que esta parte possa funcionar. E, por fim, a terceira parte foi o desenvolvimento do *front-end* da solução que exibe os dados por meio de Técnicas de Visualização de Informações com uso do *framework* APEXCHARTS.JS, além do uso do React.js e React Bootstrap para construção dos componentes das telas.

4. Desenvolvimento

Nesta Seção apresentamos os tópicos relacionado ao desenvolvimento da aplicação, que envolve a elaboração do Diagrama de Casos de Uso para expor as funcionalidades da aplicação, o Modelo Conceitual de Dados (MER) com as entidades relacionais do sistema, e por fim o Diagrama de Arquitetura com a estrutura de funcionamento e a troca de informações.

4.1 Modelagem

4.1.1 Caso de Uso

Na Figura 5, é apresentado o diagrama de Caso de Uso que, segundo Sommerville (2019, p. 108), “O conjunto de casos de uso representa todas as interações possíveis que serão descritas nos requisitos do sistema.”. O autor também informa “Os casos de uso identificam cada interação entre o sistema e seus usuários ou outros sistemas. Cada caso de uso deve ser documentado com uma descrição textual, que pode ser ligada a outros modelos — também na UML — para compor um cenário mais detalhado”.

A seguir descrevemos resumidamente os principais casos de uso. Onde se encontra os atores Gerente que é o responsável por manusear a aplicação. E o Sistema de locação de veículos que é a plataforma do cliente onde alimentam a base de dados, sobre as locações, veículos, avaliações, pessoas. E é onde o trabalho proposto neste documento está sendo criado para consumir estas informações e melhorar as visualizações.

O caso de uso "Exportar dados do gráfico" começa quando o Gerente escolhe um gráfico para realizar a exportação. O Gerente irá clicar no botão que indica as opções disponíveis para aquele gráfico. O Gerente poderá selecionar a opção de exportar no formato CSV ou PNG. O sistema irá fazer o *download* do arquivo para a máquina.

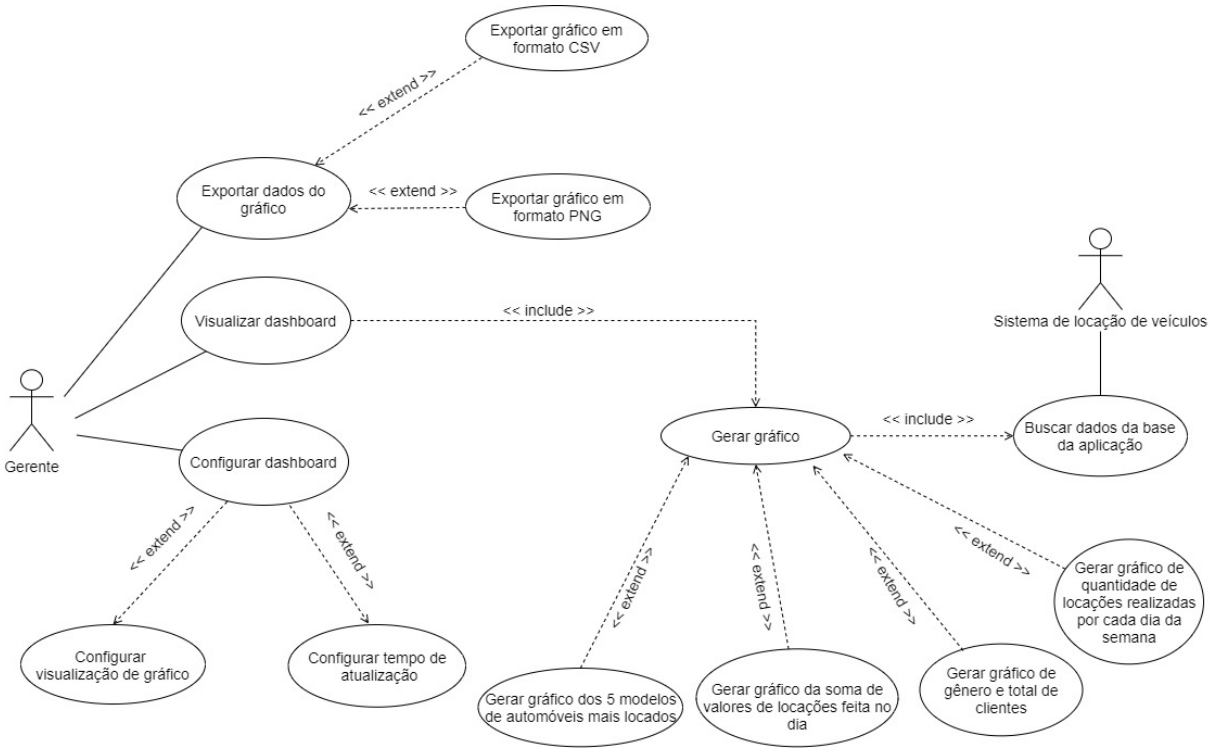

Figura 5. Diagrama de Caso e Uso gerado para expor as interações dos requisitos do sistema.

O caso de uso “Configurar visualização de gráfico” começa quando o Gerente seleciona a opção de visualização de gráfico. O Gerente poderá marcar ou desmarcar os gráficos para ser exibidos no dashboard. O Gerente irá clicar em salvar configurações e o sistema irá guardar esta configuração.

O caso de uso “Visualizar *dashboard*” começa quando o Gerente abre o *dashboard* que irá buscar e gerar os gráficos disponíveis para a visualização. Os dados e os tipos dos gráficos serão gerados de acordo com as informações buscada da base da aplicação do Gerente.

O caso de uso “Configurar o tempo de atualização do dashboard” começa quando o Gerente seleciona a opção de configurações. O Gerente irá clicar na aba de *dashboard*. O Gerente poderá selecionar a opção de tempo de atualização do *dashboard*. O Gerente irá clicar em salvar configurações e o sistema irá guardar esta configuração.

O caso de uso “Alterar perfil” começa quando o Gerente seleciona a opção de configurações. O Gerente irá clicar na aba de perfil. O Gerente poderá editar as informações de cadastro. O Gerente irá clicar em salvar configurações e o sistema irá guardar esta configuração.

