



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U. nº 198, de 14/10/2016
AELBRA EDUCAÇÃO SUPERIOR - GRADUAÇÃO E PÓS-GRADUAÇÃO S.A.

Antonio Carlos Pereira Mota Milhomens Junior

PLATAFORMA PARA EXPERIMENTAÇÃO REMOTA COM ARDUINO

Palmas – TO

2021

Antonio Carlos Pereira Mota Milhomens Junior
PLATAFORMA PARA EXPERIMENTAÇÃO REMOTA COM ARDUINO

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Esp. Fábio Castro Araújo.

Palmas – TO

2021

Antonio Carlos Pereira Mota Milhomens Junior
PLATAFORMA PARA EXPERIMENTAÇÃO REMOTA COM ARDUINO

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Esp. Fábio Castro Araújo.

Aprovado em: ____/____/____

BANCA EXAMINADORA

Prof. Esp. Fábio Castro Araújo.

Orientador

Centro Universitário Luterano de Palmas – CEULP

Profa. M.Sc. Madianita Bogo Mariotti

Centro Universitário Luterano de Palmas

Profa. Dr.a Parcilene Fernandes de Brito.

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2021

AGRADECIMENTOS

Tipo de homenagem onde se agradece **às pessoas que contribuíram para que o trabalho fosse concluído**, ou que auxiliaram em algo. (ABNT NBR 14724, 2011b, p. 02, grifo nosso).

RESUMO

O presente trabalho de pesquisa apresenta o desenvolvimento de uma Plataforma para experimentação remota com Arduino, o contexto envolve a área de aprendizagem baseada em laboratórios de experimentação remota (LER). O trabalho descreve as nove fases que foram utilizadas para sua conclusão (reuniões com o orientador, elaboração/estudo do referencial teórico, levantamento e definição dos requisitos, estudo das ferramentas, planejamento do design para a plataforma, implementação da plataforma, integração dos softwares já desenvolvidos, testes de funcionalidades e por fim, a plataforma desenvolvida), assim como, os trabalhos utilizados para complementar a plataforma. A solução foi desenvolvida utilizando uma arquitetura RLMS baseada em softwares e hardwares *open source*, resultando em uma plataforma online que possibilita o aprendizado de programação Arduino utilizando experimentos físicos criados previamente, bem como o acompanhamento desses experimentos por meio de uma webcam.

PALAVRAS-CHAVE: RLMS, laboratórios de experimentação remota; Arduino; ensino e aprendizagem.

LISTA DE FIGURAS

Figura 1 - Classificação dos tipos de laboratórios. Fonte: Zutin et al. (2010).	13
Figura 2 - Representação do modelo de um laboratório remoto.	16
Figura 3 - Visão geral do WebLab-Deusto (arquitetura). Fonte: Orduna et al. (2014).	18
Figura 4 - Ambiente de aprendizagem com experimentos remotos. Fonte: http://relle.ufsc.br/	19
Figura 5 - Metodologia do Trabalho.	20
Figura 6 - Arquitetura.	24
Figura 7 - Experimento sugerido.	25
Figura 8 - Arquitetura do Back-end para um software de experimentação remota. Fonte: Milhomens Junior (2018).	26
Figura 9 - Diagrama de banco de dados da plataforma.	28
Figura 10 - Tela de login da Plataforma.	30
Figura 11 - Mensagens com valores inválidos.	31
Figura 12 - Dashboard da área administrativa.	31
Figura 13 - Formulário de cadastro dos experimentos.	32
Figura 14 - Formulário de cadastro dos experimentos, erro de preenchimento.	33
Figura 15 - Página gerenciar experimentos, mensagem de cadastro com sucesso.	33
Figura 16 - Página gerenciar experimentos.	34
Figura 17 - Página detalhes do experimento.	35
Figura 18 - Página gerenciar experimentos, status diversos.	35
Figura 19 - Página gerenciar usuários.	36
Figura 20 - Página detalhes do usuário.	37
Figura 21 - Formulário de cadastro de usuário, erro de preenchimento.	37
Figura 22 - Página lista de experimentos, área pública.	38

LISTA DE ABREVIATURAS E SIGLAS

ProInfo	-	Programa Nacional de Tecnologia Educacional
UFSC	-	Universidade Federal de Santa Catarina
TICs	-	Tecnologias de Informação e Comunicação
HTTP	-	Hypertext Transfer Protocol
API	-	Application Programming Interface
REST	-	Representational State Transfer
AVAs	-	Ambientes Virtuais de Aprendizagem
D2L	-	Design to Learn
Moodle	-	Modular Object Oriented Dynamic Learning Environment
LO	-	Laboratório on-line
LVs	-	Laboratórios virtuais
LRs	-	Laboratórios remotos
RLMS	-	Remote Lab Management System
ER	-	Experimentação remota
UTS	-	Universidade Tecnológica de Sydney
RELLE	-	Remote Lab Learning Environment
UFSC	-	Universidade Federal de Santa Catarina
HTML	-	HyperText Markup Language
CSS	-	Cascading Style Sheets
JS	-	JavaScript
VSC	-	Visual Studio Code
PHP	-	Hypertext Preprocessor
RAM	-	Random Access Memory
SSD	-	Solid State Drive

SUMÁRIO

1	7	
2	10	
2.1	10	
2.2	12	
Laboratórios virtuais		13
Laboratórios remotos		14
2.3	15	
2.4	17	
3	19	
3.1	20	
3.2	21	
Projetos Integrados a plataforma		22
4	23	
4.1	23	
4.1.1	24	
4.1.2	25	
4.1.3	26	
4.1.3.1	26	
4.1.3.2	29	
4.1.3.3	37	
4.1.4	38	
CONSIDERAÇÕES FINAIS		40
REFERÊNCIAS		42

1 INTRODUÇÃO

A aprendizagem utilizando dispositivos eletrônicos (Computadores, Smartphones, Tablets etc.) vêm se tornando cada vez mais comum no contexto da educação brasileira e do mundo. O programa Brasileiro ProInfo (Programa Nacional de Tecnologia Educacional) é um claro exemplo disto. Ele visa à inclusão de Tecnologias de Informação e Comunicação (TICs) nas escolas públicas como uma ferramenta de apoio ao processo ensino-aprendizagem (SOUSA et al, 2011, p. 20).

De acordo com Carlos (2017), a inclusão das TICs promoveu uma série de discussões sobre as tecnologias educacionais, o que acabou disseminando novos meios para aprimorar o processo de ensino-aprendizagem, dentre eles surgem os laboratórios *online* e remotos.

A utilização desses laboratórios por universidades vem se tornando cada vez mais comum no contexto educacional. Tais laboratórios permitem o acesso aos experimentos de forma remota, apenas utilizando a internet a fim de oferecer uma compreensão das práticas laboratoriais sem a limitação de tempo ou geográfica dos próprios alunos e professores (SANTANA et al., 2013).

Conforme relata Simão (2015), um dos pioneiros desta junção de tecnologias no Brasil é a Universidade Federal de Santa Catarina (UFSC) através do projeto RexLab, que já em 2015 atendia a 4 escolas públicas na região do Vale do Araranguá, com um total de 287 professores e 3.201 alunos, oferecendo o suporte para à realização dos experimentos remotos.

É importante ressaltar que a utilização das TICs não só em escolas e universidades, como em casa, por exemplo, vem facilitando o acesso às fontes de informação, o que acaba incentivando a busca pelo próprio conhecimento, estimulando a cultura *Maker* (*do-it-yourself* ou faça você mesmo), que, segundo Anderson (2012), é a nova revolução industrial, pois proporciona a qualquer pessoa fabricar seus próprios produtos de qualquer lugar.

A cultura *maker* pode ser uma grande aliada ao aprendizado, pois proporciona um ambiente de experimentação e de práticas do conhecimento de forma colaborativa, tornando essa cultura do “faça você mesmo” uma oportunidade de interação entre os estudantes e professores (SILVA et al., 2020).

Um projeto ícone da cultura *maker* é o Arduino, uma placa de prototipagem eletrônica de código aberto (*open-source*), que possibilita o desenvolvimento de circuitos eletrônicos, baseados em hardware e software simples, segundo a própria empresa Arduino (2021, online).

Utilizando os conceitos apresentados, o presente trabalho é uma integração de dois projetos voltados para experimentação remota, ambos desenvolvidos pelo autor desse trabalho em Milhomens Junior (2018; 2017).

O primeiro projeto, *back-end* para um software de experimentação remota com Arduino possibilita a manipulação de código no microcontrolador através de uma API REST, já o segundo apresenta uma interface que possibilita a visualização e interação dos resultados obtidos pelo trabalho anterior (O *back-end* para um software de experimentação).

É importante ressaltar que os trabalhos desenvolvidos não contam com um gerenciamento para os experimentos remotos, assim, os usuários controlam apenas um experimento por vez através de um software web, no qual programaticamente desenvolvem seus próprios projetos em Arduino. Assim, tendo em vista alcançar o objetivo deste trabalho, buscou-se responder ao seguinte problema de pesquisa: como desenvolver uma plataforma de experimentação remota para Arduino?

Levantando-se como hipótese, uma plataforma de experimentação remota para Arduino pode ser desenvolvida utilizando uma arquitetura de software em camadas composta por: a) uma API REST sobre HTTP que se comunica com projetos físicos em Arduino e permite a sua operação; e b) um front-end que utiliza tecnologias de desenvolvimento web moderno como Angular e Web Sockets e permite a operação remota dos projetos físicos em Arduino.

Baseado nisto, este projeto propõe de forma geral o desenvolvimento de uma plataforma de experimentação remota para o Arduino que contemple as funcionalidades de gerenciamento e execução dos experimentos, sendo subdividido nos seguintes objetivos específicos:

- implementar funcionalidades de gerenciamento de experimentos em uma API REST sobre HTTP;
- implementar funcionalidades para execução de projetos físicos em Arduino e seu monitoramento em um software web que se comunica com a API REST HTTP;
- desenvolver um software web para gerenciamento dos experimentos da plataforma; ao implantar um laboratório remoto é importante que ocorra o gerenciamento dos experimentos fornecidos, de modo que tais experimentos possam ser utilizados com o mínimo de desperdício, uma vez que, cada experimento físico só pode ser utilizado por uma pessoa, sendo assim, seu uso fica restrito até o fim de sua utilização, impossibilitando outros usuários realizarem testes no experimento.
- integrar a API REST ao software web para execução e monitoramento dos projetos e o software web para gerenciamento dos experimentos da plataforma.

O intuito da junção dos projetos é de alcançar usuários que buscam utilizar a tecnologia para desenvolver seus próprios protótipos, proporcionado uma interação com experimentos reais em Arduino independentemente do seu nível econômico, de forma que, através da internet

o aluno (ou aprendiz) possa ter experiências semelhantes aos laboratórios locais. Tornando-se assim uma ótima ferramenta de apoio ao ensino e aprendizagem, de modo que alunos e professores possam utilizar dos experimentos remotos em Arduino para estudar princípios de química, física, programação e robótica.

2 REFERENCIAL TEÓRICO

2.1 E-LEARNING E AMBIENTES VIRTUAIS DE APRENDIZAGEM

Nos últimos anos o computador se tornou uma excelente ferramenta de comunicação para fins educativos, o que levou a uma difusão de tecnologias com o propósito de ofertar ambientes educacionais virtuais (on-line). Estes ambientes são classificados desde e-mails, chats até as plataformas de aprendizagem educacionais, essa comunicação por intermédio do computador tem sido uma ferramenta de uso crescente no ensino superior (LITTO; FORMIGA, 2009).

A educação on-line, educação a distância e *e-learning* são conceitos comuns na área do ensino, porém não são equivalentes. A educação a distância pode ser realizada por diferentes meios, por exemplo correspondência (postal ou eletrônica), rádio, televisão, telefone, computador, internet etc. Já o conceito de educação on-line, se baseia na modalidade de educação a distância especificamente pela internet, cuja comunicação ocorre de forma síncrona ou assíncrona. *E-learning* também é uma modalidade de educação a distância por meio da internet, mas que se desenvolveu principalmente após a necessidade das empresas em treinar seus funcionários (ALMEIDA, 2003).

Enap (2006) descreve o conceito de e-learning como:

E-learning não tem como foco a educação, no entendimento mais amplo do termo, como é tratado no meio acadêmico das ciências humanas. Mas focaliza o treinamento para o trabalho, aperfeiçoamento profissional ou preparação para o mercado de uma forma eficaz com uso intensivo de tecnologia.

Com o estímulo e velocidade que a tecnologia avança, novas formas de disseminar o conhecimento surgem. Os Ambientes Virtuais de Aprendizagem (AVAs) atuam no contexto educacional como uma ferramenta poderosa, capaz de aliar diferentes mecanismos de ensino (SCHNEIDER; TULHA, 2016).

