Laboratório Virtual com Dados Reais

Victor Takashi Hayashi
victor.hayashi@usp.com
Universidade de São Paulo
São Paulo, Brasil

Danilo Oliveira Martins
danilo.oliveira.martins@usp.br
Univerdade de São Paulo
São Paulo, Brasil

Reginaldo Arakaki
reginaldo.arakaki@poli.usp.br
Universidade de São Paulo
São Paulo, Brasil

Julio Carlos Teixeira
juliocarlos.teixeira@ufabc.edu.br
Universidade Federal do ABC
Santo André, Brasil

Bruno Angélico
angelico@lac.usp.br
Universidade de São Paulo
São Paulo, Brasil

Fabio Hirotsugu Hayashi
fabio.hayashi@ufabc.edu.br
Universidade Federal do ABC
Santo André, Brasil

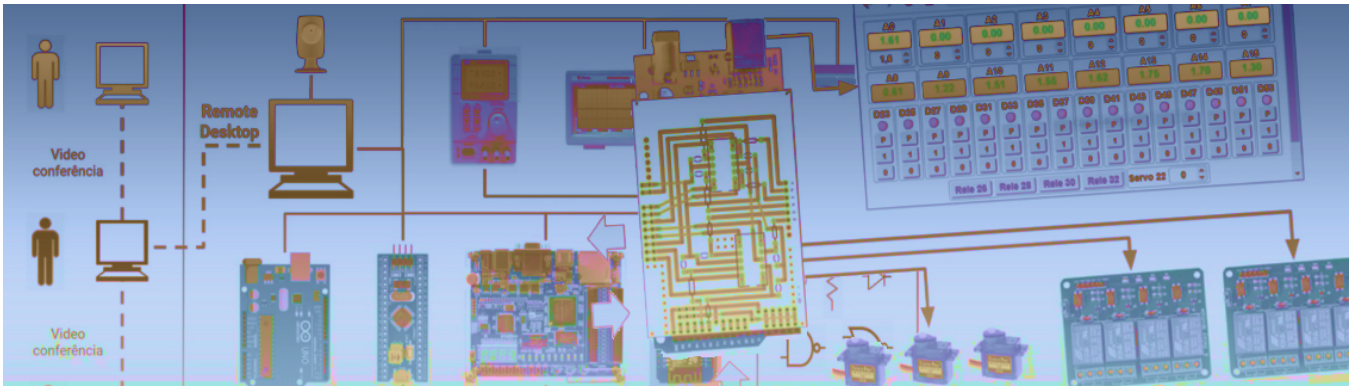

Figura 1: Divulgação do LabEAD 2020. Retirado de: https://jornal.usp.br/?p=325366

RESUMO

O laboratório virtual com dados de experimentos reais permite escalabilidade e a capacidade de comparar os dados do experimento obtidos em laboratório por professores e técnicos, disponibilizando estes dados em um portal web para análise de dados pelos alunos. Este trabalho demonstra como é possível implementar as ferramentas que permitem a prática de experimentos laboratoriais para fins de aprendizagem em Engenharia. Os resultados apresentados incluem percepções iniciais dos alunos sobre a ferramenta.

CCS CONCEPTS

• **General and reference** → Experimentation; • **Hardware** → *Communication hardware, interfaces and storage*; • **Computer systems organization** → *Distributed architectures*; *Data flow architectures*; **Embedded systems**; • **Software and its engineering** → **Software architectures**.

PALAVRAS-CHAVE

laboratório virtual, laboratório remoto, análise de dados, IoT

EduComp'21, Abril 27–30, 2021, Jataí, Goiás, Brasil (On-line)

1 INTRODUÇÃO

O uso de laboratórios é comum no processo de aprendizagem em cursos de nível superior. Entretanto, aspectos de segurança dos alunos e de reprodutibilidade dos ensaios exigem um alto investimento em equipamentos e sistemas de monitoramento dos ensaios. Um número elevado de alunos no mesmo momento em um laboratório exige uma equipe de professores e técnicos altamente capazes e atentos aos naturais erros dos alunos, e costuma criar dificuldades de acesso à infraestrutura. Em alguns momentos, como por exemplo no contexto da pandemia de COVID-19, este acesso é muito restrito. Neste contexto, este trabalho tem por objetivo apresentar uma solução que permite o ensino a distância.

O aspecto prático do ensino de habilidades relacionadas à computação é essencial para o desenvolvimento de competências. A partir da premissa que competência não se ensina, mas se constrói através da experiência, uma questão essencial para os pesquisadores do projeto LabEAD é: "Como fornecer uma ferramenta que permita a criação de situações que propiciem o desenvolvimento de competências?".

O ensino de computação possui uma vertente prática, que pode ser observada em cursos *online* através da integração de ferramentas como o *Jupyter Notebook* para a execução de exercícios em Python durante cursos de programação [3]. Um desafio para o ensino de computação surge quando sistemas ciber físicos são considerados: como integrar o ensino de computação com o monitoramento e controle de equipamentos? Tais aprendizados são necessários para a formação de profissionais capazes de atuar em projetos que envolvam Internet das Coisas, como a Indústria 4.0 [4].

Em geral, nas disciplinas de laboratório, temos bancadas de trabalho onde os alunos experimentam na prática as teorias lecionadas em classe. É comum que o experimento seja executado em grupo permitindo uma interação entre os alunos, isso exercita não somente a prática em si mas o trabalho em grupo. Visando permitir essa interação em um formato remoto utilizamos videoconferência, a aula é supervisionada pelo professor que também executará o acesso de forma remota e contará com o auxílio do técnico de laboratório que estará presente para solucionar qualquer imprevisto ou obter informações adicionais requeridas pelo professor.

Contudo, uma solução de laboratório remoto é limitada pela disponibilidade do número de bancadas conectadas disponíveis para uso. Mesmo que a interação em tempo real seja viável com o uso de um laboratório remoto, escalar o acesso da ferramenta para um número grande de alunos é um desafio.