4.1.2 Modelo Conceitual de Dados (MER)

Na Figura 6, é apresentando um diagrama conceitual para os dados representado seguindo o Modelo Entidade-Relacionamento, com intuito de representar uma visão geral do banco de dados MySQL esenvolvido. Segundo Elmasri e Navathe (2018, 31) “Modelos de dados de alto nível ou conceitual oferecem conceitos próximos ao modo como muitos usuários percebem os dados, enquanto os modelos de dados de baixo nível ou físico oferecem conceitos que descrevem os detalhes de como os dados são armazenados no computador, em geral em discos magnéticos”.

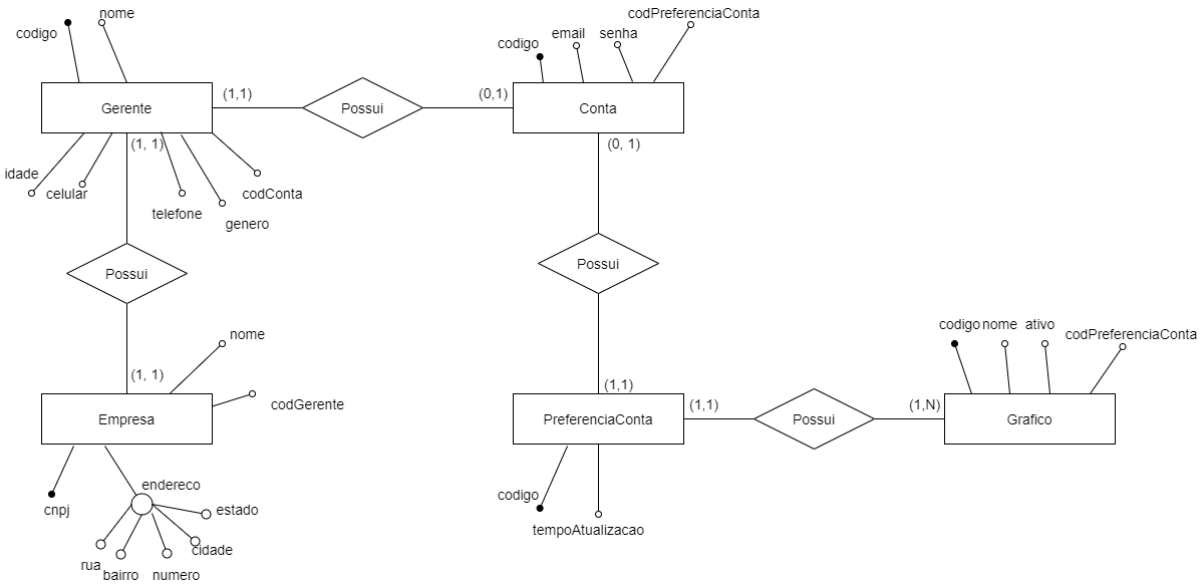

Figura 6. Diagrama com a modelagem conceitual para representar a visão geral do banco de dados da aplicação.

O banco de dados MySQL da aplicação possui 5 entidades de relacionamento, sendo elas Gerente, Empresa, Conta, PreferenciaConta e Gráfico. A entidade Conta é responsável por armazenar informações relacionadas a autenticação do usuário, como e-mail e senha, onde toda Conta criada no sistema possui no mínimo e no máximo uma preferência de conta, que

armazena o tempo de atualizações que o *dashboard* vai realizar na aplicação; esta informação é do tipo numérica e em segundos.

A entidade Gráfico é responsável por armazenar os gráficos que o administrador vai visualizar no *dashboard,* com as propriedades código, nome do gráfico e se ele está ativo ou não para se exibido em tela.

Por fim, a entidade Gerente e Empresa tem relação que diz respeito as informações pessoais do administrador e de sua empresa, como nome, endereço e CNPJ. Ambas possuem ligações que no mínimo e no máximo o administrador pode ter uma empresa dentro do sistema.

4.2 Diagrama de arquitetura

Na Figura 7, é apresentado o diagrama de arquitetura proposta para a aplicação que, de acordo com Sommerville (2019, p. 150), "Os diagramas de bloco são uma boa maneira de apoiar a comunicação entre pessoas envolvidas no processo de projeto (design) do *software*. Eles são intuitivos, e tanto os especialistas no domínio quanto os engenheiros de *software* podem se referir a eles e participar das discussões sobre o sistema".