Neste cenário, o AVA consiste em um conjunto de ferramentas e recursos tecnológicos que são utilizados para mediar o processo ensino-aprendizagem a distância, classificado pelas autoras Barros e Carvalho (2011) como:

Espaços eletrônicos construídos para permitir a veiculação e interação de conhecimentos e usuários, Esses ambientes são chamados de Sistemas de Gerenciamento de Aprendizagem (do inglês: *Learning Management Systems – LMS*). São softwares projetados para atuarem como salas de aula virtuais e têm como características o gerenciamento de integrantes, relatório de acesso e atividades, promoção da interação entre os participantes, publicação de conteúdos.

sistemas computacionais disponíveis na internet, destinados ao suporte de atividades mediadas pelas tecnologias de informação e comunicação. Permitem integrar múltiplas mídias, linguagens e recursos, apresentar informações de maneira organizada, desenvolver interações entre pessoas e objetos de conhecimento, elaborar e socializar produções tendo em vista atingir determinados objetivos.

Os Ambientes Virtuais de Aprendizagem podem ser obtidos por diversos meios, sejam eles por projetos desenvolvidos pelas próprias Instituições de Ensino (IES), adquiridos por encomendas a empresas especializadas no desenvolvimento ou os AVAs implementados em *softwares* de versões livres. São exemplos de AVAs de código privado: Blackboard, Geenio e D2L (*Design to Learn*). Quanto aos Ambientes Virtuais de Aprendizagem de código aberto, são exemplos: Teleduc (desenvolvido pela Universidade de Campinas em São Paulo), Chamilo (apoiado pela associação chamilo) e o Moodle (desenvolvido por Martin Dougiamas) um dos AVAs mais utilizados por universidades no Brasil (LOTTHAMMER, 2017).

Conforme Santos (2018), a plataforma Moodle (sigla do inglês *Modular Object Oriented Dynamic Learning Environment*), ou “Ambiente de Aprendizagem Dinâmico Orientado a Objetos Modulares”, auxilia professores na construção de cursos on-line para aprendizagem ativa e colaborativa. Litto e Formiga (2009) descrevem que, na prática, os AVAs estão incentivando e ampliando as melhores práticas nas universidades, uma vez que eles fortalecem e ampliam os benefícios derivados do *e-learning*, entre os quais:

- cursos on-line, que devidamente elaborados aguçam a curiosidade e o desejo de saber mais, facilitando assim, o autodesenvolvimento;
- prática através da simulação, que aumenta a capacidade perceptiva, diminuindo as margens de erro na prática do dia-dia;
- aprendizado em grupo;
- possibilidade do *feedback*, que diminui as prováveis barreiras causadas pela distância e pelo tempo.

Um exemplo da utilização de AVAs em universidades é o RExLab, que através do projeto IntecEdu desenvolve atividades em escolas públicas da região do município de Araranguá, utilizando como ferramenta a plataforma educacional Moodle, que funciona como um suporte para que os docentes criem cursos que integrem tecnologia às suas práticas de ensino (LOTTHAMMER, 2017).

2.2 LABORATÓRIOS ON-LINE, VIRTUAIS E REMOTOS

O professor Lopes Lambertus Hesselink de Stanford foi um dos pioneiros no desenvolvimento de laboratórios virtuais, a pouco tempo, realizou um trabalho em conjunto com a sua equipe de pesquisadores. Juntos digitalizaram todos os resultados possíveis obtidos do experimento com uma nano esteira rolante, um equipamento que é extremamente caro e que poucos estudantes têm acesso, feito isso, foi disponibilizando virtualmente todos os testes, para que qualquer estudante do mundo pudesse refazê-los via internet.

O Autor Simão (2018, apud BRINSON, 2015) define o conceito de laboratório on-line (LO) como uma ferramenta tecnológica, que por meio de dispositivos computacionais auxiliam na realização de atividades experimentais. Deste modo, alunos e professores podem conduzir práticas laboratoriais de forma mais flexível e interativa, proporcionando experiências diferentes e tendo o mesmo potencial de aprendizagem que laboratórios tradicionais.

Carlos (2017) relata que, os laboratórios on-line possibilitam a inclusão de práticas laborais mesmo que à distância, assim ao utilizar deste tipo de laboratório é possível amenizar os problemas de infraestrutura que algumas instituições de ensino possuem. Tornando-se assim uma poderosa ferramenta de ensino ao proporcionar que atividades práticas sejam realizadas inúmeras vezes.

Neste sentido, os laboratórios on-line se tornam uma excelente alternativa para instituições de ensino que não possuem laboratórios físicos, disponibilizando a elas a possibilidade de superar a falta de estrutura laboratorial. No entanto, dependendo de alguns fatores como, local do laboratório, gerenciamento e o quão virtual/real é o experimento, os LO recebem algumas classificações conforme ilustra a Figura 1: Laboratórios remotos, laboratórios virtuais e laboratórios presenciais.

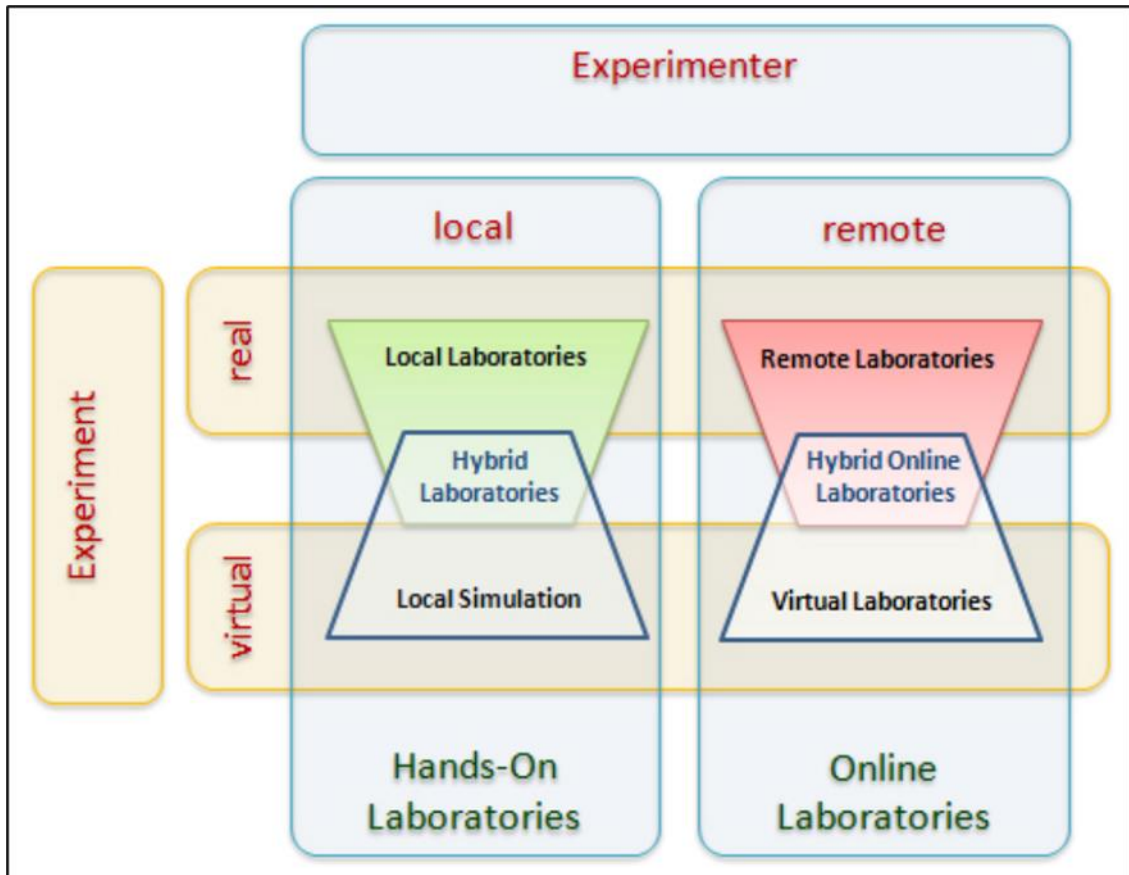


Figura 1 - Classificação dos tipos de laboratórios. Fonte: Zutin et al. (2010).

Zutin et al (2010) define que os laboratórios podem ser divididos em dois grupos principais, os reais e os virtuais. O laboratório real como o próprio nome já deixa claro é classificado como um conjunto de experimentos reais utilizados fisicamente. Já o laboratório virtual, se refere a simulações de software ou aplicativos, “imitando” o real, de modo que, os resultados serão sempre os mesmos.

Através das experiências realizadas nos laboratórios reais (*hands-on*) é possível obter dados realistas. Porém, sua utilização impõe diversas restrições, sejam elas por tempo, lugar ou interação com os equipamentos reais (tanto para professores quanto para alunos). Visto que, para um aluno utilizar o laboratório é necessário ter uma supervisão, seja ela por um técnico ou professor (NEDIC; MACHOTKA; NAFALSKI, 2003).

Laboratórios virtuais

O Surgimento dos laboratórios virtuais (LVs) veio da necessidade de utilizar experimentos em tempo real, podendo, assim, atender uma demanda maior de usuários, estando eles na mesma cidade ou geograficamente distantes. Outro ponto importante são os custos na

utilização de um laboratório real, que é bastante superior comparado ao virtual, já que não é necessário possuir os experimentos físicos no laboratório (BOTTENTUIT JUNIOR; COUTINHO, 2007).

De acordo com Chitungo (2018, apud MANCHIKANTI; KUMAR; SINGH, 2016, p. 136), a utilização das TICs em conjunto com os laboratórios virtuais revolucionou a área da educação científica de forma única. De maneira que o baixo custo das TICs utilizadas por esses laboratórios virtuais permite que os alunos aprendam os conceitos em seu próprio tempo e ritmo.

Os LVs são baseados em simulações e podem ser acessados de forma local ou remota, eles são isolados ou até mesmo integrados em sistemas para gerenciamento de *e-learning*. As simulações são frequentemente utilizadas como alternativas práticas e acessíveis para experimentação física, abordando o mesmo tipo de problemas e preocupações relacionadas à experimentação de um laboratório real (RODRIGUEZ-ANDINA; GOMES; BOGOSYAN, 2018).

Os LVs utilizam multimídia (vídeos, imagens, sons, gráficos, animações e textos) para simular o experimento em um ambiente interativo, tentando se aproximar o máximo de um real. Embora eles não permitam que novos resultados sejam descobertos, uma vez que todas as experiências são programadas para serem executadas da mesma forma, obtendo os mesmos resultados (BOTTENTUIT JUNIOR; COUTINHO, 2007).

Um exemplo de laboratório virtual é o VLab, desenvolvido na Universidade Politécnica de Bucareste, ele utiliza de equipamentos reais distribuídos entre várias universidades, de forma que os alunos remotamente realizam experimentos de física, como por exemplo o estudo das formas de onda e tensão, controlando relés e contadores de baixa tensão. A solução de software é multiusuário/cliente-servidor, já a arquitetura é desenvolvida em ambiente LabVIEW (ALBU; HOLBERT; HEYDT; GRIGORESCU; TRUC, 2004).

Laboratórios remotos

Desde o surgimento da internet e sua popularização, várias soluções foram viabilizadas nesse contexto. Por exemplo, os laboratórios remotos, experimentos reais acessados por meio da internet que complementam o ensino laboratorial (MÜLLER; ERBE, 2007).

O uso de laboratórios remotos (LRs) por universidades está se tornando cada vez mais comum, pois proporciona aos estudantes acesso aos experimentos em horários flexíveis, uma vez que os experimentos podem ficar disponíveis até 24 horas por dia, não oferecendo risco nenhum para o aluno na utilização do equipamento físico (SANTANA et al., 2013).

Autores como Carlos et al. (2001) afirmam que a maioria dos LRs são disponibilizados através de RLMS (*Remote Lab Management System* ou Sistema de Gerenciamento de Laboratórios Remotos). Os RLMSs são utilizados para suportar os laboratórios e os conteúdos de apoio, como por exemplo, manuais de utilização, guias didáticos, dentre outros (CARLOS et al. 2017).

Os laboratórios remotos, geralmente dependem de uma arquitetura cliente-servidor, permitindo através da Internet a comunicação com o equipamento real. O servidor de um laboratório pode ser um microcontrolador, um computador ou um sistema de gerenciamento de laboratório remoto RLMS (do inglês *Remote Laboratory Management System*). O servidor de laboratório tem de assegurar a interação remota do usuário, partindo de um aplicativo autônomo ou como um componente em um ambiente da Web. O usuário geralmente observa o laboratório através de um fluxo de vídeo ao vivo, defini parâmetros de configuração e visualiza os dados ou medição em tempo real (GOLINELLI et al., 2018).

Santos (2018) descreve os laboratórios virtuais e remotos como ferramentas tecnológicas, que, por meio da internet, tornam disponíveis em tempo integral o acesso aos experimentos, abrindo espaço para elaboração de atividades mais dinâmicas e atrativas por parte dos professores, ajudando no entendimento dos seus alunos o que acaba estimulando o interesse e curiosidade por parte deles.

