

ETL AUTOMATIZADO PARA SISTEMA DE MONITORAMENTO REMOTO DE BATERIAS DE CHUMBO-ÁCIDO

AUTOMATED ETL FOR A REMOTE MONITORING SYSTEM OF LEAD-ACID BATTERIES

Lays Leal Correia

laysleal13@gmail.com

Prof. Dr. Bruno Albuquerque Dias

bruno.dias@pesqueira.ifpe.edu.br

RESUMO

Atualmente, o uso de bancos de baterias é essencial para os mais diferentes tipos de negócios e seus equipamentos elétricos que necessitam de fornecimento de energia constante como, por exemplo, respiradores em hospitais. É inerente ao uso de banco de baterias de chumbo-ácido a dificuldade em realizar manutenções nos períodos de melhor oportunidade e, também, de identificar situações pontuais de falha elétrica. A manutenção desse tipo de instalação elétrica comumente ocorre somente após a necessidade de seu uso, ou seja, durante períodos sem fornecimento de energia da rede elétrica. Dessa forma, caso o sistema apresente falha entre as manutenções programadas, ela provavelmente só será resolvida em um momento crítico que pode comprometer momentaneamente a operação do local. Com o software descrito neste artigo é possível que um colaborador, com conhecimento técnico, de uma empresa ou organização tenha conhecimento do estado de operação do banco de baterias do local, pois o software é capaz de extrair de uma base de dados as informações das baterias, transformar e enviar esses dados para que posteriormente sejam exibidos em um aplicativo móvel. Este artigo descreve o funcionamento de um ETL (*Extract, Transform and Load*) desenvolvido na plataforma *Jupyter Notebook* utilizando a linguagem de programação *Python*. Este software é parte essencial de um sistema de monitoramento de banco de baterias de chumbo-ácido e possui capacidade de atender um usuário, empresa ou organização.

Palavras-chave: ETL. software. python. monitoramento remoto. bateria. aplicativo.

ABSTRACT

Nowadays the use of battery banks is a must for many kinds of businesses and their electrical equipment that requires constant power supply, such as respirators used in hospitals. It is not easy to do the maintenance of an emergency power supply made of batteries at the best times and identify situations of electrical failure. The maintenance of this kind of electrical system is frequently done after its use, when the site does not have power being supplied from the usual electrical network. That said, if the system fails between maintenances, it will probably only be detected at a critical moment that could momentarily compromise the company's operation. With the software described in this article, it is possible for an employee, with technical knowledge, from a company or organization to be aware of the operating status of its batteries, as the software is capable of extracting information from a database, transform it and send it back, so it can be shown using a mobile app. This article describes the operation of an ETL (Extract, Transform and Load) developed on Jupyter Notebook platform using Python. This software is an essential part of a monitoring system for lead-acid batteries and is capable to serve a single user, company or organization.

Keywords: ETL. software. python. remote monitoring. battery. mobile app.

1 INTRODUÇÃO

Em virtude dos esforços dos italianos Luigi Galvani e Alessandro Volta as células eletroquímicas foram apresentadas a humanidade no final do século XVIII. Em decorrência de uma disputa entre os dois cientistas os primeiros estudos e pesquisas sobre baterias foram divulgados. (SCROSATI, 2011)

Baterias recarregáveis são utilizadas para diversos fins como, por exemplo, celulares, carros e câmeras fotográficas. A bateria de chumbo-ácido foi inventada pelo cientista Gaston Planté em 1859 e após diversas melhorias nos seus materiais e em seu design, atualmente ela se tornou a tecnologia de armazenamento de energia mais utilizada em aplicações para carros, motocicletas e *backup* de sistemas. (SCROSATI, 2011; IBER, 2022; USGS, 2023)

Nas últimas décadas, avanços em tecnologias de IoT (do inglês, *Internet of Things*) abriram novas possibilidades para o monitoramento remoto em tempo real dos mais diversos ativos. Sensores e dispositivos de aquisição de dados podem ser instalados em qualquer local, coletando informações cruciais e transmitindo dados para a nuvem para análises posteriores. (UNIVERSITY OF SILICON VALLEY, 2021)

À medida que a demanda por fontes de energia mais limpas e eficientes continua a crescer, o papel das baterias estacionárias de chumbo-ácido na estabilização da rede elétrica e no armazenamento de energia se torna cada vez mais relevante. Portanto, o monitoramento da operação dessas baterias pode executar um papel crítico no desempenho de diversos serviços essenciais como, por exemplo, em torres de telecomunicação que precisam continuar a transmitir dados mesmo durante uma queda de energia da rede.

Baterias são dispositivos suscetíveis a falhas, pois podem perder capacidade elétrica por vários motivos, a maioria deles podem ser prevenidos através da realização de manutenções. Para auxiliar empresas a evitar problemas diversos em suas baterias de *backup*, sistemas de monitoramento remoto utilizam sensores e a Instituto Federal de Pernambuco. Campus Pesqueira. Curso de Bacharelado em Engenharia Elétrica. 25 de dezembro de 2023.

tecnologia IoT para medir grandezas físicas do dispositivo e a partir desses dados indicar quando uma manutenção é necessária.

Este trabalho apresenta o desenvolvimento de um software ETL para um sistema de monitoramento remoto de baterias de chumbo-ácido. A abordagem se concentra na extração de informações de uma base de dados em nuvem, na sua transformação, limpeza e na carga desses dados de volta para a nuvem. A implementação é realizada em *Python*, uma linguagem amplamente reconhecida e utilizada na comunidade de ciência de dados utilizando a IDE *Jupyter Notebook*.

2 FUNDAMENTAÇÃO TEÓRICA

Para melhor compreensão das atividades realizadas neste trabalho é necessário, primeiramente, conhecer alguns conceitos. Assim, neste capítulo serão explorados cinco conceitos. O primeiro é a descrição do que é e como funciona uma bateria estacionária de chumbo-ácido. O segundo abordará a definição de ETL. O terceiro conceito será sobre a linguagem de programação *Python*. O quarto apresenta a IDE *Jupyter Notebook*. O quinto trata sobre a ferramenta *Google Sheets*. Por fim, será desenvolvido um tópico sobre APIs.

2.1 Baterias estacionárias de chumbo-ácido

A bateria estacionária de chumbo-ácido é usada em sistemas solares, torres de telecomunicação, sistemas de iluminação, locais de armazenamento e fontes de energia elétrica para manter aparelhos elétricos e eletrônicos funcionando durante períodos em que a fonte principal de energia não estiver ativa.

Baterias de chumbo-ácido utilizam Dióxido de Chumbo (PbO_2) nas placas positivas, Chumbo (Pb) nas placas negativas e Ácido Sulfúrico (H_2SO_4) como eletrólito. Quando a bateria é utilizada, os materiais de ambas as placas se tornam Sulfato de Chumbo (PbSO_2) e o ácido sulfúrico torna-se Água (H_2O), quando a bateria é recarregada a reação contrária acontece. (HIOKI, 2020)

Este tipo de bateria contém elementos com placas positivas e negativas que juntas fornecem uma tensão de aproximadamente 2 Volts e a quantidade de elementos na bateria definirá sua tensão nominal. Na Figura 1 é possível visualizar uma bateria e os seus elementos.

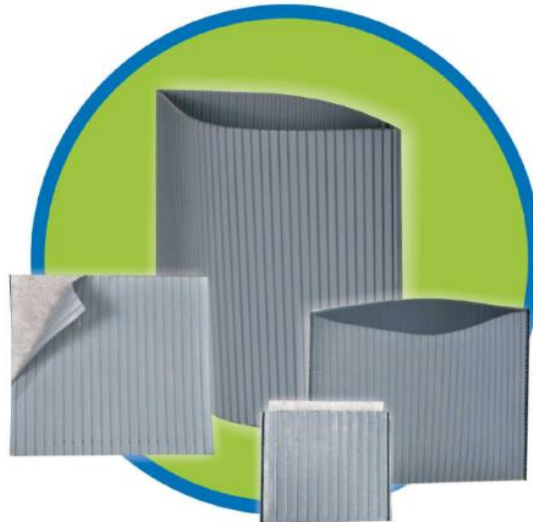
Figura 1 – Bateria estacionária TAB.