Este trabalho apresenta o desenvolvimento de um laboratório virtual construído com dados reais obtidos em experimentos práticos. Proporciona fidelidade com o comportamento real, permitindo escalabilidade e capacidade de comparar dados com laboratório remoto. Arquitetura proposta permite disponibilização de dados de experimentos com interface de osciloscópio em portal web e análise de dados entre outros experimentos. A arquitetura consiste em implementações de dispositivos embarcados para possibilitar a execução de experimentos práticos em laboratório, os dados gerados e obtidos nos experimentos são manipulados e inseridos em um banco de dados documental operando em um ambiente de computação em nuvem. Os dados armazenados são disponibilizados através de serviços em rede que possibilitam a associação de ferramentas externas para análise de dados.

É importante definir que o escopo do artigo considera o cenário em que experimentos são conduzidos remotamente pelo professor, com a ajuda de um técnico no laboratório, por exemplo em experimentos que envolvam equipamentos cuja operação a distância envolve riscos relevantes. Neste caso, os alunos só terão acesso aos dados coletados posteriormente, ou seja, sem possibilidades de interação ativa com o protótipo em laboratório.

A ferramenta está sob a licença *GNU General Public License v3.0*. Os códigos desenvolvidos para o portal *web* estão disponíveis no GitHub[1]. Os códigos do *Jupyter Notebook* em Python no ambiente Google Colab também estão disponíveis no Github[2] [12, 13].

2 TRABALHOS RELACIONADOS

Em sua obra "Sapiens - Uma breve história da humanidade"[11], Yuval Harari (2014) destaca que uma das capacidades univocamente humanas é a capacidade de transmitir informações sobre coisas que não existem. Além da capacidade de modelagem e abstração, as habilidades cognitivas permitiram a evolução do pensamento e linguagem humanos.

Linguagens de programação permitem que humanos construam abstrações interpretáveis por humanos, e exequíveis por computadores [1]. Esta bimodalidade inerente nas linguagens de programação pode facilitar ou dificultar o aprendizado de habilidades de programação.

A revisão de código é um dos métodos mais usados de revisão estática em uso na indústria (e.g. em empresas como Microsoft e Google). Revisores independentes são responsáveis por identificar defeitos e deficiências de manutenibilidade, que podem estar associados a um baixo nível de legibilidade do código. Neste sentido, um programa pode ser entendido como uma forma de literatura [6].

O ensino contextualizado de computação pode fornecer maior relevância e engajamento de estudantes, uma vez que os discentes podem aprender melhor se entenderem o valor do que estão aprendendo [10]. Mesmo com um possível impacto negativo na transferência de aprendizado resultante do conhecimento mais especializado apresentado no ensino contextualizado [10], esta abordagem possui valor para aumentar a taxa de sucesso em cursos introdutórios de computação [9].

Conforme Schulte & Knobelsdorf (2007), as pré-concepções sobre computação podem afetar o interesse e inclusive fomentar barreiras iniciais a alunos novos de computação. A percepção de que as habilidades atuais são insuficientes para aprender programação, ou que somente especialistas devem atuar no uso profissional de computadores [19].

Alguns laboratórios virtuais presentes na literatura endereçam o suporte a atividades práticas de programação. Por exemplo, o laboratório virtual VPL da Universidade de Las Palmas de Gran Canaria é integrado ao ambiente moodle, e permite a avaliação automática de exercícios, com verificações adicionais contra plágio [5].

Dentre os diferentes trabalhos encontrados sobre laboratórios virtuais, foi possível identificar semelhanças de arquitetura computacional, embora diferentes técnicas e tecnologias fossem aplicadas diferenciando os trabalhos, é comum encontrar o conceito de aquisição de dados de dispositivos, processamento e armazenamento. Em J. Sáenz et al. (2015) [18] temos uma plataforma de baixo custo, laboratório de livre acesso, código fonte aberto para replicação, possui integração com Moodle e semelhante a este trabalho suporta laboratório virtual e remoto porém como desavantagem é focado para experimentos para disciplina de Engenharia de Controle.

Em Y. Liang et al (2017) [16] é apresentado um laboratório virtual interativo 3D. Os autores justificam esta solução, que não usa uma câmera de vídeo, pelas seguintes vantagens: não necessita de uma banda de dados considerável; possui suporte a experimentos colaborativos (estudantes e professores); possui menor relação custo benefício quando comparado com um laboratório remoto. Entretanto a solução proposta não oferece controle do dispositivo.

A implementação de um laboratório virtual é abordada em S. Wilson et al (2020) [20]. Ela implica em utilizar recursos computacionais, fazendo uma escolha entre as diversas possibilidades tecnológicas disponíveis. Há entretanto desafios para manter uma plataforma em pleno funcionamento.

Pelo exposto, fica fundamentada a oportunidade de construção de uma ferramenta de laboratório virtual que possa fornecer um ensino contextualizado de programação a partir de uma disciplina de laboratório com dados reais, e assim motivar estudantes iniciantes em computação a fazer uso de suas capacidades cognitivas ao analisar, e tratar dados com algoritmos implementados em ambiente *Jupyter Notebook*, comumente usado por cientistas de dados.

[1] https://github.com/vthayashi/weblabead
[2] https://github.com/vthayashi/labvirt-ventilador

3 OBJETIVO

A questão de pesquisa abordada neste trabalho é: "Como fornecer uma ferramenta que torne possível a criação de situações que propiciem o desenvolvimento de habilidades práticas de computação e sistemas ciber físicos a distância?"

Para responder à pergunta de pesquisa, dois aspectos principais foram considerados como oportunidades de pesquisa:

- Conjunto de dados de experimentos realizados, a ser disponibilizado para estudantes - não somente para os que fizeram as aulas de laboratório, mas para outros estudos - significa que os pesquisadores sabem as condições de experimentos e podem realizar estudos históricos, simulações e análises;
- Catálogo de experimentos: as condições de realização prática para obtenção dos dados também podem ser disponibilizados para estudos, estimulando evoluções colaborativas de experimentos.