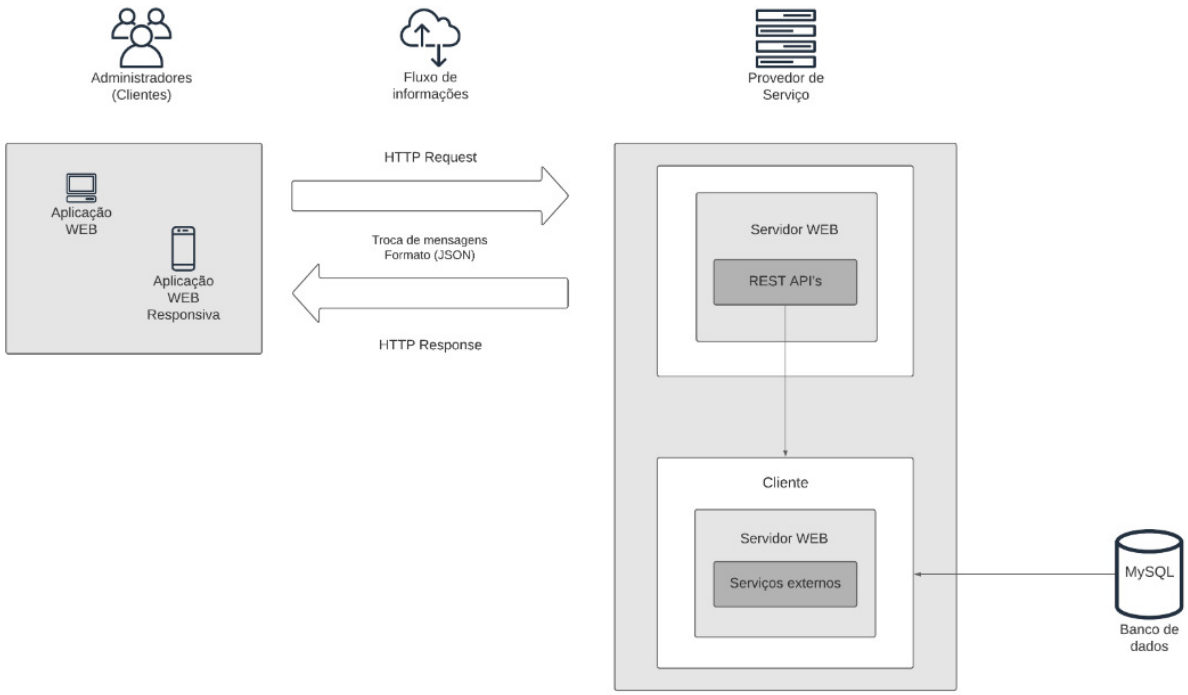

Figura 7. Arquitetura proposta para a aplicação

Os usuários, administradores da empresa de locação de veículos, interagem com uma aplicação desenvolvida para a plataforma Web. Conforme a funcionalidade, uma requisição é enviada ao servidor que, para responder a requisição solicitada, pode solicitar dados a um outro componente da aplicação, o serviço externo do cliente. Este tem como objetivo obter os dados do banco de dados MySQL da empresa de locação de veículos. Ao retornar os dados ao servidor, estes são enviados à aplicação web em formato JSON, que interpretará e atualizará os dados na tela do usuário.

Detalharemos a seguir como funciona cada parte da solução proposta, iniciando com o tópico de "Sistema para disponibilizar os serviços externos".

4.2.1 Sistema para disponibilizar serviços externos

Na Figura 8, é apresentada a primeira parte da solução que é um exemplo de serviço feito em Node.js que disponibiliza “Os modelos de automóveis mais locados”, onde é feito uma consulta ao MySQL para acessar as informações da base de dados do administrador. O retorno desta consulta é gerado uma lista que contenha para cada posição o modelo e a quantidade do automóvel visto na Figura 9.

```
1  const express = require("express");
2  const router = express.Router();
3  const mysql = require("../mysql").pool;
4
5  router.get("/getRentalCarModel", (req, res, next) => {
6    mysql.getConnection((error, conn) => {
7      if (error) return res.status(500).send({ error: error });
8      const isRented = 1;
9      conn.query(
10       `SELECT COUNT(veiculo.codigo) AS quantity, veiculo.modelo AS model
11              FROM aluguelveiculos.veiculo
12                WHERE alugado = ${isRented}
13              GROUP BY veiculo.modelo
14              ORDER BY veiculo.modelo`,
15       (error, result, fields) => {
16         conn.release();
17         if (error) return res.status(500).send({ error: error });
18         const response = {
19           length: result.length,
20           models: result.map((item) => {
21             return {
22               quantity: item.quantity,
23               model: item.model,
24             };
25           }),
26         };
27
28         return res.status(200).send({ response });
```

Figura 8. Serviço para retornar os modelos de automóveis mais locados

O código na linha 2 refere-se à definição de terminais do aplicativo (URIs) e como eles respondem às solicitações do cliente. Na linha 3 há o código para a configuração da conexão com o banco de dados MySQL da aplicação de locação de veículos. Nas linhas 5 até 27 têm-se o tratamento da chamada via método GET da URL “/getRentalCarModel”, cujo objetivo é solicitar ao servidor o retorno das informações dos modelos de automóveis mais locados. Quando este método é invocado, uma conexão com o banco de dados MySQL é criada (linha 6) e uma consulta SQL é realizada (linhas 10 até 14), que retorna a quantidade e o modelo do automóvel que estejam no estado alugado. A partir do retorno dessa consulta é gerada uma lista no formato JSON que segundo (JÚNIOR *et al*., 2018), “JSON é um formato de texto derivado da linguagem JavaScript e é representado por uma coleção de objetos compostos por pares de chave-valor”. Entretanto, a forma que este serviço retorna as informações para a exibição do gráfico, é incompatível com o componente da biblioteca de gráficos, que será tratado na segunda parte do sistema.

Na Figura 10, é representado o modelo de configuração que o componente espera para a renderização das informações. Portanto, é necessário que seja construído a segunda parte da solução proposta, que é o desenvolvimento do *middleware* para manipular essas informações.

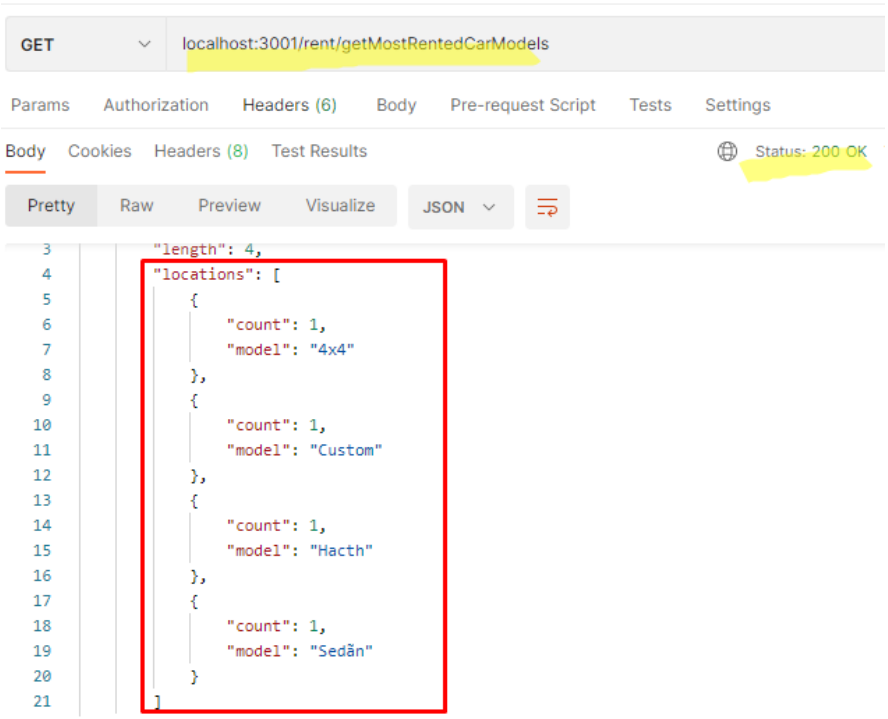

Figura 9. Retorno no formato JSON do serviço de modelos de automóveis mais locados.

```
import React from "react";
import Chart from "react-apexcharts";

export default function ChartMostRentedCarModels(props) {
  function configCharts() {
    const config = {
      options: {
        chart: {
          id: "basic-bar",
        },
        xaxis: {
          categories: [],
        },
      },
      series: [
        {
          name: "Quantidade",
          data: [],
        },
      ],
    };

    return config;
  }

  return (
    <div className="chart-container">
      <h2 className="chart-title">Modelos de automóveis mais locados</h2>
      <Chart
        options={configCharts().options}
        series={configCharts().series}
        type="bar"
        width="100%"
      />
    </div>
  );
}
```