Fonte: TAB, 2023.

As placas e o eletrólito são os componentes mais importantes de uma bateria, mas outros componentes também são necessários para que a bateria funcione e seja segura. Um desses componentes é o envelope separador, que reveste algumas das placas do elemento, evitando que as placas positivas toquem as placas negativas e provoquem um curto-circuito. Envelopes separadores estão ilustrados na Figura 2.

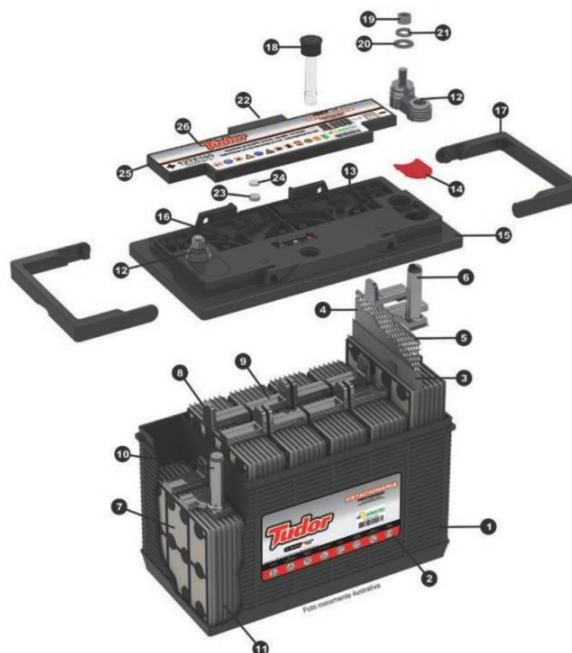
Figura 2 – Separadores de polietileno.



Fonte: Indian Batteries, 2018.

Outros componentes essenciais são a caixa, a tampa e a sobre tampa que juntas abrigam e protegem os componentes químicos que permitem o acúmulo e o fornecimento de energia elétrica. Esses e outros componentes podem ser visualizados na Figura 3.

Figura 3 – Bateria estacionária Tudor e seus componentes.



Fonte: Tudor, 2017.

O monitoramento de uma bateria como essa é realizado por meio da leitura de grandezas como tensão e corrente e, após a leitura, os dados são armazenados em nuvem para que, posteriormente, sejam processados.

2.2 ETL

Segundo Abreu (2008), ETL (do inglês, *Extract, Transform and Load*) é uma ferramenta que tem como objetivo extrair, transformar e carregar dados de uma ou mais fontes de dados de origem para um ou mais conjuntos de dados de destino. As etapas de extração e de carga são obrigatórias durante o processo, já a etapa que contempla a transformação ou limpeza dos dados é opcional.

O uso mais comum do ETL em organizações é para armazéns de dados. Um armazém de dados contém informações críticas para a tomada de decisões relacionadas as estratégias da organização. Nesta aplicação, o ETL garante que os dados que são carregados no armazém sejam precisos e confiáveis. (GAMA e ABREU, 2008; ASTERA SOFTWARE, 2020)

O processo de extração consiste em copiar dados armazenados em outros locais e convertê-los, afim de seguir para a etapa de transformação. Para isso, é necessário saber com quais objetivos o ETL será implementado, quais informações são fundamentais para o cliente e de quais fontes serão extraídos os dados (KIMBALL; CASERTA, 2004).

O processo seguinte é o de transformação e limpeza, em que os dados podem passar por conversões, correções de valores, correções de erros de digitação e outras operações que garantirão que os dados estão devidamente organizados, são confiáveis e estão no formato correto para a próxima etapa. A Tabela 1 contém um exemplo de transformação de dados.

Tabela 1 – Exemplo de transformação de datas.

| Formato Original | Formato Transformado |
|---------------------|----------------------|
| 18/05/2022 19:40:29 | 2022-05-18 19:40:29 |
| 21/10/2022 07:57:12 | 2022-10-21 07:57:12 |
| 03/11/2022 11:20:17 | 2022-11-03 11:20:17 |

Fonte: Própria, 2023.

A Tabela 1 mostra um exemplo de transformação de datas, o formato da variável deixou de ser do tipo “texto” para se tornar do tipo “data”. Esse tipo de transformação é realizada com o objetivo de aumentar a eficiência do processamento dos dados.

O último processo é o de carga, em que os dados transformados serão enviados para um ou mais conjuntos de dados de destino, substituindo ou complementando os dados já existentes. A Figura 4 representa as etapas do processo realizado pelo ETL.

Figura 4 – Etapas do processo realizado pelo ETL com exemplos.



Fonte: Própria, 2023.

Para este trabalho se fazia necessário o uso de uma linguagem de programação capaz de lidar com grandes quantidades de dados e possuísse bibliotecas para tal, de forma prática e gratuita, por esse motivo, a linguagem de programação *Python* foi escolhida para ser utilizada durante todo o desenvolvimento da ferramenta de ETL.

2.3 Linguagem de programação *Python*

Python é uma linguagem de programação interpretada, orientada a objeto, de alto nível, possui semântica dinâmica, é gratuita, fácil de aprender, tem suporte a bibliotecas e módulos que descomplicam o desenvolvimento, dão ao desenvolvedor a oportunidade de utilizar o mesmo código infinitas vezes e, além disso, os programas criados com ela são de fácil manutenção. (PYTHON, 2023a)

Ela é utilizada nas mais diversas aplicações como, por exemplo, desenvolvimento web, computação científica e desenvolvimento de jogos (PYTHON, 2023a). No campo da pesquisa e desenvolvimento, é possível programar equipamentos de bancada utilizando ela. A empresa *Keysight*, fabricante de equipamentos para testes e medições eletrônicas possui produtos, como fontes de bancada e osciloscópios, que podem ser automatizados utilizando a linguagem *Python* juntamente com a linguagem SCPI (KEYSIGHT, 2022).

Outros exemplos de empresas que adotaram o uso do Python são: *Google, Meta, Dropbox, Spotify, Reddit, Uber* e *Netflix*. Para elas a linguagem é fundamental para coleta, processamento, análise e visualização de dados, automação operacional, manutenção de servidores, administração de sistemas de recomendação, gerenciamento de sistemas de autenticação, identificação e combate a vulnerabilidades, programação assíncrona e muitas outras aplicações essenciais aos seus negócios. (SANTANA, 2023)

Python também é utilizado para pequenos projetos. Ni (2023), por exemplo, utilizou a linguagem para analisar dados sobre as preferências de usuários do *YouTube* nos Estados Unidos. Ribeiro *et al.* (2021), por sua vez, utilizou a linguagem para ler, interpretar e construir gráficos de linhas espectrais em seu trabalho, no qual construiu um espectrômetro de baixo custo com o objetivo de utilizá-lo para ensinar o modelo atômico de Bohr a estudantes.

É comum que desenvolvedores independentes ou que atuam em empresas dos mais diversos segmentos utilizem a linguagem *Python* juntamente com a plataforma *Jupyter Notebook*, pois além também ser gratuita ela é amplamente utilizada para processamento de grandes quantidades de dados, características indispensáveis para o desenvolvimento de uma ferramenta ETL.

2.4 *Jupyter Notebook*

Jupyter Notebook é uma ferramenta que faz parte do *Project Jupyter*, um projeto de código aberto iniciado em 2014 e que, nos dias atuais, é usado principalmente para projetos de ciência de dados e ciência da computação, oferecendo suporte a qualquer linguagem de programação. (JUPYTER, 2023)

É popular entre os profissionais de ciência de dados por sua flexibilidade, interatividade e também por ser possível escrever código de programação, processá-lo, visualizar os resultados e incluir textos explicativos em um único documento (PARASHAR, 2022). A Figura 5 ilustra um código de programação na linguagem *Python* no ambiente *Jupyter Notebook*.

