



UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LARISSA MOREIRA DE LISBOA

**LARA DOMÓTICA - PROTÓTIPO DE LABORATÓRIO VIRTUAL  
DE DOMÓTICA**

VITÓRIA DA CONQUISTA - BA

2023



UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LARISSA MOREIRA DE LISBOA

**LARA DOMÓTICA - PROTÓTIPO DE LABORATÓRIO VIRTUAL  
DE DOMÓTICA**

Trabalho de Conclusão de Curso apresentado à Universidade Estadual do Sudoeste da Bahia-UESB, para a disciplina de Trabalho Supervisionado II, como requisito para aprovação na disciplina, ministrada pela Prof. Dr. Maísa Soares dos Santos Lopes, do curso de Ciências da Computação, no PL 2022.2.

Prof. Orientadora: Maísa Soares dos Santos Lopes

VITÓRIA DA CONQUISTA - BA

2023

## RESUMO

Laboratórios virtuais podem ser acessados de qualquer lugar com o uso de um computador e acesso a internet. Essas ferramentas educacionais contém experimentos que são capazes de tornar a aprendizagem lúdica e incentivar o aluno a continuar utilizando a ferramenta. A adição da colaboratividade nessas ferramentas educacionais aumenta a chances de aprendizado, pois os alunos podem trocar experiências e conhecimentos entre si. A experiência proposta neste trabalho é uma casa inteligente (também conhecida como Domótica e automatização residencial) definida como um ambiente residencial que possui um sistema gerenciador de dispositivos eletrônicos. A automatização residencial tem o objetivo de prover maior conforto, segurança, praticidade e entretenimento para seus moradores. O objetivo principal deste trabalho é a criação de um protótipo de laboratório virtual contendo uma simulação de Domótica para ensino de programação. Este sistema se faz necessário para auxiliar tanto professores como alunos no processo de ensino e aprendizagem de programação, já que esta matéria é considerada difícil e isso constantemente desencoraja os alunos iniciantes na área. A interface do sistema foi desenvolvida priorizando a experiência do usuário. Para isso foram feitas avaliações de usabilidade e seus resultados foram analisados e utilizados para aprimorar a ferramenta desenvolvida.

Palavras-chave: Laboratórios virtuais, Domótica, ferramentas educacionais, programação, usabilidade.

## **ABSTRACT**

Virtual labs can be accessed from anywhere with the use of a computer and internet access. These educational tools contain experiments that are able to make learning fun and encourage the student to continue using the tool. The addition of collaboration in these educational tools increases the chances of learning, as students can exchange experiences and knowledge with each other. The experience proposed in this work is a smart home (also known as Domotics and home automation) defined as a residential environment that has an electronic device management system. Home automation aims to provide greater comfort, safety, practicality and entertainment for its residents. The main objective of this work is the creation of a virtual laboratory prototype containing a Home Automation simulation for teaching programming. This system is necessary to help both teachers and students in the process of teaching and learning programming, since this subject is considered difficult and this constantly discourages beginner students in the area. The system interface was developed prioritizing the user experience. For this, usability evaluations were carried out and their results were analyzed and used to improve the developed tool.

Keywords: Virtual laboratories, Domotics, educational tools, programming, usability.

## **AGRADECIMENTOS**

Agradeço a minha irmã, meu pai e minha mãe pela paciência e ajuda durante o desenvolvimento deste trabalho.

Agradeço aos meus amigos que sempre me incentivaram durante todo curso.

Agradeço aos avaliadores que fizeram parte e foram essenciais para a conclusão deste trabalho.

Agradeço a minha professora e orientadora Máisa Soares por todos os direcionamentos e auxílios que fizeram chegar a conclusão deste curso.

Agradeço os integrantes do projeto LARA por suas contribuições que foram de grande ajuda ao longo deste trabalho.

## LISTA DE FIGURAS

Figura 01 - Esquema demonstrativo do uso das atividades do DSR neste trabalho

Figura 02 - Primeiro esboço da simulação de uma casa inteligente

Figura 03 - a) Sensor de movimento passivo, b) Sensor de movimento ativo

Figura 04 - a) Detector de fumaça, b) Detector de fumaça com sensor de temperatura

Figura 05 - a) Sprinkler de prevenção de incêndio, b) Sprinkler de jardim

Figura 06 - Painel de controle da ferramenta

Figura 07 - Imagem código no Open Roberta utilizando programação em blocos

Figura 08 - Imagem de um circuito e robô virtual disponibilizado no Open Roberta

Figura 09 - Cadastro de questão no LabPy

Figura 10 - Área professor

Figura 11 - Diagrama de Casos de uso do usuário aluno

Figura 12 - Diagrama de Casos de uso do usuário professor

Figura 13 - Diagrama de Casos de uso do usuário administrador

Figura 14 - Diagrama de fluxo de interação do Usuário aluno

Figura 15 - Diagrama de fluxo de interação do Usuário professor

Figura 16 - Tela inicial do sistema web

Figura 17 - Tela de programação e simulação

Figura 18 - Tela “Fale conosco”

Figura 19 - Tela “Central de ajuda”

Figura 20 - Tela de login

Figura 21 - Tela de cadastro

Figura 22 - Tela “Área professor” na aba “Nova tarefa”

Figura 23 - Tela “Área professor” na aba “Nova tarefa” criando uma atividade em grupo

Figura 24 - Tela “Área professor” na aba “Histórico”

Figura 25 - Tela “Área professor” na aba “Histórico” com a área de grupo aberta

Figura 26 - Tela “Área professor” na aba “Histórico” com a área de edição aberta

Figura 27 - Seletor de atividades criadas

Figura 28 - Tela “Área professor” na aba “Alunos”

Figura 29 - Tela “Área professor” na aba “Alunos” ao clicar no nome de um aluno

Figura 30 - Tela “Área professor” na aba “Nova turma”

Figura 31 - Tela “Área do aluno”

Figura 32 - Menu superior antes das modificações

Figura 33 - Menu superior depois das modificações

Figura 34 - Console adicionado a área de programação

Figura 35 - Botão de edição do nome da turma

Figura 36 - Edição do nome da turma

Figura 37 - Mensagens informando os campos obrigatórios

Figura 38 - Pesquisa de atividades

## Sumário

<b>1.INTRODUÇÃO.....</b>	<b>10</b>
1.1. Justificativa.....	12
1.2. Objetivos.....	13
1.2.1 Objetivos gerais.....	13
1.2.2 Objetivos específicos.....	13
1.3. Metodologia.....	14
<b>2.REVISÃO BIBLIOGRÁFICA.....</b>	<b>17</b>
2.1.Laboratório virtual.....	17
2.2. Domótica.....	18
2.2.1.Dispositivos.....	19
2.2.1.1. Sensores de movimento.....	20
2.2.1.2. Sensor de fumaça e Sensor de temperatura.....	21
2.2.1.3. Sprinkler.....	22
2.2.2. Maquete residencial utilizando Arduino.....	22
2.3. Simulação.....	23
2.4. Ensino de programação.....	25
2.4.1. Open Roberta Lab.....	26
2.4.2. LabPy.....	29
<b>3. LARA DOMÓTICA.....</b>	<b>32</b>
3.1. Proposta.....	32
3.2. Requisitos.....	33
3.2.1.Requisitos funcionais.....	33
3.2.2.Requisitos não funcionais.....	39
3.2.2.1. Usabilidade.....	39
3.2.2.2.Colaborativo.....	40
3.2.2.3.Educacional.....	41
3.2.2.4. Disponibilidade.....	42
3.2.2.5. Ubiquidade (Desempenho).....	43
3.2.2.6. Documentação.....	43
3.2.2.7. Atratividade.....	44
3.2.3.Regras de negócio.....	44
3.2.4.Diagrama de fluxo de interação.....	45
3.2.5.Telas do sistema.....	47
<b>4.AVALIAÇÃO.....</b>	<b>62</b>
4.1.Heurísticas de Nielsen.....	62
4.1.1. Grau de severidade.....	64
4.1.2. Avaliadores.....	64
4.2. Resultado da avaliação.....	65
4.3. Conclusões da avaliação.....	69
4.4. Mudanças realizadas.....	71
<b>5.CONCLUSÃO.....</b>	<b>75</b>



## 1.INTRODUÇÃO

Laboratórios são ambientes educacionais fundamentais para proporcionar aos alunos experiências com a aplicação de conceitos teóricos. Um fato importante sobre laboratórios é que tendem a criar nos alunos uma maior sedimentação do aprendizado, já que a prática é o foco nesse ambiente. Juntamente com a possibilidade de um ambiente colaborativo, a prática se soma com a troca de ideias e com isso o aluno passa a internalizar os conhecimentos de maneira mais eficiente. A partir da união da teoria com a prática que se desenvolvem habilidades de resolução de problemas e entendimento pleno do assunto:

A teoria é a forma como o conhecimento se apresenta articulando-se sistematicamente em graus e especificidades, disposto a explicar ou ilustrar ações práticas; enquanto a prática é a constituição da teoria, formulada em ações concretas, podendo ser modificada e modificar as teorias. Considerando esse contexto, fica evidente que ambas se entrelaçam e que a desvinculação destas fragiliza o processo de aprendizagem do sujeito. (PACHECO;BARBOSA;FERNANDES,2017,p.334)

Os laboratórios reais tem sua importância nos ambientes de ensino, porém os custos relacionados com equipamentos, manutenção, locação e pessoal de apoio dificultam a adoção desse tipo de ferramenta educativa. Os métodos de ensino evoluíram conforme a tecnologia avançou. Com a modernização das formas de adquirir conhecimento, surgiram os laboratórios remotos e virtuais, modalidades de aprendizado que utilizam ferramentas computacionais dispensando a realização de atividades presenciais. O acontecimento da pandemia do coronavírus causou o fechamento de escolas e universidades e com isso a relevância de ferramentas de aprendizado a distância na continuidade do ensino se tornou ainda maior. Kenski (2003) discorre sobre a importância de se adaptar a mudança tecnológica na qual o homem vive:

A evolução tecnológica não se restringe apenas aos novos usos de determinados equipamentos e produtos. Ela altera comportamentos.[...] O homem transita culturalmente mediado pelas tecnologias que lhe são contemporâneas. Elas transformam suas maneiras de pensar, sentir, agir. Mudam também suas formas de se comunicar e de adquirir conhecimentos.

As vantagens desses métodos de aprendizagem são a possibilidade de acesso de qualquer local (com conexão de internet e um computador) por diversas

pessoas simultaneamente. Além de serem grandes ferramentas de estudo, também tem a utilidade de dispensar que o aluno se locomova, sendo assim mais inclusivo para pessoas com deficiência e pessoas que moram em locais de difícil acesso.

Experimentos que utilizam laboratórios virtuais como forma de interação mostram aos alunos como seria o resultado do seu trabalho de uma forma segura e sem riscos financeiros, ambientais e de saúde. Morozov et al. (2004), em seu trabalho sobre laboratório virtual de química, diz como esse tipo de experimentos virtuais permitem que estudantes se familiarizem com as técnicas e procedimentos laboratoriais antes de conduzir experimentos reais. Esse estudo relata: “Deve-se enfatizar que os experimentos virtuais de química são seguros mesmo para iniciantes. Os alunos podem realizar experimentos que podem ser perigosos e/ou caros em um laboratório real.” (tradução nossa). Em um ambiente virtual, os experimentos desenvolvidos não afetam o meio real, sendo assim uma boa opção para aprendizados que envolvam riscos, como explosões e choques, sem a possibilidade de destruir equipamentos caros ou ferir pessoas nesse processo.