A partir dos objetivos e oportunidades, a ferramenta foi especificada conforme suas funcionalidades principais:

- Permitir o acesso remoto ao laboratório oferecendo a possibilidade de executar experimentos que alimentam uma base de dados em um ambiente de computação em nuvem.
- Disponibilizar os dados coletados em ambiente virtual interativo para posterior análise pelos alunos.

Como requisitos não-funcionais da ferramenta, destacam-se:

- O sistema deve permitir o acesso simultâneo de 40 alunos;
- O sistema deve estar disponível para uso pelos estudantes e professores em 99% do tempo;
- O sistema deve permitir o uso de pelo menos 2 dispositivos embarcados diferentes para coleta de dados;
- O sistema deve ser intuitivo, de fácil entendimento, com um aprendizado de uso ferramenta inferior à 5 minutos.

4 SOLUÇÃO PROPOSTA

A especificação de um caso de uso para o laboratório remoto foi a primeira etapa para a especificação da ferramenta.

Durante uma aula de laboratório remoto, o professor inicia e encerra a aula através da ferramenta de video conferência geral que é acessível a todos os participantes. Cada experimento será executado por um grupo de alunos conectados por video conferência que permite envio de mensagens instantâneas e conta com o auxílio do técnico de laboratório. Este caso de uso está ilustrado na Figura 2.

Durante a execução do experimento, os dados amostrados pelo dispositivo são armazenados, após o seu processamento, em um ambiente computacional em nuvem. O pacote de dados do experimento é submetido através de uma interface conectada à rede mundial de computadores que permite o armazenamento de dados em um ambiente computacional em nuvem. Cada experimento possui um conjunto de dados que é armazenado no banco, e esses dados ao serem inseridos no ambiente em nuvem podem ser acessados por diversas interfaces simultaneamente, permitindo análises e visualização. O projeto LabEAD utilizou o banco de dados NoSQL MongoDB, por ser possível hospedar a massa de dados em um *cluster* de simples configuração. O diagrama da Figura 3 apresenta os compoentes e ferramentas utilizados.

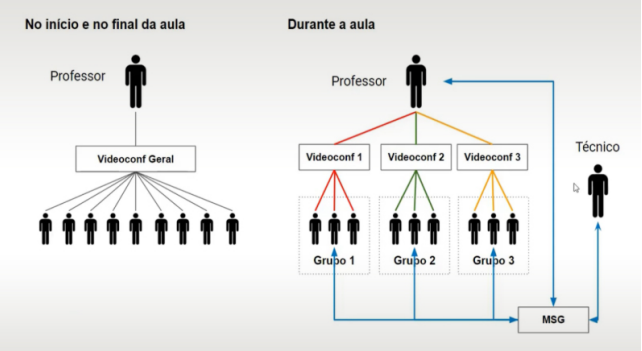

Figura 2: Fluxo de comunicação em uma aula a distância. Retirado de: https://www.youtube.com/watch?v=xnupw2cHBE

Após o armazenamento de dados em banco de dados não-relacional, os dados são acessados pela aplicação Google Colab [7]. Este ambiente de análise e desenvolvimento de algoritmos em Python é semelhante ao *Jupyter Notebook* [14], e é considerado uma das ferramentas essenciais para um cientista de dados. Com o uso desta ferramenta, a familiarização dos alunos com um ambiente novo utilizado por profissionais da área de Ciência de Dados é esperada, assim como maior engajamento e interesse dos usuários.

Figura 3: Diagrama do Sistema Proposto

O dispositivo embarcado foi construído a partir de sensor de corrente não-invasivo, relés, acoplador óptico, ventilador, sensor infravermelho de passagem, sensor de temperatura DS18B20, uma placa NodeMCU ESP32 (vide Figura 4). O NodeMCU ESP32 é uma placa de desenvolvimento para prototipação de soluções de Internet das Coisas, baseada no microcontrolador ESP32 [17]. Esta placa possui comunicação WiFi e Bluetooth integradas, com suporte a sistema de arquivos, atualização de *firmware* por comunicação sem fio (*firmware over the air*).

É importante destacar que no projeto LabEAD há o uso de outros dispositivos, como FPGA (*Field Programmable Gate Array*) e placas de desenvolvimento Arduino (e.g., Arduino Uno, Arduino Mega). Porém, neste artigo a placa utilizada é o ESP32.

As grandezas monitoradas são: rotação por meio do sensor de passagem infravermelho; corrente por meio do sensor de corrente não-invasivo; e indicador de fase da rede elétrica por meio do acoplador óptico. O ESP32 possui um servidor *web*, que é responsável por receber requisições pela rede local, controlar a partida do ventilador por meio do relé, e monitorar as grandezas de interesse com os sensores descritos. Os sinais são amostrados a uma taxa de 20 kHz e então armazenados em um *buffer* interno do dispositivo até que outra requisição de dados os obtenha.

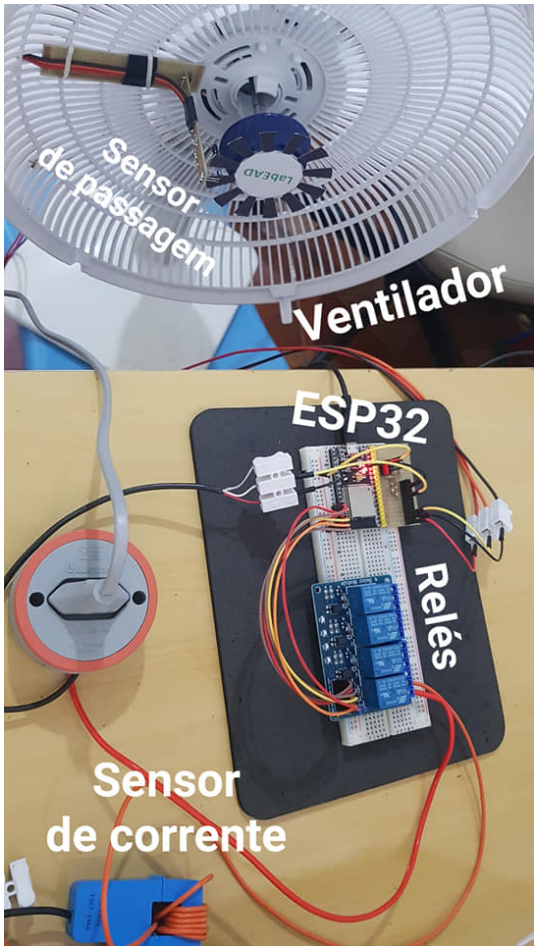

Figura 4: Montagem realizada para coleta de dados.