Figura 5 – Linhas de código executadas individualmente.

```

In [1]: 7**4
Out[1]: 2401

In [1]: s = 'Olá, Pai!'

In [2]: s.split()
Out[2]: ['Olá,', 'Pai!']

In [3]: planeta = "Terra"
        diametro = 12742

In [4]: print("O diâmetro da {} é {} quilômetros.".format(planeta,diametro))
O diâmetro da Terra é 12742 quilômetros.

In [4]: lst = [1,2,[3,4],[5,[100,200,['ola']],23,11],1,7]

In [5]: lst[3][1][2][0]
Out[5]: 'ola'

```

Fonte: Própria, 2023.

A plataforma pode ser utilizada para os mais diferentes objetivos. Bastos e Lima (2023), por exemplo, utilizaram o *Jupyter Notebook* para processamento de dados em um projeto de detecção de *fake news*, já Sigalingging *et al.* (2023), realizou toda a modelagem matemática de seu trabalho de sistemas de previsão de produtividade energética de chalotas com o *Jupyter Notebook*.

Ela também permite que as linguagens utilizem livremente suas bibliotecas e módulos nativos ou desenvolvidos pela comunidade e disponíveis para *download* através da *internet*.

2.5 Google Sheets

Segundo Silva (2013), “Uma Planilha eletrônica é um tipo de programa de computador que utiliza tabelas para realização de cálculos ou apresentação de dados. Cada tabela é formada por uma grade composta de linhas e colunas.”

O *Google Sheets* é um programa de planilhas eletrônicas que opera *online* e permite colaboração em tempo real entre usuários a partir de qualquer dispositivo com acesso à *internet*. Ele possui vários recursos para criação e edição de planilhas, como a utilização de fórmulas, geração de gráficos e preenchimento inteligente de células. (GOOGLE, 2023b)

A base de dados contendo medições realizadas em uma bateria utilizada para este trabalho está alocada em uma planilha do *Google Sheets*, por essa razão, foi fundamental a utilização de uma API para a acesso a planilha.

2.6 API

API (do inglês, *Application Programming Interface*) é uma interface de programação de aplicações. Desenvolvidas majoritariamente por empresas, são extensivamente utilizadas para acesso a dados e serviços por aplicativos e softwares. Algumas APIs são abertas a todos os desenvolvedores interessados, outras são restritas apenas a parceiros e colaboradores da empresa que desenvolveu a ferramenta. (JACOBSON; BRAIL; WOODS, 2014)

APIs são ferramentas amplamente utilizadas para desenvolvimento de software, pois são consistentes, previsíveis e possuem ampla documentação. A *AccuWeather*, por exemplo, é uma empresa estadunidense que presta serviços de previsão do tempo, suas APIs são capazes de oferecer dados de diversas localidades, enviando informações de condições climáticas que são atualizadas a cada hora. (ACCUWEATHER, 2023)

Configurar e executar um aplicativo ou software que utiliza uma API é simples, para isso o desenvolvedor deve visitar o site da API e acessar sua documentação, nesta página a empresa deve explicar sobre a implementação da API e fornecer exemplos do que a ferramenta faz e como ela deve ser chamada para a ação através do código de programação.

3 METODOLOGIA

Neste capítulo serão apresentados o equipamento utilizado para coletar dados das baterias de chumbo-ácido e a metodologia para que o ETL seja capaz de extrair, tratar, limpar e carregar os dados obtidos, incluindo as bibliotecas e módulos *Python* escolhidos, e como tudo isso foi implementado. É importante notar durante a leitura que o ETL é capaz de trabalhar com qualquer base de dados que possua formato compatível com a ferramenta.

Almeida, Pereira e Dias (2022), desenvolveram um equipamento para monitoramento remoto de baterias de chumbo-ácido utilizando diversos sensores. Para medir tensão foi utilizado o sensor INA219, para medir corrente o sensor Dallas DS18B20 e para temperatura o Neo-6M-0-001, todos conectados a um microcontrolador ESP32, que possui módulo *Wi-Fi* integrado, o que facilita o envio dos valores obtidos através das medições para uma base de dados. O protótipo do equipamento pode ser observado na Figura 6.

Figura 6 – Protótipo da unidade de aquisição de dados.



Fonte: Almeida; Pereira; Dias, 2022.

Esse equipamento foi utilizado para coletar e enviar informações para a base de dados utilizada neste trabalho para testar refinar o ETL, por isso ele é de grande importância para a viabilização deste trabalho.

3.1 Extração dos dados coletados

A unidade de aquisição de dados envia todos os valores medidos para uma planilha *Google Sheets* armazenada na nuvem, cada medição realizada pelo equipamento envia onze tipos de dados diferentes. Na planilha, cada dado permanece em uma coluna e cada medição ocupa uma linha, como ilustra a Figura 7.

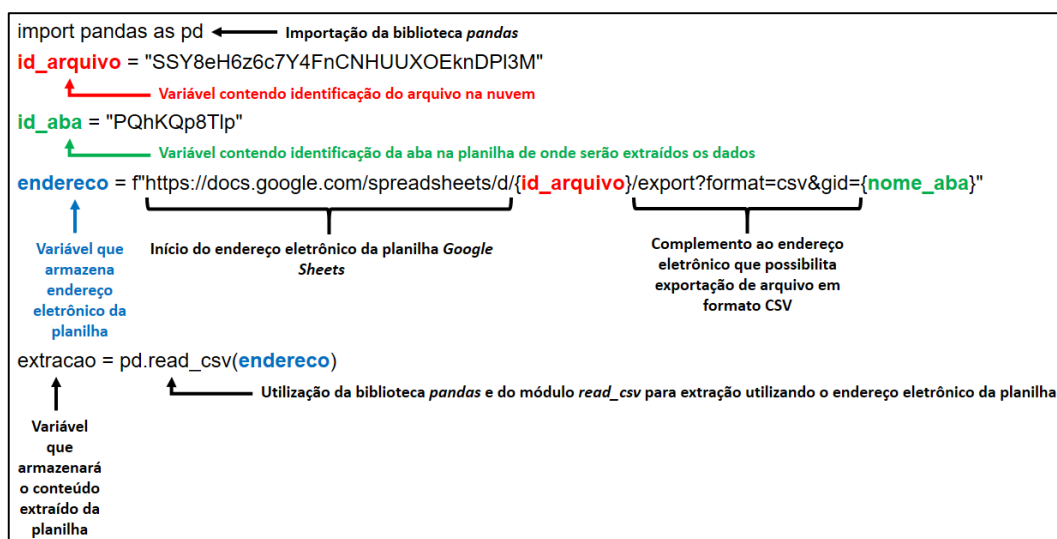
Figura 7 – Planilha com as informações enviadas pela unidade de aquisição de dados.