Criar um laboratório virtual pode se tornar um desafio. O processo depende de um profissional que idealiza a ferramenta e desenvolve o código, a modelagem da simulação e o fluxo de navegação pela ferramenta. Por vezes, é necessário o uso de máquinas virtuais, servidores, bancos de dados e software de simulação interativa. Os objetos utilizados no laboratório devem passar uma interatividade para o usuário, capazes de utilizar equipamentos virtuais de maneira similar ao mundo real. Ao utilizar o sistema, também é necessário que este ofereça facilidade no uso, para que o usuário consiga manter o foco no aprendizado. Devem ser considerados tempos de resposta do sistema para garantir que não haja atrasos significativos ao utilizá-la. No caso de laboratórios virtuais, validar a simulação é fundamental para o seu sucesso. Na etapa final os testes de validação são realizados para garantir que o sistema tenha atingido os objetivos estabelecidos. É importante manter a atualização e manutenção do sistema com o intuito de corrigir erros e melhorar a ferramenta.

Neste trabalho discutiremos os laboratórios virtuais de programação (ou de robótica), suas propostas e seus objetivos. Além disso, será apresentado um protótipo de um laboratório virtual que contém uma simulação de uma casa inteligente, cujo objetivo é o aprendizado de programação juntamente com os conceitos de Domótica. Será descrita a proposta da ferramenta e os requisitos que

devem compô-la de modo que supra as necessidades educacionais tanto para alunos como para professores.

### 1.1. Justificativa

A pandemia de COVID-19 mostrou que é necessário meios para acesso virtual aos materiais didáticos. Pensando nisso, os laboratórios virtuais são uma boa opção para sanar as barreiras em relação ao aprendizado. Esse tipo de experiência pode ser realizada de qualquer lugar, até mesmo do conforto de sua casa. Laboratórios virtuais são práticos, não demandam um ambiente ou um objeto de estudo físico e mesmo assim são ótimas ferramentas de aprendizagem :

O ambiente virtual pode ser considerado a “sala de aula” da EAD. É um espaço caracterizado pela interação entre professor e aluno através da troca, das conversações, instruções, atividades, “tira-dúvidas”, e até realização de avaliações. Este ambiente permite que os alunos tenham acesso aos conteúdos das distintas disciplinas, aos calendários, aos programas e sobre tudo, é o lugar que, de certa forma, facilita e favorece a aprendizagem através da combinação dos elementos anteriores. (Bianka Pires André,2014)

Esse projeto será desenvolvido seguindo as normas do Laboratório Remoto em Ambiente Virtual de Aprendizagem (LARA), um projeto da Universidade Estadual do Sudoeste da Bahia (UESB), que reúne diversos projetos visando uma aprendizagem lúdica através da prática. A contribuição desse projeto para o LARA está na facilidade de acesso simultâneo ao experimento, já que sendo web e virtual pode ser acessado por vários alunos ao mesmo tempo. Ademais, promove a colaboratividade ao permitir que alunos realizem atividades em grupo e evoluam em conjunto. O LARA propõe a reunião de experimentos diversos contextualizando a aprendizagem de programação. A temática dos experimentos varia criando um ambiente de aprendizado com diversas opções para o aluno.

Para Gomes, Henriques e Mendes (2008), os métodos de ensino de programação normalmente utilizados não são adequados para a maioria dos alunos. O hábito de trabalhar conceitos dinâmicos através de materiais de natureza estática é um dos motivos apontados. Além disso, segundo o artigo, os professores normalmente estão mais preocupados com o ensino da linguagem em seus aspectos sintáticos do que realmente utilizá-la para resolver problemas. Já

considerando os alunos, estes cometem erros ao não dedicarem tempo suficiente para o desenvolvimento das habilidades de programação ou não realizarem esse estudo de maneira correta (planejamento, prática e constância).

Como alternativa para amenizar esse problema, o presente trabalho propõe um ambiente de simulação de Domótica que seja capaz de retornar respostas visuais às codificações desenvolvidas pelos alunos. A ideia é que o aluno se entretenha com as interações disponíveis, empregando mais tempo nos estudos, e que as respostas visuais apresentadas o incentivem na codificação. O uso da Domótica se justifica pois o tema é interessante e um primeiro contato com os seus conceitos seriam de bom proveito para os estudantes da área. A automatização residencial vem crescendo no mercado e a sua aplicação tende a se tornar cada vez mais frequente, aumentando as possibilidades de profissionalização na área.

## 1.2. Objetivos

### 1.2.1 Objetivos gerais

O principal objetivo é desenvolver um protótipo de laboratório virtual conceituado em Domótica que seja capaz de auxiliar o ensino de programação para alunos do curso de computação e exemplificar como funciona uma casa inteligente.

### 1.2.2 Objetivos específicos

- Levantar os requisitos de um laboratório de Domótica voltado ao ensino de programação.
- Modelar a solução proposta.
- Avaliar a interface proposta.
- Melhorar a interface de acordo com o resultado da avaliação.

### 1.3. Metodologia

A metodologia de pesquisa é a área de estudo que define os melhores métodos para se produzir conhecimento. O Design Science Research (DSR) foi a metodologia utilizada no desenvolvimento deste projeto, a qual tem como foco desenvolver artefatos que solucionem ou minimizem um problema da sociedade:

Um artefato, por ser uma invenção humana, em princípio não é objeto de investigação das ciências tradicionais. Contudo, é possível produzir conhecimento científico a partir do uso dos artefatos porque eles são projetados considerando a realidade e, para funcionar, requerem conhecimento sobre o mundo natural e social em que vivemos. Portanto, o desenvolvimento de artefatos pode também ser um meio para se fazer ciência. (Pimentel et al.,2018)

O presente trabalho apresenta um protótipo de um laboratório virtual com simulador de uma casa inteligente para o ensino de programação. A proposta de se utilizar uma simulação de Domótica visa criar maior determinação para os estudos de programação. Os tópicos teóricos foram embasados por materiais como livros, trabalhos de conclusão de curso, artigos científicos e sites. Através desses materiais e das necessidades do LARA, foram definidos os principais requisitos para a criação do produto final.

A validade pragmática, ou capacidade de resolver problemas, é um conceito relacionado a esta metodologia, já que se espera do artefato a entrega das soluções propostas que precisam ser alcançáveis para a população sem grandes custos financeiros. Pensando nisso, é prioritário que seja acessível via internet com o uso de qualquer computador. O foco do DSR é desenvolver sistemas que solucionem ou minimizem um desafio da sociedade:

Já no (meta)paradigma Pragmático (design, projeto), em que a abordagem DSR se enquadra, considera-se que a realidade é projetada pelo homem, modificada, possibilitada sociotecnicamente, e nesse caso, o conhecimento é produzido pelo fazer, pela construção de artefatos contextualmente situados. (Pimentel et al.,2018)

Na metodologia DSR três ciclos são utilizados: O ciclo de engenharia, cujo objetivo é o desenvolvimento do artefato que solucione um problema social, o ciclo de rigor, que foca na obtenção de conhecimento do comportamento humano em um determinado contexto do projeto, e o ciclo de relevância, no qual é avaliado a

necessidade do desenvolvimento do artefato no contexto apresentado. É importante que, ao usar o DSR, sejam feitas pesquisas sobre o tema em desenvolvimento em busca do conhecimento já existente sobre o assunto para reafirmar a relevância do trabalho proposto na contribuição do saber científico.

Os passos seguidos neste trabalho para a aplicação da metodologia foram propostos por Peffers et al.(2007) que divide o processo em 6 atividades. Na Atividade 1 foram identificados os problemas que envolvem a pesquisa e apresentados os motivos que se faz necessário a criação do artefato proposto nesse cenário. O subtópico que melhor exemplifica essa etapa é a “Justificativa” que expõe as razões para que o projeto desenvolvido se torne relevante. Em seguida, considerando o problema identificado, o objetivo da solução é definido pensando na sua viabilidade e possibilidade. Devem ser consideradas as propostas de soluções existentes e sua eficácia. O subtópico “Objetivos” e o tópico “Revisão bibliográfica” demonstra como Atividade 2 foi realizada.

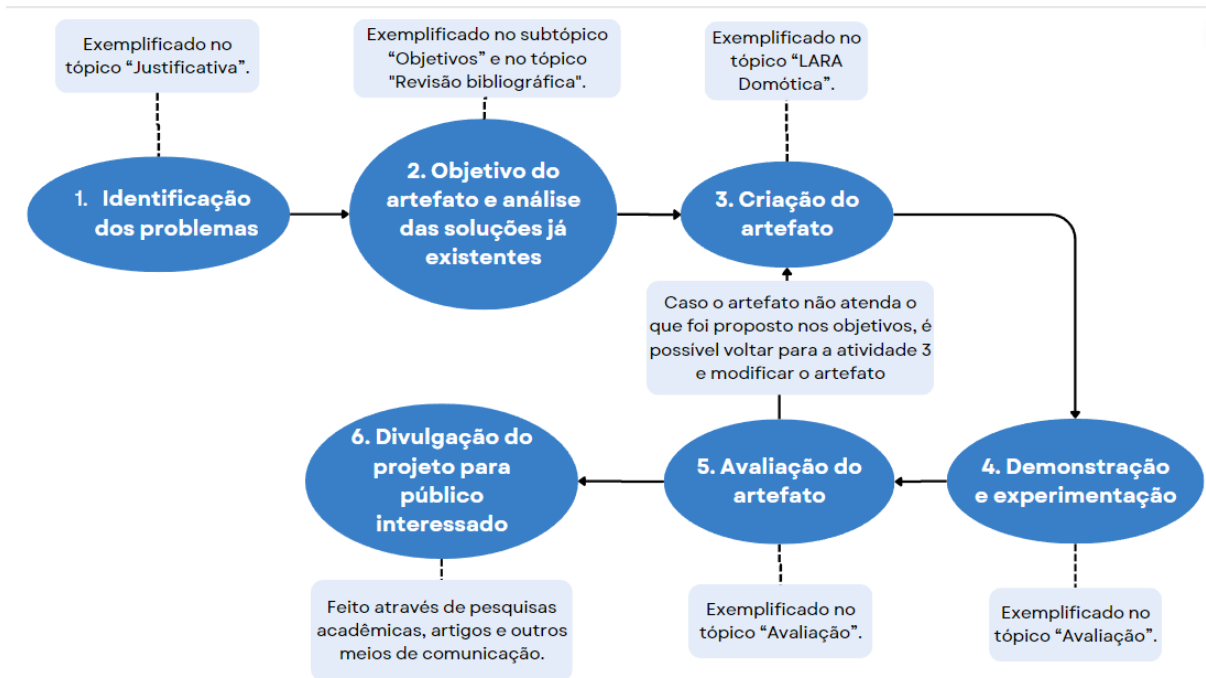
Na Atividade 3 o artefato é projetado através de modelos, métodos e construções. Nessa etapa é necessário determinar as funcionalidades desejadas e enfim criar o artefato. O tópico responsável pela demonstração dessa atividade é o “LARA Domótica”, onde são expostos os requisitos funcionais e não funcionais, regras de negócio, diagramas de fluxo de interação e as telas do sistema. Na quarta atividade o foco é a demonstração que envolve o uso em experimentação, estudos de caso ou outra atividade apropriada.

Na Atividade 5 é feita a avaliação do sistema considerando os objetivos levantados na Atividade 2 e se o artefato foi capaz de atendê-los. Se necessário o pesquisador pode voltar a Atividade 3 para melhorar seu artefato de modo que seja capaz de atingir seus objetivos. No caso deste trabalho, uma avaliação do sistema foi feita utilizando as Heurísticas de Nielsen, com o objetivo de verificar a usabilidade do sistema.