2.3 EXPERIMENTAÇÃO REMOTA

Conforme Alves et al. (2016), a experimentação remota surgiu como uma alternativa para suprir a falta de laboratórios na realização de atividades práticas, por exemplo em disciplinas como ciências. A experimentação remota possibilita ao usuário acesso a recursos que ele não possui, permitindo que experimentos práticos possam ser realizados de qualquer lugar e a qualquer hora.

A experimentação remota (ER) no ambiente educacional oferece o acesso a laboratórios físicos/reais, em que, por meio da internet, os alunos têm a possibilidade de interagir com experimentos reais e fundamentais para compreensão dos conceitos relacionados ao ensino de ciências (ROCHADEL, 2013)

De acordo com Alves et al. (2016, p. 64, apud MARCELINO et. al, 2011), mesmo que o laboratório esteja fisicamente separado do estudante, ao utilizar a experimentação remota, o aluno poderá interagir e controlar instrumentos e dispositivos reais, por meio de interfaces que intermediam a conexão, a seguinte imagem (Figura 2) representa o fluxo de um experimento remoto.

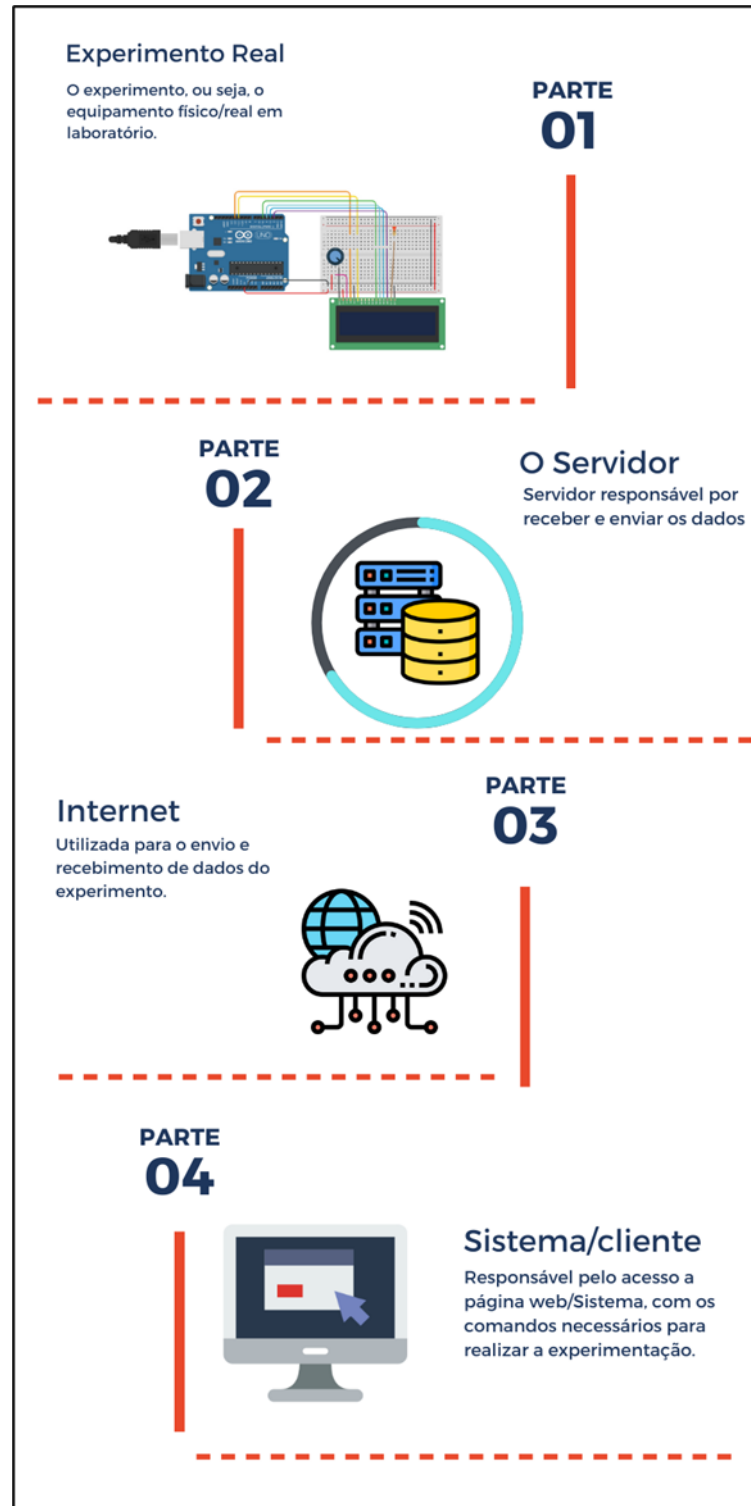


Figura 2 - Representação do modelo de um laboratório remoto.

No modelo de experimento remoto apresentado na Figura o equipamento de laboratório (Parte 1) é conectado ao servidor (Parte 2), que utiliza da internet para o envio e recebimento de dados do experimento (Parte 3), por fim, o cliente/sistema, ou seja, a interface responsável pelo acesso ao experimento (Parte 4), por onde o aluno poderá interagir com equipamento real.

2.4 MODELOS DE FUNCIONAMENTO DE LABORATÓRIOS REMOTOS

Simão (2018) relata que o gerenciamento é uma das atividades críticas no processo de desenvolvimento dos laboratórios remotos, uma vez que, ao disponibilizá-los o provedor precisa ficar atento a uma série de fatores (confiabilidade, autorização, tolerância a falhas e segurança) que podem acabar influenciando em todo o funcionamento do laboratório. Por esse motivo, as garantias acabam criando um conjunto de funcionalidades comum entre todos os laboratórios, independentemente de sua finalidade.

O *Remote Laboratory Management System* da sigla inglesa RLMS é o sistema responsável pelo gerenciamento dos laboratórios e seus experimentos, ele inclui funções como autenticação, permissão, agendamento e rastreamento de usuário (ORDUÑA et al., 2016). A seguir, são apresentados alguns exemplos de laboratórios remotos.

Labshare Sahara foi desenvolvido pela Universidade Tecnológica de Sydney (UTS) entre os anos de 2000 e 2005. O sistema Sahara oferece funcionalidades de agendamento e suporte à múltiplas instâncias de um experimento, permitindo o acesso de forma remota apenas após a autenticação, deste modo, o usuário (autenticado) inicia uma sessão e tem acesso aos experimentos durante um período (LOWE; MACHET; KOSTULSKI, 2012).

Conforme Simão (2018) o Labshare Sahara se divide em três modelos de acesso:

- Controle periférico: o acesso ao experimento ocorre por uma interface externa, assim, o Sahara não tem controle sobre a comunicação com o hardware, porém, ainda é responsável por atribuir o acesso ao programa de controle, revogar o acesso, iniciar a experiência e monitorar os resultados dos testes, junto com a detecção de atividade na plataforma.
- Controle Direto ou Primitivo: a aplicação do cliente é quem controla o hardware.
- Controle em Lotes: o usuário não interage diretamente com o experimento, neste caso, o cliente é responsável por carregar o arquivo de controle contendo as instruções e valores para a execução do experimento.

De acordo com Lima (2016), o RLMS WebLab-Deusto foi desenvolvido em meados do ano 2000 pela Universidade de Deusto (Espanha). Este RLMS passou por diversas atualizações em termos de funcionalidades e tecnologias, um dos principais motivos foi a compatibilidade a novas tecnologias de desenvolvimento web. Atualmente encontra-se na versão 5.0.

A Figura 3 apresenta a arquitetura cliente-servidor adotada pelo WebLab-Deusto, que disponibiliza laboratórios em espécie de “salas”, fazendo com que todas as conexões de um

dado laboratório sejam enviadas ao servidor. Os alunos geralmente interagem com o servidor de *login* para autenticação e com Core Server para o resto das operações (autorização, agendamento e armazenamento).

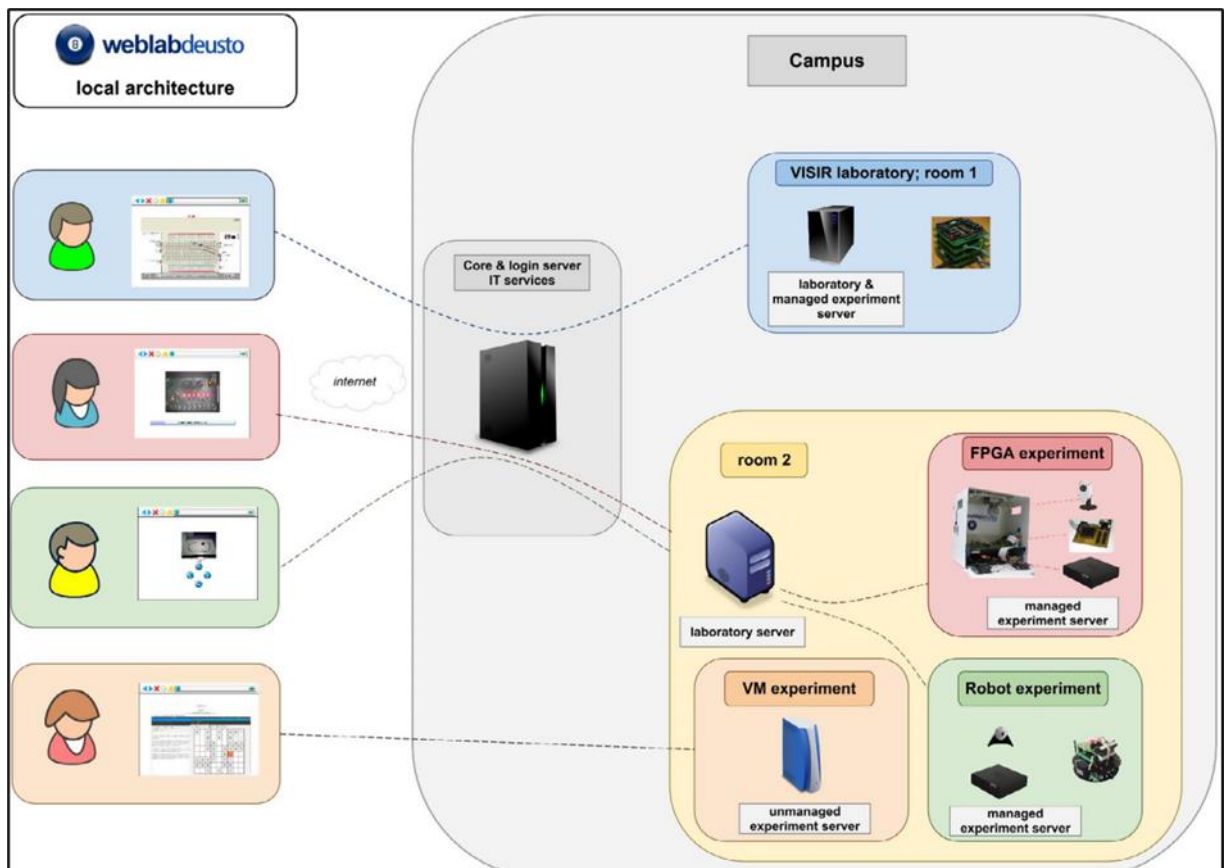


Figura 3 - Visão geral do WebLab-Deusto (arquitetura). Fonte: Orduna et al. (2014).

O WebLab-Deusto provê uma API para criação de novos experimentos, autenticação, agendamento, rastreamento e gerenciamento de usuários, além de permitir o compartilhamento de experimentos entre diferentes instituições. (SIMÃO, 2018).

Por fim, o Remote Lab Learning Environment (RELLE) desenvolvido no ano de 2015 pela Universidade Federal de Santa Catarina (UFSC), este sistema é responsável pelo gerenciamento de usuários e os experimentos do laboratório (RExLab). Segundo Simão (2018) dentre as funcionalidades do RELLE, estão:

- Gerenciamento dos usuários;
- Gerenciamento dos experimentos remotos;
- Emissão de relatórios;
- Controle de acessos à plataforma.

Ele foi construído sobre a arquitetura RLMS dividida em 3 camadas, elementos de software/hardware (mecatrônica responsável pelo acionamento de atuadores), captação de dados e tecnologias de interface gráfica dispostas por meio da aplicação para o usuário final (CARLOS, 2017). Conforme Nicolete (2016), o RELLE provê um cliente web comum para cada experimento, ele é responsável por carregar e apresentar os dados que foram inseridos no momento da publicação (armazenados em uma base de dados). Figura 4 apresenta a tela de acesso ao RELLE.