Modelos

Contadores

Figura 10. Configurações do componente de gráfico para a exibição em tela

4.2.2 *Middleware*

A construção da segunda parte foi a criação de um *middleware* feito em Node.js com REST API's para acessar os serviços externos disponibilizados pela primeira parte, com o intuito de preparar as informações para o *front-end*, que é a terceira parte da solução. Essa etapa foi necessária para permitir que o *software* seja mais facilmente modificado para outros *softwares* de locação de veículo.

Na Figura 11, temos o exemplo de um REST API consumindo o serviço externo "/rent/getRentalCarModel" requisitado pelo *endpoint* "/rentalCarModelChart" que por sua vez obtém os dados de retorno e realiza um mapeamento das propriedades de forma que fiquem agrupadas em suas respectivas variáveis, que podem ser observadas na Figura 12.

```
const express = require("express");
const router = express.Router();

const axios = require("./axios").instance;

router.get("/rentalCarModelChart", (req, res, next) => {
  axios
    .get("/rent/getRentalCarModel")
    .then(({ data }) => {
      if (data !== undefined) {
        const result = data.response.models;
        return res.status(200).send({
          seriesQuantity: result.map((item) => item.quantity),
          seriesName: result.map((item) => item.model),
          success: true,
        });
      }
    })
    .catch((error) => {
      return res.status(400).send({ success: false });
    });
});

module.exports = router;
```

Figura 11. REST API consumindo o serviço externo

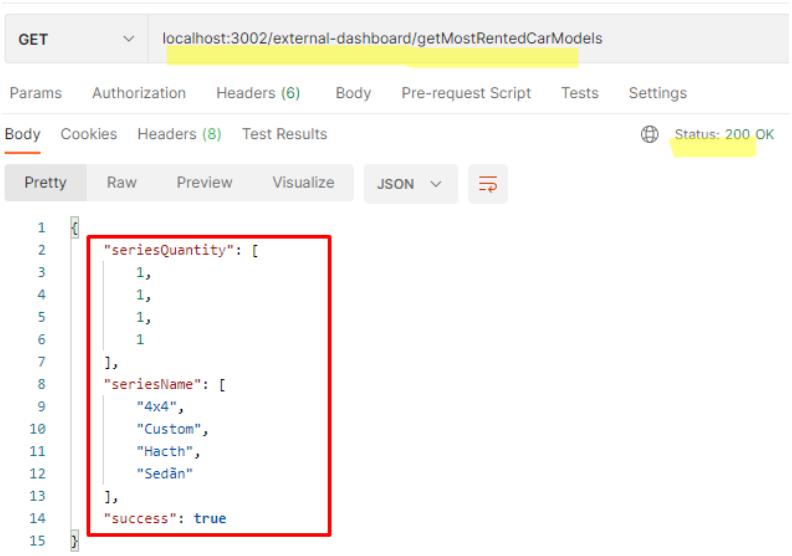

Figura 12. Retorno da REST API com as novas propriedades mapeadas

4.2.3 *Front-end*

Por fim, a parte 3 e final da solução é o *front-end* onde é exibido o gráfico com a informação retornada pela REST API. Por exemplo, os dados no arquivo JSON da Figura 12 serão exibidos de modo gráfico conforme Figura 13.

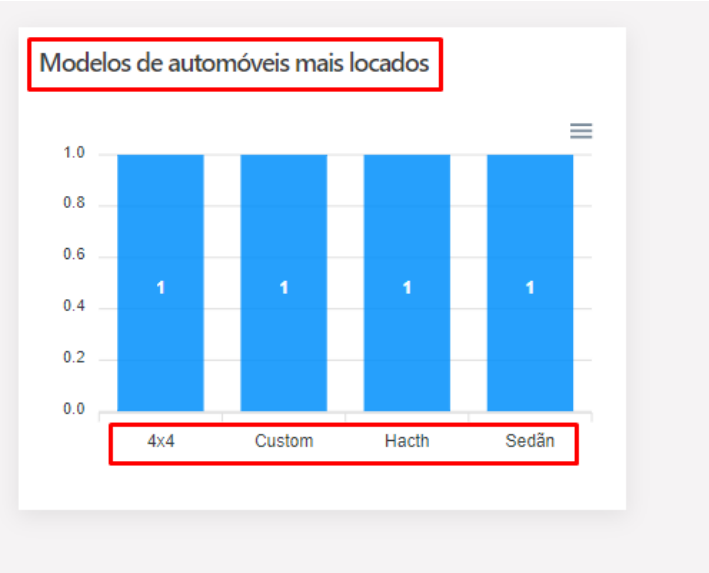

Figura 13. Gráfico de modelos de automóveis mais locados

5. Resultados

Nesta Seção apresentamos os resultados obtidos após a modelagem e arquitetura apresentado nas subseções anteriores, e o primeiro resultado é a tela de login para realizar a autenticação do administrador no sistema, e logo em seguida o dashboard com os gráficos desenvolvidos de acordo com os serviços disponíveis, e as configurações que poderão ser realizadas em tela, e pôr fim a exportação de gráficos.

5.1 Login

Considerando o diagrama entidade-relacionamento criado e o conhecimento obtido na construção do *middleware* e do gráfico para exibição dos modelos de automóveis mais locados, iniciou-se a produção das interfaces da aplicação Web. Na Figura 14, é apresentado a tela de Login para o gerente realizar a autenticação. Neste momento, o e-mail e senha cadastrado na base de dados são obrigatórios para obter acesso na aplicação.

Caso a o gerente clique no botão ‘Entrar’ sem inserir o e-mail e senha, o sistema não irá validar as informações pois perceberá que um dos campos não foram preenchidos. Os campos e-mail e senha ficarão vermelho e uma mensagem será mostrada para que seja digitado valores correspondentes aos campos.

