



EasyEV - Uma aplicação para apoiar o gerenciamento de veículos elétricos com foco no consumo de bateria

Gabriel dos Santos Nascimento

JUIZ DE FORA
SETEMBRO, 2021

EasyEV - Uma aplicação para apoiar o gerenciamento de veículos elétricos com foco no consumo de bateria

GABRIEL DOS SANTOS NASCIMENTO

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Sistemas de Informação

Orientador: José Maria Nazar David
Coorientador: Mario Antônio Ribeiro Dantas

JUIZ DE FORA
SETEMBRO, 2021

EASYEV - UMA APLICAÇÃO PARA APOIAR O
GERENCIAMENTO DE VEÍCULOS ELÉTRICOS COM FOCO NO
CONSUMO DE BATERIA

Gabriel dos Santos Nascimento

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

José Maria Nazar David
Doutor

Mario Antônio Ribeiro Dantas
Doutor

Calebe de Paula Bianchini
Doutor

Victor Ströele de Andrade Menezes
Doutor

JUIZ DE FORA
2 DE SETEMBRO, 2021

À minha noiva.

Aos meus pais.

Aos meus professores.

Resumo

Com o avanço da tecnologia, veículos elétricos passam a fazer parte da paisagem urbana. Entretanto, encontram-se problemas relacionados à autonomia destes veículos e a escassez de postos de abastecimentos. Este trabalho propõe uma aplicação para dar suporte ao motorista quanto ao consumo de um veículo elétrico como também aos modos de condução do mesmo. Simulou-se a captação de dados de um simulador de contexto capaz de reproduzir o consumo e autonomia de um veículo elétrico. As informações obtidas através das simulações serviram de base para a realização de avaliações preliminares em relação à construção do artefato produzido. Para tanto, foi utilizada a metodologia de *Design Science Research*.

Palavras-chave: autonomia veicular, nível de bateria, percepção de consumo energético, simulador de contexto.

Abstract

Due to the progress of technology, electric vehicles become part of the urban scenery. However, there are problems related to autonomy of these vehicles and a shortage of refueling stations. This work proposes an application to support the driver regarding the consumption of an electric vehicle as well as its driving modes. The vehicle's data was provided from a context simulator that could reproduce the consumption and autonomy of an electric vehicle. The information obtained through the simulations served as a basis for conducting preliminary assessments in relation to the construction of the artifact produced. For that, the DSR methodology was used.

Keywords: vehicle autonomy, battery level, perception of energy consumption, context simulator.

Agradecimentos

Agradeço primeiro a Deus por ter me mantido na trilha certa durante este curso de graduação com saúde e forças para chegar até o final.

À minha noiva, Bianca, pelo encorajamento e apoio irrestrito demonstrado durante o período do curso de graduação.

Aos meus pais, Luísa e Paulo, pelo sustento e confiança ao longo de toda a minha trajetória.

Deixo um agradecimento especial aos meus orientadores, José Maria e Mário Antônio, pela orientação, amizade e, principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos meus colegas do curso de Sistemas de Informação pelas trocas de ideias e ajuda mútua. Juntos conseguimos avançar e ultrapassar todos os obstáculos.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

Também agradeço à Universidade Federal de Juiz de Fora que me deu a oportunidade de cursar este curso de graduação.

“Temos nosso próprio tempo”.

Legião Urbana (Tempo Perdido)

Conteúdo

Lista de Figuras	7
Lista de Tabelas	8
Lista de Abreviações	9
1 Introdução	10
1.1 Problema de Pesquisa	10
1.2 Objetivos	11
1.3 Metodologia	11
1.4 Organização do Trabalho	13
2 Fundamentação Teórica	14
2.1 <i>Design Science Research</i> (DSR)	14
2.2 Internet das Coisas (IoT)	15
2.3 Veículo elétrico	16
2.4 Sustentabilidade aplicada à Engenharia de Sistemas	17
2.5 Trabalhos Relacionados	17
2.6 Considerações Finais do Capítulo	19
3 Material e Métodos	20
3.1 Planejamento	21
3.1.1 Identificação das necessidades da solução proposta	21
3.2 Solução Proposta	23
3.2.1 Regras de Negócio	24
3.2.2 Requisitos Funcionais	25
3.2.3 Requisitos Não Funcionais	25
3.2.4 Arquitetura	25
3.2.5 Ferramentas Utilizadas	27
3.2.6 A Aplicação <i>EasyEV</i>	30
3.2.7 Interface do usuário	32
3.3 Considerações Finais do Capítulo	33
4 Avaliação da solução	34
4.1 Avaliação dos requisitos propostos	34
4.2 Avaliação da aplicação em plataforma em Nuvem	39
4.3 Considerações Finais do Capítulo	41
5 Conclusão e Trabalhos Futuros	42
Bibliografia	44

Lista de Figuras

2.1	Ciclo da DSR. Fonte: Elaborada pelo autor.	15
3.1	Visão geral da solução proposta. Fonte: Elaborado pelo autor.	20
3.2	Planejamento do projeto proposto para o desenvolvimento da solução apresentada neste trabalho. Fonte: Elaborado pelo autor.	22
3.3	Captura de tela do simulador <i>Siafu</i>	23
3.4	Visão geral da arquitetura. Fonte: Elaborado pelo autor.	26
3.5	Requisição <i>POST</i> para API <i>RESTful</i> da aplicação <i>EasyEV</i> . Fonte: Elaborado pelo autor.	28
3.6	Diagrama de Classes da aplicação <i>EasyEV</i> . Fonte: Elaborado pelo autor. .	31
3.7	Diagrama de Sequência da aplicação <i>EasyEV</i> . Fonte: Elaborado pelo autor.	31
3.8	Captura da tela principal da aplicação <i>EasyEV</i> . Fonte: Elaborado pelo autor.	32
4.1	Funcionamento do módulo <i>DataEV System</i> . Fonte: Elaborado pelo autor. .	35
4.2	Diagrama de caso de uso da aplicação <i>EasyEV</i> . Fonte: Elaborado pelo autor.	38
4.3	Avaliação em ambiente de produção. Fonte: Elaborado pelo autor.	40
4.4	Identificação dos dados do veículo na aplicação <i>EasyEV</i> . Fonte: Elaborado pelo autor.	41

Lista de Tabelas

3.1	Regras de Negócio. Fonte: Elaborado pelo autor.	24
3.2	Requisitos Funcionais. Fonte: Elaborado pelo autor.	25
3.3	Requisitos Não Funcionais. Fonte: Elaborado pelo autor.	25
4.1	Cenário de teste para atestar o funcionamento do módulo <i>DataEV System</i> . Fonte: Elaborado pelo autor.	35
4.2	Relacionamento entre casos de uso e requisitos funcionais da aplicação. Fonte: Elaborado pelo autor.	39

Lista de Abreviações

DCC Departamento de Ciência da Computação

UFJF Universidade Federal de Juiz de Fora

DSR *Design Science Research*

IoT Internet das Coisas

1 Introdução

1.1 Problema de Pesquisa

No âmbito da sustentabilidade ambiental, fabricantes de automóveis buscam reduzir o uso de combustíveis fósseis, visto que eles são um recurso limitado e poluente. Aliado ao movimento de troca da fonte primária de energia dos veículos convencionais, o avanço da tecnologia abre espaço para novas pesquisas neste contexto (KUMAR; ALOK, 2020). O veículo elétrico é uma invenção do século XIX, porém, ainda encontra desafios que postergam sua aderência ao mercado consumidor. Segundo (BASIT et al., 2020), veículos elétricos tendem a substituir veículos à combustão, entretanto, a baixa autonomia de veículos elétricos cria um grande desafio de aderência ao mercado consumidor para esse tipo de veículo.

O uso de um veículo elétrico, entretanto, requer cuidados, visto que as cidades ainda não estão adaptadas a suportarem uma grande frota de veículos elétricos. Estes veículos carecem de uma estrutura elétrica robusta que deem suporte a recargas eficientes (SCHULTIS, 2021). Além disso, um veículo elétrico utiliza componentes e sistemas que funcionam, em sua maioria, com a energia elétrica provida pela bateria veicular. Com isso, diversos fatores podem influenciar na duração da bateria como, por exemplo, congestionamento, sistema de controle climático e sistema de som (LUNDSTRÖM, 2014).

Devido a falta de previsibilidade do consumo da bateria do veículo elétrico, se faz necessário um estudo mais abrangente em relação ao monitoramento e entendimento das informações providas pelo veículo (LUNDSTRÖM, 2014). Além disso, levando em conta a cidade de São Paulo, por exemplo, tem-se a precariedade de postos de recarga de veículos elétricos. Eles são encontrados, em sua maioria, em *shopping centers* (CASTRO et al., 2019).

Relacionando a indisponibilidade de postos de recarga com a falta de experiência e conhecimento do público consumidor em relação a veículos elétricos, ressalta-se novos desafios a serem superados pela comunidade acadêmica e pelos fabricantes de automóveis

como, por exemplo, a baixa autonomia dos veículos e dificuldade em recarregar estes veículos. Sendo assim, este trabalho propõe uma solução para os problemas relacionados ao consumo de bateria do veículo elétrico.

1.2 Objetivos

Este trabalho tem por objetivo geral dar suporte ao condutor em relação a visualização e ao entendimento dos dados fornecidos por um veículo elétrico, especialmente, sobre o nível de carga da bateria e ao consumo do veículo. Para tanto, foi desenvolvida uma aplicação, denominada *EasyEV*, para apoiar o condutor do veículo através do fornecimento de informações relacionadas à carga da bateria.

Como objetivos específicos, podemos destacar:

- Escolha de um *software* simulador de contexto para simular um veículo em funcionamento em um contexto de utilização.
- Integrar o simulador à aplicação objeto deste trabalho.
- Desenvolver um sistema capaz de capturar os dados gerados pelo simulador e armazená-los em um banco de dados.
- Desenvolver uma aplicação capaz de receber os dados armazenados no banco de dados da aplicação.

1.3 Metodologia

Para elucidar e dar andamento ao projeto de Pesquisa, a forma como as diferentes etapas foram identificadas e quais os procedimentos adotados para a condução dessas etapas, serão apresentados a seguir.

Primeiramente, como forma de se obter uma fonte geradora de dados, optou-se por escolher um simulador de contexto capaz de gerar dados que só seriam gerados em um veículo elétrico. A fonte de dados foi essencial para dar início a estruturação da solução apresentada neste trabalho. Seguindo, foi necessária a construção de um

módulo intermediário capaz de se conectar ao simulador e ao banco de dados da aplicação. Finalmente, seguiu-se para o desenvolvimento da aplicação, efetivamente. Como forma de guiar o desenvolvimento desde trabalho, escolheu-se uma metodologia já solidificada no nosso grupo de pesquisa.

A metodologia de pesquisa utilizada denomina-se *Design Science Research* (DSR) (HEVNER et al., 2004). Segundo (PIMENTEL; FILIPPO; SANTOS, 2020), desenvolver um artefato pode ser uma forma para gerar conhecimento científico. Neste caso, tem-se como artefato a aplicação idealizada, que tem por objetivo dar suporte ao gerenciamento de carga de um veículo elétrico.

Através da DSR, o problema foi identificado e mapeado através do Ciclo de Relevância. Neste momento, além do mapeamento do problema, desenvolveu-se os critérios de aceitação da solução proposta. No Ciclo de *Design*, foi desenvolvida a implementação da solução e a implantação de processos necessários para a correta execução de testes em ambientes experimentais. No Ciclo de Conhecimento, ocorreu a análise da solução desenvolvida e a execução em ambiente simulado.

Além disso, como forma de ressaltar a avaliação da solução desenvolvida, este trabalho fez uso de uma abordagem denominada *ScenarioIoT* proposta por (SILVA; TRAVASSOS, 2020). A abordagem propõe especificação de cenários funcionais para aplicações “Internet das Coisas”, ou em inglês, *Internet of Things* (IoT). Por meio dos cenários especificados, foi possível validar o funcionamento da aplicação com base nos critérios de aceitação.