| Carimbo de data/hora | Tensão (V) | Corrente (mA) | Potência (mW) | Temperatura (°C) | Latitude | Longitude | Altitude (m) | Dia | Mês | Ano |
|----------------------|------------|---------------|---------------|------------------|----------|-----------|--------------|-----|-----|------|
| 19/09/2023 21:41:13 | 15.90 | -2.40 | -38.17 | 25.14 | 0.00 | 0.00 | 0.00 | 19 | 09 | 2023 |
| 19/09/2023 21:41:42 | 11.65 | -1.60 | -18.64 | 25.14 | 0.00 | 0.00 | 0.00 | 19 | 09 | 2023 |
| 19/09/2023 21:42:29 | 11.62 | -1.60 | -18.59 | 25.12 | 0.00 | 0.00 | 0.00 | 19 | 09 | 2023 |
| 19/09/2023 21:43:17 | 11.61 | -1.60 | -18.57 | 25.11 | -8.36 | -36.70 | 682.30 | 19 | 09 | 2023 |
| 19/09/2023 21:43:54 | 11.61 | -1.60 | -18.57 | 25.11 | -8.36 | -36.70 | 670.40 | 19 | 09 | 2023 |
| 19/09/2023 21:44:30 | 11.60 | -1.60 | -18.56 | 25.10 | -8.36 | -36.70 | 663.30 | 19 | 09 | 2023 |
| 19/09/2023 21:45:06 | 11.60 | -1.60 | -18.56 | 25.08 | -8.36 | -36.70 | 663.70 | 19 | 09 | 2023 |
| 19/09/2023 21:45:52 | 11.60 | -1.60 | -18.56 | 25.07 | -8.36 | -36.70 | 663.70 | 19 | 09 | 2023 |
| 19/09/2023 21:46:30 | 11.60 | -1.60 | -18.56 | 25.05 | -8.36 | -36.70 | 665.50 | 19 | 09 | 2023 |
| 19/09/2023 21:47:07 | 11.60 | -1.60 | -18.56 | 25.04 | -8.36 | -36.70 | 666.80 | 19 | 09 | 2023 |
| 19/09/2023 21:47:43 | 11.60 | -1.60 | -18.56 | 25.02 | -8.36 | -36.70 | 666.80 | 19 | 09 | 2023 |
| 19/09/2023 21:48:20 | 11.60 | -1.60 | -18.56 | 25.02 | -8.36 | -36.70 | 666.60 | 19 | 09 | 2023 |
| 19/09/2023 21:48:56 | 11.60 | -1.60 | -18.56 | 25.01 | -8.36 | -36.70 | 668.00 | 19 | 09 | 2023 |
| 19/09/2023 21:49:34 | 11.60 | -1.60 | -18.56 | 25.01 | -8.36 | -36.70 | 670.20 | 19 | 09 | 2023 |
| 19/09/2023 21:50:10 | 11.60 | -1.60 | -18.56 | 25.03 | -8.36 | -36.70 | 677.30 | 19 | 09 | 2023 |
| 19/09/2023 21:50:47 | 11.60 | -1.60 | -18.56 | 25.03 | -8.36 | -36.70 | 679.10 | 19 | 09 | 2023 |
| 19/09/2023 21:51:23 | 11.60 | -1.60 | -18.56 | 25.04 | -8.36 | -36.70 | 676.80 | 19 | 09 | 2023 |
| 19/09/2023 21:52:00 | 11.60 | -1.60 | -18.56 | 25.03 | -8.36 | -36.70 | 674.50 | 19 | 09 | 2023 |
| 19/09/2023 21:52:38 | 11.60 | -1.60 | -18.56 | 25.04 | -8.36 | -36.70 | 673.80 | 19 | 09 | 2023 |

Fonte: Própria, 2023.

Para extrair esses dados é necessário o uso da biblioteca *pandas* e do módulo *read_csv* já que os dados são exportados no formato CSV (*Comma Separated Values*, em português, Valores Separados por Vírgula) e o próprio módulo os converte para *DataFrame*, um tipo de tabela da biblioteca *pandas*, após a extração. Abaixo é exibida a Figura 8 que contém uma representação do código de extração e breves comentários sobre suas características.

Figura 8 – Código de extração dos dados.



Fonte: Própria, 2023.

Nota: As letras e números utilizados como identificação do arquivo e identificação da aba na planilha são aleatórios e não representam a identificação real de uma planilha disponível no *Google Sheets*.

3.2 Limpeza e transformação dos dados

Para trabalhar com uma base de dados com muitas colunas é necessário organizar quais dados precisam de quais tipos de limpezas e transformações, por isso eles foram divididos em grupos: Data e Hora, Grandezas Físicas e Dados de Localização. O Quadro 1 ilustra quais colunas pertencem a cada grupo.

Quadro 1 – Grupos de informações e suas colunas.

| Data e Hora | Grandezas Físicas | Dados de Localização |
|----------------------|--|--|
| Carimbo de data/hora | Tensão Corrente Potência elétrica Temperatura | Latitude Longitude Altitude Dia Mês Ano |

Fonte: Própria, 2023.

Os Dados de Localização que são armazenados na base de dados são utilizados diretamente pelo aplicativo e não possuem uso para geração de gráficos como os que serão mostrados nos resultados deste trabalho, por isso, eles não têm utilidade após serem extraídos pelo ETL.

Visando diminuir o tempo de processamento tanto da limpeza inicial, quanto das posteriores transformações dos dados, foi definido que o primeiro passo seria deletar os Dados de Localização do *DataFrame* logo após sua extração conforme recomendado por Faris *et al.* (2016), que em seu trabalho concluiu que quanto mais informações precisam ser processadas durante as iterações, maior o tempo total de processamento.

Em seguida, as colunas do grupo Grandezas Físicas que durante a exportação foram identificadas como tipo *float32* ou *float64* nesta etapa devem ser transformadas para o tipo *float16*, isso é feito com o objetivo de reduzir o uso de memória do *DataFrame* no *Jupyter Notebook*, o que também contribui para a redução do tempo de processamento durante as próximas transformações dos dados, isto será abordado com mais detalhes na seção de Resultados.

A coluna Carimbo de data/hora, inicialmente do tipo *string* (representada pela biblioteca *pandas* como do tipo *object*), é transformada para o tipo *datetime64* com auxílio do módulo *datetime*, dessa forma a ferramenta é capaz de utilizar todo o potencial da biblioteca *pandas*, o que facilita o processo de filtragem dos dados posteriormente.

3.2.1 Obtendo solicitação do usuário

Após a limpeza inicial, o objetivo se torna segregar os dados que o usuário necessita. Esses dados são solicitados através de um aplicativo *Android*, atualmente intitulado *Monstack*, como mostra a Figura 9.

Figura 9 – Aplicativo *Android* em que o usuário informa quais dados deseja visualizar.

MONSTACK

?

Qual ano deseja visualizar?

- ▼

Deseja visualizar um gráfico do ano inteiro?

- ▼

Qual mês deseja visualizar?

- ▼

Deseja visualizar um gráfico do mês inteiro?

- ▼

Qual dia do mês deseja visualizar?

- ▼

Que tipo de dado gostaria de visualizar?

- ▼

VOLTAR

MANDAR DADOS

Fonte: Própria, 2023.

Em uma das telas desse aplicativo, o usuário ao responder perguntas simples, poderá escolher dentre várias opções de período de tempo e grandeza física para visualizar na forma de gráfico em seu dispositivo. O aplicativo está em desenvolvimento e será abordado em trabalhos futuros.

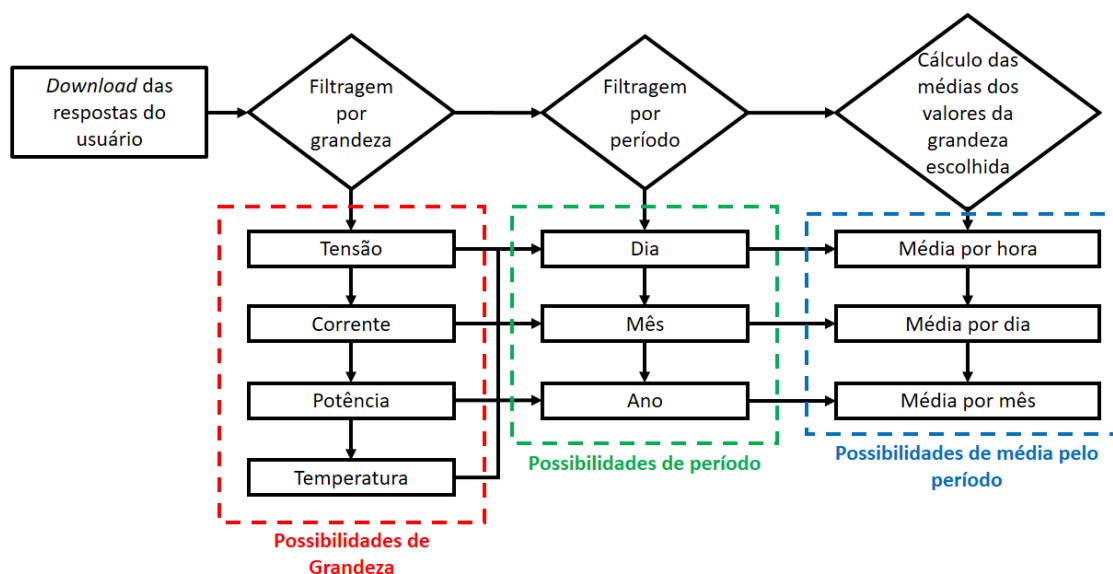
As respostas do usuário são enviadas para outra aba da planilha e a partir desse momento servem como gatilho para o ETL quando são baixadas para o *Jupyter* em formato de *DataFrame*, essas respostas são interpretadas pela ferramenta que segregará os dados através de algumas transformações fazendo com que as respostas sirvam como filtro para que somente os dados solicitados pelo usuário apareçam para ele posteriormente.