O código do dispositivo embarcado (*firmware*) foi desenvolvido na Arduino IDE, com a inclusão do suporte ao ESP32 através do gerenciador de placas da IDE.

5 EXEMPLO DE USO

Esta seção possui o detalhamento de um exemplo de uso do laboratório virtual com dados reais, conforme prova de conceito desenvolvida em agosto de 2020. As três etapas consideradas são: coleta de dados, acesso aos dados e análise dos dados.

5.1 Base de Dados e Automação da Coleta de Dados

A automação da coleta de dados foi desenvolvida em Python e implantado em um computador Acer (intel Core i3, 8GB RAM), com o uso das bibliotecas *requests*, *time* e *pymongo*.

A biblioteca *requests* é utilizada para realizar requisições HTTP para o módulo de Internet das Coisas a partir do computador, que está conectado na mesma rede WiFi do módulo. A biblioteca *time* é usada para a temporização entre requisições de partida do ventilador, que permite um intervalo configurável de 30 segundos entre partidas. Por fim, a conexão com o banco de dados é realizado a partir da biblioteca *pymongo*, e após a realização de cada partida (experimento), os dados são coletados e enviados para o banco de dados não-relacional MongoDB.

A Figura 5 ilustra o procedimento automatizado. A primeira requisição possui como argumentos o endereço local do módulo de Internet das Coisas, e configurações do experimento, como nome, notas associadas (valores do tipo *String*), taxa de amostragem desejada, além da preparação do módulo em modo de coleta de dados *single* (i.e. o módulo espera a mudança de um sinal de sincronismo para coletar os dados em sua memória).

Na segunda requisição, o motor é ligado, sendo desligado após 10 segundos com a terceira requisição.

Os dados são obtidos em formato JSON na quarta requisição, e enviados em seguida para o banco de dados não relacional MongoDB, com o uso de parâmetros específicos, como credenciais de acesso ao sistema de gerenciamento de base de dados, e informações que identificam qual instância deve ser atualizada. O módulo coletor de dados espera 30 segundos para iniciar nova partida do motor.

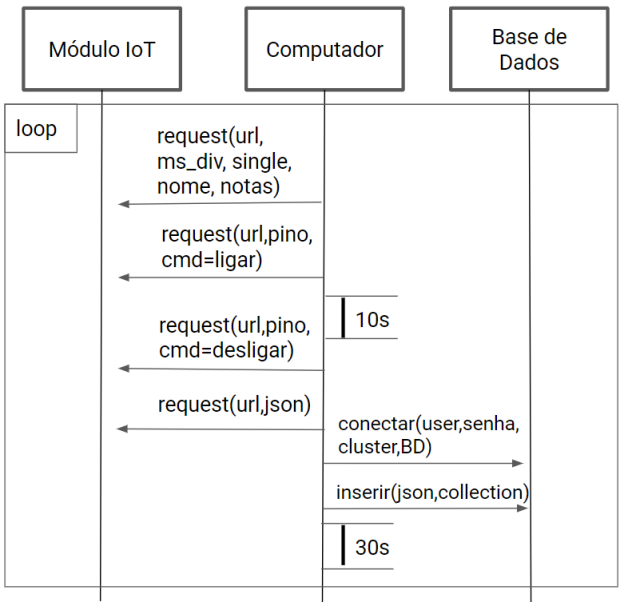

Figura 5: Diagrama de Sequência da Coleta de Dados

Com o módulo coletor de dados implantado no computador, dados de 150 experimentos foram armazenados na base de dados MongoDB, sendo 50 experimentos em cada uma das três velocidades do ventilador (mínimo, médio e máximo). A Figura 6 ilustra a interface de gerenciamento do MongoDB observada pelo administrador do sistema, onde é possível verificar a existência de 150 documentos de dados, na base específica do ventilador.

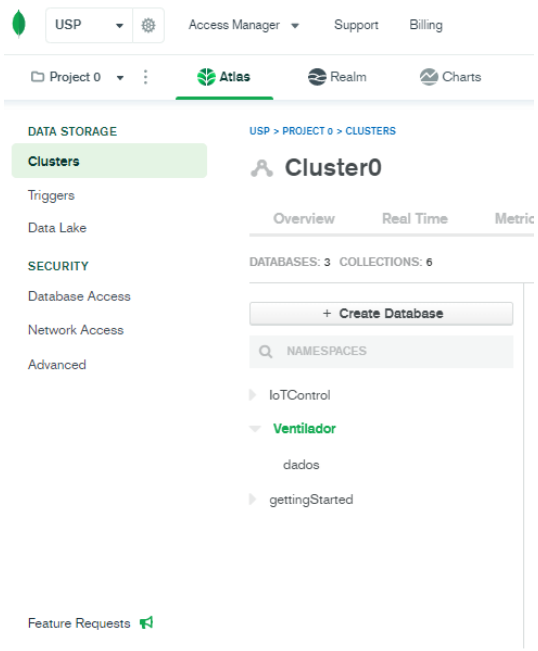

Figura 6: Interface do MongoDB

A Figura 7 ilustra o detalhe de um documento de dados de um experimento, observada pelo administrador do sistema. O formato é JSON, e os dados armazenados são dos três canais do módulo IoT, configuração de taxa de amostragem do experimento, nome e notas em linguagem natural, e a temperatura no momento de coleta dos dados.

```
_id: ObjectId("5f2342d34e48f9b6b0d36fa6")
w_us: 200
> wa: Array
nome: "velocidade 3"
notas: "CH0: velocidade, CH1: referencia da fase
temp: "21.4"
```

Figura 7: Exemplo de documento JSON no MongoDB

A implementação da dinâmica de inserção de dados provenientes do dispositivo embarcado no ambiente computacional em nuvem possibilita posterior acesso remoto aos dados históricos, gerados pelo experimento, possibilitando um estudo comparativo entre diversas execuções e permite expandir o aprendizado do aluno em métodos avançados de análises de dados.