Por fim, a pesquisa realizada, e descrita neste trabalho, pode ser caracterizada como Trabalho Científico Original. Como descrito na Seção 1.1, o problema de pesquisa deste trabalho se relaciona ao suporte a uma nova realidade presente na sociedade, ou seja, a adoção de veículos elétricos. Para tanto, pretende-se contribuir com a comunidade acadêmica e a sociedade, visto que a arquitetura de *software* desenvolvida pode contribuir para o fomento de novas pesquisas e estudos relacionados ao uso e adoção de veículos elétricos.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte forma: o Capítulo 2 realiza um embasamento teórico de conceitos tratados neste trabalho. No Capítulo 3, é descrita e justificada a arquitetura de *software* proposta. Também são abordados problemas e dificuldades encontradas na implementação e configuração das camadas propostas na arquitetura de *software*. O Capítulo 4 discute a avaliação da solução. E finalmente, no Capítulo 5, são feitas as conclusões e elencadas possibilidades de trabalhos futuros.

2 Fundamentação Teórica

O presente capítulo aborda os principais conceitos utilizados no desenvolvimento desde trabalho. Devido a multidisciplinaridade desde trabalho, as próximas seções apresentam os conceitos técnicos do artefato em estudo, no caso, o veículo elétrico. Além disso, discutimos os conceitos técnicos e arquiteturais envolvendo a solução desenvolvida. A primeira seção traz a definição das etapas da DSR, a segunda seção traz o conceito de veículo elétrico. Seguindo, discorre-se sobre aspectos de Sustentabilidade aplicados à Engenharia de Sistemas. Por fim, apresenta-se trabalhos relacionados com o problema de pesquisa proposto.

2.1 *Design Science Research (DSR)*

A Figura 2.1, desenvolvida com base na metodologia proposta por (HEVNER, 2007), ilustra as etapas abordadas na DSR. As etapas apresentadas se caracterizam como tarefas que foram desempenhadas no contexto desta pesquisa.

O ambiente está relacionado com a sociedade e seus problemas e, a partir deles, cria oportunidades diante às necessidades enfrentadas. Seguindo o Ciclo de Relevância, relaciona-se com o contexto ao qual o artefato está sendo projetado. Neste ciclo, são elencados os requisitos necessários para a concepção do artefato e os testes necessários para validar a solução proposta, ou seja, a partir de desafios encontrados no contexto de utilização, cria-se artefatos para resolver os problemas identificados.

O Ciclo de *Design*, volta-se para a implementação do artefato, ou seja, as fases e os processos necessários para a sua concepção. A implementação de um artefato pode ocorrer de diversas maneiras. Para cada artefato desenvolvido, enumera-se passos obrigatórios para a execução do ciclo. A avaliação da solução deve ocorrer conforme especificado no Ciclo de Relevância. Cabe ao autor identificar o correto ambiente experimental.

No último ciclo, o Ciclo de Conhecimento, está relacionado com teorias e métodos científicos que garantem que a pesquisa desenvolvida irá seguir o rigor teórico e metodo-

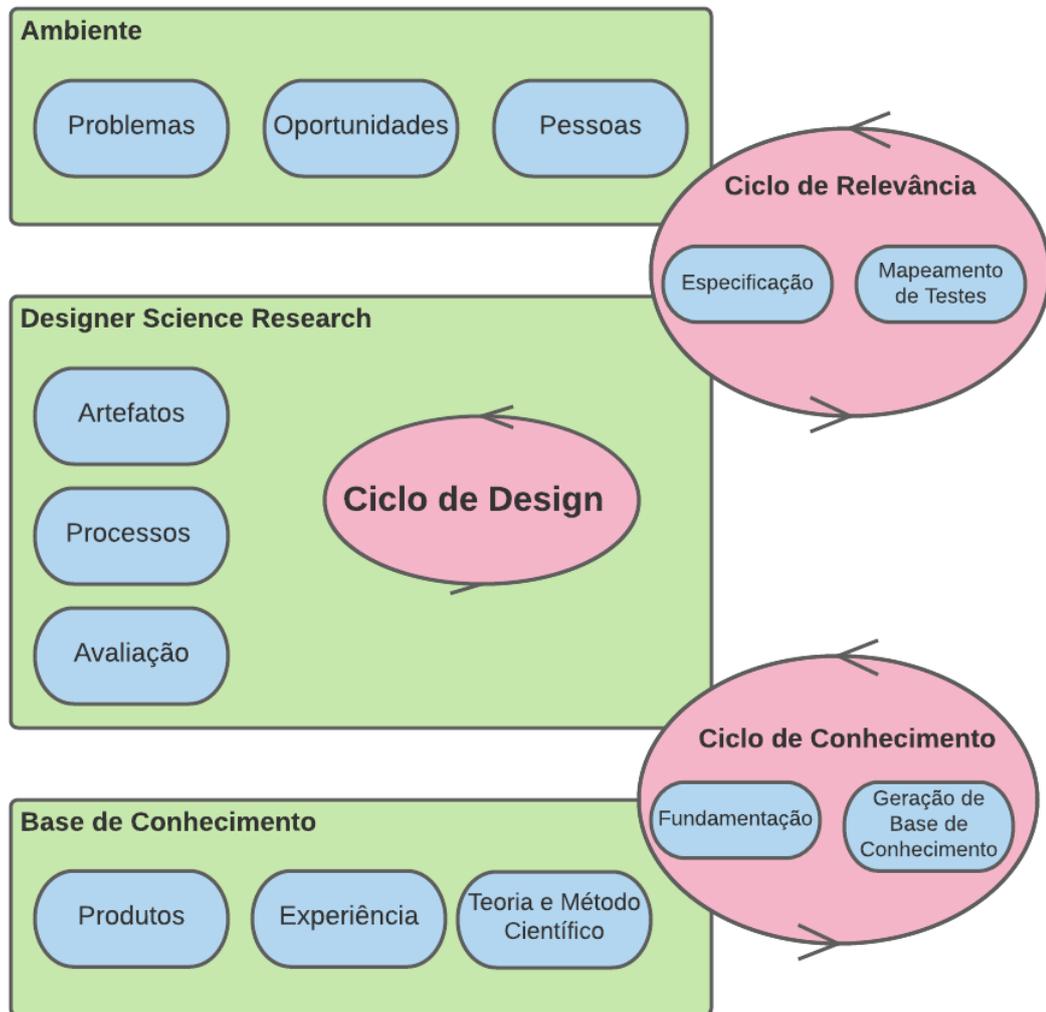


Figura 2.1: Ciclo da DSR. Fonte: Elaborada pelo autor.

lógico na investigação científica. Nesta metodologia, um dos objetivos é a geração de um artefato que agregue valor à área de conhecimento em estudo, ou seja, que, de fato, o artefato produzido tenha um emprego e valor real na sociedade (HEVNER, 2007). Ao construir e avaliar o artefato – e valor para a área -, a DSR busca gerar conhecimento científico ao final de cada ciclo.

2.2 Internet das Coisas (IoT)

O conceito de IoT pode ser considerado como uma evolução de dispositivos tecnológicos como, por exemplo, sensores, *hardware*, semântica, processamento, armazenamento, comunicação, entre outros. Esses dispositivos, ao serem conectados em um mesmo ambiente,

representam o futuro da computação e das comunicações (TAN; WANG, 2010).

Em decorrência da possibilidade de conexão de diversos dispositivos, deu-se início a era da ubiquidade, sendo esta a principal característica e base fundamental para o surgimento da IoT (TIBURSKI et al., 2015). Na IoT, emerge-se uma gama de aplicações ou funcionalidades que podem ser exploradas pela sociedade, como, por exemplo, a coleta de dados de pacientes e monitoramento de idosos, sensoriamento de ambientes de difícil acesso e inóspitos, entre outras (SUNDMAEKER et al., 2010).

2.3 Veículo elétrico

O veículo elétrico foi desenvolvido no século XIX e apresenta, por concepção, uma bateria que armazena energia suficiente para alimentar o motor elétrico, que tem por finalidade mover o veículo. (BARAN; LEGEY, 2011) mencionam que, apesar do longínquo tempo de criação do veículo elétrico, sua adoção perdeu forças devido a rápida ascensão da *Ford Motor Company*, idealizada por Henry Ford. Naquela época, o veículo a combustão tinha um desempenho melhor e percorria maiores distâncias que os veículos elétricos daquela época.

O renascimento de projetos relacionados à adoção de um veículo elétrico aconteceu no século XX. Isso se deve ao surgimento de temáticas envolvendo a exploração de recursos naturais, dando ênfase à exploração de petróleo. Neste período, foi considerado que o veículo a combustão era um dos maiores contribuintes para a poluição atmosférica (SITARZ, 1993). Embora o veículo elétrico necessite de energia elétrica para ser recarregado, segundo (NOVAIS, 2016), numa hipótese de converter toda frota de veículos a combustão para elétrico e, posteriormente, utilizar o combustível fóssil destes veículos para alimentar usinas termoelétricas, teríamos uma economia de 50% dos gastos com locomoção da frota mencionada.

Nesse sentido, aliado às questões de Sustentabilidade, o veículo elétrico ressurgiu como uma forma de dar continuidade a mobilidade urbana, dado que o veículo elétrico não emite qualquer poluente nocivo à atmosfera. A característica de emissão zero tem justificativa com base em avanços tecnológicos alcançados em seus componentes (BARASSA et al., 2015).

2.4 Sustentabilidade aplicada à Engenharia de Sistemas

O termo Sustentabilidade pode ser entendido como “a habilidade de atender às necessidades presentes sem comprometer a habilidade de futuras gerações de satisfazer suas próprias necessidades” (WCED, 1987). Essa habilidade inclui a satisfação de quatro dimensões: ambiental, social, econômica e humana (WCED, 1987). A sustentabilidade, portanto, tem-se tornado um aspecto fundamental para a manutenção da sociedade como um todo.

Para tanto, é perceptível um aumento das preocupações envolvendo os impactos das atividades humanas, bem como a adoção de práticas que visem a redução do consumo e aumento da eficiência energética de vários dispositivos. Segundo (BOMFIM et al., 2013), os sistemas computacionais estão intimamente ligados às áreas críticas para o gerenciamento da Sustentabilidade. Com isso, tais sistemas são fundamentais para apoiar e controlar o uso eficiente de recursos em diversas áreas como, por exemplo, *Smart Cities*¹.

Por fim, tem-se que sistemas computacionais tem um papel fundamental para a adoção de práticas sustentáveis, visto que, atualmente, a tecnologia está presente nos diversos setores da economia. Neste contexto, a Engenharia de *software* sustentável não atua somente em sistemas de *software*, mas também, em processos de negócios e tomadas de decisão relacionados aos aspectos de Sustentabilidade. Sendo assim, este trabalho objetiva-se relacionar aspectos de Sustentabilidade com o suporte ao uso de veículos elétricos, visto que, dar suporte à adoção de veículos elétricos pode ser considerada uma forma de apoiar a minimização dos índices de poluição encontrados na atmosfera.

2.5 Trabalhos Relacionados

Com o objetivo de facilitar o entendimento do indicador de carga de um veículo elétrico, (LUNDSTRÖM, 2014) propôs o desenvolvimento de um aplicativo *mobile* capaz de fornecer informações mais precisas a respeito do indicador de carga da bateria veicular. Nesta solução, o dispositivo móvel foi conectado diretamente à porta de transmissão de

¹<https://ois.sebrae.com.br/publicacoes/smart-cities/>. Acesso em: Agosto de 2021.

dados do veículo por meio de um adaptador *bluetooth*. O autor ressalta que este indicador sofre variação devido a fatores como, por exemplo, uso de sistema de climatização e congestionamentos urbanos. Seguindo o estudo desta problemática, este trabalho propõe o desenvolvimento de uma aplicação web para captar e enviar dados de consumo e nível de bateria em tempo real de um veículo elétrico em funcionamento. Nossa solução, diferente da solução apresentada por (LUNDSTRÖM, 2014), permite ao usuário ter uma flexibilidade em relação ao momento de usar a aplicação, ou seja, o usuário não carece de estar no veículo para visualizar os dados captados pelo aplicativo.