Os resultados obtidos e as mudanças no sistema (decorrentes do resultado da avaliação) podem ser encontrados no tópico “Avaliação” que emprega os objetivos das atividades 4 e 5. A última atividade é relativa à comunicação do projeto para pesquisadores e públicos interessados, incluindo o problema relacionado, as soluções levantadas, demonstração da utilidade do sistema e as avaliações. A divulgação pode ser feita através de artigos, pesquisas acadêmicas e outros meios de comunicação.

No esquema abaixo (Figura 01) é possível observar a sequência de atividades utilizada para desenvolver o presente trabalho e seus resultados:

Figura 01 - Esquema demonstrativo do uso das atividades do DSR neste trabalho



Fonte: De autoria própria

## 2.REVISÃO BIBLIOGRÁFICA

Nesse tópico são abordados conceitos nos quais este trabalho se baseia. Inicialmente é apresentada a diferença entre os laboratórios virtuais e remotos, a arquitetura utilizada para construir ferramentas educacionais a distância e os conhecimentos necessários para o desenvolvimento desse tipo de ferramenta. Em seguida é explicado o conceito de Domótica e as vantagens que seus usuários possuem. O texto também justifica o uso da Domótica neste trabalho e apresenta os dispositivos normalmente utilizados na sua construção. Em seguida são discutidas as vantagens do uso de simulações na área educacional e as diferenças entre as tecnologias 2D e 3D em seu projeto. Por fim, são apresentadas as dificuldades encontradas no ensino de programação e exemplos de ferramentas educacionais que objetivam amenizar estes problemas.

### 2.1.Laboratório virtual

Laboratórios são ambientes de aprendizagem onde os alunos realizam testes e experimentos, embasados em teorias, para visualizar na prática o que já foi aprendido. É uma maneira de demonstrar conhecimentos em um ambiente propício para tentativas e ocasionais erros. Atualmente a modalidade de aprendizado em laboratório também pode ser feita à distância com as versões remotas e virtuais.

Um laboratório virtual é um ambiente digital de estudo, onde o aluno pratica e aprende uma determinada área de foco, similar ao que aconteceria em um laboratório real. Diferente do laboratório remoto, este não possui um experimento real e sim uma ferramenta virtual que simula a realidade. É um meio de simular um laboratório sem a utilização de objetos físicos. Tudo é feito digitalmente imitando a realidade e reduzindo o risco de saúde e segurança ao qual os usuários se expõem ao utilizar esses dispositivos eletrônicos os quais possivelmente não possuem familiaridade e conseqüentemente, poderiam causar acidentes.

Os laboratórios remotos são uma combinação entre ambientes virtuais de ensino com um objeto real que sofre modificações remotamente através desse ambiente virtual. Em outras palavras, existe uma ferramenta real (um robô, por exemplo) que foi criada com o intuito de ser controlada remotamente por vários usuários. Esses se assemelham mais a laboratórios reais e sua construção é feita por uma camada de software, onde o usuário interage com o experimento através da internet, e uma camada de hardware, onde o equipamento realiza os comandos, recebe as configurações e informa os dados do seu sistema (Silva et al.,2020).



Assim como os laboratórios virtuais, esses também possuem acesso através da internet. Alguns laboratórios exigem um requerimento para acesso no intuito de manter somente um usuário por vez utilizando o experimento. Nesses casos é feito um sistema de controle para que cada um tenha um tempo definido e agendado para utilizar o sistema.

Um laboratório virtual, como uma ferramenta de aprendizado a distância, deve possuir algumas características em sua arquitetura. A arquitetura básica proposta por Nick van Dam (DAM,2004) aponta a necessidade de sistemas de gerenciamento de ensino para promover um acompanhamento do progresso de aprendizagem de cada aluno. Além disso, é fundamental a gestão do conteúdo tanto em relação a adição de atividades como a administração de usuários. Ademais, é indispensável a inclusão de ferramentas de desenvolvimento que permitam adição de conteúdo por pessoas que não possuam habilidades de programação. É necessário também a inclusão de ferramentas que permitam a colaboratividade através da troca e compartilhamento de informações e edição conjunta de trabalhos. Outro fator importante é a disposição de ferramentas síncronas e assíncronas para interação entre alunos e professores. É importante que os ambientes implementados possam ser usados para acessar informações sobre os tópicos e atividades, registros de cursos e suporte. O laboratório virtual deve permitir a interação do aluno com o professor, com o conteúdo e com outros alunos (MOORE,1989).

O processo de construção de um ambiente de aprendizagem web envolve habilidades como desenvolvimento web, conteúdo educacional, experiência de usuário, design da interface, banco de dados, segurança de sistemas web, gestão de projetos, entre outros. Durante e após o desenvolvimento é necessário a aplicação de testes de validação do sistema. Dependendo do número de pessoas envolvidas, esse projeto pode levar bastante tempo, porém a sua importância na área da educação faz deste um projeto valioso.

## 2.2. Domótica

Também conhecida como automatização residencial ou casa inteligente, a Domótica é a tecnologia de gerenciamento para automatizações residenciais. Uma casa inteligente consiste em uma moradia que dispõe de diversos aparelhos

eletrônicos que buscam facilitar a vida do morador tanto em questões de segurança como em relação ao conforto. Cada um desses aparelhos pode reagir de maneira automática a alguma interferência no ambiente, seja temperatura, luz solar ou capturas de sensores de movimento. É possível também programar alguns aparelhos com o tempo em que se deseja ligá-los. Na simulação proposta neste trabalho, o usuário poderá programar o dispositivo para realizar suas ações em relação a algum desses fatores presente na simulação. Além disso, outras funcionalidades também serão discutidas ao longo do projeto.

Entre as vantagens presentes na adoção da casa inteligente se encontram os benefícios ambientais, financeiros, psicológicos e relacionados à saúde. A literatura atual mostra como esses fatores são positivos pontos para a adoção da automatização residencial (Nascimento e Fettermann, 2021). A redução do consumo da energia é um exemplo de como o uso da automatização residencial pode afetar a vida financeira do usuário e o meio ambiente, já que é possível que o sistema da casa faça uma análise do que foi gasto e diminua o consumo de energia ao desligar aparelhos de acordo com o comportamento do morador, observado pelo sistema. Já como benefícios psicológicos tem-se o entretenimento de fácil acesso e o conforto. Em relação à saúde pode-se citar a melhor comodidade que pessoas com necessidades especiais e idosos podem ter em um ambiente automatizado.

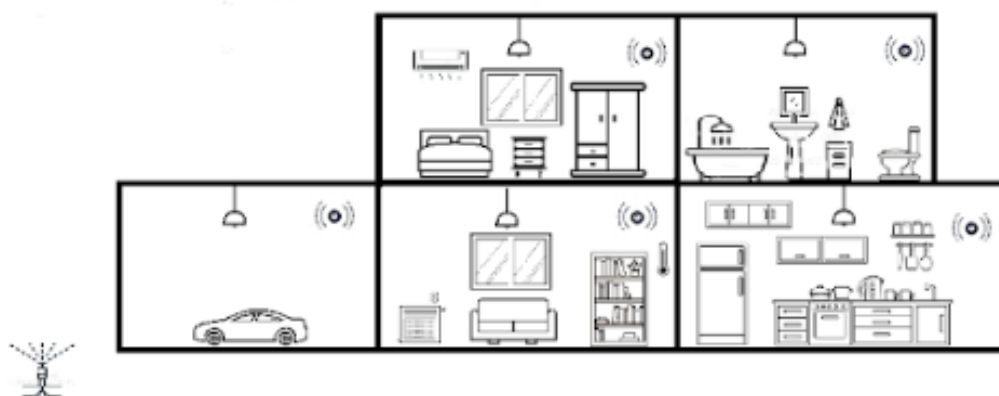
Buscando o menor custo e a praticidade, o presente trabalho optou pelo uso de uma simulação de Domótica, cujo objetivo secundário além da programação é a utilização dos conceitos e dispositivos que a compõem como introdução a essa área da robótica/computação que está em evidência. O custo financeiro de criar uma casa automatizada ainda é inacessível para a maioria das pessoas. Por esse motivo, para fins didáticos, as opções alcançáveis ainda se limitam a maquetes e casas automatizadas virtualizadas. O aluno poderá interagir com o ambiente da casa e criar novos cenários de acordo com a sua criatividade, com a possibilidade de aplicar os conhecimentos de determinada linguagem e gerar resultados diferentes.

### 2.2.1. Dispositivos

Para a realização desse projeto é necessário o entendimento de algumas ferramentas comumente utilizadas na automatização de residências. A Figura 02

mostra a distribuição dos dispositivos que serão abordados neste capítulo e incorporados no protótipo da simulação da casa.

Figura 02 - Primeiro esboço da simulação de uma casa inteligente



Fonte: De autoria própria

#### 2.2.1.1. Sensores de movimento

Sua função se dá de forma que, se houver algum movimento na sua área de visão, o dispositivo informará ao usuário ou fará alguma ação. É uma ferramenta muito útil na segurança e melhora a dinamicidade da rotina dos usuários. Se houver uma invasão na residência do usuário, o sensor de movimento, acrescido de outros equipamentos, pode soar um alarme e avisar empresas de segurança imediatamente promovendo uma maior tranquilidade para os moradores. Outro exemplo de sua utilidade seria a automatização de portas, luzes e câmeras que podem ser ativadas de acordo com a movimentação nas áreas de visão do sensor.

Os sensores de movimento podem ser ativos e passivos (Figura 03). Os sensores ativos possuem dispositivos que são posicionados em extremidades opostas, um emitindo e outro recebendo um feixe de luz infravermelho. Quando esse sinal é interrompido o sensor informa que houve uma movimentação na área. Já os sensores de movimento passivo fazem a leitura do ambiente, sendo ativados ao detectar a radiação infravermelha que emana dos seres vivos.

Figura 03 - a) Sensor de movimento passivo      b) Sensor de movimento ativo



Fonte: <https://www.intelbras.com/pt-br/seguranca-eletronica/sensores>

#### 2.2.1.2. Sensor de fumaça e Sensor de temperatura

Muito utilizados para evitar acidentes durante incêndios, esses dispositivos disparam luzes e/ou sons como forma de alerta de que um incêndio pode estar acontecendo. Existem alguns dispositivos que, ao serem acionados, informam a central de incêndio sobre o ocorrido para evitar danos às pessoas. Combinados com *sprinkles*, formam um conjunto de dispositivos muito eficientes na prevenção de acidentes desse tipo.

Uma simples diferenciação entre o sensor de fumaça e o sensor de temperatura (Figura 04) é que o primeiro inspeciona o ambiente em busca de sinais de fumaça, já o segundo só será acionado no caso de a temperatura ambiente ultrapassar o valor estabelecido como limite de perigo. Existe atualmente um dispositivo que possui os dois sensores ao mesmo tempo.

Figura 04 - a) Detector de fumaça      b) Detector de fumaça com sensor de temperatura



Fonte: <https://zeusdobrasil.com.br/produtos/detalhes/detector-optico-de-fumaca-ip20-24v-enderecavel>

### 2.2.1.3. Sprinkler

Sprinklers são dispositivos conectados em tubulações que, em contato com altas temperaturas, são ativados, liberando água no ambiente para prevenir ou apagar incêndios. Existem diferentes tipos de sprinklers com mecanismos e temperatura de ativação próprios, mas o princípio permanece o mesmo.

Além dessa funcionalidade, os sprinklers (Figura 05) também podem ser usados para irrigação de jardins. Isso pode se dar de modo automático e programado através da medição da temperatura, da umidade do solo e da possibilidade de chuva, formando um sistema eficiente e prático.