Na Figura 16, é apresentado o erro que será exibido ao Gerente caso o e-mail e senha fornecidos não correspondam com nenhum registrado na base de dados.

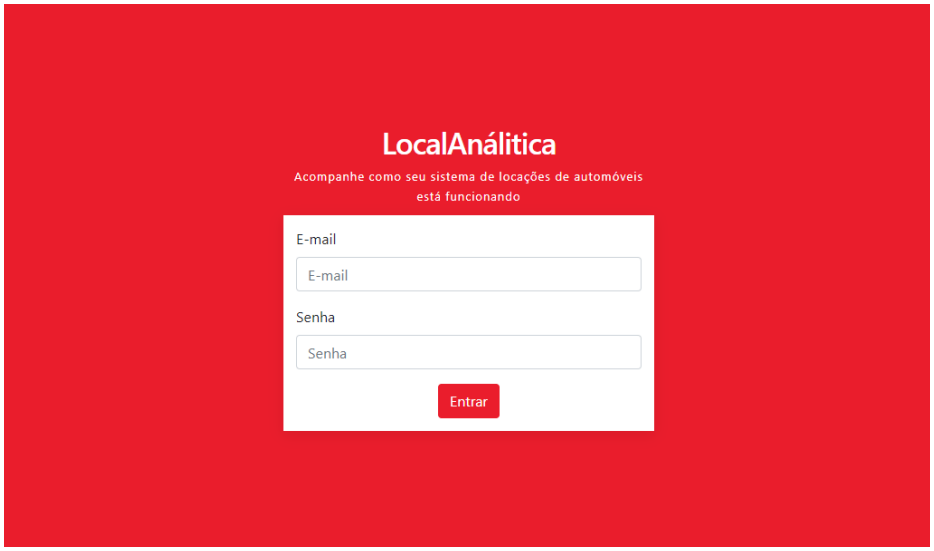

Figura 14. Tela de Login para o gerente realizar a autenticação no sistema

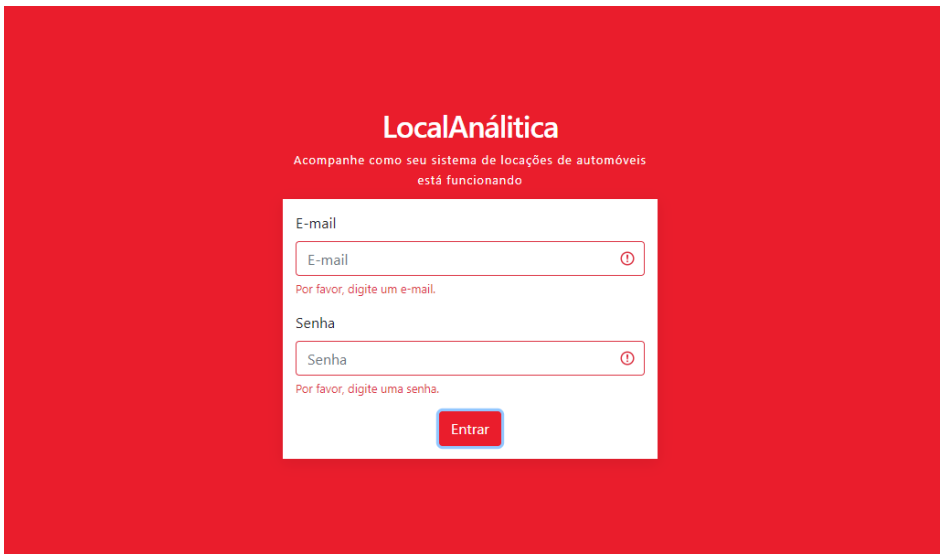

Figura 15. Tela de Login com a validação de campos obrigatórios

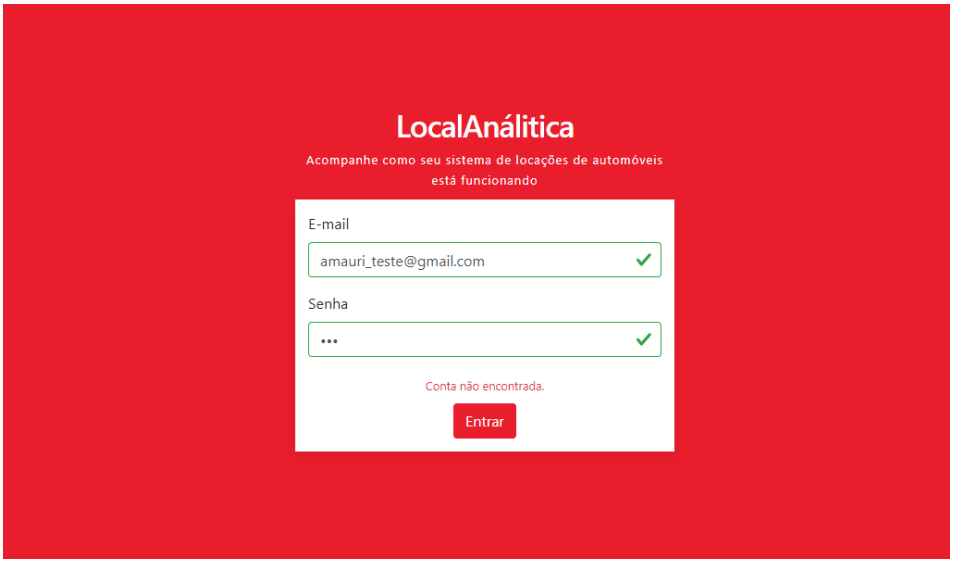

Figura 16. Tela de Login com a validação de conta não encontrada

5.2 Visualização do dashboard

Na Figura 17, é apresentado o *dashboard*, a principal funcionalidade do sistema, que a exibição dos gráficos preenchidos com as informações baseada no banco de dados MySQL do administrador com os passos desenvolvidos explicados nas subseções anteriores.

Para cada tipo de gráfico renderizado em tela foi realizado uma escolha baseada no tipo da informação disponível para ser exibida, desta forma, foram escolhidos tipos de gráficos para uma melhor visualização, dentre os gráficos “bar”, “pie”, “área”, “radialBar” e informações quantitativas.