Através de um estudo de níveis de estresse relacionado à condução de um veículo elétrico, (FRANKE et al., 2012) conduziram uma pesquisa de campo que mapeou, por meio de um *framework*, a existência de um nível de autonomia confortável e variável pelo perfil do condutor, ou seja, a autonomia é caracterizada como um dos fatores fundamentais ao se optar por um veículo elétrico. Os autores concluíram que, considerando o contexto de veículo elétrico, se faz necessária uma readequação no conceito de uso e reabastecimento. Levando em conta os resultados obtidos pelos autores, este trabalho propõe uma aplicação para dar suporte ao gerenciamento de um veículo elétrico. A aplicação recebe dados gerados durante a condução que, posteriormente, poderão ser usados para gerar informações relevantes que irão apoiar o condutor no gerenciamento da carga e uso do veículo.

Em sua pesquisa, (SCHULTIS, 2021) aborda a problemática envolvida no aumento de estações de recarga de veículos elétricos e o impacto deste movimento crescente no sistema energético. Segundo o autor, este aumento viola limites operacionais em redes de baixa tensão, o que afeta diretamente o carregamento de um veículo elétrico. Com isso, o autor propõe um método para detecção de congestionamentos energéticos e redução do consumo de energia de estações de recargas envolvidas nessas situações. O método proposto mostrou uma eficácia de 73,03% em simulações computacionais. Como contribuição ao trabalho de (SCHULTIS, 2021), este trabalho visa a construção de um aplicativo para dar suporte ao uso de veículos elétricos objetivando o não esgotamento total da carga e uma diminuição em relação ao uso de estações de recarga.

2.6 Considerações Finais do Capítulo

Com o estudo destes conceitos teóricos e trabalhos relacionados, percebe-se uma conexão entre veículos elétricos, IoT e Sustentabilidade. Além disso, é possível constatar novos desafios que surgem com a adoção de veículos elétricos. Esses desafios podem ser desde uma readequação em redes elétricas de alta tensão, até o gerenciamento que cada condutor precisa realizar para conduzir o veículo de maneira segura. Sendo assim, busca-se explorar o problema de pesquisa na concepção de uma solução para apoiar a condução de um veículo elétrico que será apresentada no Capítulo 3.

3 Material e Métodos

Neste capítulo, apresenta-se o planejamento para o desenvolvimento e testes experimentais da solução proposta neste trabalho. A última Seção traz uma reflexão a respeito da longevidade almejada para esta solução.

A Figura 3.1 ilustra a solução real idealizada, ou seja, caso ela fosse aplicada em um contexto real. A idealização deste cenário tornou-se essencial para que se conseguisse concluir o mapeamento de funcionalidades e regras de negócio desta solução.

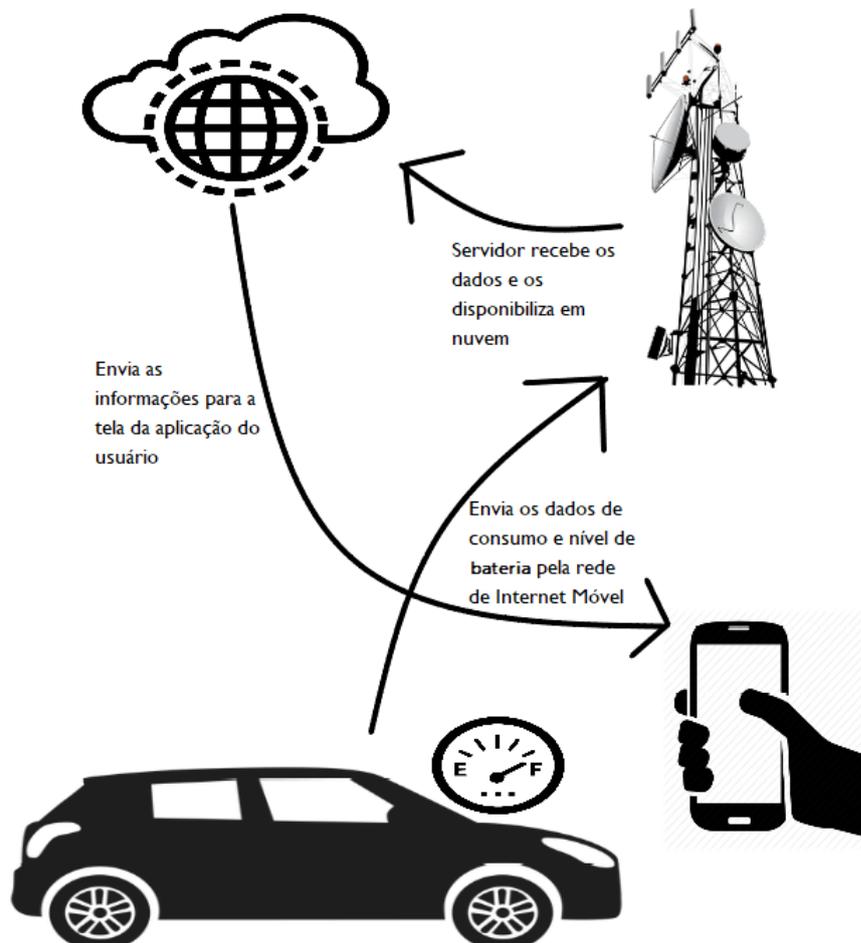


Figura 3.1: Visão geral da solução proposta. Fonte: Elaborado pelo autor.

A Figura 3.1 ilustra um cenário no qual o veículo elétrico, dotado de um dispositivo com acesso a Internet, envia os dados gerados pelo veículo em movimento para um serviço hospedado em nuvem. A aplicação, também hospedada em um servidor na nuvem, realiza o acesso ao servidor que provêm os dados e os disponibiliza na tela do usuário.

Nesta concepção, é possível realizar um gerenciamento dos dados do veículo *anywhere*, ou seja, em qualquer lugar que se tenha acesso a Internet. Por meio do acesso à aplicação via dispositivo móvel, o condutor do veículo elétrico poderá ter acesso às informações de seu veículo e, posteriormente, tomar decisões assertivas no que é tangente ao gerenciamento do veículo.

Ademais, por meio de um monitoramento constante dos dados de consumo e nível de bateria, nossa solução tende dar suporte ao condutor do veículo, visto que, com o uso da aplicação desenvolvida, o condutor irá conseguir obter dados que o irão auxiliar na tomada de decisão de recarregar o veículo ou, simplesmente, usá-lo.

3.1 Planejamento

No Planejamento para o desenvolvimento da solução proposta neste trabalho, pensou-se no escopo do projeto, ou seja, qual seria a contribuição deste trabalho e qual a importância do mesmo para o meio. Além disso, identificou-se limitações técnicas, visto que para a fonte de dados não tínhamos um veículo elétrico real. Foi necessário, portanto, o uso de um *software* de simulação de contexto capaz de gerar tal fonte de dados. Sanadas as limitações relacionadas à fonte e geração de dados, decisões de projeto foram tomadas para prover um ambiente experimental satisfatório.

A Figura 3.2 ilustra através de uma modelagem BPMN², o planejamento do projeto.

3.1.1 Identificação das necessidades da solução proposta

Para identificar e mapear os requisitos do projeto, levou-se em conta a necessidade de gerar dados de consumo e nível de bateria de um veículo elétrico em funcionamento. Neste projeto, buscou-se simular os parâmetros veiculares em um *software* de simulação de contexto. Este simulador também foi utilizado como meio para a gravação dos dados simulados em uma base de dados externa.

O *software* de simulação escolhido, denominado (SIAFU, 2007), dispõe de agentes

²<http://www.dpo.unb.br/images/phocadownload/dpr/biblioteca/BPMN.pdf>. Acesso em: Agosto de 2021.

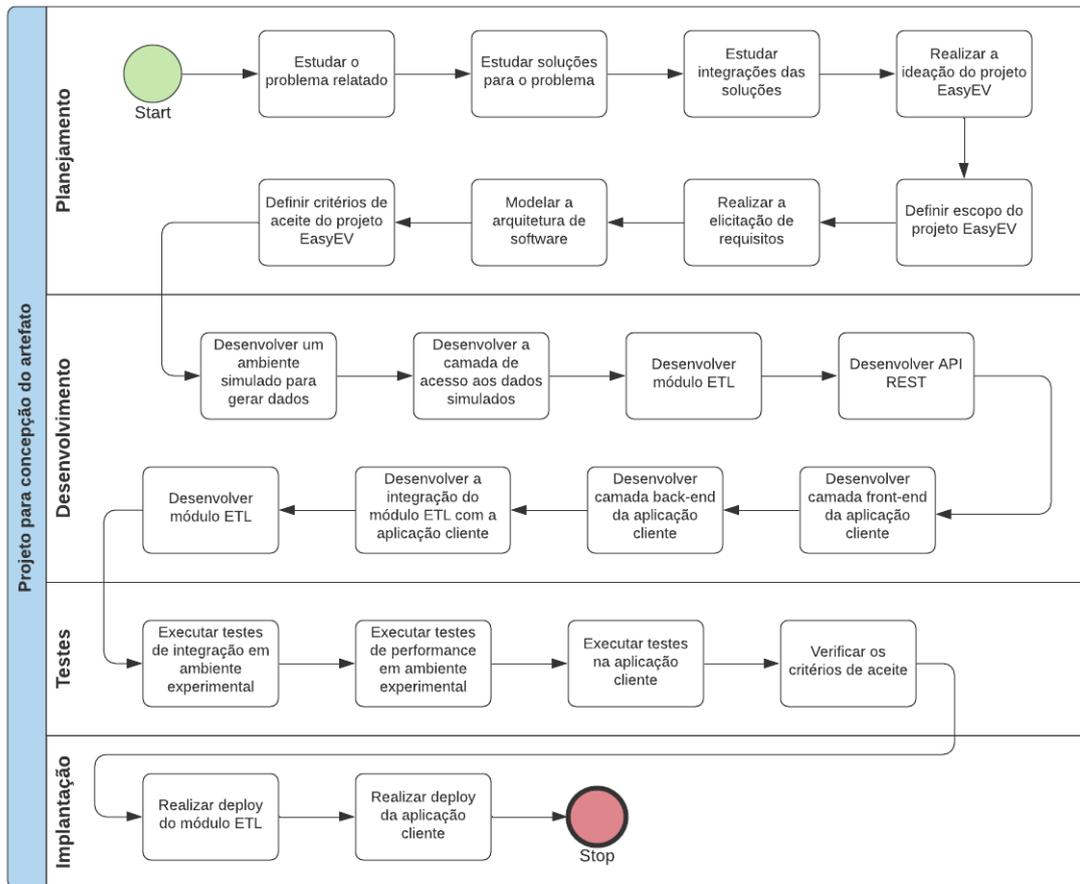


Figura 3.2: Planejamento do projeto proposto para o desenvolvimento da solução apresentada neste trabalho. Fonte: Elaborado pelo autor.

customizáveis que interagem entre si e com o ambiente simulado. Além disso, o *software* possibilita simulações em ambientes reais, dando credibilidade para projetos na área de IoT. Este *software* tem sido usado com frequência em trabalhos de grupos de pesquisa como, por exemplo, em (NASCIMENTO et al., 2020). A Figura 3.3, abaixo, ilustra a tela de funcionamento do simulador *Siafu*. No canto superior direito, é possível perceber a data e hora da simulação em execução. Além disso, logo abaixo, é possível perceber as informações do veículo simulado como, por exemplo, o nível de bateria atual, o consumo de bateria registrado naquele instante, placa e posição geográfica do veículo.