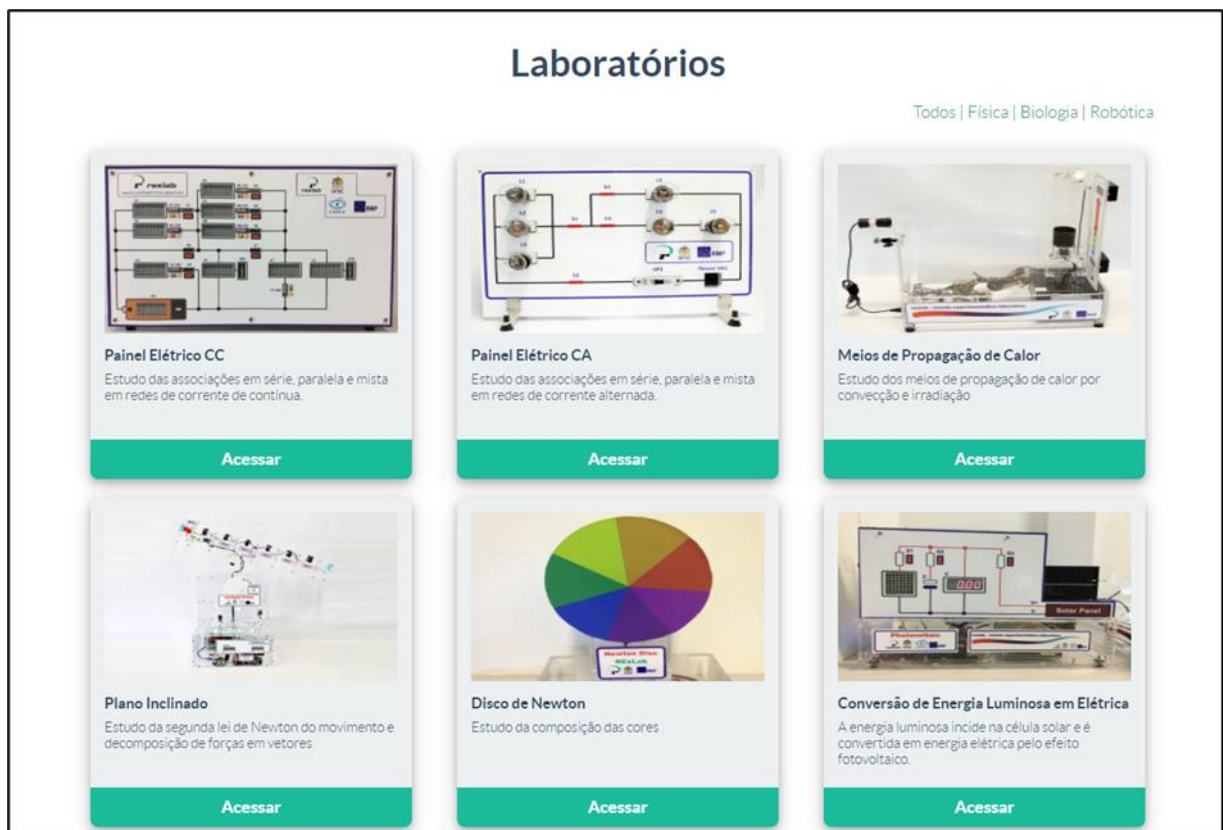


Figura 4 - Ambiente de aprendizagem com experimentos remotos. Fonte: <http://relle.ufsc.br/>

O acesso aos experimentos é aberto a usuários visitantes, porém algumas funções do RELLE são disponibilizadas apenas para usuários autenticados. Usuários com permissões de administrador têm acesso às funções de gerenciamento dos usuários e experimentos registrados. Sem estas permissões, os usuários podem apenas editar e excluir o seu perfil (NICOLETE, 2016).

3 METODOLOGIA

3.1 MÉTODOS

Os métodos adotados para o desenvolvimento deste trabalho são apresentados neste capítulo. A Figura a seguir (Figura 5) apresenta as nove etapas da metodologia que foram utilizadas para o desenvolvimento da plataforma de experimentação remota com Arduino.



Figura 5 - Metodologia do Trabalho.

A metodologia de desenvolvimento apresentada teve início a partir das reuniões com o professor orientador Fábio Castro Araújo (fase 1), essas reuniões aconteceram de forma remota pelo menos três vezes na semana, o intuito desses encontros virtuais foi de planejar e definir as fronteiras do projeto. O acompanhamento do orientador esteve presente em todas as fases (nove fases), apoiando e seguindo o cronograma das atividades que foram desenvolvidas.

A segunda fase, estudo e elaboração, foi responsável pelo desenvolvimento do referencial teórico, de modo que os conceitos apresentados, *e-learning*, ambientes virtuais de aprendizagem (AVA), laboratórios on-line, virtuais e remotos, experimentação remota e modelos de funcionamento de laboratórios remotos, foram escritos nesta etapa.

Com base na pesquisa realizada foi possível dar continuidade a fase 3 (levantamento e definição dos requisitos), que consistiu em construir e elaborar os artefatos destinados ao desenvolvimento da plataforma, diagramas de modelagem, modelo de dados e diagramas de

caso de uso. Na quarta fase (definição das ferramentas para implementação) definiu-se as ferramentas e materiais que foram utilizadas para o desenvolvimento da plataforma, que estão descritas na seção [3.2 MATERIAIS](#).

A fase 5 (planejamento do *design* da plataforma) foi responsável pelo detalhamento do formato de saída, de modo que foi produzido *wireframes* para interface com usuário, relatórios, telas de cadastro (usuário ou experimentos), assim como a tela de utilização dos experimentos remotos.

Já as fases 6 e 7 tiveram início depois da elaboração dos artefatos construídos nas fases anteriores, utilizando desses documentos e ferramentas para a produção da plataforma. Desenvolvendo o processo de gerenciamento dos usuários, experimentos, permissões e as funcionalidades para emissão de relatórios. O gerenciamento disponibiliza as quatro operações básicas (CRUD): *create* (criar), *read* (ler), *update* (atualizar) e *delete* (excluir).

Para os testes de funcionalidades (fase 8) durante o processo de desenvolvimento foram adotados os testes de unidade ou *unit test*, o intuito da sua utilização foi de revelar possíveis falhas, possibilitando a correção dos erros. Testes para a performance da plataforma também foram adotados (testes de carga e estresse), a fim de verificar a quantidade de acessos suportada pelo servidor onde está o sistema.

3.2 MATERIAIS

Esta seção apresenta as tecnologias e materiais utilizados para o desenvolvimento deste trabalho:

- *HyperText Markup Language* (HTML): de acordo com a W3SCHOOLS (2021), HTML é uma linguagem de marcação padrão para a criação de páginas da web. Consiste em uma série de elementos e estruturas que informam ao navegador como exibir o conteúdo, seja eles por blocos ou *tags*. O *HyperText Markup Language* identifica as partes do conteúdo como "isto é um título", "isto é um parágrafo", "isto é um link" etc;
- *Cascading Style Sheets* (CSS): conforme a *World Wide Web Consortium* (2021), CSS ou Folhas de Estilo em Cascata é um mecanismo para gerenciar estilos (fontes, cores, espaçamento etc.) em um documento da web, por exemplo HTML. "CSS é usado para definir estilos para suas páginas da web, incluindo o design, layout e variações de exibição para diferentes dispositivos e tamanhos de tela." (W3SCHOOLS, 2021);
- JavaScript (JS): segundo Docs (2021) JavaScript é uma linguagem de programação leve baseada em objetos, mais conhecida como a linguagem de script para páginas Web, não precisando de pré-processamento (por exemplo, compilação) antes de ser executada. O

JavaScript é baseado em protótipos, multiparadigmas etc. suportando estilos de orientação a objetos, imperativos e declarativos;

- JQuery: é uma biblioteca rápida e leve que mudou a maneira como milhões de pessoas escrevem JavaScript. Ela permite a passagem e manipulação de documentos HTML, eventos, animações etc. de forma simples. A biblioteca utiliza uma interface de programação de aplicações (API) compreensível, que funciona em vários navegadores atuais (FOUNDATION; JQUERY, 2021);
- *Visual Studio Code* (VSC): Code (2021) define o VSC como um editor de código simples, mas poderoso, com suporte integrado a várias linguagens de programação como JavaScript e TypeScript. Essa ferramenta possui um rico ecossistema de extensões para outras linguagens (como C ++, C #, Java, Python, PHP, Go), além de temas, fontes etc;
- *Hypertext Preprocessor* (PHP): O PHP é uma linguagem de script open source (código aberto) voltada para o desenvolvimento web no lado do servidor, podendo ser embutida dentro do código HTML (PHP, 2021 online). Para a implementação da plataforma de experimentação remota com Arduino foi aplicado o padrão cliente-servidor, sendo que, no lado do servidor a linguagem de programação definida foi o PHP.

Os materiais e ferramentas recém apresentados foram escolhidos para o desenvolvimento da interface com o cliente, de modo que, o software *Visual Studio Code* foi selecionado para auxiliar no processo de codificação. A linguagem de marcação HTML foi utilizada para a construção das páginas web, trabalhando em conjunto com o CSS para a estilização. O JavaScript e JQuery foram responsáveis pela manipulação das páginas, por meio de eventos, animações etc.

Projetos Integrados a plataforma

O *back-end* para um software de experimentação remota com Arduino, já desenvolvido pelo autor Milhomens Junior (2018), foi incorporado à plataforma. O trabalho de pesquisa mencionado proporciona ao usuário (aluno ou aprendiz) a utilização de experimentos com o microcontrolador Arduino, usando como base os conceitos de experimentação remota, do mesmo modo, a aplicação permite que qualquer usuário que a utilize tenha a interação com experimentos reais de forma simples e econômica. Os materiais que foram utilizados para execução da aplicação consistem em:

- software: linguagem de programação PHP, framework Silex, PlatformIO, linguagem de programação Python; e
- hardware: placa Arduino e um computador utilizado como servidor.

Outro trabalho de pesquisa vinculado a plataforma é o software web para experimentação remota com Arduino, também desenvolvido pelo autor Milhomens Junior (2017). O projeto citado apresenta uma interface que possibilita a visualização e interação dos resultados obtidos pelo trabalho anterior (O *back-end* para um software de experimentação).

4 RESULTADOS E DISCUSSÃO

Esta seção apresenta os resultados obtidos no desenvolvimento do presente trabalho de pesquisa, bem como as discussões referentes aos resultados. Apresentando a arquitetura adotada, os serviços utilizados em conjunto com as ferramentas.

4.1 DESENVOLVIMENTO

A arquitetura que compõe a plataforma de experimentação remota para Arduino foi subdividida em módulos de utilização, esses módulos são compostos pela integração dos trabalhos correlacionados e as tecnologias utilizadas para o desenvolvimento. Tal arquitetura está ilustrada na Figura 6.

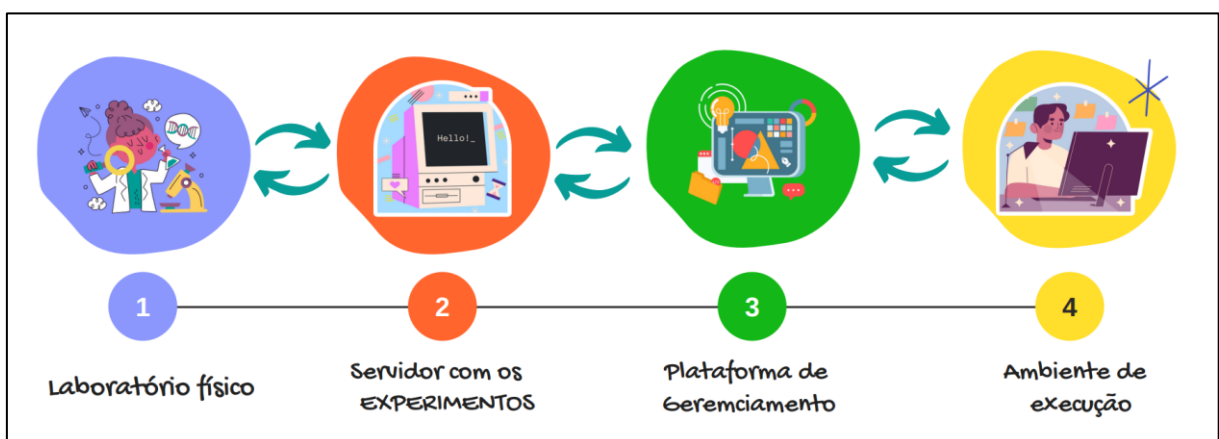


Figura 6 - Arquitetura.

A arquitetura foi planejada e dividida em 4 módulos principais: laboratório físico, que é o ambiente no qual os projetos em Arduino (reais) estarão previamente montados; Servidor

com os experimentos, responsável pela disponibilização dos experimentos na rede; Plataforma de gerenciamento, que é o sistema responsável pelo gerenciamento dos usuários e experimentos; e, por fim, o ambiente de execução responsável pela interface de manipulação dos projetos em Arduino.

A comunicação entre os módulos é representada pelas setas bidirecionais, isso indica que a informação vai de um módulo ao outro de forma hierárquica, permitindo que as instruções fornecidas pelo usuário sejam processadas em seu devido módulo retornado ao ambiente de execução com o resultado esperado.