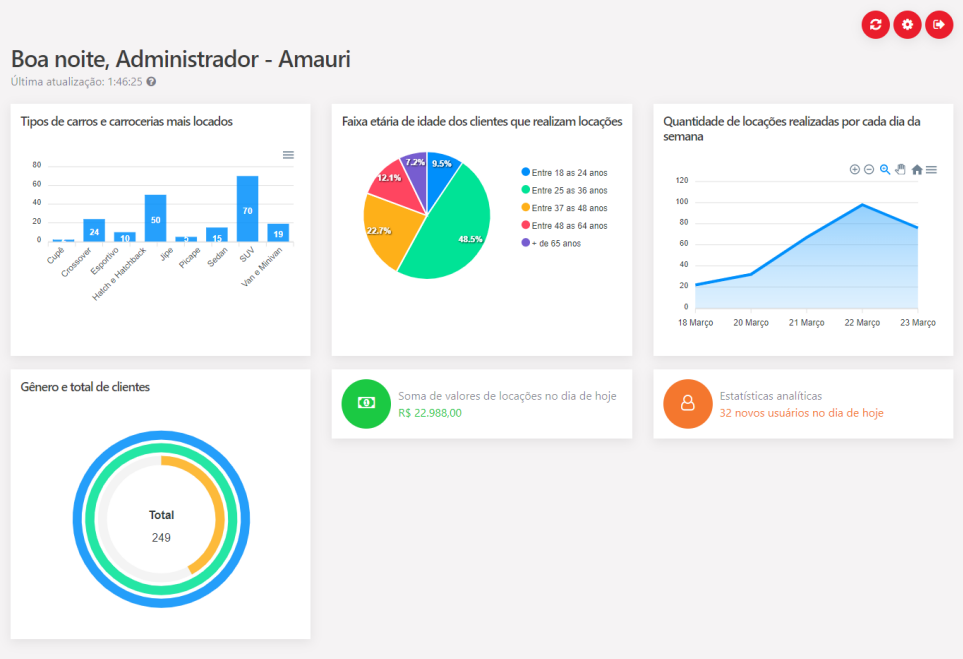

Figura 17. Tela de Dashboard com gráficos disponíveis para o gerente

5.3 Configurações do dashboard

Na Figura 18, é representado o modal de configurações do *dashboard* que contém a opção de “Visualização de gráficos” funcionalidade descrita no diagrama de casos e uso (Figura 5).

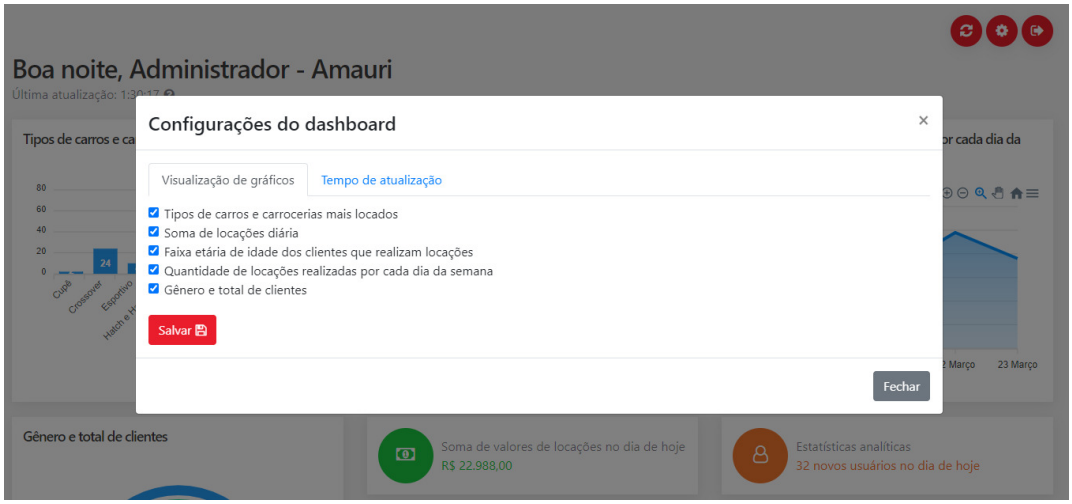

Figura 18. Modal de Configurações do dashboard – Opção de Visualização de gráficos.

O armazenamento das informações dos tipos de gráficos fica na entidade Gráfico comentada no tópico de “Modelo Conceitual de Dados (MER)”, visto que é renderizado uma lista de caixa de seleção para a escolha do gráfico que será exibido no *dashboard,* na Figura 19 é apresentado o serviço construído para buscar estas informações.

```
router.get("/getchartstypes/:email", (req, res, next) => {
  const { email } = req.params;

  mysql.getConnection((error, conn) => {
    if (error) return res.status(500).send({ error: error });
    conn.query(
      "SELECT * FROM grafico " +
        "WHERE grafico.codPreferenciaUsuario = " +
        "(SELECT conta.codPreferenciaUsuario FROM conta WHERE conta.email = ?)",
      [email],
      (error, result, fields) => {
        conn.release();
        if (error) return res.status(500).send({ error: error });
        if (result.length === 0) {
          return res
            .status(404)
            .send({ errorMessage: "Gráficos não encontrado." });
        }
        const response = {
          charts: result.map((chart) => {
            return {
              chart_id: chart.codigo,
              name: chart.nome,
              active: chart.ativo,
            };
          }),
        };
        return res.status(200).send({ response });
      }
    );
  });
});
```

Acessando via parâmetro o e-mail do administrador

Figura 19. Serviço para buscar as informações dos tipos de gráficos com o e-mail do Administrador.

Com este serviço é possível obter o seguinte resultado representado na Figura 20, que se têm uma lista de objetos que possuem propriedades de chart_id, que é o código armazenado do gráfico, a propriedade name, é o nome do gráfico e por fim, active, que define se o gráfico vai ser exibido no dashboard.

Após o administrador realizar a escolha das opções dos gráficos que deseja exibir ou não no *dashboard* ele deverá clicar no botão de Salvar onde é feito uma atualização no registro dentro da entidade Gráfico para ativo ou inativo dependendo de sua escolha e a mensagem exibida na Figura 21, será apresentada.