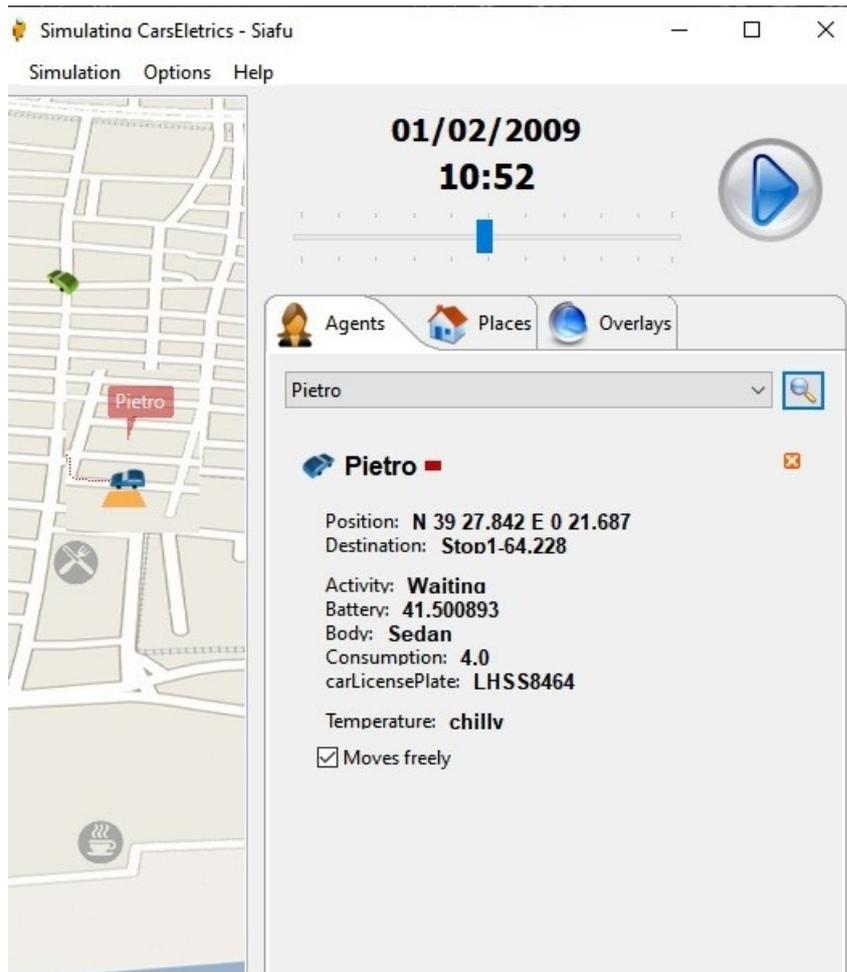


Figura 3.3: Captura de tela do simulador *Siafu*.

3.2 Solução Proposta

A solução, denominada *EasyEV*, foi desenvolvida para apoiar o condutor em direção a um melhor entendimento dos dados providos pelo veículo. Para tanto, isto implica em (i) utilizar um simulador de contexto para a geração de dados de um veículo elétrico, (ii) implementar uma camada *middleware*, denominada *DataEV System* com objetivo receber, transformar e persistir os dados disponibilizados pelo simulador. Por fim, (iii) desenvolver uma aplicação cliente para recuperar os dados armazenados e gerar informações através de gráficos e relatórios.

Para o desenvolvimento, as regras de negócios, os requisitos funcionais e não funcionais, a arquitetura de *software* e as decisões técnicas de projeto, foram especificadas.

3.2.1 Regras de Negócio

A Tabela 3.1, detalha as regras de negócio elencadas neste projeto:

Regra de Negócio	Descrição
RN01	O sistema armazenará os seguintes dados do usuário: <ul style="list-style-type: none">● Nome● E-mail● Senha
RN02	O sistema armazenará os seguintes dados do veículo: <ul style="list-style-type: none">● Marca● Modelo● Placa● Código de segurança
RN03	O sistema gerará um relatório com as seguintes colunas: <ul style="list-style-type: none">● Condutor● Veículo● Consumo● Nível de Bateria
RN04	O sistema irá trabalhar o tempo de sincronização com o módulo ETL em milissegundo.
RN05	O sistema terá um registro de conexão com o módulo ETL com as seguintes informações: <ul style="list-style-type: none">● Nome do módulo ETL● Condutor● Data da última atualização

Tabela 3.1: Regras de Negócio. Fonte: Elaborado pelo autor.

3.2.2 Requisitos Funcionais

A Tabela 3.2, detalha os requisitos funcionais elencados neste projeto:

Requisito Funcional	Descrição
RF01	O sistema deve permitir o cadastro de usuário através de um link na página inicial da aplicação
RF02	O sistema deve permitir o cadastro de um veículo elétrico por usuário
RF03	O sistema deve permitir a alteração de dados cadastrais do usuário
RF04	O sistema deve permitir a alteração e exclusão de veículos elétricos sincronizados com usuário ativo na aplicação
RF05	O sistema deve exibir na tela principal um gráfico de consumo de bateria do veículo elétrico
RF06	O sistema deve exibir na tela principal um gráfico de nível de bateria do veículo
RF07	O sistema deve exibir na tela principal as informações de conexão com o módulo ETL
RF08	O sistema deve permitir que o usuário manipule o intervalo de atualização dos dados persistidos pelo módulo ETL
RF09	O sistema deve fornecer um relatório com as informações coletadas do módulo ETL
RF10	O sistema deve permitir a recuperação de senha de um usuário

Tabela 3.2: Requisitos Funcionais. Fonte: Elaborado pelo autor.

3.2.3 Requisitos Não Funcionais

A Tabela 3.3, detalha os requisitos funcionais elencados neste projeto:

Requisito Não Funcional	Descrição
RNF01	A aplicação deverá ter uma tela com <i>layout</i> responsivo para apoiar a usabilidade da aplicação em diferentes tipos de dispositivos
RNF02	Os gráficos da aplicação deverão receber dados em formato <i>JSON</i> para promover a portabilidade dos dados
RNF03	A aplicação deverá usar um banco de dados relacional para dar garantia à escalabilidade da solução
RNF04	A aplicação deverá exibir <i>Tooltips</i> , caso ocorra alguma exceção, para promover a usabilidade .
RNF05	A aplicação deverá ser implantada em plataforma em nuvem para dar suporte à escalabilidade no processamento dos dados
RNF06	A aplicação deverá usar, entre os módulos da aplicação, o protocolo de comunicação HTTP como forma a garantir a interoperabilidade .

Tabela 3.3: Requisitos Não Funcionais. Fonte: Elaborado pelo autor.

3.2.4 Arquitetura

O projeto foi desenvolvido em módulos, aplicando uma arquitetura em camadas, onde cada camada é responsável por desempenhar uma função específica. Esta arquitetura foi

baseada no trabalho proposto por (GOMES et al., 2020). A Figura 5 ilustra a arquitetura desenvolvida para este projeto. Nesta figura, tem-se três camadas principais, sendo a primeira, composta pelo simulador *Siafu*, a segunda, composta pelo módulo ETL e, por fim, a última camada, composta pelo módulo cliente.

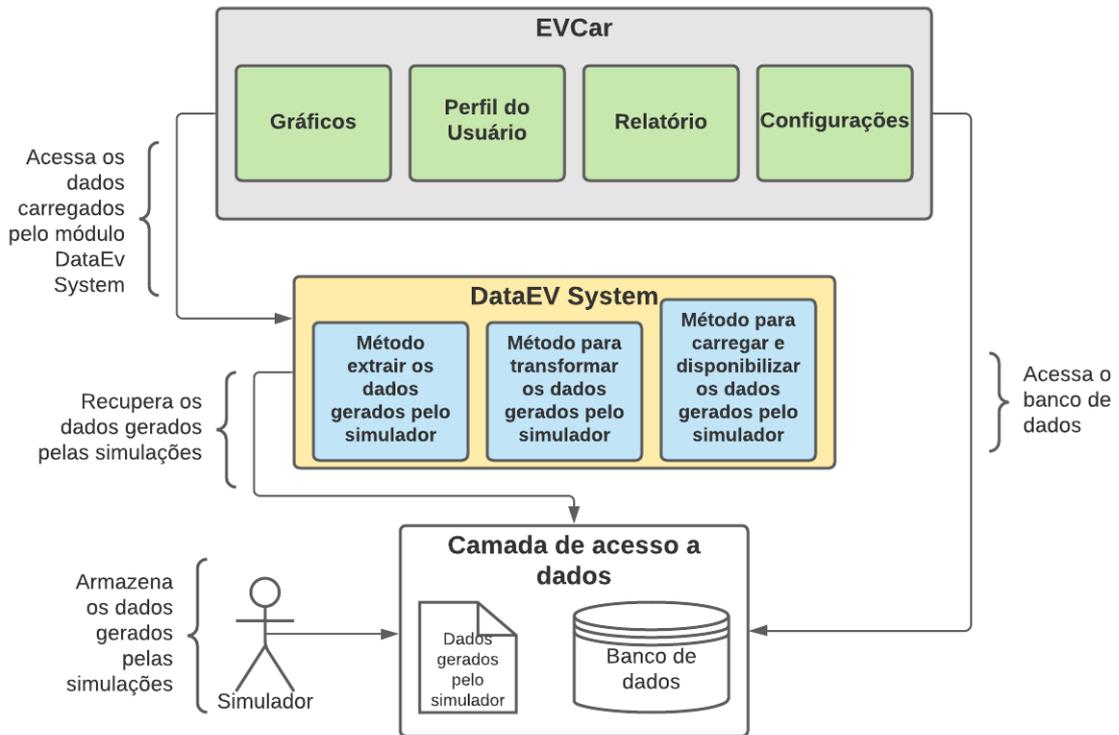


Figura 3.4: Visão geral da arquitetura. Fonte: Elaborado pelo autor.

A Figura 3.4 traz em uma perspectiva alto nível, a arquitetura proposta para o artefato criado. Primeiramente, o simulador, representado pelo item “*Simulador*”, gera e persiste dados gerados pelas simulações em um arquivo. O módulo *DataEV System* é responsável pela extração dos dados deste arquivo e pela disponibilização destes dados. Neste ponto, vale destacar os conceitos de **interoperabilidade** e **escalabilidade** que a solução possui, visto que tais dados explorados podem ser acessados por qualquer sistema - desde que o mesmo possua o *token* de acesso. Por fim, a aplicação *EasyEV*, recupera os dados fornecidos pelo módulo *DataEV System* e os exibe em forma de gráficos e relatórios personalizados. Finalizando, todos os dados de acesso de usuários e sincronização da aplicação *EasyEV* são armazenados em um banco de dados.

3.2.5 Ferramentas Utilizadas

A seguir, apresenta-se as ferramentas utilizadas durante o desenvolvimento da solução proposta. A escolha das ferramentas aqui elencadas deu importância aos estudos feitos nas seções anteriores. Além disso, visou-se aspectos técnicos aplicados à Sustentabilidade descritos em (BECKER, 2014).