4.1.1 Laboratório físico

Os experimentos utilizados no desenvolvimento da plataforma possuem a mesma arquitetura básica, levando em conta o aprendizado inicial na utilização do Arduino, assim o intuito é facilitar o processo de montagem e de codificação. Cada um é composto por um Arduino UNO, visor LCD 16x2, potenciômetro, sensor de luminosidade, três LEDs e uma *protoboard*, conforme ilustra a Figura 7. Importante salientar que é totalmente possível modificar os itens acoplados ao microprocessador Arduino, de modo que não está restrito apenas a esses componentes apresentados, podendo adicionar ou remover itens conforme a necessidade de utilização.

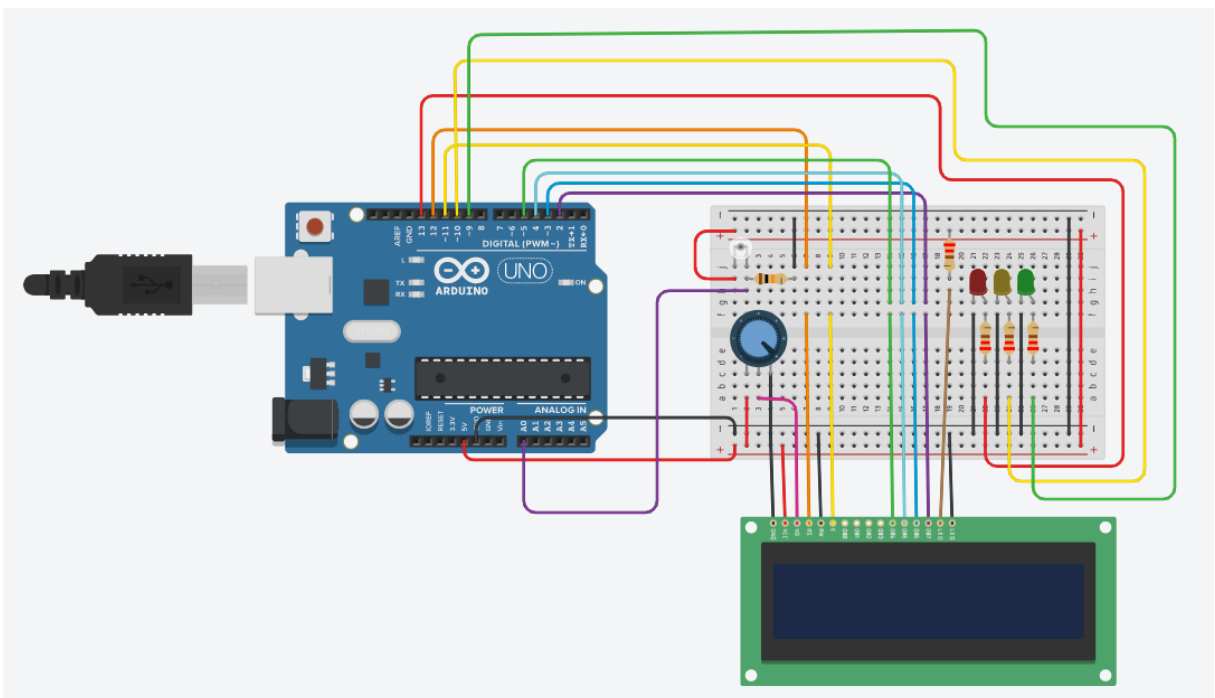


Figura 7 - Experimento sugerido.

O experimento sugerido na Figura 7 é utilizado na plataforma para o processo de experimentação remota com Arduino. Com base no experimento sugerido, o usuário pode, via internet, realizar a verificação e carregamento de códigos remotamente, além do processo de codificação é possível monitorar o experimento em tempo real por meio de uma *webcam*.

4.1.2 Servidor com os experimentos

Esta seção é destinada a exemplificação e explicação do processo de implementação do servidor de experimentos para a plataforma. O servidor citado foi desenvolvido utilizando o trabalho do autor Milhomens Junior (2018). A aplicação referenciada possibilita a manipulação de código no microcontrolador Arduino, um processo crucial para experimentação remota da plataforma.

Primeiramente, foi utilizado um computador Desktop com o sistema operacional *Windows* 64bits (v. 10 pro), com o processador 11th Gen Intel(R) Core i5, 8 gigas de memória RAM e um SSD com capacidade de armazenamento de 500 gigas, ele foi configurado e utilizado como servidor APACHE funcionando de forma semelhante a um servidor matriz. Foi acoplado a ele um microprocessador Arduino uno com as especificações sugeridas no tópico anterior (4.1.1 Laboratório físico), ao incorporar o *back-end* para um software de experimentação remota com Arduino no servidor APACHE podemos enviar um conjunto de instruções para o microcontrolador, esse fluxo está representado pela arquitetura da aplicação como ilustra a Figura 8.

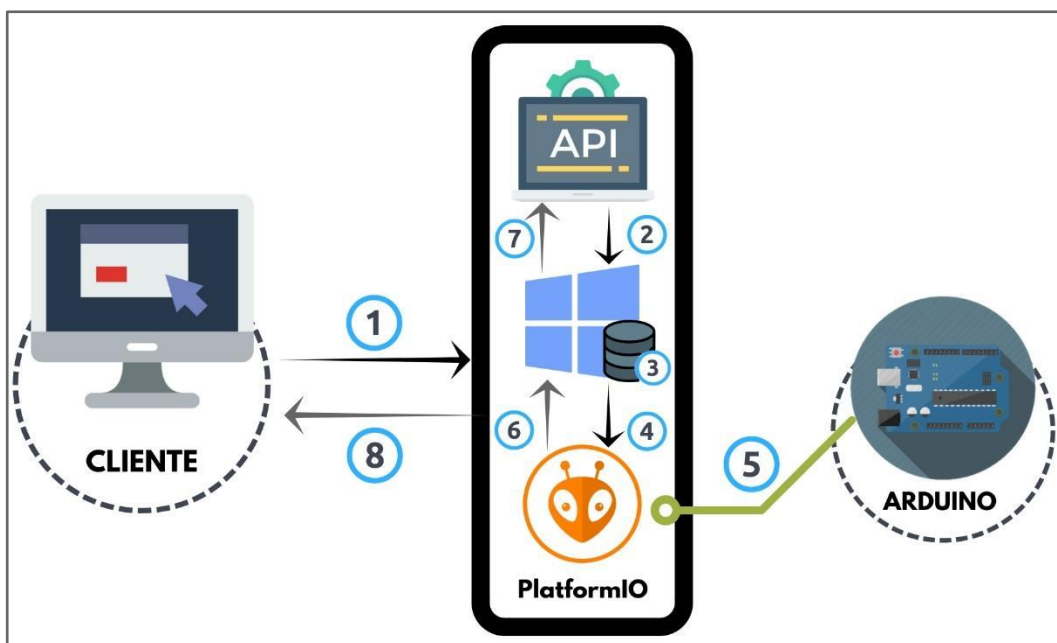


Figura 8 - Arquitetura do Back-end para um software de experimentação remota. Fonte: Milhomens Junior (2018).

Conforme ilustra a Figura 8 o fluxo da aplicação foi dividido em 8 passos:

No primeiro passo (1) ocorre a requisição da API por parte da aplicação do cliente, já no segundo passo a API trata e trabalha as informações recebidas conforme a solicitação. O próximo passo (terceiro passo) é situacional, ele é responsável pelo armazenamento do código arduino no servidor, porém, só ocorre quando se utiliza a rota verifica que está encarregada de receber esse código-fonte Arduino via requisição. O quarto passo corresponde a utilização da PlatformIO através de procedimentos realizados no shell do SO (sistema operacional), esses procedimentos são realizados de forma programática pela API utilizando PHP.

O quinto passo dá início ao procedimento de entrada e saída de dados a placa arduino uno, de modo que, está dividido em duas etapas, a compilação e a saída de retorno. Na primeira etapa o código-fonte localizado no servidor é convertido para binário e, então, gravado na EEPROM da placa Arduino utilizando a biblioteca PlatformIO (por exemplo, por meio da conexão USB). A segunda parte já faz parte do sexto passo, responsável por emitir um resultado pela PlatformIO através do shell do sistema operacional.

O próximo passo (sétimo passo) é designado a coletar as informações de saída geradas pelo shell (passo 6), esse processo de coleta de informações é realizado através de dois comandos, um responsável pela saída de sucesso e outro a saída de erro, por fim, no sentido de gerar uma saída mais “amigável” para o usuário da API o conteúdo dos arquivos é processado e transformado. Por último, o oitavo passo responsável por retornar um json estruturado para a aplicação do cliente conforme sua requisição.

4.1.3 Plataforma de gerenciamento

A plataforma de gerenciamento está subdividida em duas partes principais, uma responsável pela gerência de experimentos e usuários, outra responsável pela exibição, manipulação e controle de acesso dos experimentos. No decorrer desta seção serão demonstrados os fluxos e telas de cada uma dessas partes.

4.1.3.1 Banco de Dados

A camada de banco de dados é a parte da aplicação na qual são armazenadas todas as informações não temporárias da ferramenta, ela foi desenvolvida para trabalhar com o banco de dados MySQL, um sistema para o gerenciamento de banco de dados que utiliza a linguagem

SQL como interface. Ele é responsável por armazenar e gerenciar os dados referentes aos experimentos e usuários da plataforma, sendo assim foi elaborado o modelo de banco de dados conforme ilustra a Figura 9.

tccexperimento experimentos	
ID	int(10) unsigned
titulo	varchar(100)
tema	varchar(10)
descricao	text
tipoDeAcesso	binary(1)
linkWebCam	varchar(200)
linkRotaUpload	varchar(200)
linkRotaVerifica	varchar(200)
srcImagemExperimento	varchar(200)
tipoSessao	binary(1)
ativo	binary(1)
dataCadastro	timestamp

tccexperimento usuarios	
ID	int(10) unsigned
nome	varchar(100)
senha	varchar(100)
email	varchar(100)
tipoUsuario	binary(1)
ativo	binary(1)
dataCadastro	timestamp

Figura 9 - Diagrama de banco de dados da plataforma.

A Figura 9 ilustra o diagrama de banco de dados da plataforma, essa estrutura é composta por 2 tabelas, cada uma contendo um conjunto de colunas para armazenar as informações necessárias para utilização da ferramenta. O diagrama é composto pelas tabelas “Experimentos” e “Usuários”. Os tópicos a seguir descrevem a função de cada uma delas.

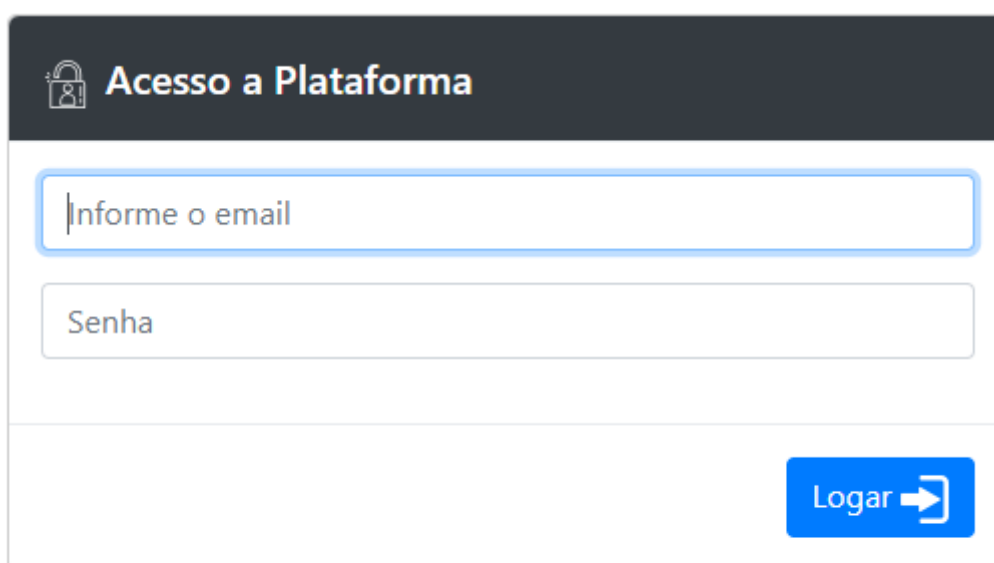
- **Experimentos** - Essa tabela contém todos os dados dos Experimentos cadastrados na plataforma, as principais informações são: “ID”, “titulo”, “tema”, “descrição”, “tipoDeAcesso”, “linkWebCam”, “linkRotaUpload”, “linkRotaVerifica”, “srcImagemExperimento”, “tipoSessao”, “ativo” e “dataCadastro”.
 - “ID”: armazena o código identificador de cada experimento, esse código é único e autoincrementado.
 - “titulo”: armazena a *string* referente ao título do experimento.
 - “tema: armazena a *string* contendo o código hexadecimal das cores (#ff0000 cor vermelha) que serão utilizadas na interface.
 - “descricao”: armazena o texto descrevendo o experimento.