A filtragem é realizada através da grandeza e do período escolhido. Para as grandezas existem quatro opções como anteriormente ilustrado no Quadro 1: Tensão, corrente, potência e temperatura. Para o período, existem três opções: Dia, mês e ano. Dessa forma, o usuário pode escolher entre doze diferentes possibilidades de visualização dos dados.

Após segregarem os dados solicitados, é necessário transformá-los de forma que o aplicativo *Android* possa baixá-los e exibi-los na forma de um gráfico de linhas, pois o aplicativo trabalha com um número limitado de pontos no eixo das ordenadas.

O método escolhido para adequação dos dados foi de calcular várias médias a partir de subperíodos, o tamanho desses subperíodos é definido de acordo com o período originalmente escolhido pelo usuário. Dessa forma, foi definido que ao ser solicitado o período de um dia seriam calculadas médias para cada hora do dia, para o período de um mês seriam calculadas médias de cada dia do mês e para o período de um ano seriam calculadas médias para cada mês do ano. A figura 10 ilustra, através de um fluxograma, o método utilizado e as possibilidades de escolhas no aplicativo.

Figura 10 – Fluxograma do filtro de dados solicitados.



Fonte: Própria, 2023.

Após a finalização da filtragem, o *DataFrame* contém somente as médias dos valores obtidos durante o período solicitado pelo usuário. Os dados que formarão o gráfico precisam ser carregados para onde possam ser baixados.

3.3 Carga dos dados

Após finalizada a etapa de filtragem os dados são organizados em formato de lista para serem carregados, para isso é utilizada a API *Google Sheets*, que permite criar planilhas, ler e modificar seus dados, atualizar sua formatação, dentre outras funcionalidades (GOOGLE, 2023c).

Seguindo o guia de início rápido do *Python* disponibilizado por Google (2023a), é possível configurar facilmente todas as etapas de autorização de acesso, gerenciamento de autenticação, conexão entre o ETL e a planilha e por fim carregamento dos dados para a planilha.

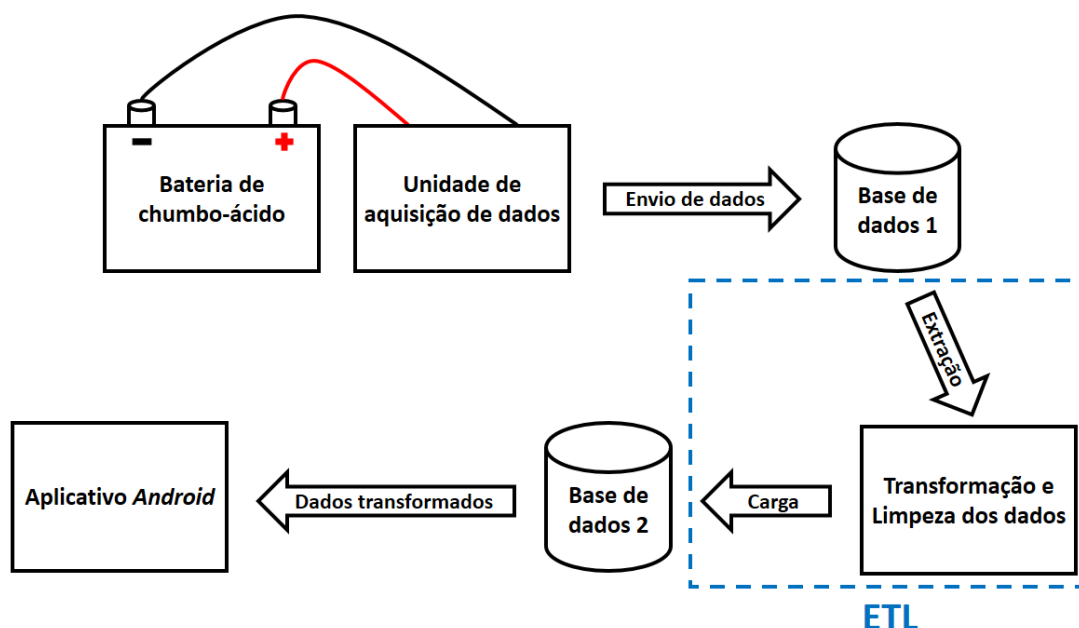
Os dados transformados são enviados para uma outra aba na mesma planilha onde estão os dados originais. A organização e utilização desses dados faz parte dos resultados deste trabalho e serão esclarecidas a seguir.

4 RESULTADOS E DISCUSSÕES

Este ETL faz parte de um sistema de monitoramento de baterias que também compreende uma unidade de aquisição de dados de baterias e um aplicativo para visualização de tais dados, ambos parte de outros trabalhos. Tal sistema visa suprir

uma demanda no mercado brasileiro de empreendimentos que utilizam bancos de baterias como reserva de energia para momentos de falha da rede elétrica local, ao utilizar o sistema de monitoramento a empresa é capaz de entender quais as necessidades de manutenção e substituição de suas baterias e verificar se o banco está pronto para a próxima queda de energia. A Figura 11 auxilia na compreensão dos componentes e etapas que fazem parte do sistema de monitoramento.

Figura 11 – Fluxograma de funcionamento básico do sistema de monitoramento de baterias.



Fonte: Própria, 2023.

É necessário que um computador esteja com o código em *loop* no *Jupyter Notebook* para que a ferramenta possa operar realizando a extração dos dados, a sua limpeza, a coleta das respostas do usuário no aplicativo, a transformação da base de dados para *DataFrame* e de suas colunas para otimização do tempo de processamento, a filtragem desses dados e, por fim, o carregamento dos dados para a planilha.

4.1 Extração dos dados coletados

Quando os dados são convertidos para *DataFrame* eles permanecem em formato de tabela e podem ser trabalhados com facilidade durante as transformações e as limpezas que serão realizadas em seguida. A Figura 12 exhibe um *DataFrame* com os mesmos dados ilustrados na Figura 7, e a Figura 13 ilustra as informações básicas do *DataFrame* assim que os dados são extraídos.

Figura 12 – DataFrame dos dados extraídos da planilha.

Out[9]:

| | Carimbo de data/hora | Tensão (V) | Corrente (mA) | Potência (mW) | Temperatura (°C) | Latitude | Longitude | Altitude (m) | Dia | Mês | Ano |
|--------|----------------------|------------|---------------|---------------|------------------|----------|-----------|--------------|------|------|------|
| 0 | 19/09/2023 21:41:13 | 15.90 | -2.4 | -38.17 | 25.14 | 0.00 | 0.0 | 0.0 | 19.0 | 9.0 | 2023 |
| 1 | 19/09/2023 21:41:42 | 11.65 | -1.6 | -18.64 | 25.14 | 0.00 | 0.0 | 0.0 | 19.0 | 9.0 | 2023 |
| 2 | 19/09/2023 21:42:29 | 11.62 | -1.6 | -18.59 | 25.12 | 0.00 | 0.0 | 0.0 | 19.0 | 9.0 | 2023 |
| 3 | 19/09/2023 21:43:17 | 11.61 | -1.6 | -18.57 | 25.11 | -8.36 | -36.7 | 682.3 | 19.0 | 9.0 | 2023 |
| 4 | 19/09/2023 21:43:54 | 11.61 | -1.6 | -18.57 | 25.11 | -8.36 | -36.7 | 670.4 | 19.0 | 9.0 | 2023 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 121858 | 05/11/2023 16:18:37 | 12.49 | -2.0 | -24.98 | 27.65 | -8.36 | -36.7 | 678.5 | 5.0 | 11.0 | 2023 |
| 121859 | 05/11/2023 16:19:13 | 12.50 | -2.0 | -24.99 | 27.66 | -8.36 | -36.7 | 675.2 | 5.0 | 11.0 | 2023 |
| 121860 | 05/11/2023 16:19:50 | 12.49 | -2.0 | -24.98 | 27.65 | -8.36 | -36.7 | 674.9 | 5.0 | 11.0 | 2023 |
| 121861 | 05/11/2023 16:20:28 | 12.50 | -2.0 | -24.99 | 27.67 | -8.36 | -36.7 | 674.2 | 5.0 | 11.0 | 2023 |
| 121862 | 05/11/2023 16:21:05 | 12.50 | -2.0 | -24.99 | 27.67 | -8.36 | -36.7 | 672.9 | 5.0 | 11.0 | 2023 |

121863 rows × 11 columns

Fonte: Própria, 2023.