Figura 05 - a) Sprinkler de prevenção de incêndio



Fonte: <https://www.multiseg.com.br/2805/sprinkler-tipo-pendentes-68-1-2-fator-k80-vermelho>

b) Sprinkler de jardim



Fonte: <https://www.amazon.com.br/Aspersor-Sprinklers-rotativos-irrigacao-A3o-Generico/dp/B0B5LQRT3P>

### 2.2.2. Maquete residencial utilizando Arduino

Como exemplo da utilização dos conceitos de Domótica temos uma maquete residencial desenvolvida por Tófoli (2014). Neste projeto foi criada uma maquete de uma casa com dispositivos automatizados que podem ser controlados remotamente através de uma página web. A maquete foi desenvolvida com o uso de Arduino, servos motores, sensores e luzes que, combinados, foram capazes de criar esse experimento. A programação realizada em cada dispositivo foi feita através do ambiente de desenvolvimento do Arduino, utilizando a linguagem C++.

O projeto possui luzes, condicionadores de ar, portões automatizados e sensores de presença com alarme. Além disso, foi desenvolvido um software que interage com esses dispositivos, desligando e ligando estes remotamente, através de uma página web. Nesta página também é possível o cadastramento de usuários comuns e administradores. Os administradores possuem acesso privilegiado e podem monitorar as atividades feitas pelos usuários comuns, através da aba de registros, para prevenir o sistema das ações nocivas dos mal intencionados.

Apesar de ser sistema web e trabalhar com os conceitos de Domótica, esta plataforma não tem o foco no aprendizado de programação, diferentemente do projeto proposto por esse trabalho.

Figura 06 - Painel de controle da ferramenta

Bem-Vindo Admin Hoje é: 15/09/2014 - 20:54:28 [Sair](#)

: Painel de controle

ID Dispositivo	Nome	Tipo	Pino	Liga	Desliga	Ligar	Desligar	Status
1	Luz da sala	Illuminacao	2	1	0	Ligar	Desligar	
2	Luz da cozinha	Illuminacao	3	3	2	Ligar	Desligar	
3	Luz do quarto 1	Illuminacao	4	7	6	Ligar	Desligar	
4	Luz do quarto 2	Illuminacao	6	9	8	Ligar	Desligar	
5	Portão Eletrônico	Seguranca	7	r	1	Ligar	Desligar	
6	Ar Condicionado	Ventilacao	8	b	a	Ligar	Desligar	
7	Alarme	Ventilacao	9	f	c	Ligar	Desligar	
14	Luz do quarto 3	Illuminacao	13	y	z	Ligar	Desligar	

.....Copyright © RT Automação Residencial - Todos os direitos Reservados-WebMaster By RJT.....

Fonte: TÓFOLI (2014)

### 2.3. Simulação

Os simuladores podem ser definidos como ambientes de aprendizagem que apresentam uma representação de algum fenômeno real com o qual os alunos podem manipular, explorar e experimentar (Jonassen 1996). A simulação pode ser

usada como aliada no ensino pois permite que os alunos tenham contato com qualquer tipo de experimento de maneira segura. Hoss et al. (2009) cita que:

O uso de simuladores é vantajoso em se tratando de ambientes que podem trazer algum tipo de risco para o operador e/ou desgaste do equipamento, bem como evita as preparações requeridas para o uso do equipamento real.

Outra vantagem apresentada por Hoss et al. (2009) no uso de simulações se dá no fato de que os alunos podem ter contato com o experimento, mesmo que de maneira virtual, e assim adquirem conhecimentos básicos necessários antes de utilizar uma versão real dele. Existem vários temas desenvolvidos em simuladores educacionais, entre eles os mais comuns são aqueles voltados para o ensino de química e física. O protótipo de laboratório virtual desenvolvido neste trabalho contém uma simulação de uma casa automatizada.

O propósito da utilização de simuladores neste trabalho se justifica pelo alto custo financeiro envolvendo equipamentos encontrados em uma casa inteligente. Entretanto, esse motivo não explica porque não usar uma maquete com versões menores dos dispositivos apresentados no subtópico anterior. Para desenvolver maquetes funcionais contendo dispositivos eletrônicos que representem uma casa inteligente seriam necessários conhecimentos acerca de circuitos e componentes robóticos, entre outros. Além disso, uma maquete possui algumas limitações que não existem na simulação. Como exemplo, é possível citar uma maior variedade de temperatura e clima que um ambiente simulado pode dispor.

A simulação pode ser utilizada como atrativo para os alunos por ser um meio diferente de aprendizado pois permite interação em um ambiente controlado e virtual que busca auxiliar o ensino dos temas abordados. A experiência interativa disponível na utilização de uma simulação permite que os alunos tenham uma relação ativa com o conteúdo e os conceitos envolvidos nesse ambiente, podendo implicar em momentos de diversão que promovam aprendizado. Braga (2001) discorre sobre a relação entre educação e realidade virtual (conceito amplo no qual a simulação se insere):

A introdução da Realidade Virtual na educação demonstra um novo paradigma que relata uma educação de forma dinâmica, criativa, colocando o aluno no centro dos processos de aprendizagem e buscando uma formação de um ser crítico, independente e construtor de seu conhecimento.

Ao desenvolver uma simulação é necessário definir as tecnologias que serão utilizadas. Uma representação bidimensional (2D) ou tridimensional (3D) do ambiente pode afetar a simulação de diversas maneiras. As vantagens do uso da representação 2D é a facilidade de implementação em relação a tecnologia 3D. Outro ponto positivo da representação 2D são os recursos mínimos de hardware necessários para a sua execução se comparados com o 3D. Com relação ao nível de imersão, as simulações em 3D são superiores e por isso podem manter o usuário mais comprometido com o processo de aprendizagem.

Para escolher a melhor opção é necessário considerar os objetivos da simulação e os recursos computacionais do público-alvo. Tendo em vista o objetivo da simulação neste projeto, um ambiente 2D seria mais vantajoso pois além de não exigir dos usuários máquinas de alto custo também são mais fáceis de serem desenvolvidas.

#### 2.4. Ensino de programação

O objetivo de aprender programação é permitir que o aluno consiga desenvolver códigos que sejam capazes de solucionar um problema apresentado. Essa matéria, presente nos cursos de computação, tem sido motivo de desistência de alguns alunos pelo seu alto grau de dificuldade. Para Gomes et al. (2008), os problemas começam na fase inicial da aprendizagem, causadas pela necessidade de aplicação de conceitos abstratos, como estruturas de controle. Ele também aponta a necessidade de conhecimentos prévios em raciocínio lógico, capacidade de resolução de problemas, entre outros, que a maioria dos estudantes não possui.

O ensino da programação é feito de maneira inadequada para atender as necessidades dos alunos. Gomes et al. (2008) cita alguns problemas encontrados no ensino dessa matéria, dentre eles a falta de ensino personalizado que atenda cada aluno de maneira individual, que dificulta a evolução dos alunos pois não possui foco nos seus pontos fracos. Outro problema apresentado são as estratégias de ensino utilizadas pelos professores que não contemplam as necessidades dos alunos, já que cada um aprende de forma e com ritmos diferentes, sendo ideal que o professor contemple as individualidades dos alunos. Além disso, o ensino dos conceitos de programação de modo estático, com o uso de slides e gráficos, por



exemplo, ao invés de dinâmicos, afeta o entendimento da matéria. Outro problema encontrado é que durante as aulas os professores normalmente priorizam o ensino da sintaxe da linguagem utilizada em detrimento da resolução de problemas que pode ser criada com o uso de tal linguagem.

De acordo com Jenkins (2002), os métodos de estudos realizados pelos alunos também não são adequados para a matéria. Durante a vida escolar, muitos utilizam o mesmo padrão de aprendizado que consiste em ler e decorar livros. Esse modo não pode ser aplicado na programação, já que para internalizar os conceitos da matéria é necessário muita prática e constância na resolução de problemas através do desenvolvimento de códigos. Os alunos que não dedicam tempo o suficiente normalmente não conseguem atingir seus objetivos de estudo.

Para amenizar os problemas apresentados é necessário que a evolução dos estudantes seja acompanhada pelos professores para que estes possam identificar os pontos onde o aluno tem mais dificuldade e auxiliar na melhoria das suas deficiências. Por isso, a interface do laboratório virtual proposto conta com ferramentas de acompanhamento do desempenho na programação e no planejamento dos códigos dos alunos.

Como forma de diminuir o impacto da falta de materiais dinâmicos para ensino de programação, o presente trabalho propõe uma simulação de Domótica que mostra aos alunos o resultado de seus códigos. Esta solução também tem potencial de aumentar o tempo que o aluno permanece programando, já que uma simulação é capaz de atrair a atenção do aluno, possibilitando a aprendizagem de maneira lúdica. Os trabalhos apresentados a seguir exemplificam ferramentas para ensino que amenizam os problemas encontrados no ensino de programação.

#### 2.4.1. Open Roberta Lab

O Open Roberta Lab<sup>1</sup>(Ketterl et al.,2015) é um website de aprendizagem de programação de robôs totalmente gratuito que utiliza programação em blocos. É classificado como laboratório virtual de ensino, pois possui características como aprendizado a distância e segurança por não envolver máquinas reais. Também

---

<sup>1</sup> Disponível em: <https://lab.open-roberta.org/>

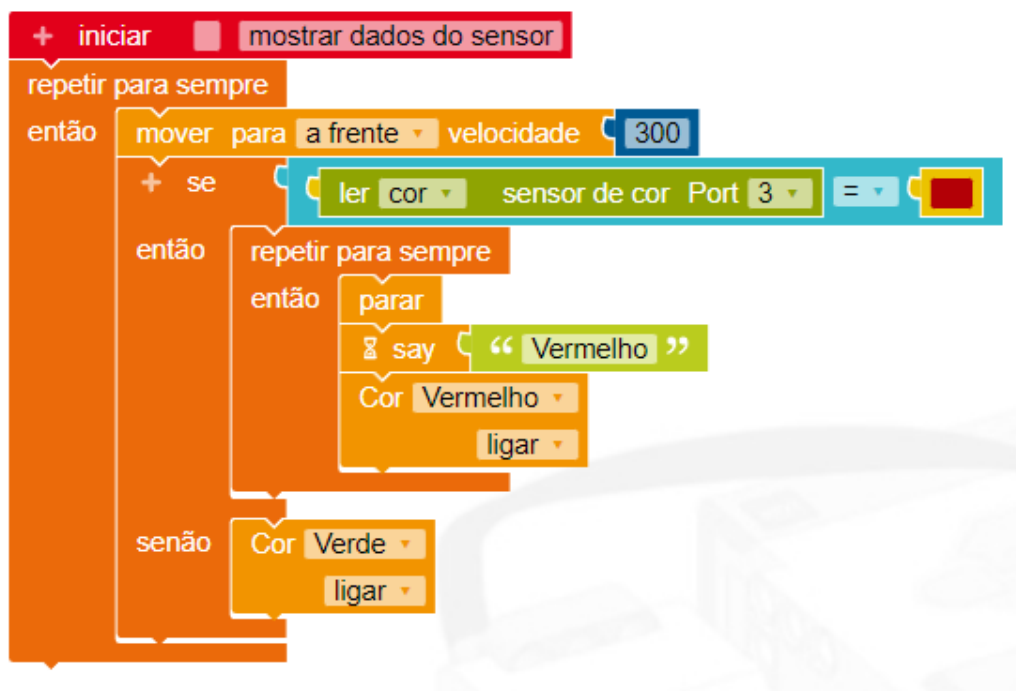
possui características de laboratório remoto já que possibilita a conexão do sistema com um robô real.

Ao entrar no site do Open Roberta o usuário pode fazer um pequeno tutorial ou ler o manual de uso. Existem diversas opções de robôs móveis diferentes disponíveis no sistema para serem utilizados. O objetivo é construir uma lista de ações que o robô irá seguir, dentre eles é possível andar por todas as direções, reconhecer obstáculos e cores, criar percursos, modificar a velocidade do robô, entre outras funcionalidades.