- “tipoDeAcesso”: armazena o código referente ao tipo de experimento da plataforma, através desse atributo os experimentos com o “tipoDeAcesso” serão classificados como públicos ou Privados.
 - “linkWebCam: armazena a *string* contendo a url (*Uniform Resource Locator*) para acesso da webcam responsável pela captura de imagem do experimento.
 - “linkRotaUpload: armazena a *string* contendo a url responsável pelo carregamento do código fonte para o microcontrolador Arduino.
 - “linkRotaVerifica: armazena a *string* contendo a url responsável pela verificação da sintaxe e semântica do código fonte.
 - “srcImagemExperimento: grava o caminho da imagem armazenada no servidor, imagem responsável pela visualização da estrutura do circuito.
 - “tipoSessao: armazena o booleano referente ao tipo de sessão do experimento, *true* para experimentos que a sessão pode ser renovada e *false* experimentos com tempo de utilização limitado sem renovação.
 - “ativo”: armazena o booleano do status de cada experimento, *true* para experimentos ativos e *false* para inativos da plataforma.
 - “dataCadastro”: armazena a data de cadastro do registro no banco.
- **Usuário** - Essa tabela contém todos os dados dos usuários cadastrados na plataforma, as principais informações armazenadas são: “ID”, “nome”, “senha”, “email”, “ativo”, “tipoUsuario” e “dataCadastro”.
 - “ID”: armazena o código identificador de cada usuário, esse código é único e autoincrementado.
 - “nome”: armazena a *string* referente ao nome do usuário.
 - “senha: armazena a *string* criptografada pela família de algoritmos de criptografia SHA-2.
 - “email”: armazena a *string* referente ao e-mail de cada usuário, esse e-mail é único e responsável pela autenticação na plataforma.
 - “ativo”: armazena o booleano do status de cada usuário, *true* para usuários ativos e *false* para usuários inativos da plataforma.
 - “tipoUsuario”: armazena o código referente ao tipo de usuário da plataforma, através desse atributo os usuários com o “tipoUsuario” administrador poderão gerenciar os experimentos, já os classificados como público poderão acessar os experimentos Privados na plataforma.

- “dataCadastro”: armazena a data de cadastro do registro no banco.

4.1.3.2 Área administrativa da plataforma

Ao acessar a área administrativa é necessário realizar a autenticação/login através dos dados cadastrados, um usuário pode se autenticar através do e-mail e senha. Após sucesso na autenticação o usuário administrador poderá ter acesso aos recursos e funcionalidades de gerência do sistema e o usuário público poderá ter acesso a manipulação de experimentos privados, tendo em vista que, experimentos públicos não necessitam de cadastro na plataforma para sua utilização.

A imagem mostra a interface de login da plataforma. No topo, há um cabeçalho escuro com um ícone de cadeado e o texto "Acesso a Plataforma". Abaixo, há dois campos de entrada: "Informe o email" e "Senha". Um botão azul com o texto "Logar" e uma seta para a direita está localizado no canto inferior direito da área de login. Abaixo do formulário, há um link azul que diz "Não possui uma conta? Criar conta".

Não possui uma conta? [Criar conta](#)

Figura 10 - Tela de login da Plataforma.

A tela de login ilustrada na Figura 10 demonstra o primeiro contato com a parte administrativa da plataforma, onde um usuário administrador realiza a sua autenticação podendo então ter acesso a área administrativa. De modo mais técnico essa comunicação é feita entre Front-end e Back-end, mais especificamente a *controller login* responsável pela gerência dos métodos de pesquisa no banco de dados, nela ocorre o tratamento e busca das informações referentes ao usuário informado.

A princípio os dados são capturados na tela de Login nos campos Email e Senha (Figura 9) e a *controller login* verifica se eles foram informados corretamente ou não, sendo

comunicado em cada campo a sua obrigatoriedade e no caso de inserção de dados inconsistentes são retornadas mensagens instrutivas como ilustrado na Figura 11.

The figure shows two states of a login form titled "Acesso a Plataforma".

- Left Screenshot:** The form has two input fields: "Informe o email" and "Senha". Below the email field is a red error message: "Por favor, informe o email.". Below the password field is another red error message: "Por favor, informe a senha.". A blue "Logar" button is at the bottom right. Below the form is a link: "Não possui uma conta? [Criar conta](#)".
- Right Screenshot:** The form has the same fields. The "Informe o email" field has a blue border. A pink error message at the bottom reads: "Email ou senha inválidos". The "Logar" button is still present. The "Criar conta" link is also present.

Figura 11 - Mensagens com valores inválidos.

Em caso de Sucesso, quando as informações fornecidas pelo usuário são encontradas e verificadas na base dados é gerada uma nova sessão de acesso ao sistema, de modo que o usuário é redirecionado para a tela inicial da área administrativa, conforme ilustra a Figura 12.

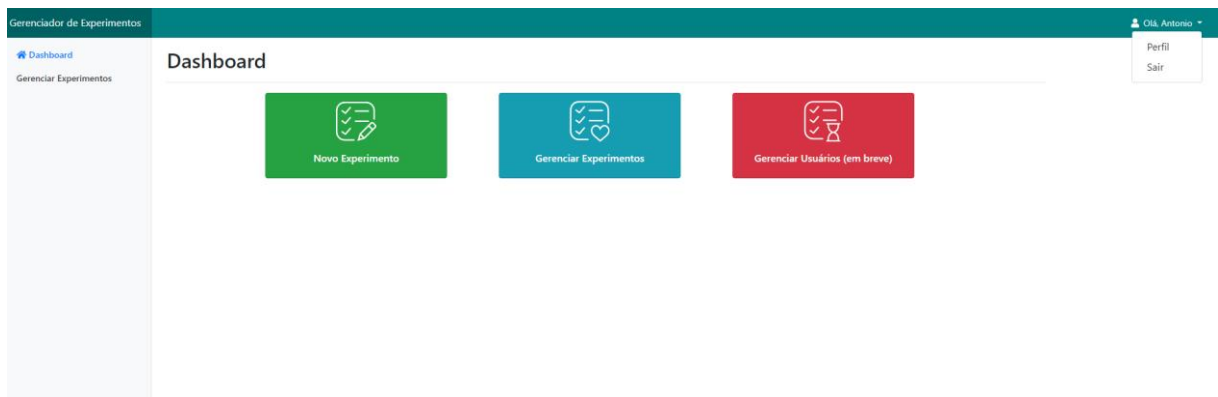


Figura 12 - Dashboard da área administrativa.

Na tela inicial são apresentados três *cards* de acesso aos serviços da área administrativa. O card ilustrado com a cor verde (Novo Experimento) dá acesso direto ao formulário de cadastro dos experimentos, conforme ilustra a Figura 13.

The screenshot shows the 'Novo experimento' form in the 'Gerenciador de Experimentos' application. The form is titled 'Novo experimento' and is located in the 'Gerenciar Experimentos' section. The form contains several fields and options:

- Título:** A text input field with a red asterisk indicating it is required. The placeholder text is 'Título do experimento'.
- Descrição:** A text area with a red asterisk indicating it is required. The placeholder text is 'Descrição do experimento'.
- Tema:** A dropdown menu with a red asterisk indicating it is required. The selected option is 'Tema'.
- Tipo de Acesso:** Radio buttons for 'Público' (selected) and 'Autenticado'.
- Sessão Renovável?:** Radio buttons for 'Sim' (selected) and 'Não'.
- Link da WebCam:** A text input field with a red asterisk indicating it is required. The placeholder text is 'Link para acesso a webCam'.
- Rota Upload:** A text input field with a red asterisk indicating it is required. The placeholder text is 'Link para acesso a rota'.
- Rota Verifica:** A text input field with a red asterisk indicating it is required. The placeholder text is 'Link para acesso a rota'.
- Foto da Arquitetura:** A file upload field with a red asterisk indicating it is required. The placeholder text is 'Escolher arquivo' and 'Nenhum arq...o selecionado'. There is a checkbox for 'Ativo' which is checked.

A green 'Salvar' button is located at the bottom right of the form.

Figura 13 - Formulário de cadastro dos experimentos.

Após o usuário ser redirecionado à tela responsável pelo cadastro de um experimento, são apresentados os campos contendo os dados requeridos para o gerenciamento. Cada campo obrigatório é marcado por um asterisco em vermelho como ilustrado pela Figura 13. A validação dos campos se dá no momento em que o formulário é enviado da aplicação (front-end) para o método *save* localizado na *controller Project* (back-end) através do botão Salvar.

Ao clicar no botão Salvar é feita uma requisição para a *controller Project*, conforme citado anteriormente, caso as informações fornecidas pelo usuário precisem de alguma correção é retornada a mensagem de erro e apresentada uma validação visual para cada campo em ordem de correção, como ilustrado na Figura 14.

Figura 14 - Formulário de cadastro dos experimentos, erro de preenchimento.

Caso a requisição feita ao método *save (controller Project)* tenha todos os campos preenchidos corretamente é gerado um novo registro no banco de dados com as informações fornecidas pelo usuário administrador. Logo em seguida é apresentada uma mensagem de sucesso redirecionando o usuário a página de gerenciamento de experimentos, conforme a Figura 15.

ID	Nome	Descrição	Acesso	Data de Cadastro	Ativo	Ações
1	TESTANDO CADASTRO ARDUINO	Mussum Ipsum, cacilds vidis litro abertis. A ordem dos tratores não altera o pão duris. Si num tem...	Público	2021-10-18 22:47:06	Sim	
2	EXPERIMENTO 2	TESTANDO NOVO EXPERIMENTO COM TEXTO	Autenticado	2021-10-18 20:49:35	Sim	
3	Testando novo experimento TCC	Testando novo experimento TCC texto descritivo armazenado. Testando novo experimento TCC texto descr...	Público	2021-11-22 16:46:03	Sim	

Figura 15 - Página gerenciar experimentos, mensagem de cadastro com sucesso.

O segundo *card* listado na *dashboard* inicial (Figura 12), está relacionado à tela Gerenciar Experimentos, apresentada na Figura 16. O usuário administrador poderá acessá-la para cadastrar um novo experimento (Parte A), filtrar os registros cadastrados (Parte B), alterar um experimento já existente (Parte C), como também desativar um registro (Parte D).

ID	Título	Descrição	Acesso	Data de Cadastro	Ativo	Ações
1	TESTANDO CADASTRO ARDUINO	Mussum Ipsum, cacilds vidis litro abertis. A ordem dos tratores não altera o pão duro. Si num tem...	Publico	2021-10-18 22:47:05	Sim	[Editar] [Excluir]
2	EXPERIMENTO 2	TESTANDO NOVO EXPERIMENTO COM TEXTO	Autenticado	2021-10-18 20:49:35	Sim	[Editar] [Excluir]
3	Testando novo experimento TCC	Testando novo experimento TCC texto descritivo armazenado. Testando novo experimento TCC texto descr...	Publico	2021-11-22 16:46:03	Sim	[Editar] [Excluir]

Figura 16 - Página gerenciar experimentos.

A Figura 16 ilustra as quatro ações principais da tela de gerenciamento, indicadas com as letras de A à D. A parte marcada com a letra A corresponde a tela de cadastro de experimentos, tela ilustrada pela Figura 13 e exemplificada anteriormente.

Em sequência, a parte indicada pela letra B, que corresponde ao filtro de pesquisa, é responsável por realizar a busca dos experimentos com base no texto fornecido pelo usuário. Ao clicar no botão filtrar é disparada uma requisição ao método *list* localizado na *controller Project*, retornando a lista com os registros encontrados para página de gerência de experimentos.

Já a ação de editar um registro existente está indicada pela letra C, ao clicar no botão editar é disparado uma requisição com o parâmetro ID do experimento, essa requisição está direcionada ao método *Details* localizado na *controller Project*, por meio desse método são repassados os dados do experimento a tela, preenchendo as informações dos campos com seus respectivos valores, como ilustrado pela Figura 17.

Figura 17 - Página detalhes do experimento.

Por meio da página detalhes do experimento é possível modificar os valores referentes ao registro no banco, contudo algumas informações não podem ser modificadas após o cadastro, por exemplo o “ID” e a “dataCadastro”. As validações dos campos seguem o mesmo padrão utilizado na tela de cadastro, descrita nesta seção.

A última parte destacada pela Figura 16 (D) se refere a opção “desativar/ativar registro”, recurso destinado a modificação do status da coluna “ativo” para o determinado experimento. Quando o usuário administrador clica em uma das opções de modificação, seja ela a de ativar um registro ou desativar é feita uma requisição ao método *modifyStatus* localizado na *controller Project*, responsável por inverter o valor booleano armazenado na coluna “ativo”.