Interface de Comunicação de Aplicações

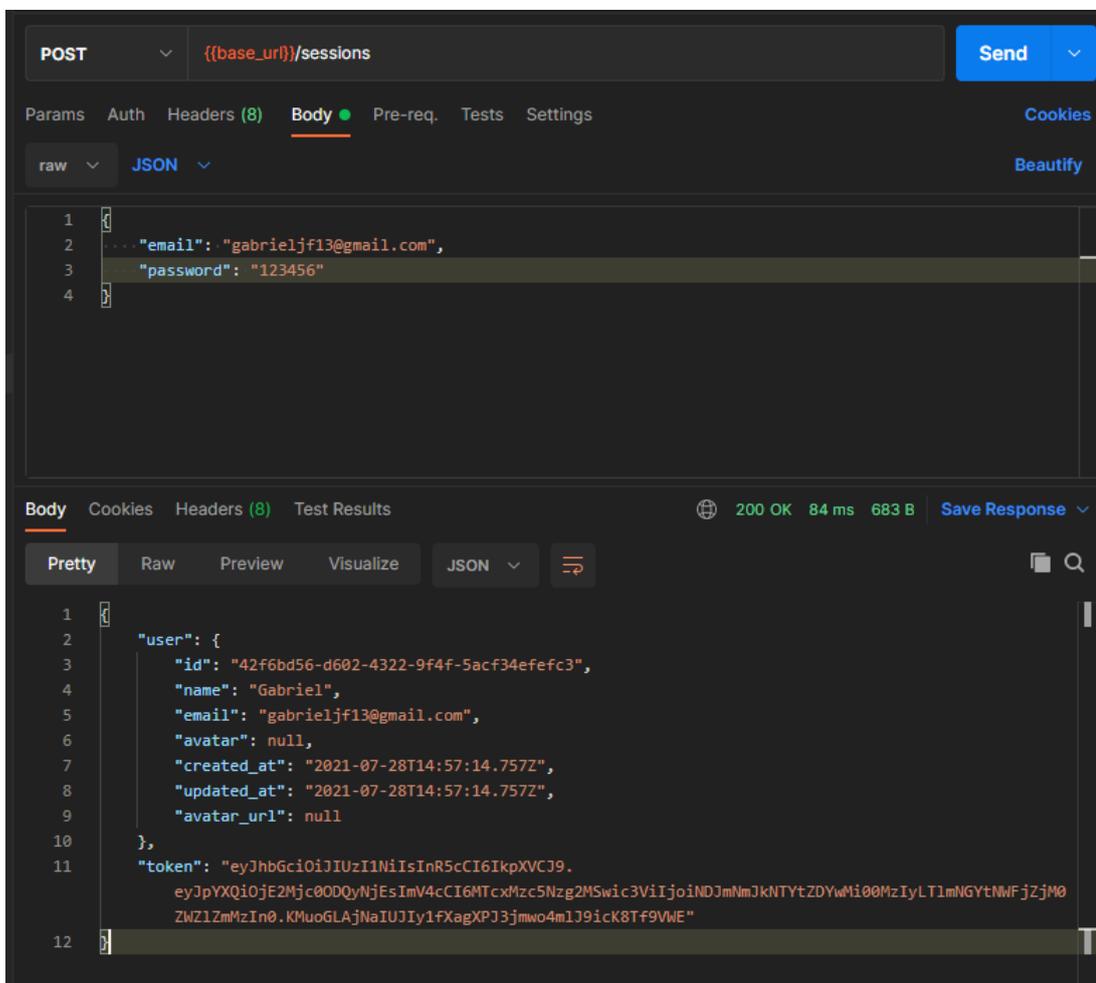
A Interface de Comunicação de Aplicações ou, em inglês “*Application Programming Interface*” (API), pode ser definida como um padrão de rotinas, instruções e configurações que servem para fornecer dados e informações relevantes para uma aplicação. As APIs desempenham um papel fundamental atualmente, visto que garantem a interoperabilidade de sistemas *web* e, neste caso, tem-se a denominação “API de Integração”. Segundo (SILVA, 2021), “APIs de Integração” permitem integrar de forma rápida e segura dois ou mais sistemas que usam linguagens de programação distintas, com o objetivo de realizar uma nova ação.

Além disso, dentre as formas de se construir uma API, existe a denominação API *RESTful*. Segundo (ANTUNES, 2021), uma API *RESTful* é uma interface que fornece dados em um formato padronizado, baseado em requisições HTTP³. Além disso, segundo o autor, as API *RESTful* aumentam a performance para situações de concorrência, quando existe um grande número de requisições, visto que utilizam uma sintaxe de construção baseada em verbos elencados abaixo.

- **GET:** A requisição é um pedido de dados, ou seja, a API recebe o pedido, realiza o acesso ao banco de dados e retorna o resultado da busca em formato JSON.
- **POST:** A requisição é uma solicitação de criação de algum recurso na *API*. Por exemplo, a criação de um novo registro na base de dados.
- **PUT:** A requisição é uma solicitação de atualização de algum recurso indicado por alguma informação.
- **DELETE:** A requisição é uma solicitação de deleção de algum recurso.

³<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>. Acesso em: Agosto de 2021.

Neste trabalho, o desenvolvimento de uma API *RESTful* possibilitou a interoperabilidade entre o simulador de contexto e os módulos da aplicação objeto deste trabalho. Além disso, este desenvolvimento garante possíveis melhorias em trabalhos futuros. A Figura 3.5 exibe o funcionamento da API *RESTful* criada neste trabalho. Nesta figura, é executada uma requisição *POST* à API *RESTful* que tem por objetivo realizar o login na aplicação. O corpo da requisição é composto por dados no formato JSON e o retorno da solicitação também é feito em formato JSON. A requisição retorna um *token* único e só válido para o usuário em questão. Desta forma, tem-se garantida a compatibilidade e interoperabilidade dos módulos criados nesta aplicação, visto que, é possível colocar a aplicação funcionamento sem que exista uma dependência entre os módulos.



```
POST {{base_url}}/sessions

{"email": "gabrieljf13@gmail.com", "password": "123456"}

200 OK 84 ms 683 B

{"user": {"id": "42f6bd56-d602-4322-9f4f-5acf34efefc3", "name": "Gabriel", "email": "gabrieljf13@gmail.com", "avatar": null, "created_at": "2021-07-28T14:57:14.757Z", "updated_at": "2021-07-28T14:57:14.757Z", "avatar_url": null}, "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IWRXVj99.eyJpYXQiOiJlMjc0ODQyNjEsImV4cCI6I6MTcxMzc5Nzg2MSwic3ViIjoiaW50NDJmNmJkNTYtZDYwMi00MzIyLTlmNGYtNWZjZjM0ZWZlZmZlIn0.KMu0GLAjNaIUJIy1fXagXPJ3jmw04m1J9icK8Tf9VWE"}
```

Figura 3.5: Requisição *POST* para API *RESTful* da aplicação *EasyEV*. Fonte: Elaborado pelo autor.

React.Js

React.js⁴, desenvolvida em Javascript⁵, é uma biblioteca para construção de interfaces de usuário. Por meio desta biblioteca, o desenvolvimento de UIs⁶ se torna mais facilitado graças a reutilização de componentes, ou seja, um botão criado para uma tela pode ser aproveitado em outra tela. Além disso, a utilização da biblioteca possibilita um ganho de performance ao carregar as telas da aplicação, visto que não é necessário que se atualize toda a tela para o carregamento de um novo dado, por exemplo.

Neste trabalho, a biblioteca é utilizada para a construção de telas, gráficos e, além disso, é utilizada para a realização de requisições a API *RESTful* mencionada na Seção anterior.

Typescript

Typescript, segundo (MELO, 2021), é um conjunto de ferramentas e formas mais eficientes de se escrever códigos *Javascript*⁷ e, além disso, adiciona recursos não presentes de maneira nativa na linguagem. O arcabouço de ferramentas presentes no *Typescript* favorece a construção de projetos em larga escala, pois contém recursos importantes como, por exemplo, tipagem estática, forte e automática, orientação a objetos e a possibilidade de mapear erros em ambiente de desenvolvido.

Neste projeto, a escolha do *Typescript* foi pautada na escalabilidade da solução desenvolvida e no tempo escasso, que acarretou na utilização de processos de desenvolvimento utilizados no mercado.

Node.js

Segundo (LENON, 2018), *Node.js* pode ser definido como um “ambiente de execução *Javascript back-end*”. Por meio desta ferramenta, é possível desenvolver aplicações e as executar sem a necessidade de se ter um *browser*. Como características principais, a ferramenta é apta em projetos de alta capacidade de escala, visto que apresenta uma arquitetura sólida e uma alta flexibilidade. Fato este que corrobora para o desenvolvi-

⁴<https://pt-br.reactjs.org/>. Acesso em: Agosto de 2021.

⁵<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: Agosto de 2021

⁶<https://www.raffcom.com.br/blog/o-que-e-ui/>. Acesso em: Agosto de 2021.

⁷<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso: Agosto de 2021.

mento de *Microservices*⁸ e arquitetura *Serverless*⁹. Neste projeto, a escolha do *Node.js* foi pautada na aplicação da solução desenvolvida e possibilidade de aplicá-la em larga escala. Com isso, foi desenvolvido um sistema que fosse apto a ser executado em ambiente *Node.js*.

3.2.6 A Aplicação *EasyEV*

A solução proposta teve como base a arquitetura descrita na Subseção 3.2.4. A primeira fase de desenvolvimento do projeto prático teve por objetivo configurar o simulador de contexto *Siafu* para que ele ficasse apto a gerar os parâmetros necessários para o desenvolvimento da aplicação.

Após a primeira fase de desenvolvimento, iniciou-se o desenvolvimento do módulo ETL *DataEV System*. A proposta desde módulo foi tornar os dados, gerados pelo simulador, acessíveis na plataforma em nuvem escolhida para o projeto. Destaca-se ainda, o fato de o simulador de contexto escolhido ser escrito em linguagem de programação distinta da linguagem de programação elencada para a concepção da aplicação. Como forma de garantir a interoperabilidade dos sistemas envolvidos na solução, desenvolveu-se a API *RESTful* mencionada anteriormente.

O módulo *back-end* da aplicação *EasyEV* foi desenvolvido seguindo a arquitetura *Model-View-Controller*¹⁰ (MVC) e escrito em *Typescript*. Tal módulo é responsável por dar suporte ao módulo *front-end* da aplicação *EasyEV*. Além disso, a comunicação padrão entre os módulos foi tratada através da API *RESTful*, conforme mencionada anteriormente.

A Figura 3.6, abaixo, ilustra o Diagrama de Classes que compõe o módulo *DataEV System* e o módulo *back-end* da aplicação *EasyEV*.

Na Figura 3.7, é exposto o Diagrama de Sequência que exemplifica o fluxo dos dados na aplicação. Neste diagrama, a sequência se inicia com a geração dos dados por meio de simulações no simulador *Siafu*. Posteriormente, o módulo *DataEV System* realiza a extração dos dados gerados pelo simulador e realiza um processo de persistência destes

⁸<https://aws.amazon.com/pt/microservices/>. Acesso em: Agosto de 2021.

⁹<https://www.redhat.com/pt-br/topics/cloud-native-apps/what-is-serverless>. Acesso em: Agosto de 2021.

¹⁰<https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc>. Acesso em: Agosto de 2021

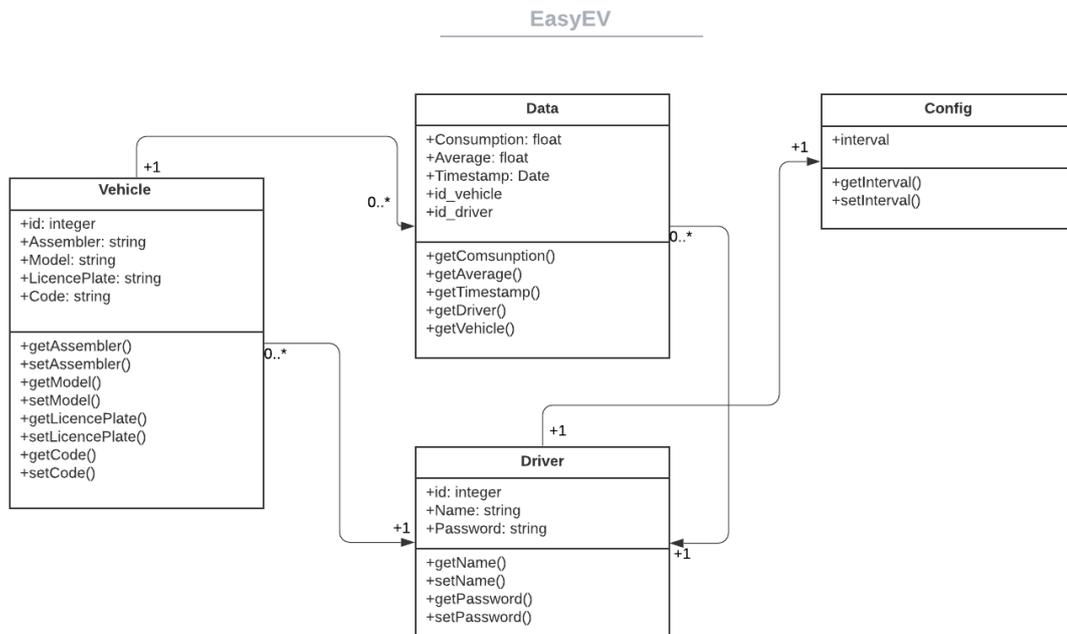


Figura 3.6: Diagrama de Classes da aplicação *EasyEV*. Fonte: Elaborado pelo autor.

dados em servidor em nuvem. Seguindo, o módulo *EasyEV* realiza requisições à API *RESTful* de acordo com as interações realizadas pelo usuário em tela.