É uma ferramenta simples de ser usada por apresentar programação em blocos que auxilia os iniciantes no aprendizado da lógica por trás das linguagens de programação. Os atributos positivos na programação em bloco são: uma melhor experiência do usuário, a facilidade de entrada nos paradigmas da programação e a capacidade de evitar erros sintáticos típicos, pois as declarações já estão implementadas em formas visuais (Ketterl et al.,2015). A programação em bloco utilizada chama-se NEPO e funciona da seguinte forma: O usuário escolhe um bloco que já vem determinado as ações e os parâmetros necessários para ocorrer a ação proposta por aquele bloco. Desse modo, basta ao usuário preencher os espaços com parâmetros que deseja e o programa se encarrega de reproduzir a ação visualmente, na aba *sim* do Open Roberta.

Esse tipo de programação é pensado para ajudar novos programadores a entender a lógica por trás de variáveis, condicionais e loops sem se preocupar com a sintaxe presente nas linguagens (Ketterl et al.,2015). Juntamente com esses conceitos de programação o usuário desse laboratório virtual também aprende conceitos de robótica e como usar sensores sem ter a necessidade de ter contato com um robô real.

Figura 07 - Imagem código no Open Roberta utilizando programação em blocos

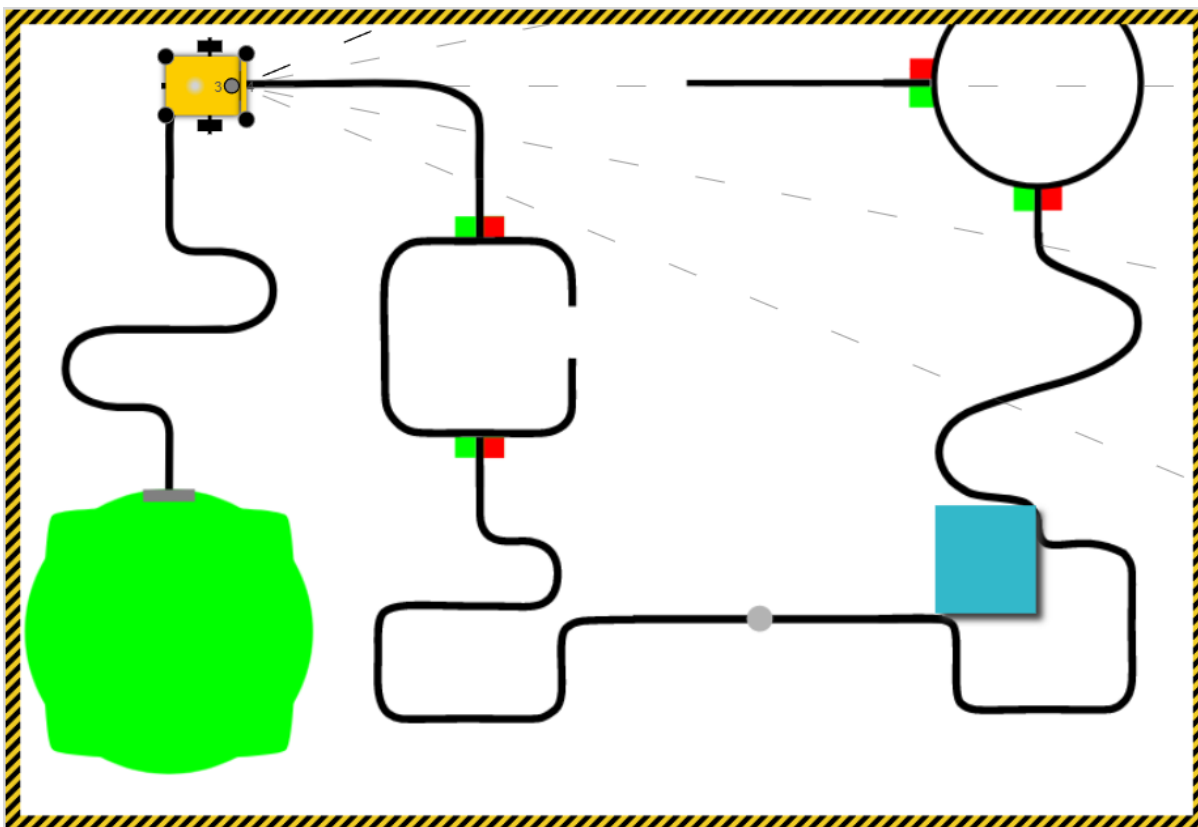


Fonte: Produzido pelo autor no laboratório remoto disponível em <https://lab.open-roberta.org/>

Caso o usuário decida utilizar um robô real, só é necessário conectá-lo ao sistema através do *Wi-fi* ou, no caso de não haver conexão com a internet, através do *Bluetooth* ou USB. O código desenvolvido pelo usuário é executado no robô.

Além disso, o Open Roberta ainda permite que seus usuários contribuam com o desenvolvimento do laboratório ao disponibilizar seu código no *GitHub* e ensinar o passo a passo para que qualquer programador possa criar seu clone e modificar esse laboratório virtual do seu jeito.

Figura 08 - Imagem de um circuito e robô virtual disponibilizado no Open Roberta



Fonte: <https://lab.open-roberta.org/>

#### 2.4.2. LabPy

O *LabPy* (Balieiro,2018) é uma proposta de um ambiente web que facilita o ensino de Python. Nele os alunos podem aprender um tópico do assunto e logo depois resolver questões práticas que ajudam a sedimentar o conhecimento. A ferramenta também ajuda professores já que os exercícios propostos nela são corrigidos automaticamente e o resultado do trabalho de cada aluno fica registrado no sistema. Dessa forma, o professor tem mais controle do desempenho de cada aluno e pode tomar as melhores decisões para o seu desenvolvimento.

A ferramenta possui dois tipos de acesso: o de professor (administrador) e o de aluno. É possível, com o login de administrador, realizar modificações, como excluir, adicionar e editar alunos, questões e turmas de cada professor. Já no acesso do aluno, é permitido selecionar e estudar o assunto separadamente na ordem que preferir, resolver questões e provas, e assistir vídeos com explicações de um

determinado tópico, bem como ler referências externas de livros e artigos. Resumindo, o *LabPy* oferece um ambiente virtual de ensino voltado para aprendizagem da linguagem de programação *Python*:

Dessa forma, um sistema deste tipo também apresenta um ganho significativo para o professor, liberando-o da atividade mais pesada de corrigir exercícios e acompanhar de perto cada estudante. Por outro lado como todo exercício receberá um feedback imediato; o estudante ganha tempo; não existe a necessidade de uma espera para saber se ele está indo na direção certa. (Sirotheau et al.,2018)

Abaixo é possível observar como uma questão é cadastrada no sistema e a área do professor:

Figura 09 - Cadastro de questão no LabPy

## Dados da Questão

### Módulo

Variaveis, Expressoes e Comandos

### Dificuldade

Facil

### Enunciado

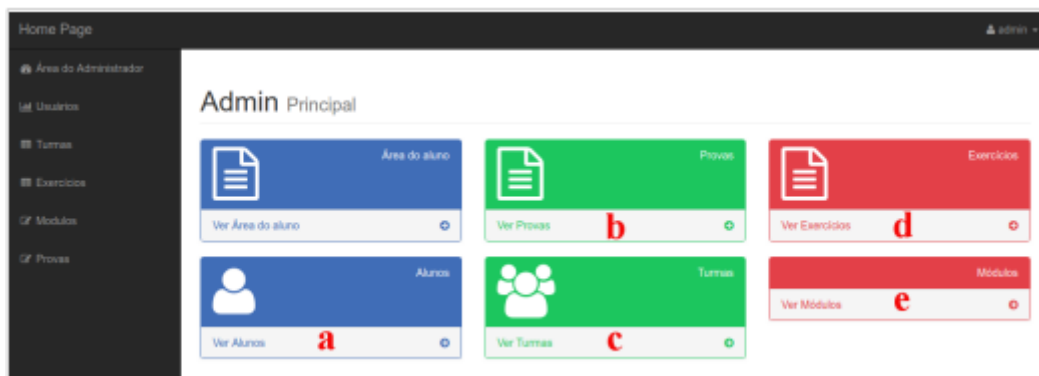
Faça um programa para somar dois números.

### Resposta

```
a = 1
b = 2
soma = a + b
print "A soma vale: ", soma
```

Fonte : Balieiro/2018

Figura 10 - Área professor



Fonte: Balieiro/2018

### 3. LARA DOMÓTICA

Nesta seção serão apresentados a proposta da ferramenta, os requisitos funcionais e não funcionais definidos, os diagramas de casos de uso, os fluxogramas de interação e as regras de negócio do sistema. Por fim são apresentadas as telas do sistema e suas funcionalidades.

#### 3.1. Proposta

O experimento desenvolvido neste trabalho consiste em uma simulação de uma casa inteligente com dispositivos automatizados em vários cômodos. A ideia principal é o ensino de programação de uma maneira simples que prenda a atenção do aluno e faça ele se interessar cada vez mais pelo assunto. Os estudantes precisam sentir que o desenvolvimento nessa área não é tão difícil quanto parece. O trabalho desenvolvido pelo Open Roberta Lab (ketterl *et al.*,2015) enfatiza que:

A maioria dos novos estudantes não têm interesse particular na programação. Nós temos em média 20 segundos para prender sua atenção antes que eles fiquem entediados e saiam para jogar videogame. Cada barreira de entrada (instalando Java, um download lento, aprender inglês, ou uma interface confusa) representa uma significativa perda de audiência.

Para um iniciante que está acostumado com métodos de aprendizado mais teóricos, a programação pode se tornar um desafio, gerando desmotivação e desistência. O processo de aprendizagem de programação possui métodos diferentes em relação às formas usadas desde o início da vida escolar de cada um. Trata-se de muita prática e repetição até que o estudante consiga um bom nível de proficiência na área.

O método de ensino usado em sala de aula (leitura intensa, aplicação de fórmulas e memorização) não condiz com o modo como a aprendizagem de programação deve ser feita. As etapas envolvem o entendimento e reconhecimento do problema, pensamento lógico para sua resolução e tradução dessa solução em linhas de código numa linguagem de programação. Diferentemente de vários outros assuntos, a compreensão da programação não se busca somente em livros. O modo como o aluno absorve os conceitos e internaliza os métodos é através da repetição e intensidade na prática de codificar.

Os trabalhos apresentados na seção “Revisão bibliográfica”, embora sejam ótimas contribuições em relação a laboratórios virtuais ou remotos, não contemplam todas as características necessárias em um só software para manter um ambiente de aprendizado ideal. Este deve atrair a atenção dos usuários através de interações deste com o ambiente, podendo modificá-lo e criar diversos cenários do modo que achar conveniente. Além disso, é necessário permitir o seu uso em ambientes escolares e acadêmicos para auxiliar a avaliação de professores, além de permitir a interatividade entre os alunos através da comunicação síncrona (que acontece em tempo real) e assíncrona (participantes não precisam estar online ao mesmo tempo) por chat e a realização de atividades em grupo, de modo síncrono ou assíncrono, ou individual.

Considerando esses e outros fatores já abordados é evidente que a ferramenta pretendida neste trabalho preza por um sistema que seja fácil de usar, além de ter toda a estrutura educacional bem definida. Como um dos objetivos é romper as barreiras de entrada do novo usuário, este poderá utilizar o sistema através de um navegador web com ou sem a criação de uma conta. Pensando nisso são priorizadas as especificações dos requisitos do projeto, definindo suas características principais e desejáveis.