5.2 Acesso aos dados pela Interface Web

O protótipo para visualização de dados em interface *web* foi implementado a partir da biblioteca Python Flask[3] (*framework* específico para desenvolvimento de aplicações *web*), e desenvolvimento de interface visual em HTML/CSS. Sua implantação foi realizada com a ferramenta Render[4]. O protótipo foi compartilhado no Github[5] sob a licença de código aberto *GNU General Public License v3.0*.

Conforme disponível no repositório do GitHub, a aplicação *web* Flask desenvolvida possui a seguinte arquitetura:

- LICENSE: informações de licença para uso;
- Motor6.htm e Motor6b.htm: interface de osciloscópio para visualização de dados;
- README.md: informações gerais;
- app.py: responsável pelo tratamento das requisições, baseado no *framework* Flask;
- logo.png: logo presente na página inicial do portal;
- requirements.txt: bibliotecas necessárias para a implantação do projeto;
- verdados.htm: interface de visualização dos dados em formato tabular.

Com o projeto disponível no GitHub, é possível realizar sua implantação com a ferramenta Render[6]. Com o endereço do repositório, a ferramenta se baseia no arquivo requirements.txt para realizar a implantação de forma automática. Há suporte para que qualquer atualização no repositório resulte em uma atualização da interface, e para que o desenvolvedor solicite a atualização da interface com o botão "*Manual Deploy*"(implentação manual em tradução livre) presente na captura da tela da Figura 8, observada pelo administrador do sistema.

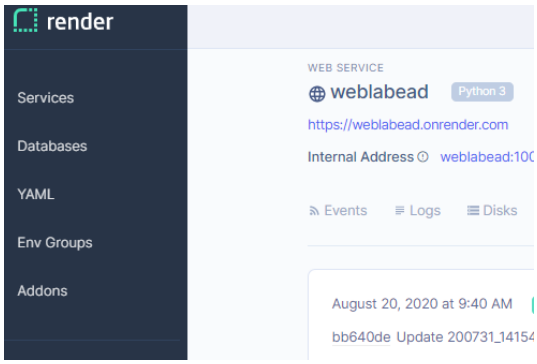

Figura 8: Interface do Render

A interface de osciloscópio apresentada na Figura 9 apresenta uma visualização interativa dos dados reais coletados, conforme observado pelo aluno e professor. Por exemplo, o aluno pode ajustar parâmetros de visualização (e.g. *zoom*, ms/div e V/div), selecionar o canal de interesse, visualizar o nome do experimento, e descrição em linguagem natural, que pode ser adicionada como metadado do experimento pelo professor ou técnico. Uma funcionalidade

[3]https://flask.palletsprojects.com/en/1.1.x/
[4]https://render.com/
[5]https://github.com/
[6]https://render.com/

adicional é a medição nos dois eixos com o cursor, o que pode facilitar a estimativa de frequência e amplitude do sinal pelo aluno.

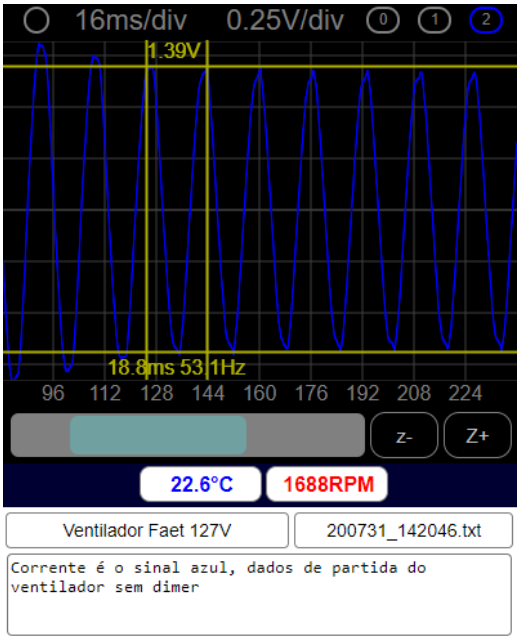

Figura 9: Interface de osciloscópio para interação com dados reais no portal *web* do laboratório virtual

5.3 Análise dos dados

Nesta seção, a análise dos dados disponíveis no banco de dados não-relacional é demonstrada com a ferramenta Google Colab, conforme observada pelo aluno, e configurada pelo professor. Na Figura 10, com a biblioteca Python pymongo, um teste inicial de conexão e obtenção dos dados de um experimento é realizado.

```
db = client.Ventilador
experiments = db.dados

dbdata = experiments.find_one({ "nome": "velocidade 3" })

print(dbdata)
```

```
{'_id': ObjectId('5f2342d34e48f9b6b0d36fa6'), 'w_us': 200, 'wa': [[0.14, 0.16,
```

```
[ ] dbdata['_id']
```

```
ObjectId('5f2342d34e48f9b6b0d36fa6')
```

```
[ ] # notas do experimento

    dbdata['notas']
```

```
'CH0: velocidade, CH1: referencia da fase da rede eletrica, CH2: corrente'
```