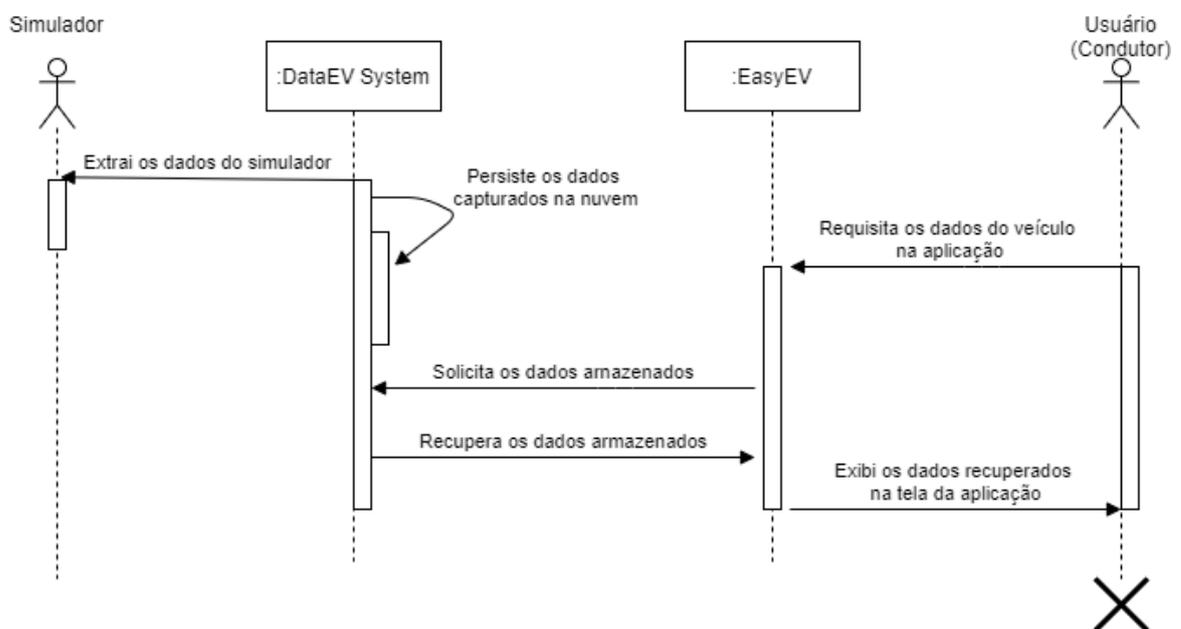


Figura 3.7: Diagrama de Sequência da aplicação *EasyEV*. Fonte: Elaborado pelo autor.

3.2.7 Interface do usuário

Para desenvolver um layout responsivo que se adequa-se na maioria dos dispositivos, esta solução fez uso do *framework front-end Bootstrap*¹¹. A utilização de componentes de interface presentes neste *framework* possibilitou que o desenvolvimento da aplicação ocorresse de maneira satisfatória, atendendo aos padrões de mercado existentes.

A Figura 3.8, abaixo, exibe a tela principal da aplicação, nesta tela, o usuário tem acesso à informações relevantes para planejar a condução do veículo elétrico. No primeiro gráfico, o condutor tem acesso ao nível de bateria levando em conta o decorrer do tempo. Neste exemplo, o condutor inicia o trajeto, às oito horas da manhã, com a carga completa e conduz o veículo até o esgotamento total da carga às oito e meia da noite. Já no segundo gráfico, o condutor tem acesso a melhor média de consumo de bateria registrada em cada dia da semana. Por meio deste gráfico, o condutor poderá optar por usar ou não o veículo elétrico em determinado dia da semana, ou seja, poderá optar por usar o veículo elétrico em um dia da semana mais propício.

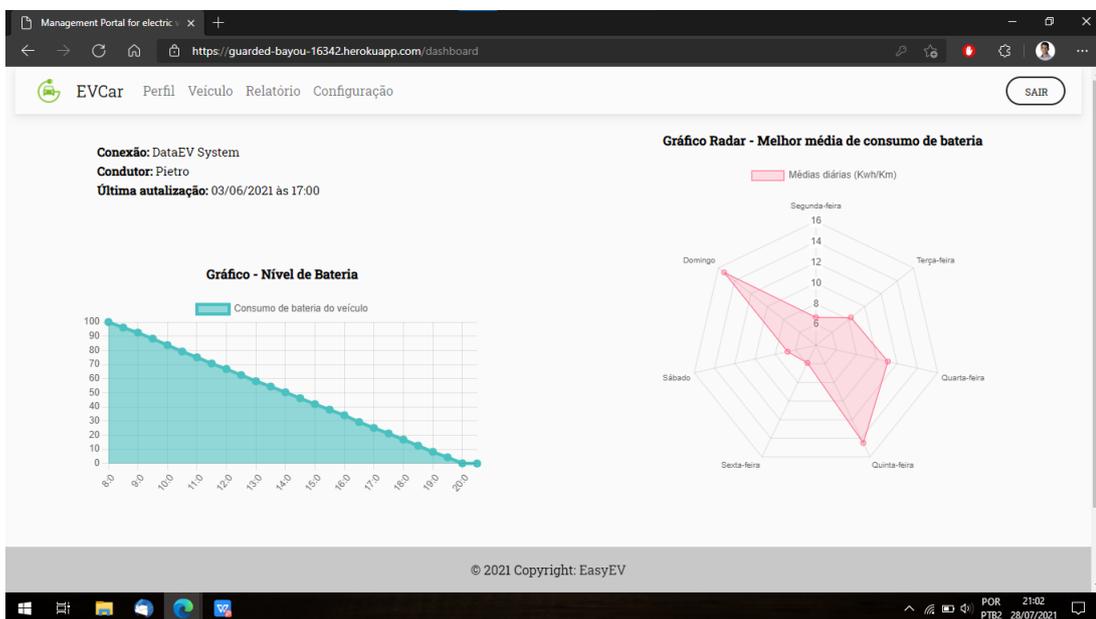


Figura 3.8: Captura da tela principal da aplicação *EasyEV*. Fonte: Elaborado pelo autor.

¹¹<https://getbootstrap.com/>. Acesso em: Agosto de 2021.

3.3 Considerações Finais do Capítulo

Este capítulo relatou o processo de ideação e desenvolvimento da aplicação proposta neste trabalho. A partir da elaboração da arquitetura do projeto, conseguiu-se definir quais ferramentas seriam apropriadas para o desenvolvimento da aplicação. Além disso, tais escolhas foram fundamentadas na possibilidade de aplicação em larga escala desta solução. É importante salientar o caráter exploratório e experimental deste trabalho que, apoiado na metodologia de pesquisa DSR, visa construir um artefato que possa ter uma real aplicabilidade na vida cotidiana. O Capítulo 4, a seguir, destaca os testes realizados em ambientes experimentais e apresenta a avaliação dos critérios de aceitação criados com base no trabalho de (SILVA; TRAVASSOS, 2020).

4 Avaliação da solução

A conclusão do desenvolvimento da aplicação *EasyEV* apresentado no Capítulo 3, permitiu traçar um método de avaliação para a solução apresentada neste trabalho. Segundo (PIMENTEL; FILIPPO; SANTOS, 2020), a avaliação de um artefato construído, seguindo a metodologia DSR, deve seguir três avaliações, sendo elas: se o artefato satisfaz aos requisitos elencados no início do projeto; se o problema foi resolvido de maneira satisfatória; e se as conjecturas teóricas parecem válidas. Neste trabalho, realizou-se as duas primeiras avaliações e a terceira será contemplada em trabalhos futuros, por meio de mais testes em ambientes experimentais.

A Seção 4.1 apresenta a avaliação dos requisitos propostos, a avaliação e execução da aplicação *EasyEV* são expostos na Seção 4.2. Por fim, a Seção 4.3 conclui o Capítulo.

4.1 Avaliação dos requisitos propostos

Para prosseguir com a avaliação dos requisitos, conduziu-se testes específicos para cada módulo da aplicação. Para o módulo *DataEV System*, conduziu-se testes com base na especificação de cenários de teste proposto por (SILVA; TRAVASSOS, 2020). A especificação de um cenário de teste tende a auxiliar a avaliação de sistemas que utilizam captação de dados de sensores. Neste trabalho, a captação de dados ocorre em ambiente simulado, através de sensores veiculares criados no *software Siafu*. Sendo assim, estipulou-se, conforme Tabela 4.1, o seguinte cenário de teste para o módulo *DataEV System*.

Cenário 1: O módulo DataEV System recebe dados gerados pelo simulador Siafu e os disponibiliza em plataforma em nuvem
Os sensores veiculares irão coletar dados considerando o <u>nível de bateria, o consumo médio de bateria, a data e hora atual e a posição geográfica do veículo</u> . Os dados capturados do veículo simulado irão ser extraídos pelo módulo <i>DataEV System</i> e serão carregados, transformados. O módulo <i>DataEV System</i> irá <u>verificar a integridade dos dados e, se estiverem corretos, irão ser persistidos no banco de dados da aplicação</u> . Se o carregamento dos dados ocorrer com sucesso, os dados <u>serão disponibilizados através da API RESTful da aplicação EasyEV</u> .

Tabela 4.1: Cenário de teste para atestar o funcionamento do módulo *DataEV System*. Fonte: Elaborado pelo autor.

Considerando o cenário exposto no na tabela 4.1, o objetivo do teste consistiu em extrair o nível de bateria, consumo médio de bateria, data, hora e posição geográfica do veículo. Posteriormente, o módulo *DataEV System* transformou e carregou os dados no banco de dados da aplicação. Por fim, após a execução deste cenário de teste, conseguiu-se verificar o correto funcionamento do módulo *DataEV System*. A Figura 4.1 ilustra o funcionamento e retorno dos dados através de requisição *GET* à API *RESTful* da aplicação.

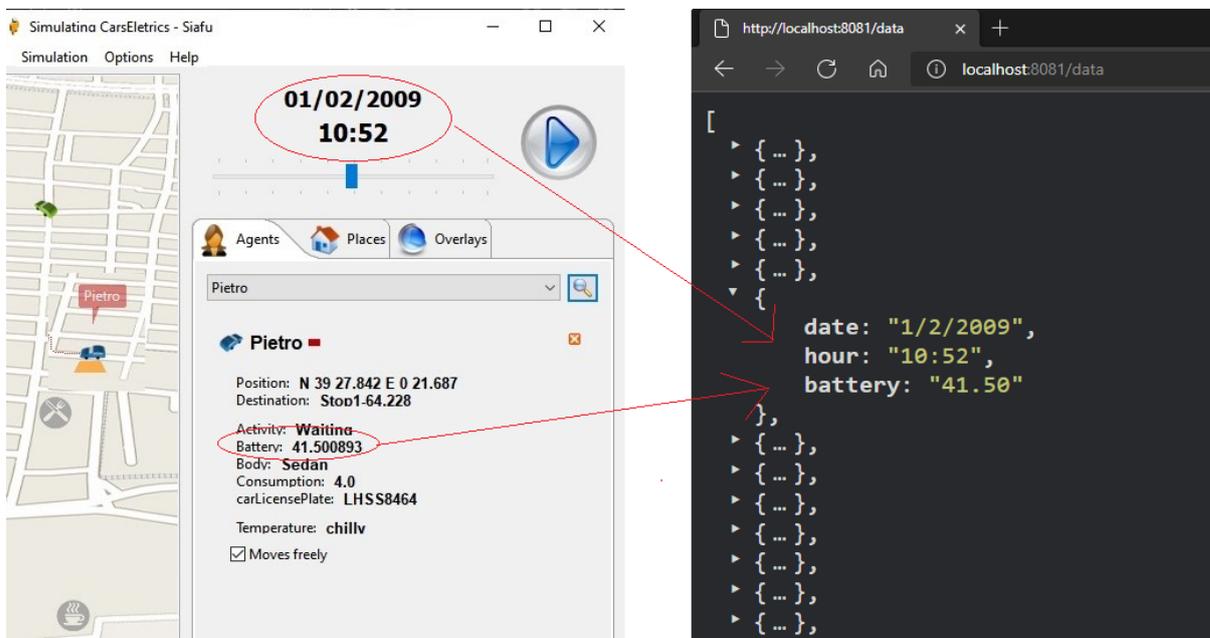


Figura 4.1: Funcionamento do módulo *DataEV System*. Fonte: Elaborado pelo autor.