Na Figura 22, a opção de “Tempo de atualização” é renderizado uma caixa de opções com valores pré-definidos em segundos para atualização do dashboard, o usuário deverá selecionar uma opção e após esta escolha irá clicar em Salvar onde é persistido a informação na base de dados para a entidade PreferenciaConta.

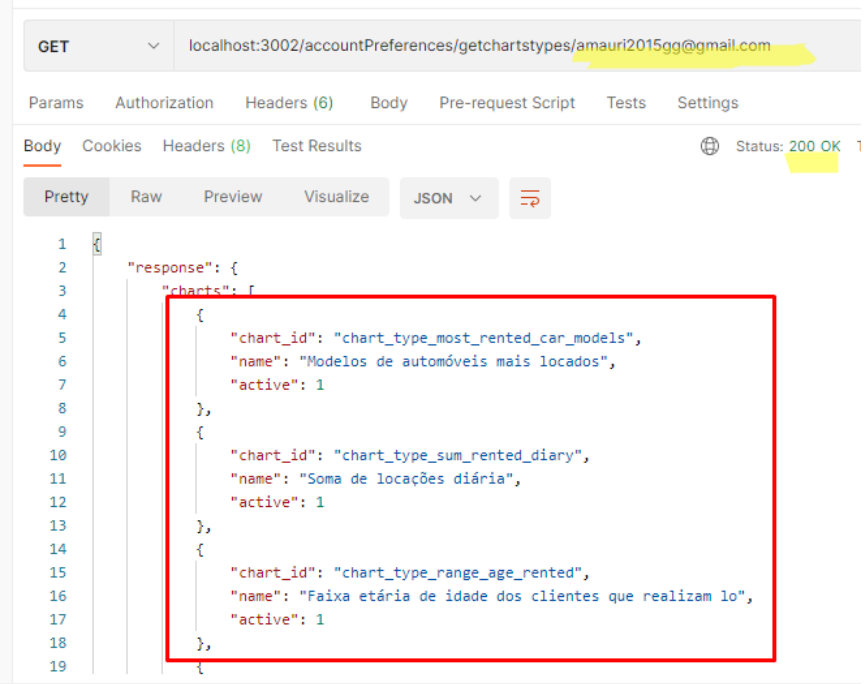

Figura 20. Resultado do serviço para buscar as informações dos tipos de gráficos com o e-mail do Administrador.

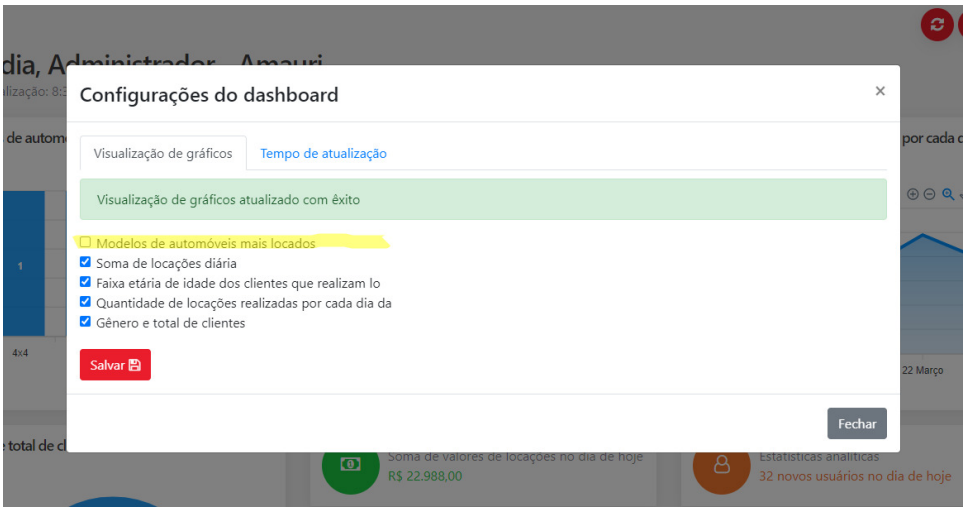

Figura 21. Mensagem de sucesso para atualização da escolha do tipo de gráfico para serem exibidos no *dashboard*.

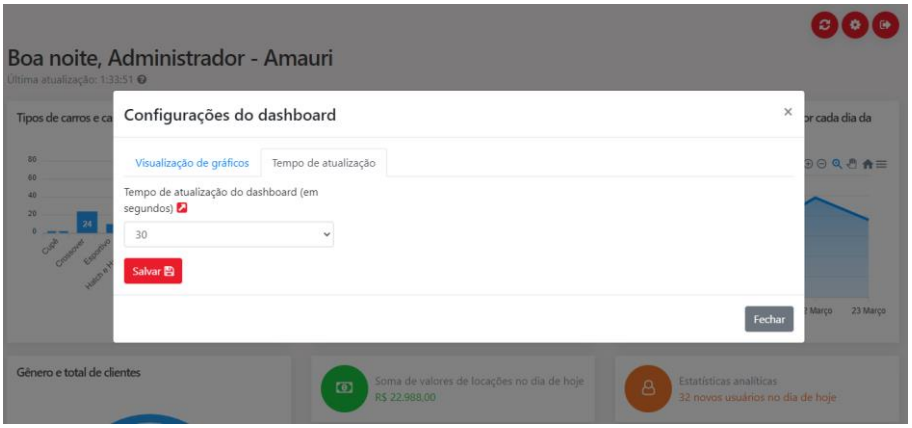

Figura 22. Modal de Configurações do dashboard – Opção de Tempo de atualização

5.4 Exportação para CSV e PNG

Na Figura 23, é apresentado um exemplo da funcionalidade de exportação de gráfico disponibilizada pela própria biblioteca de gráficos APEXCHARTS.JS utilizada no desenvolvimento. A exportação pode ser realizada para CSV e PNG, conforme exposto no diagrama de casos e uso (Figura 5).