## 3.2. Requisitos

### 3.2.1. Requisitos funcionais

O sistema permitirá ao usuário uma modificação do ambiente em que a casa automatizada se encontra. O aumento e a diminuição da temperatura ambiente, a modificação do horário, e a integração entre os equipamentos serão algumas das funcionalidades disponíveis.

Cada componente interativo presente na casa terá uma resposta visual quando corretamente programado. Como exemplo, tem-se que, se o usuário conseguir programar corretamente a porta automatizada combinado com o sensor de movimento, será exibida uma animação na qual esta reconhece a movimentação e abre uma porta. Para representar um humano que, numa casa real, seria



responsável por acionar esses dispositivos de presença, o software disponibilizará um boneco modelo humano que poderá ser direcionado por toda a casa.

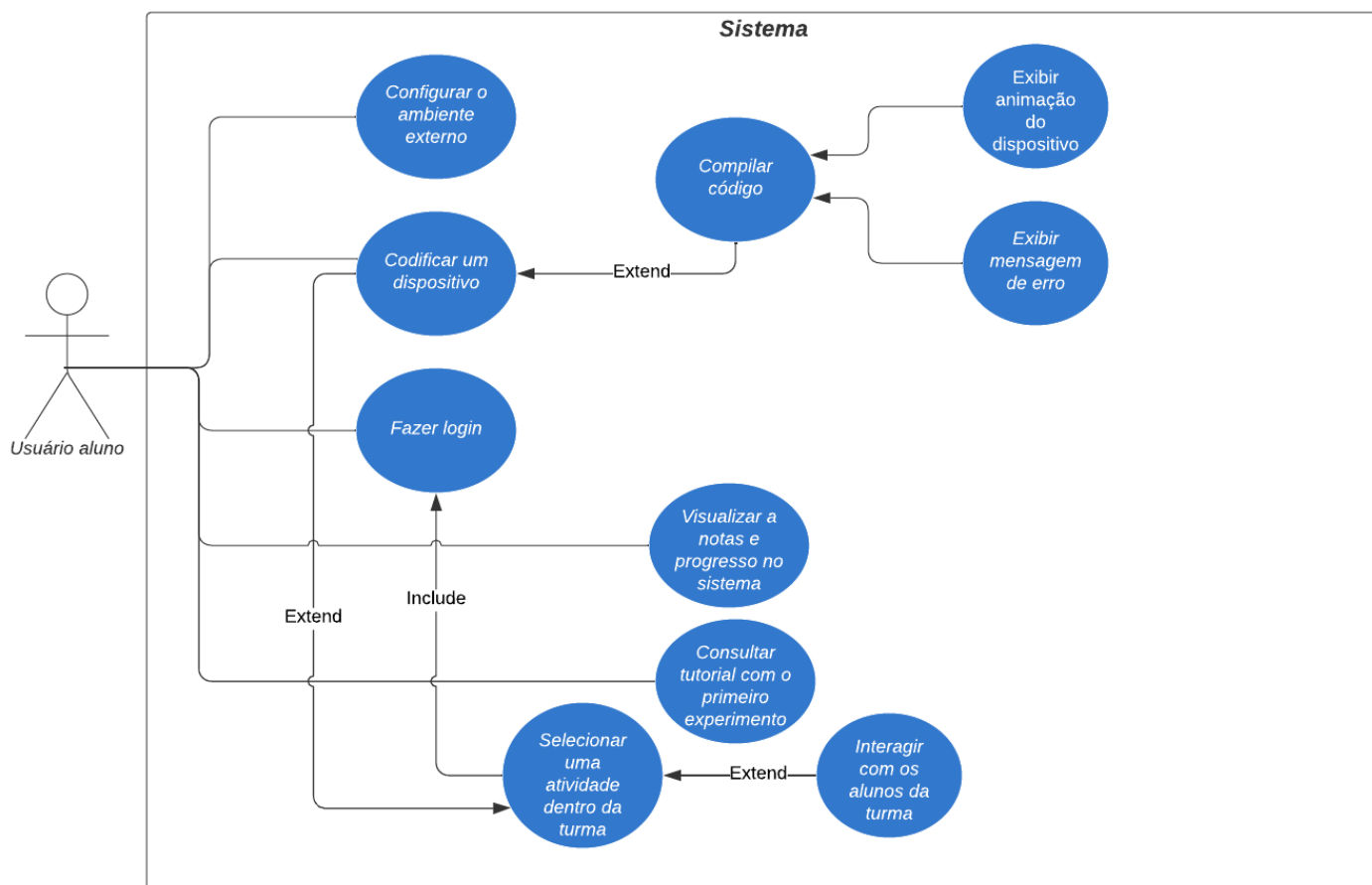
O sistema apresentará uma tela contendo a casa e ferramentas de modificação do ambiente (anteriormente descritas), objetos não programáveis (como móveis e a estrutura física da casa) e objetos programáveis (como as ferramentas mencionadas no tópico "Revisão bibliográfica"). Além disso, botões ficarão posicionados no canto superior da tela para facilitar a experiência da simulação de cada usuário.

Com o código desenvolvido, o usuário deverá apertar o botão que inicia a compilação do código. No caso de não serem encontrados erros, uma animação dos dispositivos utilizados será exibida. No caso do código digitado pelo usuário estar incorreto, será exibida uma mensagem de erro e o usuário poderá tentar novamente até que seu código esteja correto.

O sistema possui três tipos de usuários: o professor, o aluno e o administrador. Para criar a conta será obrigatório um e-mail, um nome de usuário e uma senha. Na conta do tipo professor será possível criar/excluir turmas, adicionar/remover alunos em cada turma, criar/excluir tarefas individuais ou em grupo, adicionar/remover material educacional alternativo (links de livros e vídeos), e avaliar a participação de cada aluno. Já na conta do tipo aluno será possível a resolução tanto de atividades propostas pelo professor como também a livre utilização das ferramentas. Além disso, quando o aluno for incluído em uma atividade pelo professor, será possível a comunicação entre os alunos que fazem parte de uma mesma atividade e entre o aluno e o professor. Na conta do tipo administrador será possível modificar as senhas e email das contas existentes. Ademais, é possível modificar os tipos de outras contas, de aluno para professor, por exemplo.

Na Figura 11 é possível observar o diagrama de Caso de uso demonstrando algumas interações do projeto.

Figura 11 - Diagrama de Casos de uso do usuário aluno



Fonte: De autoria própria criado no Lucidchart<sup>2</sup>

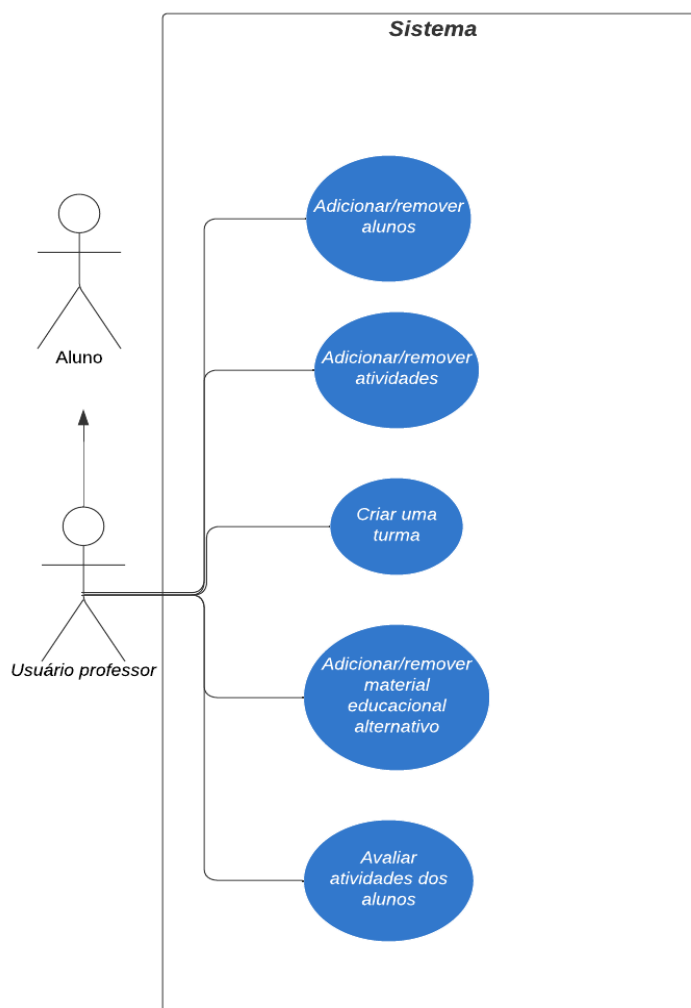
- Configurar o ambiente externo: O sistema deverá possibilitar ao usuário a modificação do ambiente a qual este irá programar. Será possível modificar as variáveis de ambiente como temperatura, luminosidade e horário para que estas interfiram no modo como os sensores atuam. A partir dessas informações, o usuário poderá alterar o comportamento dos dispositivos dependendo de como as variáveis de ambiente estiverem definidas. O dispositivo que abre a cortina pode ser programado para abrir às 7:00 da manhã, por exemplo. Essa funcionalidade tem o objetivo de aumentar as possibilidades de desenvolvimento do código e os resultados de cada experimento.
- Codificar um dispositivo: Como o objetivo desse laboratório é o aprendizado de programação, essa funcionalidade seria a mais básica e importante para o projeto. Desse modo, cada dispositivo programável disposto na área da casa

<sup>2</sup> Disponível em: <https://www.lucidchart.com>

será clicável. Quando clicado abrirá uma aba onde o usuário poderá observar características como ID e bibliotecas mais importantes para escrever seu código.

- Compilar o código: Assim que o código for escrito na área própria disponibilizada será possível compilar o código desenvolvido pelo usuário.
- Exibir animação do dispositivo: Depois de compilado, caso o código esteja sintaticamente e semanticamente correto, o sistema deverá exibir uma animação do dispositivo utilizado no código.
- Exibir mensagem de erro: Depois de compilado, caso o código esteja sintaticamente ou semanticamente incorreto, o sistema deverá exibir uma mensagem de erro.
- Fazer login: Com um nome de usuário e uma senha o aluno poderá fazer login no sistema.
- Visualizar as notas e progresso no sistema : Caso logado, o aluno poderá ver o seu progresso na aba “Área do aluno”. Nesse espaço serão disponibilizados o nome, a data e a pontuação de cada uma das atividades que o aluno já tenha feito. Além disso, se for o caso de uma atividade em grupo, também serão dispostos o nome dos outros alunos que fizeram parte do seu grupo.
- Consultar tutorial com o primeiro experimento: Em caso de dúvida em como começar a utilizar o laboratório, o aluno poderá realizar um tutorial que consiste em um pequeno projeto inicial onde o sistema o guiará, passo a passo, através da ferramenta apresentando suas disponibilidades e opções. No final do tutorial o usuário deverá conseguir utilizar as funcionalidades mais importantes do sistema e terá desenvolvido um código com seu primeiro programa.
- Selecionar uma atividade dentro de uma turma: Caso logado, o aluno poderá selecionar uma atividade no qual foi adicionado. Dentro da área da atividade ele deverá desenvolver seu código de acordo com o objetivo proposto pelo professor que criou a atividade.
- Interagir com os alunos de uma turma: Caso logado e dentro de uma atividade, o aluno poderá se comunicar com os outros alunos dentro da mesma atividade através de um chat assíncrono. O objetivo desta comunicação é que os alunos discutam a divisão das etapas de realização da atividade e que haja uma troca de conhecimentos entre eles.