Figura 13 – Informações gerais do DataFrame logo após a extração.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121863 entries, 0 to 121862
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Carimbo de data/hora  121863 non-null object
1   Tensão (V)           121863 non-null float64
2   Corrente (mA)        121863 non-null float64
3   Potência (mW)        121863 non-null float64
4   Temperatura (°C)     121863 non-null float64
5   Latitude              121863 non-null float64
6   Longitude             121863 non-null float64
7   Altitude (m)         121863 non-null float64
8   Dia                  121863 non-null float64
9   Mês                  121863 non-null float64
10  Ano                   121863 non-null int64
dtypes: float64(9), int64(1), object(1)
memory usage: 10.2+ MB
```

Fonte: Própria, 2023.

Esses dados foram coletados entre os dias 19 de setembro de 2023 e 5 de novembro de 2023, por isso a partir daqui serão mencionados como possuindo uma quantidade fixa de valores e de espaço ocupado. Isso ocorre para garantir que o leitor deste artigo entenda com clareza a seção de Resultados e também porque a quantidade de dados que existe na planilha muda a cada 30 segundos, tornando uma explicação que utilize todos os dados inviável.

4.2 Limpeza e transformação dos dados

Utilizando o método de exclusão de colunas que não são essenciais citado na seção de Metodologia para redução do uso de memória do *DataFrame* iniciam-se as transformações neste *DataFrame* de tamanho 10,2 megabytes que contém 11 colunas e 121.863 linhas.

Após as exclusões o resultado obtido é de um *DataFrame* de 4,6 megabytes que contém 5 colunas e 121.863 linhas, obtendo uma redução de cerca de 55% do espaço

originalmente ocupado pelos dados. A Figura 14 ilustra as informações básicas do *DataFrame* após as exclusões.

Figura 14 – Informações gerais do *DataFrame* logo após exclusão de algumas colunas.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121863 entries, 0 to 121862
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Carimbo de data/hora  121863 non-null object
1   Tensão (V)           121863 non-null float64
2   Corrente (mA)        121863 non-null float64
3   Potência (mW)        121863 non-null float64
4   Temperatura (°C)     121863 non-null float64
dtypes: float64(4), object(1)
memory usage: 4.6+ MB
```

Fonte: Própria, 2023.

Dando continuidade ao trabalho de redução do uso de memória, todas as colunas que contém dados numéricos são transformadas do tipo *float64* para tipo *float16*, além disso existem comandos para apagar quaisquer linhas que contenham dados vazios de medição e possam impedir a biblioteca *pandas* de realizar cálculos. Os nomes das colunas também são levemente modificados para retirar espaços entre palavras, pois esta é considerada uma boa prática ao trabalhar com *DataFrame*.