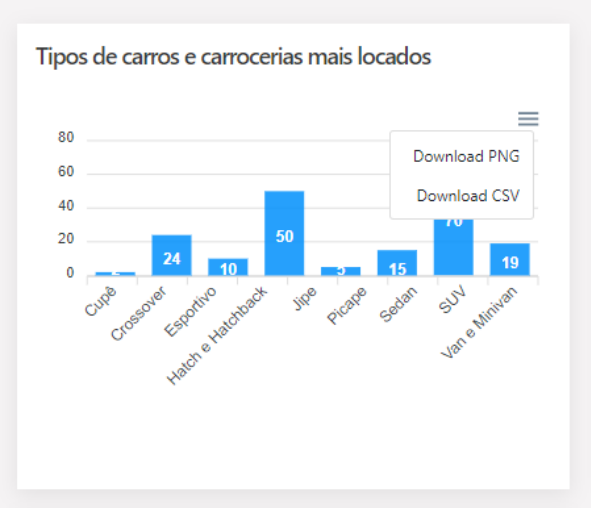

Figura 23. Exemplo de gráfico de Tipos de carros e carrocerias mais locados

6. Conclusão

O objetivo deste trabalho foi o desenvolvimento de um *software* para proporcionar uma visão analítica das informações e uma melhor análise realizada pelo administrador do estabelecimento de locação de automóveis: o LocalAnalitica. Para o seu desenvolvimento foi empregada uma abordagem incremental dividida em 3 partes, bem como técnicas de Visualização de Informações.

Acreditamos que o LocalAnalitica é capaz de fornecer uma plataforma moderna, robusta e adaptável para determinado contexto da base de dados do cliente (gerente), sendo somente obrigatório serviços de acordo com que é necessário conforme o *middleware* espera, para o fornecimento das informações do estabelecimento do gerente para a construção do *dashboard.*

Vale ressaltar que, conforme a bibliografia consultada (que foi de grande benefício para o desenvolvimento deste trabalho), e com a metodologia desenvolvida em 3 partes para desenvolver a parte prática, sendo elas: a primeira parte, o sistema simples para disponibilizar serviços externos com informações da base dados do administrador; a segunda parte, o middleware para consumir estes serviços externos e devolver as informações para o f*ront-end*; e a terceira parte e última, *o front-end*, para construção dos componentes e gráficos.

O *dashboard* e as funcionalidades desta aplicação podem ajudar o gerente na análise e gerenciamento de seu estabelecimento. Entretanto é necessário verificar a adaptação da solução proposta com *softwares* reais de locação de automóveis, visto que isto foi uma limitação para este trabalho, necessitando neste caso, construir a primeira parte, conforme explicado na Seção 4.1. Também sugerimos como trabalhos futuros aplicar outras possibilidades tipos de gráficos que estão disponíveis no *framework* APEXCHART.JS para a construção de um *dashboard* ainda mais completo.

Para realizar este Trabalho de Conclusão de Curso foram utilizados os conhecimentos adquiridos nas disciplinas: Banco de Dados I e II, Desenvolvimento Web, Engenharia de *Software*, Metodologia de Pesquisa Científica, Inglês, Inglês Avançado, Algoritmos e Lógica de Programação, Arquitetura de *Software* e Empreendedorismo.

Referências

ASSOCIAÇÃO BRASILEIRA DAS LOCADORAS DE AUTOMÓVEIS. Semana do trabalho: Locadoras mantêm mais de 80 mil empregos no Brasil. Disponível em <https://www.abla.com.br/semana-do-trabalho-locadoras-mantem-mais-de-80-mil-empregos-no-brasil/>. Acesso em 05 out. 2020.

CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN, B.; CARD, M. Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann Series in Interactive Technologies, Academic Press, 1999.

CRUZ, F. Scrum e PMBOK unidos no Gerenciamento de Projetos. 1. ed., Rio de Janeiro: Editora Brasport, 2013.

ELMASRI, R.; NAVATHE, S. B. Sistemas de Banco de Dados. 7. ed., São Paulo: Pearson Education do Brasil, 2018.

FIGO, A. Empresas criam Netflix de carros. Vale a pena assinar? Disponível em <https://exame.abril.com.br/seu-dinheiro/empresas-criam-netflix-de-carros-vale-a-pena-assinar>. Acesso em 05 out. 2020.

FREITAS, C. M. D. S; CHUBACHI. O. M; LUZZARDI. P. R. G; CAVA. R. A. Introdução à Visualização de Informações. Revista de Informática Teórica e Aplicada, Rio Grande do Sul, v. 8, n. 2, 2001, p. 143-158. Disponível em: < http://www.lume.ufrgs.br/bitstream/handle/10183/19398/000300210.pdf?sequence=1 >. Acessado em 05 out. 2020.

JÚNIOR, J. B. S; SILVA. P. C. Análise da representação semântica de modelos de dados do formato JSON. Revista de Sistemas e Computação, Salvador, v. 8, n. 1, 2018, Disponível em: < https://revistas.unifacs.br/index.php/rsc/article/view/5487 >. Acessando em 20 fev. 2021.

MOZZATO, A. R.; GRZYBOVSKI, D. Análise de conteúdo como técnica de análise de dados qualitativos no campo da Administração: potencial e desafios. Revista de Administração Contemporânea, v. 15, n. 4, p. 731-747, 2011.

NASCIMENTO; H. FERREIRA; C. Uma Introdução à Visualização de Informações. Visualidades, v. 9, n. 2, p. 13-44, 2011. Goiânia-GO: UFG, FAV, 2011.

OLIVEIRA, R. P.; WILDNER, M. C. S; PRETTO F. Técnicas de Visualização de Informações como Apoio à Gestão Estratégica. Revista Destaque Acadêmicos, v. 10, n. 1, p. 235-253, 2018.

SOMMERVILLE, I. Engenharia de Software 10. ed., São Paulo: Pearson Education do Brasil, 2019.

VAZ, F. R.; CARVALHO, C. L. DE. Visualização de Informações. Minas Gerais: UFG, 2004. 1 p. (Série Texto Técnico, INF _003/04)

VISUALIZAÇÃO. In: MICHAELIS, Dicionário Online de Português. São Paulo: Melhoramentos, 2020. Disponível em: <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/visualizacao/>. Acesso em 05 out. 2020.

Documento Digitalizado Público

Artigo de TCC

Assunto: Artigo de TCC
Assinado por: Andre Constantino
Tipo do Documento: Relatorio
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- **Andre Constantino da Silva**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 12/08/2021 22:37:24.

Este documento foi armazenado no SUAP em 12/08/2021. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifsp.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 742624
Código de Autenticação: 4aeca35979