Figura 12 - Diagrama de Casos de uso do usuário professor



Fonte: De autoria própria criado no Lucidchart

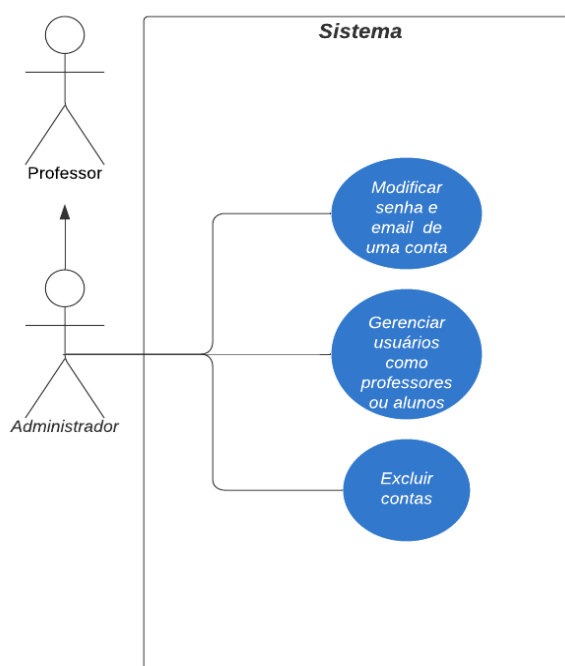
Além de realizar as ações do perfil aluno, o perfil do professor também poderá:

- Criar uma turma: O professor poderá criar turmas e definir os alunos e atividades que essa turma terá.
- Adicionar e remover alunos: Com uma turma criada, o professor poderá adicionar e remover alunos nela.
- Adicionar e remover atividades: Dentro de uma turma, o professor poderá criar e excluir atividades. Essas atividades terão uma pontuação e podem ser em grupo ou individuais. No caso de uma atividade individual, o aluno é avaliado pela correção do código. No caso de uma atividade em grupo, além

do código ser analisado, o professor poderá avaliar o comportamento de cada aluno na construção do resultado final do trabalho. Através do tempo dedicado de cada aluno e a interação deles entre si será atribuído o nível de contribuição de cada participante de um grupo.

- Adicionar e remover material educacional alternativo: Na área da turma, o professor terá a opção de adicionar materiais adicionais como vídeos, slides e livros para que o aluno tenha acesso a mais fontes de conhecimento.
- Avaliar atividades dos alunos: A conta do tipo professor contará com uma aba para analisar o desempenho de cada aluno em suas atividades, de acordo com as atividades realizadas, o tempo dedicado a cada atividade e o nível de participação em tarefas em grupo.

Figura 13 - Diagrama de Casos de uso do usuário administrador



Fonte: De autoria própria criado no Lucidchart

Além de realizar as ações do perfil professor, o administrador também poderá:

- Modificar senha e email de uma conta: O administrador terá acesso aos dados das contas e poderá modificá-los caso o usuário tenha solicitado.
- Gerenciar usuários como professores ou alunos: O administrador poderá modificar o tipo de contas. Como a conta aluno tem menos permissões que a conta professor, as novas contas criadas são definidas por padrão como

alunos. Caso um professor crie uma conta no sistema inicialmente essa conta será do tipo aluno e o professor poderá solicitar ao administrador a mudança da sua conta para o tipo professor.

- Excluir contas: Caso haja necessidade, o administrador pode excluir uma conta, desse modo suas atividades, turmas e notas dentro do sistema também serão excluídas.

### 3.2.2.Requisitos não funcionais

No processo de desenvolvimento de um software, antes mesmo de se iniciar a fase de codificação, é necessário definir as principais características que esta ferramenta deve apresentar, visando o público alvo e o objetivo principal para qual foi pensada.

Uma pesquisa realizada no ano de 2013 intitulada “Requisitos Não Funcionais no Desenvolvimento de Software Educacional: Um Estudo Utilizando Grounded Theory“ apontou, através de entrevistas com 20 alunos de Licenciatura em Informática, os requisitos não funcionais mais desejados em softwares voltados para aprendizagem:

O processo de concepção e levantamento inicia-se com a identificação dos interessados e reconhecimento dos seus pontos de vista com a finalidade de confrontar as inconsistências e apontar áreas em comum. No que se refere ao projeto de um software educacional, a falta de clareza ou equívoco nesta fase pode ser demasiadamente danoso ao ciclo de desenvolvimento em sua totalidade, principalmente quando o público-alvo contempla pessoas que usarão a ferramenta para adquirir conhecimentos basilares para a sua formação.

Entre os requisitos apresentados na pesquisa foram selecionados os mais relevantes para o projeto proposto neste trabalho.

#### 3.2.2.1. Usabilidade

O termo diz respeito a uma propriedade que define o grau de facilidade que o usuário tem ao utilizar um software ou site. Quanto maior for a usabilidade melhor será a experiência de quem usa a aplicação, evitando assim frustrações e evasões de usuários.

Com o tempo, os padrões de usabilidade mudam para se adaptar a melhor atender as necessidades dos usuários, de modo que sua experiência entendendo o funcionamento da ferramenta ou site seja simples e rápida. As cores, os tamanhos dos botões, a posição dos elementos na tela, entre outras coisas, são utilizados para garantir uma experiência intuitiva e uma interface agradável para manter usuários satisfeitos com a ferramenta.

Esse requisito será utilizado para que a única preocupação do usuário seja resolver os exercícios e aprender a programar. Para isso, é necessário que a interface seja amigável de modo que cada etapa, mesmo que nova para o usuário, seja fácil e rápida de ser realizada. Elementos de maior importância relativos à programação em si devem ser de fácil acesso e a ferramenta deve contar com um tutorial para novos usuários que pode ser acessado a qualquer momento durante o seu uso. Também é importante que a interface seja simples e direta, reduzindo cliques desnecessários para executar tarefas e sem muitas informações pouco importantes que possam confundir o estudante.

#### 3.2.2.2.Colaborativo

Colaboração é definida como um conjunto de pessoas desempenhando atividades com o mesmo objetivo final. Somada ao ambiente educacional, a colaboração aumenta as chances de aprendizado de todo um conjunto de pessoas. Um estudo feito comparando o aprendizado individual com o aprendizado colaborativo concluiu que alunos que aprendem em grupo conseguem desenvolver um maior pensamento crítico sobre o assunto abordado (Gokhale, 1995). Além disso, os participantes do estudo disseram que essa prática colaborativa facilitou o aprendizado, ampliou as perspectivas do assunto abordado e manteve o ambiente descontraído.

Um software educacional colaborativo tem como objetivo a união entre os estudantes para o desenvolvimento de suas habilidades. Desse modo, aqueles que possuem maior dificuldade podem ter assistência daqueles que não encontram muitos problemas. Além disso, essa interação entre os alunos cria, em um ambiente virtual, uma maior semelhança com a realidade da sala de aula e com laboratórios reais. Essa prática pode facilitar o ensino para o professor, já que os alunos

poderiam se ajudar e não ficariam tão dependentes deste, sendo assim, seriam menos tarefas para o educador.

Essa característica fará parte do software proposto neste trabalho. Nele o usuário poderá fazer atividades tanto individualmente como em conjunto. O professor poderá criar atividades e definir o número de colaboradores em cada uma delas. Os participantes poderão se comunicar através de um grupo de conversa disponível para todos os participantes da mesma atividade. Além disso, as contribuições dos alunos para o projeto serão registradas e o professor terá acesso a essas informações para observar o nível de participação de cada um no desenvolvimento do projeto, facilitando assim a sua avaliação final. Isso será feito da seguinte forma: O aluno que modificar a atividade terá o usuário registrado no sistema e o professor poderá visualizar esses dados. Além disso, o código digitado por cada aluno terá uma cor diferente. Dessa forma a contribuição de cada um ficará bem definida e o professor poderá atribuir avaliações individuais mesmo em trabalhos em grupo.

### 3.2.2.3.Educacional

Um *software* educacional nada mais é do que um sistema de ensino que é desenvolvido e pensado com objetivo de auxiliar o usuário no entendimento de algum assunto. Essa ferramenta de estudo possibilita aos seus usuários uma forma de aprender no seu ritmo e no momento que julgar conveniente. Para que o foco do uso desse tipo de sistema seja apenas o aprendizado, este deve ser fácil de usar e muitas vezes necessita de um tutorial para novos usuários. Para atender ao requisito educacional um software pode ser instrutivo, informativo, motivacional, investigativo, expressivo, lúdico e inovador.

O uso desse tipo de ferramenta combinado com o ambiente educacional amplia o modo como o indivíduo consegue o acesso ao conhecimento. Para o professor, esse tipo de software pode ajudar tanto como complemento e prática de cada assunto abordado em aula, quanto para introduzir um novo conteúdo para além do ambiente acadêmico e escolar.

Existem diferentes tipos de softwares educativos e podem conter pequenas particularidades. Como exemplo, ferramentas para estudantes com necessidades



especiais podem conter leitores de texto em voz alta. Outro exemplo são as plataformas de jogos educativos que focam em passar o conteúdo de forma lúdica fazendo a experiência de aprendizagem mais divertida.

Pensando na programação como foco do projeto desenvolvido, a importância de um *web software* voltado para a necessidade de aprender a codificar se dá de forma que a falta de habilidades nessa área tem sido grande motivo de evasão de alunos nos cursos de tecnologia por apresentar uma grande curva de aprendizagem. Os alunos que encontram dificuldades acabam se desestimulando e desistindo por não lhes terem sido apresentados uma forma eficiente de internalizar esse tipo de conteúdo.

Como o objetivo do *software* proposto neste trabalho é o de desenvolver o conhecimento dos alunos em programação através de um laboratório virtual com elementos de simulação de uma casa inteligente, as características mais importantes a serem observadas são a necessidade de exercitar e praticar os conceitos já vistos em aula. Além disso, por ser uma simulação em um laboratório virtual, o aluno poderá interagir com elementos didaticamente, que não teria acesso de maneira fácil, como dispositivos automatizados encontrados em casas inteligentes.

#### 3.2.2.4. Disponibilidade

Como já foi definido, um dos objetivos propostos para essa ferramenta é facilitar o acesso ao conhecimento de qualquer lugar com acesso a internet. Por isso é importante que não haja quedas constantes no sistema.

É importante para o sistema que este esteja funcional, acessível e livre de erros durante a maior parte do tempo. Este deve ser mantido livre de *bugs* e disponível através do constante monitoramento do estado da aplicação.

Para auxiliar a constância da qualidade da aplicação foi pensado a área “Fale conosco” onde o usuário pode compartilhar problemas e experiências negativas que possam vir a enfrentar ao utilizar a aplicação. A identificação das causas desses erros deve ser da mais alta prioridade para os responsáveis pela manutenção do sistema. Em caso de mudanças na ferramenta, é recomendável o uso de *backups*

do código com cada uma das versões funcionais que garantirão a continuidade da plena execução do sistema.

#### 3.2.2.5. Ubiquidade (Desempenho)

Uma das metas da ferramenta proposta é que possa ser utilizada de qualquer lugar. Esta pode ser acessada através de qualquer navegador web sem a necessidade de instalação de qualquer software adicional.

É importante esclarecer que como a proposta da ferramenta é aprender programação, esta não poderá ser utilizada em aparelhos móveis, como smartphones, pois seria complicado para o usuário criar um algoritmo utilizando um celular.

#### 3.2.2.6. Documentação

Essa é uma prática que não afeta o sistema diretamente, mas que pode facilitar a sua manutenção e aperfeiçoamento. No processo de desenvolvimento de um sistema, é uma boa prática a criação de documentos que contêm informações sobre todo o processo de criação. A arquitetura, os recursos disponíveis, o modo como o sistema funciona, entre outras informações, são descritas e armazenadas para uso futuro tanto pela equipe de suporte quanto para os usuários finais do produto.