Conforme a Figura 4.1, foi possível avaliar o Cenário 1 mencionado anteriormente. Neste teste, foi possível realizar simulações que gerassem dados e, posteriormente, foi possível constatar que o módulo *DataEV System* foi capaz de extrair, transformar e carregar esses dados em um banco de dados hospedado em um servidor em nuvem.

Nesta etapa, o conhecimento científico gerado possibilitou extrair dados de um simulador de contexto em tempo de execução e, além disso, persistir e disponibilizar estes dados em nuvem. Este progresso tornou-se relevante para o desenvolvimento da aplicação. Entretanto, foram observados desafios relacionados a quantidade de dados gerados em tempo de execução pelas simulações, pois em cada simulação, a captura de dados ocorre a cada minuto no tempo decorrido na simulação. Fato este, que estava ocasionando um problema de desempenho no módulo *DataEV System*.

Como forma de sanar o problema de desempenho citado, tomou-se a decisão de capturar dados em intervalos delimitados de tempo, ou seja, ao invés de realizar a captura a cada minuto, o simulador iria enviar dados que estivessem delimitados dentro do intervalo de tempo. Destaca-se ainda, que a decisão de delimitar a captura dos dados foi tomada levando em conta os recursos disponíveis para a realização deste projeto. Desta forma, nesta avaliação, a decisão de delimitar o intervalo de captura dos dados foi tomada como base nas configurações do computador - utilizado para executar as simulações - e servidor em nuvem utilizados.

Nesta avaliação, observou-se a importância e o impacto da quantidade de dados gerados em ambiente de simulação. Além disso, a delimitação do espaço de captura dos dados contribuiu para um menor tempo de resposta às requisições feitas a API *RESTful*. Por fim, após a descoberta de conhecimento e resolução dos desafios encontrados, foi possível concluir a primeira avaliação da solução proposta.

Seguindo, para a avaliação do módulo *EasyEV* - aplicação do usuário - conduziu-se testes funcionais. Segundo (NETO; DIAS, 2007), testes funcionais podem ser entendidos como “testes caixa-preta”, ou seja, o software é testado sem considerar nenhum comportamento interno. O teste é concluído com sucesso caso o teste do software termine conforme idealizado a priori.

Para a realização de testes funcionais, considerou-se a observação e experiência do

usuário ao utilizar a aplicação. Desde modo, neste trabalho, não se adotou o formalismo adotado no contexto de Teste de *Software*. A contribuição, levando em conta o contexto ao qual este trabalho está inserido, advém da utilização do artefato para resolver um desafio presente na sociedade.

Para guiar a avaliação mencionada acima, estipulou-se casos de uso para guiar o avaliador durante o teste de utilização da aplicação. Os casos de uso foram criados com base nos requisitos funcionais mencionados a priori.

A Figura 4.2, a seguir, traz o diagrama de caso de uso desenvolvido para esta aplicação. Nesta figura, o simulador é representado por um ator que realiza uma determinada ação - armazenar os dados gerados na simulação. Do outro lado, o usuário é representado por um ator que realiza ações através da aplicação *EasyEV*. Ações estas, representadas pelos casos de uso expostos na Figura 4.2.

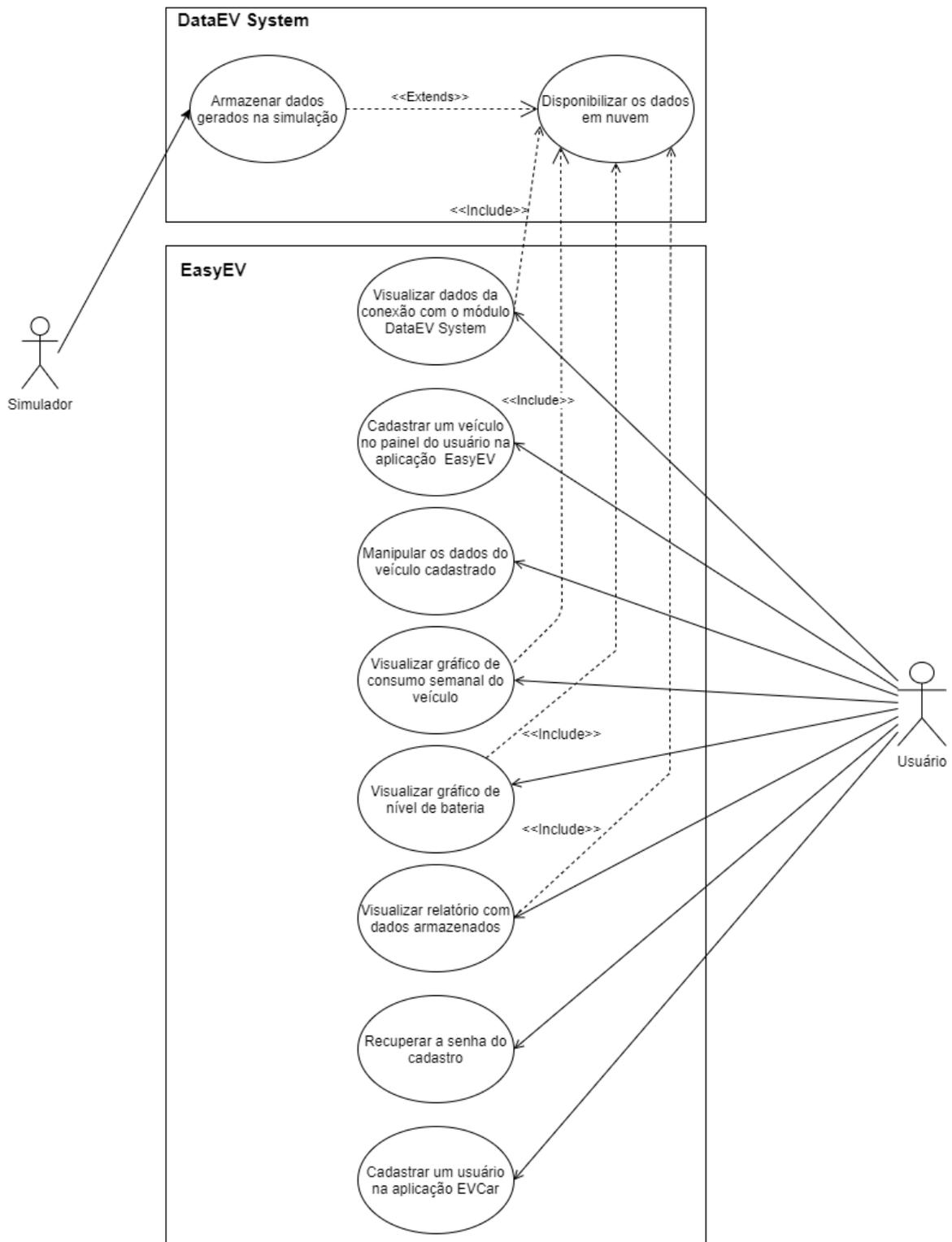


Figura 4.2: Diagrama de caso de uso da aplicação *EasyEV*. Fonte: Elaborado pelo autor.

A Tabela 4.2, a seguir, apresenta a relação entre os casos de uso e os requisitos funcionais. Por meio desta Tabela, todos os requisitos funcionais elencados na etapa de projeto da aplicação foram mapeados em casos de uso. Este mapeamento possibilitou que a avaliação fosse realizada em todos os recursos oferecidos pela aplicação *EasyEV*.

Requisito Funcional	Caso de uso
RF07	Visualizar dados da conexão com o módulo <i>DataEV System</i>
RF02	Cadastrar um veículo no painel do usuário na aplicação <i>EasyEV</i>
RF04	Manipular os dados do veículo cadastrado
RF05	Visualizar gráfico de consumo semanal do veículo
RF06	Visualizar gráfico de nível de bateria do veículo
RF09	Visualizar relatório com dados armazenados
RF010	Recuperar a senha do cadastro
RF01, RF03, RF08	Cadastrar um usuário na aplicação <i>EasyEV</i>

Tabela 4.2: Relacionamento entre casos de uso e requisitos funcionais da aplicação. Fonte: Elaborado pelo autor.

Após a conclusão dos testes funcionais, pode-se perceber que o diagrama de casos de uso exerceu um papel fundamental, visto que guiou o testador durante a etapa de teste. Durante os testes, observou-se dificuldades relacionadas à adequação dos gráficos em dispositivos móveis, visto que era necessário garantir a usabilidade da aplicação em dispositivos com resoluções de tela distintas. Como forma a garantir uma compreensão satisfatória dos dados expostos nos gráficos, buscou-se utilizar uma biblioteca de gráficos presente no *React.Js*. Com a implantação da biblioteca de gráficos e a reavaliação dos casos de uso relacionados com os requisitos funcionais RF05 e RF06, foi possível constatar uma visualização uniforme dos gráficos em diferentes dispositivos. Seguindo a avaliação, executou-se a implantação da solução em ambiente em nuvem. Tal ação e teste serão descritos na próxima seção.

4.2 Avaliação da aplicação em plataforma em Nuvem

Para conduzir os testes referentes a implantação da aplicação em ambiente de produção, escolheu-se a plataforma em nuvem *Heroku*¹². Por meio desta plataforma, foi possível realizar a implantação da aplicação em ambiente remoto. Conforme a arquitetura descrita no Capítulo 3, elencou-se que cada módulo da aplicação funcionaria um sistema individual, ou seja, cada sistema poderia ser executado separadamente.

¹²<https://www.heroku.com/home>. Acesso em: Agosto de 2021.

A Figura 4.3, abaixo, ilustra o teste realizado para verificar se a aplicação foi implantada com sucesso, ou seja, que a sua execução em ambiente remoto está funcionando conforme esperado.

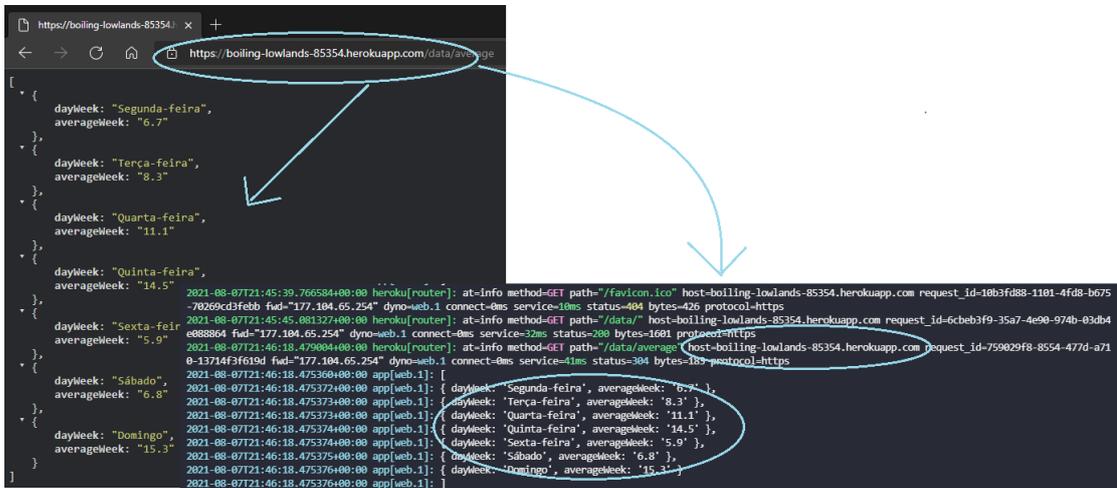


Figura 4.3: Avaliação em ambiente de produção. Fonte: Elaborado pelo autor.