Figura 10: Teste inicial de conexão com MongoDB no Google Colab

Com os dados obtidos em formato JSON, a separação das séries temporais em listas é facilitada, assim como uma visualização inicial dos dados sem tratamento (i.e. dados "brutos"). A separação e visualização do parâmetro que indica as rotações estão presentes na Figura 11.

```
# CH0: velocidade

velocidade = dbdata['wa'][0]
```

```
[ ] print(velocidade)
```

```
[0.14, 0.16, 0.16, 0.16, 0.16, 0.16, 0.16, 0.16,
```

```
[ ] plt.plot(velocidade)
```

```
[<matplotlib.lines.Line2D at 0x7fd4cb5a2c88>]
```

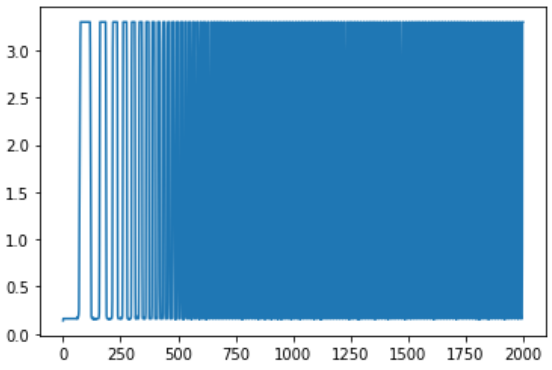

Figura 11: Visualização inicial dos dados de velocidade angular do ventilador

A visualização das grandezas rotação, indicador de fase da rede elétrica e corrente podem ser visualizadas na Figura 12.

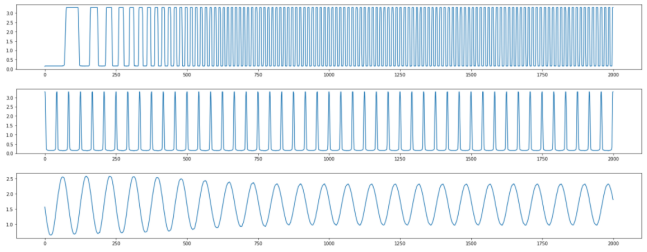

Figura 12: Visualização inicial das séries de dados coletados. De cima para baixo: velocidade angular, fase da rede elétrica e corrente.

Após alguns tratamentos implementados em Python nos dados obtidos, é possível obter parâmetros de velocidade e amplitude da corrente ao longo do tempo. Vide visualizações de corrente, amplitude e velocidade na Figura 13.

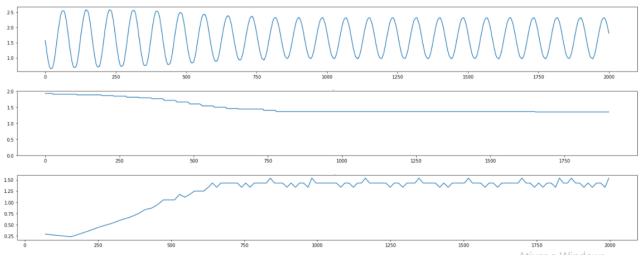

Figura 13: Visualização dos dados após tratamento. De cima para baixo: corrente, amplitude da corrente, velocidade.

Outro parâmetro que pode ser obtido com alguns outros tratamentos em Python é o instante em que o sinal alcançou o regime estacionário. Através do monitoramento da diferença entre o dado atual e o último dado da série, é possível computar sua diferença, e detectar quando esta diferença se torna menor que determinado valor (*threshold*). O procedimento para obtenção destes dados para cada experimento está ilustrado na Figura 14.

Data Labeling

```
[ ]  for i in range(len(corrente_amp)):
       datapoint = corrente_amp[i]
       delta = datapoint - corrente_amp[len(corrente_amp)-1]

       if (delta < 0.1):
         start_time = corrente_amp_time[i]
         print(start_time)
         break
```

```
650
```

```
[ ]  print(rede_inicio)
```

```
39
```

```
[ ]  print(max(velocidade_proc))
```

```
1.5384615384615385
```

Figura 14: Rotulagem automática de experimentos

O instante de partida de cada um dos 150 experimentos coletados foi determinado com o algoritmo ilustrado na Figura 14, e os dados foram inseridos em uma entidade do tipo *dataframe* da biblioteca Python pandas, comumente usada por cientistas de dados. A Figura 15 ilustra esta estrutura de dados.

```
df
```

	start	network	velocity	id	velocidade_anotada
0	650	39	1.538462	5f2342d34e48f9b6b0d36fa6	3
1	646	75	1.538462	5f2343004e48f9b6b0d36fa7	3
2	606	36	1.538462	5f23432a4e48f9b6b0d36fa8	3
3	611	39	1.538462	5f2343544e48f9b6b0d36fa9	3
4	686	75	1.538462	5f23437e4e48f9b6b0d36faa	3
...	...	...	...	...	...
145	1272	43	1.428571	5f235efc07939aab042bd64c	1
146	1273	43	1.428571	5f235f2607939aab042bd64d	1
147	1305	75	1.428571	5f235f5007939aab042bd64e	1
148	1273	43	1.428571	5f235f7907939aab042bd64f	1
149	1316	47	1.428571	5f235fa307939aab042bd650	1

150 rows × 5 columns

Figura 15: Estrutura de dados para aplicação de algoritmo de clusterização

Com o *dataframe*, é possível utilizar a biblioteca Python sklearn para realizar uma classificação simples dos experimentos em três agrupamentos, com o uso do algoritmo de clusterização *K Nearest Neighbours*, a partir do instante de partida calculado, conforme ilustrado na Figura 16.