ID	Título	Descrição	Acesso	Data de Cadastro	Ativo	Ações
1	TESTANDO CADASTRO ARDUINO	Mussum Ipsum, cacilds vidis litro abertis. A ordem dos tratores não altera o pão duris. Si num tem...	Publico	2021-10-18 22:47:06	Sim	[Desativar] [Ativar]
2	EXPERIMENTO 2	TESTANDO NOVO EXPERIMENTO COM TEXTO	Autenticado	2021-11-23 23:06:14	Não	[Desativar] [Ativar]
3	Testando novo experimento TCC	Testando novo experimento TCC texto descritivo armazenado. Testando novo experimento TCC texto desc...	Publico	2021-11-23 23:06:12	Sim	[Desativar] [Ativar]

Figura 18 - Página gerenciar experimentos, status diversos.

Sendo assim, o botão se comporta conforme o registro no banco de dados, quando o status da coluna se encontra ativo a opção para o usuário é de desativar o registro, através do

botão destacado pela cor vermelha, caso o registro se encontre inativo a opção habilitada é reativar o registro (cor verde), ambos ilustrados pela Figura 18.

O terceiro *card* listado na *dashboard* inicial (Figura 12), está relacionado a tela Gerenciar Usuários, representada na Figura 19. O administrador poderá acessá-la para filtrar os registros cadastrados (Parte A), alterar um usuário já existente (Parte B), como também desativar um registro (Parte C).

The screenshot shows the 'Gerenciar Usuário' page. At the top, there is a search bar with the text 'Buscar Usuários' and a blue 'Filtrar' button. Below the search bar is a table with the following columns: ID, Nome, Email, Tipo de usuário, Data de Cadastro, Ativo, and Ações. The table contains three rows of user data. The 'Ações' column for each row contains three buttons: a yellow 'Editar' button, a red 'Excluir' button, and a green 'Ativar' button. The 'Filtrar' button is labeled 'A', the 'Editar' button for the first user is labeled 'B', and the 'Ativar' button for the third user is labeled 'C'.

ID	Nome	Email	Tipo de usuário	Data de Cadastro	Ativo	Ações
1	Antonio Carlos	a@gmail.com	Administrador	2021-11-28 00:57:20	Sim	Editar Excluir Ativar
2	Persona um	persona@gmail.com	Público	2021-11-28 12:32:11	Sim	Editar Excluir Ativar
3	Persona dois	persona2@gmail.com	Público	2021-11-28 12:32:17	Não	Editar Excluir Ativar

Figura 19 - Página gerenciar usuários.

A Figura 19 ilustra as três ações principais da tela de gerenciamento, elas estão indicadas com as letras de A à C. A parte marcada com a letra A corresponde ao filtro de pesquisa, responsável por realizar a busca dos usuários com base no texto fornecido pelo administrador. Ao clicar no botão filtrar é disparado uma requisição ao método *list* localizado na *controller User*, retornando a lista com os registros encontrados para página de gerência de usuários.

Já a ação de editar um registro existente está indicada pela letra B, ao clicar no botão editar é disparado uma requisição com o parâmetro ID do usuário, essa requisição está direcionada ao método *Details* localizado na *controller Project*, por meio desse método é repassado os dados do experimento a tela, preenchendo as informações dos campos com seus respectivos valores, como ilustrado pela Figura 20.

Gerenciador de Experimentos Olá, Antonio ▾

Dashboard
Gerenciar Experimentos

Detalhes do Usuário - ID: 1

* Nome:
Antonio Carlos

* Email:
a@gmail.com

Tipo de Usuário:
 Administrador Público
 Ativo

Salvar

Figura 20 - Página detalhes do usuário.

Por meio da página detalhes do usuário é possível modificar os valores referentes ao registro no banco, contudo algumas informações não podem ser modificadas após o cadastro, por exemplo o “ID”, “Senha” e a “dataCadastro”. A validação dos campos se dá no momento em que o formulário é enviado da aplicação (front-end) para o método *save* localizado na *controller User (back-end)* através do botão Salvar.

Ao clicar no botão Salvar é feita uma requisição para a *controller User*, conforme citado anteriormente, caso as informações fornecidas pelo administrador precisem de alguma correção é retornada a mensagem de erro e apresentada uma validação visual para cada campo em ordem de correção, como ilustrado na Figura 21.

Gerenciador de Experimentos Olá, Antonio ▾

Dashboard
Gerenciar Experimentos

Detalhes do Usuário - ID: 1

* Nome:
Antonio Carlos Pereira Mota Milhomens Junior

* Email:
Email

Por favor, informe um Email para o usuário.

Tipo de Usuário:
 Administrador Público
 Ativo

Salvar

Figura 21 - Formulário de cadastro de usuário, erro de preenchimento.

A última parte destacada pela Figura 19 (C) se refere a opção “desativar/ativar registro”, recurso destinado a modificação do status da coluna “ativo” para o determinado usuário. Quando o administrador clica em uma das opções de modificação, seja ela a de ativar um

registro ou desativar é feita uma requisição ao método *modifyStatus* localizado na *controller User* responsável por inverter o valor booleano armazenado na coluna “ativo”.

Sendo assim, o botão se comporta conforme o registro no banco de dados, quando o status da coluna se encontra ativo a opção para o usuário é de desativar o registro, através do botão destacado pela cor vermelha, caso o registro se encontre inativo a opção habilitada é reativar o registro (cor verde), ambos ilustrados pela Figura 19.

4.1.3.3 Área Pública

No decorrer desta seção serão demonstrados os fluxos e telas da área pública na plataforma. É nessa parte pública que os usuários terão acesso a lista de experimentos para manipulação, conforme ilustrado pela Figura 22.



Figura 22 - Página lista de experimentos, área pública.

A Figura 22 ilustra a lista de experimentos na área pública da plataforma, nela são apresentadas as informações essenciais para o entendimento de usabilidade aos usuários do sistema. Dentre as informações apresentadas se encontram: título do experimento, data de cadastro, descrição, tipo de acesso, tipo de sessão e o tempo da sessão.

Durante o acesso a tela inicial é feita uma requisição ao método *listPublic* localizado na *controller Project (back-end)*, ele é responsável por retornar a lista de objetos ativos registrados no banco de dados. Os experimentos classificados como público por meio da variável tipo de acesso são disponibilizados a qualquer usuário que acesse a plataforma, já os classificados como privados só serão acessados por usuários registrados.

Outra classificação feita aos experimentos é referente ao tipo de sessão, a sessão padrão para uso de cada experimento foi definida com o tempo de sete minutos, para os experimentos registrados como Sessão Renovável o usuário poderá (caso necessite) renovar a sessão por mais 7 minutos, porém, os registros classificados como sessão não renovável terão o tempo de uso restrito padrão.

4.1.4 Ambiente de Execução

Os próximos parágrafos são destinados à explicação do processo de implementação do ambiente de execução, ele é responsável pela utilização e manipulação dos experimentos da plataforma. O ambiente de execução foi desenvolvido e incorporado a plataforma baseado no projeto de iniciação científica também desenvolvido pelo autor Milhomens Junior (2017), com o tema “DESENVOLVIMENTO DE UM SOFTWARE WEB PARA EXPERIMENTAÇÃO REMOTA COM ARDUINO”.

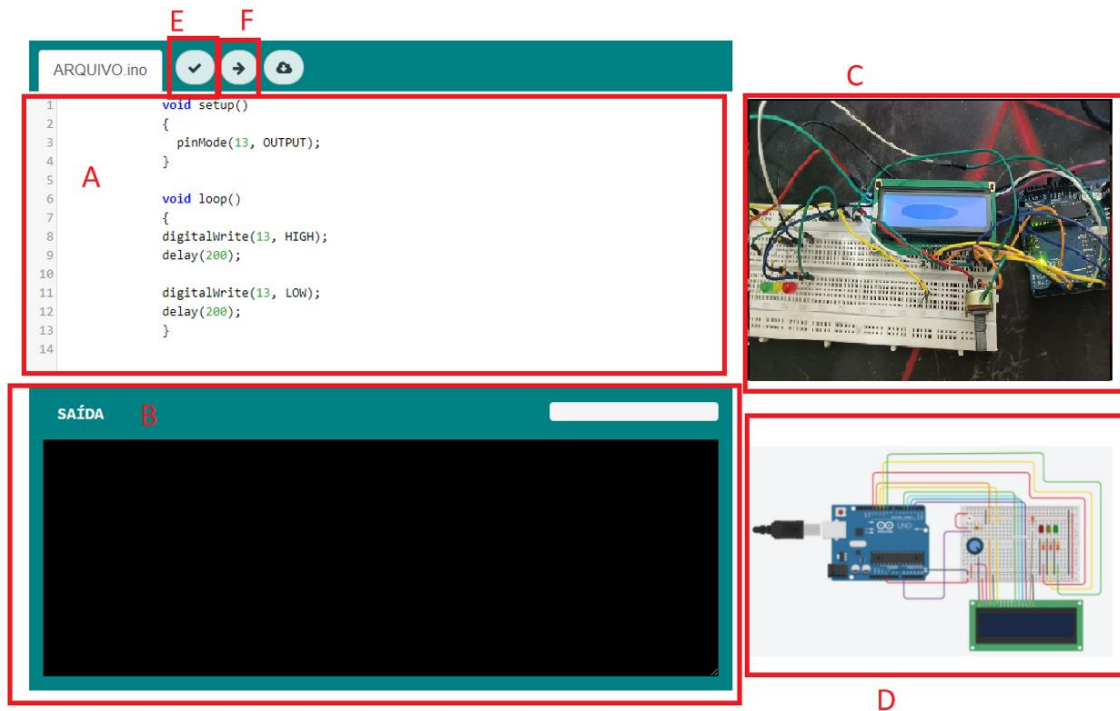


Figura 23 - Ambiente de Execução.

O ambiente de execução apresentado pela Figura 23 possui dois campos de texto: um responsável pela entrada de código fornecida pelo usuário da plataforma (Figura 23-A), de modo que o campo de texto será o local onde o usuário irá realizar o desenvolvimento do código para o Arduino. O campo de texto citado utiliza da biblioteca *codemirror* para identificar a sintaxe do código produzido, além de adicionar os identificadores de linha para facilitar o reconhecimento de erros durante a compilação.

O segundo campo de texto (Figura 23-B) é responsável por apresentar os resultados obtidos através das rotas *upload* e *verifica*, ambas disponibilizadas pelo *back-end* de experimentação remota. Além disso, a tela inclui uma área para apresentação em tempo real das imagens obtidas pela *WebCam* (Figura 23-C), ela está apontada para o experimento físico e acoplada no servidor matriz. Por fim, a Figura 23-D apresenta um diagrama esquemático do circuito utilizado no experimento.

O ambiente de execução comunica-se com o *back-end* de experimentação remota utilizando requisições assíncronas, a partir da interação do usuário com os botões “verificar” (Figura 23-E) e “carregar” (Figura 23-F). O botão “verificar” faz com que aconteça a validação do código Arduino, realizando a validação da sintaxe e semântica do código, já o botão “carregar” realiza a conversão do código produzido para binário e o upload para o microcontrolador Arduino.

5 CONSIDERAÇÕES FINAIS

Este trabalho de pesquisa teve por objetivo desenvolver uma plataforma para experimentação remota com Arduino, que contemplasse as funcionalidades de gerenciamento e execução dos experimentos. Foram definidos os objetivos específicos com o intuito de modelar e analisar o contexto aplicado à plataforma, de modo que fosse possível definir e contextualizar os principais propósitos da ferramenta.

As metas propostas no projeto de pesquisa foram realizadas, possibilitando responder ao problema de pesquisa proposto “Como desenvolver uma plataforma de experimentação remota para Arduino?”. A resposta para o problema citado é a plataforma de experimentação remota produzida, todos os processos para o desenvolvimento da plataforma que atendessem os requisitos propostos foram descritos e apresentados neste documento.

O objetivo da plataforma para experimentação remota com Arduino é de alcançar usuários que busquem aprender e utilizar a tecnologia para desenvolver seus próprios protótipos programaticamente, proporcionando uma interação com experimentos reais em Arduino independentemente do seu nível econômico, de modo que, através da internet o aluno (ou aprendiz) terá experiências semelhantes aos laboratórios locais.

É importante ressaltar que os resultados desenvolvidos e apresentados foram embasados no que é disponibilizado em diversas fontes relacionadas aos objetivos do projeto, todos eles citados na parte do referencial teórico. Foram utilizadas diversas técnicas, conceitos, ferramentas e tecnologias, como por exemplo Python, PHP, PlatformIO, API Rest, Xampp e Visual Studio Code para produzir uma plataforma simples e agradável para quem irá utilizá-la.

Para o futuro, acredita-se nas seguintes possibilidades propostas: um serviço de agendamento que possibilita marcar uma data para utilização de um ou mais experimentos, tendo em vista que professores e alunos possam agendar os serviços de experimentação remota para aplicar estudos em grupo ou individualmente. Outro serviço que pode ser incorporado à plataforma são conteúdos de estudo e entendimento sobre Arduino, facilitando o aprendizado e utilização dos experimentos remotamente, como exemplos, aulas, livros, artigos etc.