Figura 15 – Informações gerais do *DataFrame* ao final da limpeza.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121863 entries, 0 to 121862
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Carimbo_de_data/hora  121863 non-null datetime64[ns]
1   Tensão                 121863 non-null float16
2   Corrente               121863 non-null float16
3   Potência               121863 non-null float16
4   Temperatura            121863 non-null float16
dtypes: datetime64[ns](1), float16(4)
memory usage: 1.9 MB
```

Fonte: Própria, 2023.

Esta mudança faz com que o *DataFrame* passe a ocupar somente 1,9 megabytes de espaço, o que representa uma diminuição de cerca de 81% no espaço de 10,2 megabytes ocupado originalmente, garantindo que durante as próximas transformações realizadas pela ferramenta o tempo de segregação para posterior envio dos dados seja o menor possível.

Foram conduzidos testes de tempo de processamento utilizando o filtro de dados configurado no aplicativo para "Dia inteiro". Os resultados obtidos ao filtrar a base de dados para obter as medições do dia 21 de setembro de 2023 são os seguintes: No caso do *DataFrame* sem a aplicação de transformações, observou-se um tempo médio de processamento para filtragem dos dados de 261 milissegundos. Por outro lado, ao aplicar transformações ao *DataFrame*, constatou-se que o tempo médio foi reduzido para 215 milissegundos de processamento.

Estes resultados indicam uma redução do tempo de aproximadamente 18%. Esta diminuição é especialmente relevante considerando que a base de dados tende a

crescer continuamente à medida que o monitoramento da bateria se mantém ativo, com a possibilidade de ter *Gigabytes* ou *Terabytes* de dados futuramente.

A melhoria do tempo necessário para processamento dos dados contribui para a experiência do usuário no aplicativo, pois ele não precisa aguardar vários segundos para que o gráfico apareça na tela de seu *smartphone*. Além disso, também reduz a quantidade de dados transmitidos via *internet*, reduz a quantidade de energia elétrica utilizada pelos aparelhos eletrônicos envolvidos e possivelmente reduz inclusive sua temperatura durante a operação.

4.3 Carga dos dados

O usuário do aplicativo *Monstack* ao realizar a solicitação através das respostas das perguntas ilustradas na Figura 9 ativará o ETL que atuará como descrito na Metodologia deste trabalho e enviará os dados transformados de volta para a base de dados.

Um exemplo de resultado deste processo é retratado na Figura 16, onde valores de horas do dia, tensão e corrente foram calculados e gerados pelo ETL. A figura exibe o resultado do processamento dos dados coletados pela unidade de aquisição de dados a pedido do usuário através do aplicativo, para isso o ETL extraiu 1.340.493 valores diferentes da base de dados e a partir destes segregou 158.400 dados relativos somente ao dia 21 de setembro de 2023, assim como solicitado pelo usuário e, por fim, realizou consecutivos cálculos para que somente 24 valores distintos, cada valor equivalente a uma hora do dia, fossem enviados a planilha e, posteriormente, exibidos no aplicativo em formato de gráfico como exposto na Figura 17.

Figura 16 – Dados de tensão (a) e corrente (b) carregados para a planilha pelo ETL.

| Horas | Tensão | Horas | Corrente |
|-------|------------|-------|---------------|
| 0h | 11,3203125 | 0h | -1,599609375 |
| 1h | 11,296875 | 1h | -1,599609375 |
| 2h | 11,2890625 | 2h | -1,599609375 |
| 3h | 11,2578125 | 3h | -1,599609375 |
| 4h | 9,5078125 | 4h | -1,315429688 |
| 5h | 9,078125 | 5h | -1,200195313 |
| 6h | 8,6875 | 6h | -1,1875 |
| 7h | 6,8828125 | 7h | -1,200195313 |
| 8h | 6,87890625 | 8h | -1,200195313 |
| 9h | 4,8671875 | 9h | -0,8208007813 |
| 10h | 4,5546875 | 10h | -0,7998046875 |
| 11h | 0 | 11h | 0 |
| 12h | 0 | 12h | 0 |
| 13h | 0 | 13h | 0 |
| 14h | 0 | 14h | 0 |
| 15h | 0 | 15h | 0 |
| 16h | 0 | 16h | 0 |
| 17h | 0 | 17h | 0 |
| 18h | 11,5 | 18h | -1,598632813 |
| 19h | 11,7109375 | 19h | -1,599609375 |
| 20h | 11,75 | 20h | -1,599609375 |
| 21h | 11,78125 | 21h | -1,599609375 |
| 22h | 11,8125 | 22h | -1,599609375 |
| 23h | 11,8515625 | 23h | -1,598632813 |

(a)

(b)

Fonte: Própria, 2023.

Figura 17 – Aplicativo com gráfico de tensão (a) e gráfico de corrente (b) formados com os dados carregados pelo ETF.



Fonte: Própria, 2023.

Como observado por Almeida, Pereira e Dias (2022), a drástica redução da tensão da bateria enquanto sua corrente continua praticamente constante possibilita que o usuário compreenda que utilizou toda a energia armazenada na bateria e que ela necessita de uma recarga, um usuário do aplicativo *Monstack* que possui conhecimento técnico sobre baterias de chumbo-ácido ou que recebeu um treinamento para utilizar o sistema de monitoramento, pode facilmente interpretar a ocorrência ilustrada na Figura 17 como uma descarga profunda. Essa e outras situações podem ser diagnosticadas através do gráfico gerado no aplicativo com os dados carregados pelo ETL.

5 CONSIDERAÇÕES FINAIS

A proposta do trabalho foi desenvolver um *software* capaz de coletar de uma base de dados medições realizadas em baterias de chumbo-ácido, a fim de torná-las apropriadas para visualização através de um gráfico gerado em um aplicativo móvel. Este software é capaz de fazer a coleta a partir de uma base de dados, realizar o tratamento dos dados coletados e enviar os resultados de volta para a nuvem para a leitura do aplicativo móvel.

É importante destacar que o código desenvolvido nesse trabalho foi reconhecido por sua inovação e valor prático, resultando na concessão de uma patente pelo INPI, que está ilustrada no Anexo A deste artigo. Esta conquista reflete a relevância e originalidade do presente trabalho.

Também é importante reconhecer limitações. O ETL atualmente trabalha com somente uma base de dados devido a restrições de recursos durante a execução deste trabalho. Além disso, é necessário que um computador esteja disponível para executar o código em *loop* durante todo o dia aguardando a solicitação do usuário para visualização de seus dados no aplicativo, o que demanda recursos para manter a ferramenta. Futuros trabalhos podem abordar tais limitações utilizando outras soluções como um servidor, por exemplo, além de atualizar o código fonte para que o ETL possa extrair, transformar e carregar dados de múltiplas bases de dados.

Além disso, o fato da base de dados estar alocada em uma planilha do *Google Sheets* a torna limitada, pois ela é capaz de armazenar somente 10 milhões de células (GOOGLE, 2022). Considerando que cada leitura dos dados da bateria ocupa 11 células a planilha é capaz de armazenar até 909.090 leituras, aproximadamente 315 dias sem interrupção. Portanto, outra melhoria possível para o trabalho é um estudo para utilização de outro tipo de base de dados relacional ou de banco de dados relacional.

Conclui-se, então, que o ETL desenvolvido é completamente funcional e capaz de trabalhar com os diversos dados de um sistema de armazenamento de energia – data, hora, tensão, corrente, potência elétrica, temperatura e dados de localização geográfica –, com a possibilidade de ser utilizado como base para o desenvolvimento de um ETL mais robusto em trabalhos futuros.

AGRADECIMENTOS

Primeiramente agradeço meu orientador Bruno Albuquerque Dias, pela boa vontade e prontidão que demonstrou durante todo o desenvolvimento do projeto, orientando para que este Trabalho de Conclusão de Curso fosse finalizado com sucesso.

A todos os colaboradores da Startup Monstack, em especial Gabriel, Giovanna, Emmanuel, Kelvyn, Leandro e Patrícia que prestaram todo seu suporte e apoio durante a realização desse trabalho.

Aos professores do curso de Bacharelado em Engenharia Elétrica do IFPE *campus* Pesqueira que participaram da minha formação acadêmica, gostaria de expressar minha sincera gratidão por todo o conhecimento e orientação oferecidos ao longo dessa jornada.

Aos meus pais Jailson e Lucilene, por sua compreensão e apoio durante toda a minha vida escolar e acadêmica. Seu amor, esforço e dedicação na minha criação e educação são parte essencial da mulher que sou hoje.

A minha madrinha Vera, por seus ensinamentos, seus conselhos e sua paciência. Minha segunda mãe, minha querida amiga. Sou grata a Deus e aos meus pais por ser sua afilhada.

Aos meus avós Eriberto e Maria do Socorro, pela compreensão de minha ausência devido ao tempo destinado para elaboração desse projeto. As minhas tias Ana e Nena, por investirem na minha educação durante a infância. Aos meus tios Renato e Roberto, com quem posso contar para tirar dúvidas sobre tecnologia. As minhas primas Danielle e Isabelle, por sua agradável companhia desde a infância até os dias de hoje. Aos meus sobrinhos Anahí e Daniel, que foram motivo dos meus mais sinceros sorrisos durante os dias mais difíceis.