Clustering with start and velocidade_anotada

```
[ ]  df2 = df[['start', 'velocidade_anotada']].copy()

     kmeans = KMeans(n_clusters=3).fit(df2)
     centroids = kmeans.cluster_centers_

     plt.figure(figsize = (8,8))

     plt.scatter(df2['start'], df2['velocidade_anotada'],
                 c= kmeans.labels_.astype(float), s=50, alpha=0.5)
     #plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)
     plt.xlabel("time to start")
     plt.ylabel("labeled velocity")

     plt.show()
```

Figura 16: Algoritmo de clusterização por meio da biblioteca Python sklearn

Uma validação com as anotações de velocidade de cada experimento validam os resultados obtidos. Na Figura 17, é possível observar o tempo de partida no eixo horizontal, e a velocidade do ventilador anotada (por um professor ou técnico no momento de coleta dos dados reais) no eixo vertical. Para a velocidade 1 (mínima), o tempo de partida foi maior, e para a velocidade 3 (máxima), o tempo de partida foi menor, como era esperado.

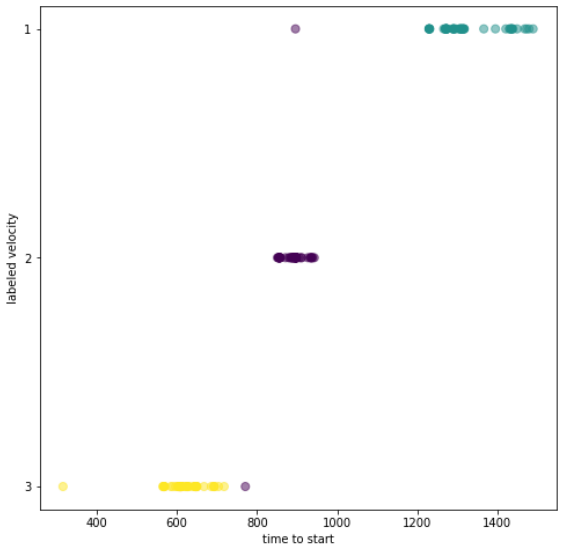

Figura 17: Resultado do algoritmo de clusterização com tempo para partida como parâmetro. No eixo horizontal está o tempo de partida calculado, e no eixo vertical está a velocidade anotada manualmente

Com os códigos desenvolvidos no Google Colab, é possível realizar tratamento nos dados coletados, e realizar uma rotulagem automática dos dados. Desta forma, o dado de velocidade pode ser inferido a partir do dado de rotação, retirando a necessidade do técnico ou professor anotar a velocidade do experimento de forma manual.

6 AVALIAÇÃO PRELIMINAR DO PORTAL

Um questionário foi aplicado pela Internet através da ferramenta Google Forms. O total de participantes da avaliação preliminar do portal foi 10, e o período de coleta foi de 20/08/2020 a 25/08/2020.

6.1 Quantitativo

Em relação ao perfil de entrevistados, as instituições de ensino mais relevantes foram UFABC e USP. Alunos são a maioria dos entrevistados (60%), conforme Figuras 18 e 19.

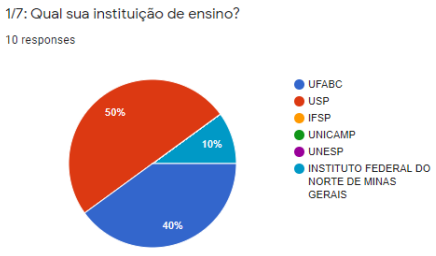

Figura 18: Instituição de ensino dos entrevistados

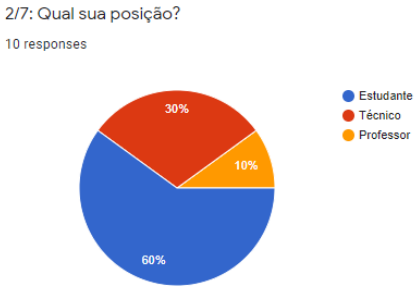

Figura 19: Perfil dos entrevistados

Somente um entrevistado respondeu erroneamente a questão sobre o comportamento da corrente durante a partida, vide Figura 20. O objetivo desta questão foi observar se após a navegação pelo portal, o usuário entendeu que um dos gráficos apresentados é da corrente, e que sua amplitude é maior nos instantes iniciais.

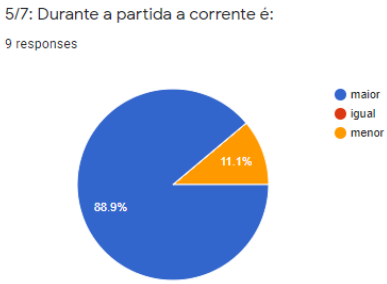

Figura 20: Pergunta de entendimento após navegação pelo portal *web*

A ferramenta foi bem-avaliada em geral, com notas 4 e 5 em sua maioria, conforme ilustrado pela Figura 21.

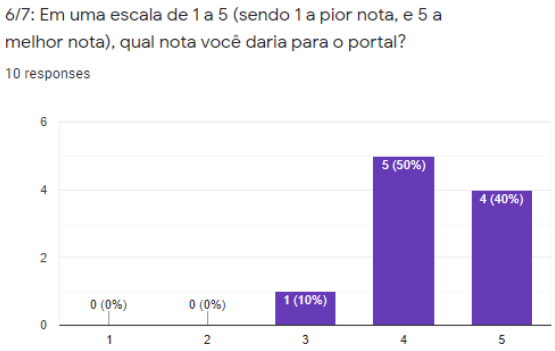

Figura 21: Avaliação da ferramenta

O entrevistado que respondeu de forma errônea também foi o responsável pela avaliação menor (3). Uma hipótese levantada pelos autores é que o entendimento sobre a ferramenta foi um aspecto

crítico para a avaliação deste usuário, o que motiva a priorização de mecanismos de ajuda e explicação das ferramentas presentes no portal *web*.

6.2 Qualitativo

A seguir, são apresentados alguns comentários livres dos entrevistados.

O primeiro comentário destaca que o aspecto de responsividade da interface pode ser melhorado: "O aspecto visual da página web ficou melhor no celular do que no computador".