REFERÊNCIAS

ALMEIDA, Maria Elizabeth Bianconcini de. **Educação a distância na internet: abordagens e contribuições dos ambientes digitais de aprendizagem.** 2003. Disponível em:

<https://www.scielo.br/j/ep/a/dSsTzcBQV95VGCf6GJbtpLy/?lang=pt&format=pdf>. Acesso em: 10 jun. 2021.

ALVES, João Bosco da Mota et al. **MICROSCÓPIO REMOTO: experimentação remota aplicada ao ensino de ciências.** In: SILVA, Juarez Bento da; BILESSIMO, Simone Meister Sommer; LUNARDI, Giovani Mendonça. *Tecnologias da Informação e Comunicação: pesquisa e inovação.* Araranguá: Universidade Federal de Santa Catarina, 2016. p. 1-179.

Disponível em: <https://posticsenasp.paginas.ufsc.br/files/2016/06/Ebook-Tecnologias-da-Informa%C3%A7%C3%A3o-e-Comunica%C3%A7%C3%A3o-Pesquisa-e-Inova%C3%A7%C3%A3o.pdf>. Acesso em: 12 jun. 2021.

ANDERSON, Chris. **A nova revolução industrial: Makers.** Elsevier Brasil, 2012.

ARDUINO. **What is Arduino?** Disponível em:

<<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 23 abr. 2021.

BARROS, Maria das Graças; CARVALHO, Ana Beatriz Gomes. **As concepções de interatividade nos ambientes virtuais de aprendizagem.** In: SOUSA, Robson Pequeno de; MOITA, Filomena M. C. da S. C.; CARVALHO, Ana Beatriz Gomes; ORGS. *Tecnologias digitais na educação.* Campina Grande: Eduepb, 2011. p. 209-232. Disponível em:

<https://static.scielo.org/scielobooks/6pdyn/pdf/sousa-9788578791247.pdf>. Acesso em: 20 jun. 2021.

BOTTENTUIT JUNIOR, João Batista; COUTINHO, Clara Pereira. **PROJECTO E DESENVOLVIMENTO DE UM LABORATÓRIO VIRTUAL NA PLATAFORMA MOODLE.** 2007. Disponível em:

<http://repositorium.sdum.uminho.pt/bitstream/1822/6504/1/015.pdf>. Acesso em: 09 jun. 2021.

BRINSON, J. R. **Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research.** *Computers & Education*, v. 87, p. 218 – 237, 2015. ISSN 0360-1315.

CARLOS, Lucas Mello et al. **Um olhar sob a interatividade em laboratórios online.** 2018.

Disponível em: <http://tecedu.pro.br/wp-content/uploads/2018/06/Art9-vol.24-Edi%C3%A7%C3%A3o-Tem%C3%A1tica-VII-Junho-2018.pdf>. Acesso em: 23 abr. 2021.

CARLOS, Lucas Mello. **PROPOSTA DE ARQUITETURA PARA LABORATÓRIOS MÓVEIS BASEADA NO PARADIGMA DE DISPOSITIVOS INTELIGENTES.** 2017.

Disponível em:

<https://repositorio.ufsc.br/bitstream/handle/123456789/181876/final.pdf?sequence=1&isAlloved=y>. Acesso em: 10 jun. 2021.

CHITUNGO, Herculano Henriques Chingui. **O USO DE LABORATÓRIOS REMOTOS NO ENSINO DE FÍSICA NA EDUCAÇÃO BÁSICA: ESTUDO DE CASO EM**

ESCOLA DA REDE PÚBLICA. 2018. Disponível em: <https://rexlab.ufsc.br/wp-content/uploads/2018/05/Herculano-Disserta%C3%A7%C3%A3o.pdf>. Acesso em: 07 jun. 2021.

CODE, Visual Studio. **Documentação**. 2021. Disponível em: <https://code.visualstudio.com/>. Acesso em: 20 jun. 2021.

DOCS, Mdn Web. **JavaScript Tutoriais**. 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 12 jun. 2021.

ENAP. **Educação a Distância em organizações públicas**: mesa redonda de pesquisa-ação. Brasília: Escola Nacional de Administração Pública, 2006. 200 p. Disponível em: https://repositorio.enap.gov.br/bitstream/1/376/1/Livro_EAD.pdf. Acesso em: 22 jun. 2021.

FOUNDATION, Openjs; JQUERY. **“Página inicial”**. 2021. Disponível em: <https://jquery.com/>. Acesso em: 22 jun. 2021.

GOLINELLI, Marcos Henrique de Moraes et al. **ARQUITETURA DE DISPOSITIVOS INTELIGENTES PARA LABORATÓRIOS REMOTOS: UMA REVISÃO SISTEMÁTICA DA LITERATURA**. 2018. Disponível em: <https://www.fatecourinhos.edu.br/retec/index.php/retec/article/view/306>. Acesso em: 23 abr. 2021.

LIMA, Joao Paulo Cardoso de. **DESENVOLVIMENTO DE SERVIDORES PARA LABORATORIOS REMOTOS BASEADO NO PARADIGMA DE DISPOSITIVOS INTELIGENTES**. 2016. Disponível em: <https://repositorio.ufsc.br/handle/123456789/172515>. Acesso em: 20 jun. 2021.

LITTO, Fredric M.; FORMIGA, Marcos. **Educação a Distância. O Estado da Arte**. São Paulo: Pearson Education, 2009. 480 p. Disponível em: http://www.abed.org.br/arquivos/Estado_da_Arte_1.pdf. Acesso em: 20 jun. 2021.

LOPES, Marina. **Laboratórios virtuais levam experimentação a Moocs**. 2014. Disponível em: <http://porvir.org/laboratorios-virtuais-levam-experimentacao-moocs/>. Acesso em: 23 abr. 2021.

LOTTHAMMER, Karen Schmidt. **CRIAÇÃO DE UM CURSO ONLINE ABERTO E MASSIVO PARA CAPACITAÇÃO DE PROFESSORES EM RELAÇÃO À INTEGRAÇÃO DE TECNOLOGIA NA EDUCAÇÃO**. 2017. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/177456/TCC_vers%C3%A3o_final.pdf?sequence=1&isAllowed=y. Acesso em: 14 jul. 2021.

LOWE, D.; MACHET, T.; KOSTULSKI, T. **Uts remote labs, labshare, and the sahara architecture. Using Remote Labs in Education: Two Little Ducks in Remote Experimentation**, Universidad de Deusto, v. 8, p. 403, 2012.

Albu M. M., Holbert K. E., Heydt G. T., S. Grigorescu Dan, Truc V., **Embedding remote experimentation in power engineering education**, IEEE Transactions on Power Systems, vol. 19, pp. 11-132, 2004.

MANCHIKANTI, P; KUMAR, B. R.; SINGH, V. **Role of Virtual Biology Laboratories in Online and Remote Learning**. 2016 IEEE Eighth International Conference on Technology for Education (t4e), [s.l.], p. 136-139, dez. 2016.

MARCELINO, Roderval et al. **Virtual 3D Worlds and Remote Experimentation: a Methodology Proposal Applied to Engineering Students**. In: UNIVERSITY OF DEUSTO (Espanha). Using Remote Labs in Education: Two Little Ducks in Remote Experimentation. Bilbao. 2011. Cap. 17. p. 349-374.

MILHOMENS JUNIOR, Antonio Carlos Pereira Mota. **DESENVOLVIMENTO DE UM BACK-END PARA UM SOFTWARE DE EXPERIMENTAÇÃO REMOTA COM ARDUINO**. 2018. 30 f. Estágio Supervisionado - Curso de Ciência da Computação, Centro Universitário Luterano de Palmas (Ceulp/Ulbra), Palmas, 2018.

MILHOMENS JUNIOR, Antonio Carlos Pereira Mota. **DESENVOLVIMENTO DE UM SOFTWARE WEB PARA EXPERIMENTAÇÃO REMOTA COM ARDUINO**. 2017. Projeto de iniciação científica - Curso de Ciência da Computação, Centro Universitário Luterano de Palmas (Ceulp/Ulbra), Palmas, 2017.

MÜLLER, D.; ERBE, H.-H. **Collaborative remote laboratories in engineering education: Challenges and visions. Advances on remote laboratories and e-learning experiences**, Bilbao, p. 35– 59, 2007.

NICOLETE, Priscila Cadorin. **INTEGRAÇÃO DE TECNOLOGIA NA EDUCAÇÃO: GRUPO DE TRABALHO EM EXPERIMENTAÇÃO REMOTA MÓVEL (GT-MRE) UM ESTUDO DE CASO**. 2016. Disponível em: <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/171704/343054.pdf?sequence=1&isAllowed=y>. Acesso em: 20 jun. 2021.

ORDUNA, P. et al. **Labsland: A sharing economy platform to promote educational remote laboratories maintainability, sustainability and adoption**. 2016. IEEE Frontiers in Education Conference (FIE). 2016. p. 1–6.

ORDUNA, P. et al. **Towards a microrlms approach for shared development of remote laboratories**. *Computers in Human Behavior*, v. 30, p. 389 – 395, 2014. ISSN 0747-5632. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0747563213001416>.

PHP. **O que é o PHP?** Disponível em: <https://www.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 20 jun. 2021.

ROCHADEL, Willian. **Rexmobile: Integrando Experimentação Remota Na Educação Básica**. 2013. 140 f. TCC (Graduação) - Curso de Tecnologias da Informação e Comunicação, Universidade Federal de Santa Catarina, Araranguá, SC, 2013.

RODRIGUEZ-ANDINA, J. J.; GOMES, L.; BOGOSYAN, S. **Current Trends in Industrial Electronics Education**. IEEE Transactions on Industrial Electronics, [s.l.], v. 57, n. 10, out. 2010.

SANTANA, I. et al. **Remote laboratories for education and research purposes in automatic control systems**. Industrial Informatics, IEEE Transactions on, IEEE, v. 9, n. 1, 2013.

SANTOS, Aline Coêlho dos. **INTEGRAÇÃO DE TECNOLOGIA NA EDUCAÇÃO BÁSICA: UM ESTUDO DE CASO NAS AULAS DE BIOLOGIA UTILIZANDO LABORATÓRIOS ON-LINE**. 2018. Disponível em: <https://rexlab.ufsc.br/wp-content/uploads/2018/05/aline-dissertacao.pdf>. Acesso em: 07 jun. 2021.

SCHNEIDER, Andréia Panchera; TULHA, Carinna Nunes. **PERCEPÇÃO DA USABILIDADE DO MOODLE COMO FERRAMENTA PARA O ENSINO DE CIÊNCIAS BASEADO EM INVESTIGAÇÃO**. 2016. Disponível em: https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/171390/TCC_VF.pdf?sequence=1&isAllowed=y. Acesso em: 18 jun. 2021.

SILVA, Juarez Bento da et al. **Integração de Tecnologias na Educação: práticas inovadoras na educação básica**. 4. ed. Araranguá: Hardtech, 2020. 196 p. Disponível em: https://rexlab.ufsc.br/wp-content/uploads/2020/05/Volume_4_Final.pdf. Acesso em: 18 jun. 2021.

SIMÃO, José Pedro Schardosim. **Modelo para Registro de Dados de Experiência de Aprendizagem em Laboratórios Remotos**. 2018. Disponível em: https://www.researchgate.net/publication/328393512_Modelo_para_Registro_de_Dados_de_Experiencia_de_Aprendizagem_em_Laboratorios_Remotos. Acesso em: 15 jul. 2021.

SIMÃO, José Pedro Schardosim. **RELLE: sistema de gerenciamento de experimentos remotos. SISTEMA DE GERENCIAMENTO DE EXPERIMENTOS REMOTOS**. 2015. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/157371/final.pdf?sequence=1&isAllowed=y>. Acesso em: 20 jun. 2021

SOUSA, Robson Pequeno et al., orgs. **Tecnologias digitais na educação [online]**. Campina Grande: EDUEPB, 2011. 276 p. ISBN 978-85-7879-124-7. Available from SciELO Books.

W3SCHOOLS. **Tutorial CSS**. 2021. Disponível em: https://www.w3schools.com/css/css_intro.asp. Acesso em: 12 jul. 2021.

W3SCHOOLS. **Tutorial HTML**. 2021. Disponível em: https://www.w3schools.com/html/html_intro.asp. Acesso em: 11 jul. 2021.

World Wide Web Consortium. **CSS “Página inicial do Cascading Style Sheets”**. 2021. Disponível em: <https://www.w3.org/Style/CSS/>. Acesso em: 11 jul. 2021.

ZUTIN, Danilo Garbi et al. **Lab2go — A repository to locate educational online laboratories**. IEEE Educon 2010 Conference, [s.l.], abr. 2010. IEEE. Disponível em: <https://ieeexplore.ieee.org/document/5492412>. Acesso: 20 jun. 2021.