Nesta figura, ilustra-se o teste realizado mediante um acesso ao endereço URL que provém os dados extraídos do simulador *Siafu*. Após o acesso, verificou *logs* gerados pela aplicação em ambiente de produção. Os *logs* confirmaram que os dados persistidos pela aplicação foram exibidos na tela do navegador. Entretanto, em alguns testes, observou-se uma lentidão ocasionada pela quantidade de dados retornados na requisição.

Como forma de tratar a eficiência observada no tempo de resposta da requisição, foram adicionados à solução, métodos que realizam a filtragem e simplificação dos dados, visto que, nem todos os dados retornados eram relevantes para o contexto ao qual a requisição foi solicitada. A Figura 4.3 exemplifica a requisição que tem por objetivo fornecer dados relacionados à maior média de consumo para cada dia da semana. Nesta figura, percebe-se que são retornados dados estritamente necessários para a aplicação do usuário. Esta melhoria, realizada durante o processo de avaliação, contribuiu para um resultado mais satisfatório. Posteriormente, concluiu-se os testes de usabilidade da aplicação em servidor em nuvem.

Por fim, com o objetivo de avaliar a aplicação *EasyEV* e a sincronia da mesma com o módulo *DataEV System*, executou-se testes de sincronização com a base de dados gerada através do fornecimento de dados do simulador *Siafu*. Nesta etapa, buscou-se atestar que os dados gerados pelo simulador eram coerentes com o usuário “logado” na

aplicação. O objetivo desta avaliação era garantir a segurança dos dados, de modo que cada condutor tivesse acesso somente aos dados de seu veículo. Como exemplo, a Figura 4.4 exemplifica o teste realizado nesta etapa. Os dados exibidos em tela coincidem com os dados exibidos pela tela do simulador - exibida na Figura 4.1.



Figura 4.4: Identificação dos dados do veículo na aplicação *EasyEV*. Fonte: Elaborado pelo autor.

4.3 Considerações Finais do Capítulo

Com a finalização das avaliações propostas para a aplicação apresentada neste trabalho, foi possível concluir que a solução, de fato, atendeu às necessidades levantadas em Capítulos anteriores. Recapitulando, esta aplicação nasceu com o objetivo de dar suporte ao condutor de um veículo elétrico tendo como base a disponibilização destes dados em ambiente em nuvem. Com o decorrer das avaliações, pode-se verificar que a aplicação recebeu dados advindos de sensores veiculares providos pelo simulador *Siafu* e, posteriormente, o módulo *DataEV System* foi capaz de realizar a captura destes dados, os tratar e disponibilizar em nuvem. Finalizando, a aplicação *EasyEV* conseguiu recuperar estes dados e prover ao usuário. Por fim, cabe ressaltar que as avaliações propostas pela DSR foram providenciais para o decorrer das avaliações deste protótipo, visto que, a avaliação da solução foi seguida por uma análise de conhecimento gerado como, por exemplo, na primeira etapa de avaliação, tem-se o conhecimento gerado ao extrair os dados de um simulador de contexto existente no mercado. Já na segunda etapa, elenca-se o conhecimento gerado ao tratar a performance das requisições à API *RESTful*. Por fim, o Capítulo 5 realiza as conclusões e faz um breve comentário sobre futuras melhorias.

5 Conclusão e Trabalhos Futuros

Neste trabalho, foi proposta uma solução para dar suporte ao uso de veículos elétricos. Para tanto, buscou-se mapear problemas e desafios diretamente ligados à condução destes veículos. Através de uma análise da literatura relacionada, pode-se perceber diversos problemas ocasionados pela falta de conhecimento do público consumidor, pela baixa autonomia destes veículos e pela falta de fidelidade dos mostradores presentes no painel de instrumentos do veículos. Além disso, estudos mostraram que a autonomia de um veículo elétrico sofre alterações que impactam diretamente na condução do veículo.

Como contribuição, este trabalho buscou entender o contexto e propor a construção de uma aplicação que pudesse dar suporte aos desafios presentes neste contexto. Para tanto, foi utilizada a metodologia *Design Science Research*. Por meio desta metodologia, o trabalho foi conduzido e foi possível conceber um artefato que pudesse, de fato, atender a uma necessidade da sociedade. Neste caso, o artefato foi caracterizado como a aplicação disponível em plataforma em nuvem¹³. Ademais, ressalta-se a importância do artefato ser acessível através da *Internet*, visto que, dessa forma, o mesmo se torna capaz de atender às necessidades de um grupo maior de perfis de usuário.

A aplicação atendeu às necessidades que foram propostas e, de fato, conseguiu exibir dados reais gerados através de simulador *Siafu*. A presença de gráficos e relatórios foram de suma importância para a real compreensão dos dados gerados. Além disso, este trabalho buscou tratar os aspectos de escalabilidade e interoperabilidade, visto que se preocupou em desenvolver uma interface de comunicação que pode ser acessada por outra aplicação, caso necessário.

Como Trabalhos Futuros, pretende-se realizar avaliações adicionais em ambiente experimental com o objetivo de simular situações do cotidiano, como, por exemplo, a simulação de congestionamentos e simulações envolvendo rotas distintas. Além disso, pretende-se aprimorar a aplicação *EasyEV* com o objetivo de fornecer informações específicas para cada público, ou seja, dar a opção do usuário escolher como deseja ter acesso

¹³<https://guarded-bayou-16342.herokuapp.com/>. Acessado em: Agosto de 2021.

às informações providas pelo seu veículo. Entende-se que devido a abrangência do público condutor de veículos elétricos, se faz necessário desenvolver uma interface que seja adequada a cada tipo de usuário. Por fim, neste grupo de pesquisa, pretende-se continuar a pesquisa realizada em (NASCIMENTO; DAVID; DANTAS, 2021).

Bibliografia

- ANTUNES, M. *API Restful: conceito, princípios e como criar*. 2021. Disponível em: <<https://www.hostgator.com.br/blog/api-restful>>.
- BARAN, R.; LEGEY, L. F. L. Veículos elétricos: história e perspectivas no brasil. *BNDES Setorial, Rio de Janeiro, n. 33, p. 207-224, mar. 2011.*, Banco Nacional de Desenvolvimento Econômico e Social, 2011.
- BARASSA, E. et al. Trajetória tecnológica do veículo elétrico: atores, políticas e esforços tecnológicos no brasil. [sn], 2015.
- BASIT, A. et al. Electric vehicles interactions for efficient energy performance within smart grid. In: IEEE. *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. [S.l.], 2020. p. 1–6.
- BECKER, C. Sustainability and longevity: Two sides of the same quality? *mental, Cite-seer*, v. 20, p. 21, 2014.
- BOMFIM, C. et al. Integrando aspectos de sustentabilidade à engenharia de sistemas. In: CEUR WORKSHOP PROCEEDINGS. *Requirements Engineering@ Brazil, ER@ BR: 2013, 16 July 2013, Rio de Janeiro, Brazil*. [S.l.], 2013. p. 62–67.
- CASTRO, L. A. F. et al. Análise de veículos elétricos no setor de logística em centros urbanos. Universidade Nove de Julho, 2019.
- FRANKE, T. et al. Experiencing range in an electric vehicle: Understanding psychological barriers. *Applied Psychology*, Wiley Online Library, v. 61, n. 3, p. 368–391, 2012.
- GOMES, E. et al. An approach of time constraint of data intensive scalable in e-health environment. In: SPRINGER. *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. [S.l.], 2020. p. 158–169.
- HEVNER, A. R. A three cycle view of design science research. *Scandinavian journal of information systems*, v. 19, n. 2, p. 4, 2007.
- HEVNER, A. R. et al. Design science in information systems research. *MIS quarterly*, JSTOR, p. 75–105, 2004.
- KUMAR, R. R.; ALOK, K. Adoption of electric vehicle: A literature review and prospects for sustainability. *Journal of Cleaner Production*, Elsevier, v. 253, p. 119911, 2020.
- LENON. *Node.js – O que é, como funciona e quais as vantagens*. 2018. Disponível em: <<https://www.opus-software.com.br/node-js/>>.
- LUNDSTRÖM, A. Differentiated driving range: Exploring a solution to the problems with the "guess-o-meter" in electric cars. In: *Proceedings of the 6th international conference on automotive user interfaces and interactive vehicular applications*. [S.l.: s.n.], 2014. p. 1–8.
- MELO, D. *O que é TypeScript? [Guia para iniciantes]*. 2021. Disponível em: <<https://tecnoblog.net/426754/o-que-e-typescript-guia-para-iniciantes/>>.

- NASCIMENTO, G. dos S.; DAVID, J. M. N.; DANTAS, M. A. Suporte à dirigibilidade de um veículo elétrico através do gerenciamento de consumo de bateria. In: SBC. *Anais Estendidos do XVII Simpósio Brasileiro de Sistemas de Informação*. [S.l.], 2021. p. 129–132.
- NASCIMENTO, M. G. do et al. Covid-19: A digital transformation approach to a public primary healthcare environment. In: IEEE. *2020 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.], 2020. p. 1–6.
- NETO, A.; DIAS, C. Introdução a teste de software. *Engenharia de Software Magazine*, v. 1, p. 22, 2007.
- NOVAIS, C. R. B. d. Mobilidade elétrica: desafios e oportunidades. 2016.
- PIMENTEL, M.; FILIPPO, D.; SANTOS, T. M. Design science research: pesquisa científica atrelada ao design de artefatos. *RE@ D-Revista de Educação a Distância e eLearning*, v. 3, n. 1, p. 37–61, 2020.
- SCHULTIS, D.-L. Sparse measurement-based coordination of electric vehicle charging stations to manage congestions in low voltage grids. *Smart Cities*, Multidisciplinary Digital Publishing Institute, v. 4, n. 1, p. 17–40, 2021.
- SIAFU. *An open source context simulator*. 2007. Disponível em: <<https://siafusimulator.org>>.
- SILVA, D. da. *O que é API? Vantagens e aplicações*. 2021. Disponível em: <<https://www.zendesk.com.br/blog/o-que-e-api>>.
- SILVA, V. M. D.; TRAVASSOS, G. H. Scenariot: Support for scenario specification of internet of things-based software systems. In: SBC. *Anais Estendidos do XI Congresso Brasileiro de Software: Teoria e Prática*. [S.l.], 2020. p. 195–209.
- SITARZ, D. Agenda 21: The earth summit strategy to save our planet. Boulder, CO (United States); EarthPress, 1993.
- SUNDMAEKER, H. et al. Vision and challenges for realising the internet of things. *Cluster of European research projects on the internet of things, European Commission, Citeseer*, v. 3, n. 3, p. 34–36, 2010.
- TAN, L.; WANG, N. Future internet: The internet of things. In: IEEE. *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*. [S.l.], 2010. v. 5, p. V5–376.
- TIBURSKI, R. T. et al. The importance of a standard security architecture for soa-based iot middleware. *IEEE Communications Magazine*, IEEE, v. 53, n. 12, p. 20–26, 2015.
- WCED, S. W. S. World commission on environment and development. *Our common future*, v. 17, n. 1, p. 1–91, 1987.