A Danyelle, Davi, Débora, Isabela, Letícia, Maria Clara, Patrícia e Thallyta muito obrigada por sua amizade durante os anos da graduação. Presencialmente ou remotamente vocês foram peça fundamental para essa conquista, pois com seu carinho, companheirismo e palavras de compreensão me ajudaram a manter a saúde mental em dia.

A Deus por colocar todas essas pessoas no meu caminho, por sempre me manter na melhor direção e por me dar sabedoria e inteligência para chegar até aqui. Como escreveu Salomão “Bem-aventurado aquele que acha sabedoria, e aquele que adquire conhecimento / Pois a sabedoria é mais proveitosa que a prata, e rende mais do que o ouro / Mais preciosa é do que os rubis, e tudo o que mais possas desejar não se pode comparar a ela”.

A SETEC (Secretaria de Educação Profissional e Tecnológica), a FACTO (Fundação de Apoio ao Desenvolvimento da Ciência e Tecnologia), ao Programa Centelha e ao Programa Startup NE por fomentar a Startup Monstack e seu projeto de Gestão de Sistemas de Armazenamento de Energia que foi pontapé inicial para desenvolvimento desse projeto.

A todos que direta ou indiretamente contribuíram para a realização desse trabalho, meus mais sinceros agradecimentos.

REFERÊNCIAS

- ACCUWEATHER. 2023. AccuWeather APIs. Disponível em: <<https://developer.accuweather.com/>>. Acesso em: 25 set. 2023.
- ALMEIDA, A. P. F.; PEREIRA, J. A. T.; DIAS, B. A. Desenvolvimento de um protótipo para monitoramento remoto de baterias de chumbo-ácido. Artigo (Trabalho de Conclusão de Curso) – Bacharelado em Engenharia Elétrica, Instituto Federal de Pernambuco. Pesqueira, 2022. Disponível em: <<https://repositorio.ifpe.edu.br/xmlui/handle/123456789/673>>. Acesso em: 17 mar. 2023.
- ASTERA SOFTWARE. 2020. O que é uma ferramenta ETL: definição, usos e casos de uso. Disponível em: <<https://www.astera.com/pt/type/blog/what-is-etl-tool/>>. Acesso em: 28 jan. 2023.
- BASTOS, Lucas Monteiro; LIMA, Rodrigo Monteiro de. DETECÇÃO DE FAKE NEWS USANDO OS ALGORITMOS DECISION TREE, SUPPORT VECTOR MACHINE E K-NEAREST NEIGHBORS. Revista de Estudos Multidisciplinares, v. 3, n. 1, p. 1-13, São Luís, 2023. Disponível em: <<https://periodicos.undb.edu.br/index.php/rem/article/view/83>>. Acesso em: 04 fev. 2023.
- FARIS, Hossam *et al.* EvoloPy: An Open-source Nature-inspired Optimization Framework in Python. In: International Joint Conference on Computational Intelligence, v. 1, p. 171-177, 2016, Porto. Disponível em: <<https://www.scitepress.org/Papers/2016/60482/60482.pdf>>. Acesso em: 04 abr. 2023.
- GAMA e ABREU, Fábio Silva Gomes da. Desmistificando o Conceito de ETL. Revista de Sistemas de Informação, Macaé, 2008. Disponível em: <http://www.fsma.edu.br/si/Artigos/FSMA_SI_2008_2_Principal_1.html>. Acesso em: 28 jan. 2023.
- GOOGLE. 2022. Google Sheets doubles cell limit. Disponível em: <<https://workspaceupdates.googleblog.com/2022/03/ten-million-cells-google-sheets.html>>. Acesso em: 26 jun. 2023.
- _____. 2023a. Guia de início rápido do Python. Disponível em: <<https://developers.google.com/sheets/api/quickstart/python?hl=pt-br>>. Acesso em: 18 jun. 2023.

_____. 2023b. Tome decisões baseadas em dados com o Google Sheets. Disponível em: <<https://www.google.com/intl/pt-BR/sheets/about/>>. Acesso em: 13 dez. 2023.

_____. 2023c. Visão geral da API Google Sheets. Disponível em: <<https://developers.google.com/sheets/api/guides/concepts?hl=pt-br>>. Acesso em: 13 mai. 2023.

HIOKI. User's Guide: Lead-acid Battery Handbook. Plano: 2020, 23 p. Disponível em: <https://www.hioki.com.sg/wp-content/uploads/2021/02/A_UG_BT0002E01.pdf>. Acesso em: 05 jan. 2023.

IBER – INSTITUTO BRASILEIRO DE ENERGIA RECICLÁVEL, 2022. Raio-x do setor de baterias chumbo-ácido: confira o desempenho do mercado em 2021. Disponível em: <<https://iberbrasil.org.br/blog/2022/06/21/raio-x-do-setor-de-baterias-chumbo-acido-confira-o-desempenho-do-mercado-em-2021/>>. Acesso em: 23 set. 2023.

INDIAN BATTERIES. 2018. PE Battery Separator. Disponível em: <<https://indianbatteries.com/battery-separator/pe-battery-separator/>>. Acesso em: 05 jan. 2023.

JACOBSON, D. BRAIL, G. WOODS, D. APIs: A Strategy Guide. 1 ed. Sebastopol: O'Reilly Media Inc., 2013. 134 p.

JUPYTER. [2023?]. About Us. Disponível em: <<https://jupyter.org/about>>. Acesso em: 31 jan. 2023.

KEYSIGHT. Instrument Automation with Python. Keysight Technologies: 2022, 18 p. Disponível em: <<https://www.keysight.com/us/en/assets/7018-06894/white-papers/5992-4268.pdf>>. Acesso em: 11 fev. 2023.

KIMBALL, Ralph. CASERTA, Joe. The Data Warehouse ETL Toolkit. 1 ed. Indianapolis: Wiley Publishing Inc., 2004, 465 p.

NI, Shuangdan. Exploring the Preferences of US YouTube Users and Factors Related to YouTube Uploader's Revenue. Journal of Studies in Social Sciences and Humanities, v. 2, n. 1, p. 43-54, Periódico online. 2023. Disponível em: <<https://www.paradigmpress.org/SSSH/article/view/398>>. Acesso em: 04 fev. 2023.

PARASHAR, Abhay. 2022. Jupyter Notebook 101: Everything You Need To Know. Disponível em: <<https://medium.com/pythoneers/jupyter-notebook-101-everything-you-need-to-know-56cda3ea76ef>>. Acesso em: 31 jan. 2023.

PYTHON. 2023a. Applications for Python. Disponível em: <<https://www.python.org/about/apps/>>. Acesso em: 10 fev. 2023.

_____. 2023b. What is Python? Executive Summary. Disponível em: <<https://www.python.org/doc/essays/blurb/>>. Acesso em: 10 fev. 2023.

RIBEIRO, Adeilda Simão *et al.* Construção de um aparato experimental de baixo custo para análise espectral e inserção do modelo atômico de Bohr no ensino de física moderna. Artigo (Trabalho de Conclusão de Curso) – Licenciatura em Física, Instituto Federal de Pernambuco. Pesqueira, 2021. Disponível em: <<https://repositorio.ifpe.edu.br/xmlui/handle/123456789/252>>. Acesso em: 06 mar. 2023.

SANTANA, Bruno. 2023. O que é Python: Conheça Uma das Linguagens de Programação mais Populares do Mundo. Disponível em: <<https://www.hostinger.com.br/tutoriais/python-o-que-e>>. Acesso em: 12 fev. 2023.

SCROSATI, Bruno. History of lithium batteries. Journal of Solid State Electrochemistry, v. 15, n. 7-8, p. 1623-1630, Periódico online, 2011. Disponível em: <<https://link.springer.com/article/10.1007/s10008-011-1386-8#article-info>>. Acesso em: 23 set. 2023.

SIGALINGGING, Riswanti *et al.* Modelling of energy productivity prediction systems of shallots classification growth phase system using convolutional neural network. In: International Conference on Computer Science and Computational Intelligence. 2023, Hong Kong. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050922022219>>. Acesso em: 04 fev. 2023.

SILVA, Maicon Hervertino Lino Ferreira da. Técnico em Informática: Planilha Eletrônica. Secretaria de Educação e Esportes de Pernambuco: 2013, 62p. Disponível em: <<https://sisacad.educacao.pe.gov.br/bibliotecavirtual/bibliotecavirtual/texto/Cadernod eINFOPlanilhaEletrnicaRDDI.pdf>>. Acesso em: 13 dez. 2023.

TAB. 2023. TAB UPS BATTERIES. Disponível em: <<https://www.tab.si/products/ups/>>. Acesso em: 05 jan. 2023.

TUDOR. Manual técnico bateria estacionária. 2017, 35 p. Disponível em: <https://www.tudor.com.br/assets/upload/manual_tecnico_baterias_estacionarias_rev_03.pdf>. Acesso em: 22 out. 2023.

UNIVERSITY OF SILICON VALLEY. 2021. How is the internet of things (IoT) used today?. Disponível em: <<https://usv.edu/blog/how-is-the-internet-of-things-iot-used-today/>>. Acesso em: 24 set. 2023.

USGS – UNITED STATES GEOLOGICAL SURVEY. 2023. Mineral commodity summaries 2023. Washington: USGS, 2023. Disponível em: <<https://www.usgs.gov/publications/mineral-commodity-summaries-2023>>. Acesso em: 23 set. 2023.

ANEXO A – Patente do software registrada no INPI**REPÚBLICA FEDERATIVA DO BRASIL**

MINISTÉRIO DA ECONOMIA

INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL

DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

Certificado de Registro de Programa de ComputadorProcesso Nº: **BR512022000102-0**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 02/11/2021, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: Monstack Software**Data de publicação:** 02/11/2021**Data de criação:** 01/11/2021**Titular(es):** INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE PERNAMBUCO**Autor(es):** BRUNO ALBUQUERQUE DIAS; FILIPE LUCENA M. DE ANDRADE; GIOVANNA PAULA OLIVEIRA DE LIMA; ANTONIO PAULO FRANÇA DE ALMEIDA; JOÃO ALISSON TAVARES PEREIRA; ANNA KÁTREN DIAS DOS SANTOS; ALERRANDRO RAY GALINDO CABRAL; GABRIEL BATISTA SILVEIRA DA SILVA; LAYS LEAL CORREIA; PATRÍCIA TATIANE LOPES DO NASCIMENTO; DAVID LUCAS ALVES DE ALMEIDA; VITÓRIO AUGUSTO DE ESPINDOLA**Linguagem:** PYTHON; MYSQL**Campo de aplicação:** EN-01**Tipo de programa:** FA-04; GI-02; GI-04; GI-06; IA-01**Algoritmo hash:** SHA-512**Resumo digital hash:**

5fef2c88067c9f6ceb171b755644ea85572b3045747cec1fe3fdd0f5dcec36638782192954c27648fe0e6a10258c145ccbb6dd6318f1f331cd7a4dc520a7a60

Expedido em: 25/01/2022**Aprovado por:**Carlos Alexandre Fernandes Silva
Chefe da DIPTO