O segundo comentário reforça a necessidade de melhoria na responsividade:

> Talvez dar mais ênfase nos botões que o usuário pode interagir na interface, e na parte de rolagem ser em tempo real em vez de atualizar apenas quando solta o scroll. O espaço na página inicial de desktop podia ser melhor aproveitado, o vídeo fica bem pequeno e tem bastante espaço não utilizado (resolução da tela 1080p). De resto, ficou bacana poder ver a evolução dos dados e interagir com o display.

O comentário seguinte evidencia que um mecanismo de ajuda, indicações e explicações podem facilitar o aprendizado inicial da interface.

"Parabéns pela iniciativa, excelente o site, como sugestão, poderia estar indicado na tela dos sinais, o que cada cor significa, facilitaria mais na compreensão dos dados (como uma legenda)".

O próximo comentário destaca uma confusão do usuário, algo que também pode ser evitado com a implementação de mecanismos de ajuda: "Não entendi muito bem qual a função do botão no canto superior esquerdo".

Por fim, o último comentário destaca que a funcionalidade adicional de cálculo de diferença usando o cursor foi bem-recebida pelo entrevistado: "Gostei da função de comparar o delta de voltagem e tempo entre dois pontos usando o mouse".

De acordo com a percepção do grupo reduzido de usuários, a implementação de um mecanismo de ajuda e explicações na interface podem motivar sua implementação e teste de usabilidade com pelo menos 15 usuários, envolvendo métricas de qualidade tais como tempo de aprendizado, memorização de comandos, erros de usuário e satisfação subjetiva, conforme Nielsen [8], questionários padrão como *system usability scale* [15], e os princípios descritos no livro "Interação Humano-Computador"[2].

7 CONSIDERAÇÕES FINAIS

Este trabalho demonstrou ser possível criar um laboratório virtual que proporciona exercitar habilidades práticas de computação e sistemas ciber físicos. Obter dados de experimentos reais, armazenados em um ambiente computacional que, permite integração com outras ferramentas, possibilita expandir a experiência prática do aluno.

Utilizar o laboratório virtual com ferramentas analíticas possibilita ter ciência dos dados e consequentemente eleva o conhecimento prático para além das fronteiras do laboratório. O padrão aberto de dados e de processamento, e disponibilização das experiências realizadas permite criar uma rede de experimentos colaborativos para aprendizado prático.

O ambiente educacional moderno é baseado em sistemas de gerenciamento de ensino, como o Moodle, executar integração do laboratório virtual com uma ferramenta de gerenciamento é certamente um trabalho a ser desenvolvido que facilitará a disponibilização de acesso no ambiente educacional. Como outros trabalhos futuros, há a possibilidade de integração de mecanismos de ajuda, especialização da interface para suportar a interação com dados reais coletados de máquinas elétricas de potência instaladas na instituição de ensino. Uma investigação relevante seria verificar se os benefícios vindouros da ferramenta podem além do contexto e demandas do período de isolamento social, por exemplo auxiliando programas de ensino híbrido.

REFERÊNCIAS

[1] Miltiadis Allamanis, Earl T Barr, Premkumar Devanbu, and Charles Sutton. 2018. A survey of machine learning for big code and naturalness. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–37.

[2] Simone Barbosa and Bruno Silva. 2010. *Interação humano-computador.* Elsevier Brasil.

[3] Coursera. 2020. Natural Language Processing. https://www.coursera.org/specializations/natural-language-processing

[4] Ministério da Indústria Comércio e Serviços. 2020. http://www.industria40.gov.br/

[5] Juan Carlos Rodríguez del Pino, Enrique Rubio Royo, and Zenón Hernández Figueroa. 2012. A virtual programming lab for Moodle with automatic assessment and anti-plagiarism features. In *Proceedings of The 2012 Internacional Conference on e-Learning, e-Business, Entreprise Information Systems, & e-Government.*

[6] Benjamin Floyd, Tyler Santander, and Westley Weimer. 2017. Decoding the representation of code in the brain: An fMRI study of code review and expertise. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE).* IEEE, 175–186.

[7] Google. 2020. https://colab.research.google.com/

[8] Nielsen Norman Group. 2020. Usability 101: Introduction to Usability. https://www.nngroup.com/articles/usability-101-introduction-to-usability/

[9] Mark Guzdial. 2009. Contextualized computing education of programming. In *Proceedings of the Eleventh Australasian Conference on Computing Education-Volume 95.* 3–3.

[10] Mark Guzdial. 2010. Does contextualized computing education help? *ACM Inroads* 1, 4 (2010), 4–6.

[11] Yuval Noah Harari. 2014. *Sapiens: A brief history of humankind.* Random House.

[12] Victor Hayashi. 2021. *vthayashi/labvirt-ventilador v1.0.0.* https://doi.org/10.5281/zenodo.4455268

[13] Victor Hayashi. 2021. *vthayashi/weblabead: First Release.* https://doi.org/10.5281/zenodo.4455311

[14] Jupyter. 2020. https://jupyter.org/

[15] James R Lewis. 2018. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction* 34, 7 (2018), 577–590.

[16] Yuxin Liang and Guo-Ping Liu. 2017. Design of large scale virtual equipment for interactive HIL control system labs. *IEEE Transactions on Learning Technologies* 11, 3 (2017), 376–388.

[17] NodeMCU. 2020. NodeMCU Documentation Overview. https://nodemcu.readthedocs.io/en/dev-esp32/

[18] Jacobo Sáenz, Jesús Chacón, Luis De La Torre, Antonio Visioli, and Sebastián Dormido. 2015. Open and low-cost virtual and remote labs on control engineering. *Ieee Access* 3 (2015), 805–814.

[19] Carsten Schulte and Maria Knobelsdorf. 2007. Attitudes towards computer science-computing experiences as a starting point and barrier to computer science. In *Proceedings of the third international workshop on Computing education research.* 27–38.

[20] Sean Wilson, Paul Glotfelter, Li Wang, Siddharth Mayya, Gennaro Notomista, Mark Mote, and Magnus Egerstedt. 2020. The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems. *IEEE Control Systems Magazine* 40, 1 (2020), 26–44.