A documentação é essencial para o detalhamento do sistema. Apesar de ser um trabalho a mais no processo de desenvolvimento, vale muito a pena a prática dessa organização, já que as informações necessárias para o pleno entendimento do sistema se encontram ali. No caso do usuário que encontrar alguma dúvida na utilização do software, a documentação pode ser de grande ajuda. Ele poderá recorrer a esse documento para obter informações com relação à visão geral da ferramenta, ao modo correto de uso e seus recursos.

Para os desenvolvedores o peso de uma documentação é ainda maior já que no processo de manutenção existe uma grande necessidade de conhecer o sistema de maneira que as suas alterações sejam para melhorar a sua proposta e não modificar a essência do software. Além disso, em casos de mudança na equipe de

desenvolvimento, os novos integrantes conseguem ter uma visão completa do software desconhecido para que possam executar o trabalho de maneira correta.

### 3.2.2.7. Atratividade

Esse requisito diz respeito à relevância da ferramenta em relação à concorrência e ao nível de aceitação pela comunidade. É importante para que uma ferramenta web tenha acessos constantes e mantenha sua importância no mercado.

Pensando na necessidade de manter o usuário satisfeito com a ferramenta web, a atratividade se torna uma das prioridades. Para que o laboratório remoto tenha essa característica, a sua interface deve ser limpa, sem muitos elementos para que não deixe o usuário confuso com excesso de opções. Como o site tem foco na aprendizagem, as cores devem ser agradáveis e a navegação deve ser fácil e intuitiva. Quando o usuário decidir o que pretende ao acessar o site, o número de etapas necessárias para a realização dessa tarefa deve ser o mínimo possível. Em relação ao código da ferramenta, deve ser priorizada a otimização de modo que não haja travamentos ou um carregamento excessivamente lento.

A simulação de Domótica é uma das particularidades que essa ferramenta terá e tem o objetivo de entreter o aluno enquanto ele aprende. Dessa forma as animações presentes na simulação devem ser agradáveis e divertidas. Essas animações só serão realizadas no momento que o usuário já tenha conseguido compilar o código de maneira correta, pois além de ser uma resposta visual positiva para o usuário (como uma recompensa pelo seu trabalho), também não será capaz de tirar sua atenção no momento de codificação. É importante que não haja elementos animados na tela pois seriam uma distração para o foco do usuário (Keeker, 1997), que nesse caso, deve ser a programação.

Para se manter como uma ferramenta relevante, o laboratório virtual proposto deve estar atualizado disponibilizando os recursos importantes de acordo com a avaliação dos usuários. Por isso, a equipe de manutenção deve se atentar aos feedbacks da comunidade.

### 3.2.3. Regras de negócio

Na tabela abaixo são descritas regras que determinam como cada usuário interage com o sistema.

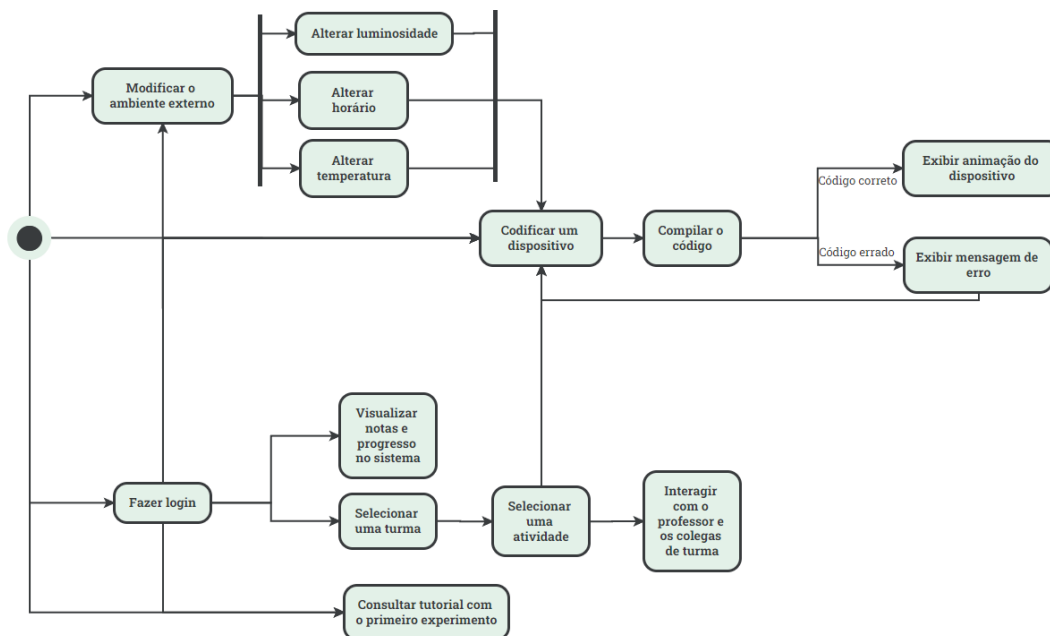
Tabela 01 - Regras de negócio

Nome	Descrição
Usuário sem conta	Um usuário não logado ou sem conta não poderá se comunicar com outros usuários e também não terá acesso a atividades
Usuário aluno	Em atividades individuais o aluno logado não poderá adicionar usuários no seu grupo de conversa
Usuário professor	O sistema não registra atividades e pontuação de um usuário professor.
Grupo	Um grupo de conversa dentro de uma atividade pode ter até 5 integrantes.
Administrador	Só essa conta pode modificar o tipo das outras contas.
Criar atividade	Só o professor pode criar atividades
Atribuir notas	Somente o professor pode atribuir notas e essas notas só podem ser atribuídas a alunos de uma turma criada por esse professor
Adicionar usuário no chat	O usuário aluno só pode adicionar alunos no chat de uma atividade quando esta não for avaliativa
Chat em atividade grupal	Os alunos que fazem parte do chat são definidos pelo professor ao criar o grupo do trabalho
Chat	O professor sempre fará parte do chat em atividades criadas por ele
Contribuição em atividades grupais	Tanto a área de código como a área de rascunho terão seu código separado por cores identificando cada aluno para permitir a avaliação do grupo pelo professor.
Chat e professor	O professor sempre terá acesso às mensagens do chat em atividades criadas por ele.

#### 3.2.4. Diagrama de fluxo de interação

As Figuras 14 e 15 apresentam os diagramas de fluxo de interação realizados pelos usuários do tipo aluno e professor, respectivamente.

Figura 14 - Diagrama de fluxo de interação do Usuário aluno

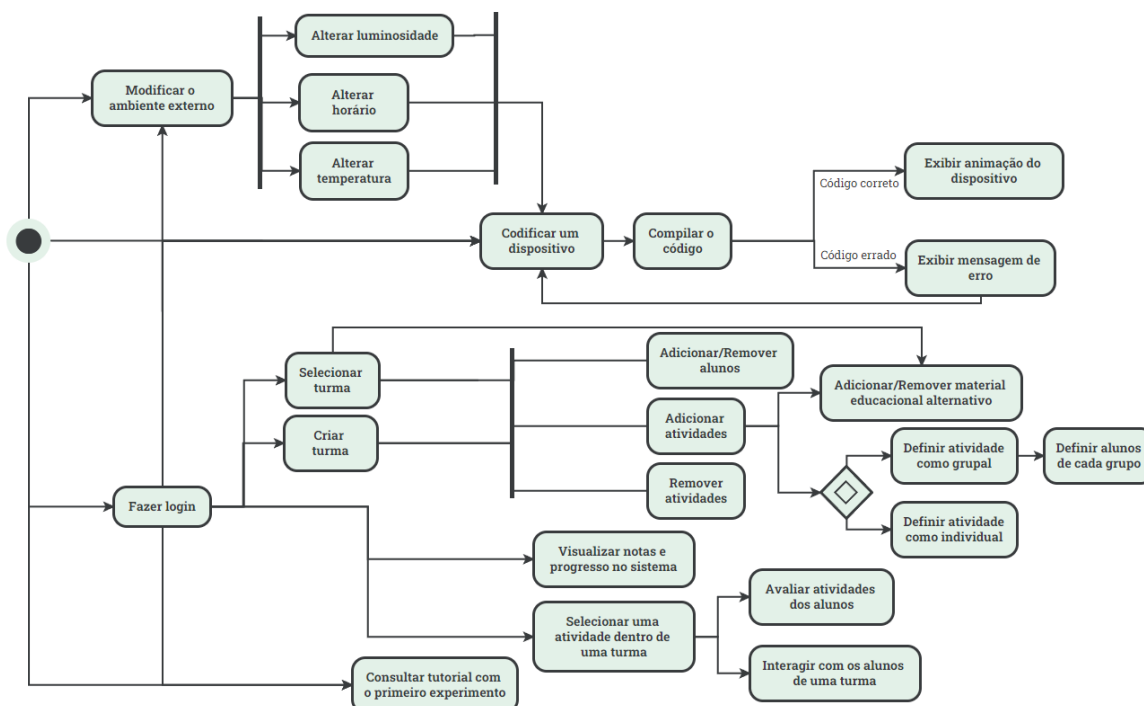


Fonte: De autoria própria criado no Venngage<sup>3</sup>

Caso o aluno sinta dificuldade ao utilizar o sistema ele pode consultar o tutorial disponibilizado. Ao utilizar a simulação é possível modificar variáveis do ambiente (luz, tempo e horário) e codificar dispositivos utilizando uma linguagem de programação. Se o código estiver correto, a animação do dispositivo é exibida. Caso contrário uma mensagem de erro é mostrada no console. Ao fazer login, o aluno pode visualizar suas notas, selecionar uma atividade dentro de uma turma e nela interagir com os colegas.

<sup>3</sup> Disponível em: <https://pt.venngage.com/>

Figura 15 - Diagrama de fluxo de interação do Usuário professor



Fonte: De autoria própria criado no Venngage

Também para o professor, o fluxo de interação se inicia pelo login no sistema. Ele pode selecionar uma turma ou criar uma nova turma e então é possível adicionar e/ou remover alunos e/ou atividades. Ao adicionar uma atividade o professor pode adicionar e/ou remover material educacional e tem a opção de definir a atividade como grupal ou individual. Caso a nova atividade seja grupal é necessário definir os alunos de cada grupo. O professor também pode visualizar as notas e progresso dos alunos no sistema. Selecionar uma atividade abre opções de avaliar o trabalho dos alunos e interagir com eles. Além dessas funcionalidades, assim como no perfil de aluno, o perfil do professor também tem acesso a todas as opções de interação disponíveis na simulação e tutoriais.

### 3.2.5. Telas do sistema

Esse subtópico apresenta as telas do sistema e as funcionalidades presentes nela. As telas foram desenvolvidas utilizando o Bootstrap Studio, um aplicativo desktop para criação de sites. A Figura 16 mostra a primeira tela do sistema.

Figura 16 - Tela inicial do sistema web

**LARA Domótica** Início Programar Fale conosco Central de ajuda Login Cadastre-se

## PROGRAMAÇÃO NA PRÁTICA

Bem vindos ao LARA Domótica! Aqui você aprende a programar de um jeito diferente. Se já conhece nossa ferramenta, clique no botão "Começar". Caso seja novo por aqui, clique no botão "Ajude-me" e te guiaremos em um passo a passo para desenvolver seu primeiro projeto.

[Começar](#) [Ajude-me](#)

### Programe uma simulação de domótica

Erat netus est hendrerit, nullam et quis ad cras porttitor iaculis. Bibendum vulputate cras aenean.

[Ver mais →](#)

### Trabalhe com amigos

Erat netus est hendrerit, nullam et quis ad cras porttitor iaculis. Bibendum vulputate cras aenean.

[Ver mais →](#)

### Seja avaliado

Erat netus est hendrerit, nullam et quis ad cras porttitor iaculis. Bibendum vulputate cras aenean.

[Ver mais →](#)

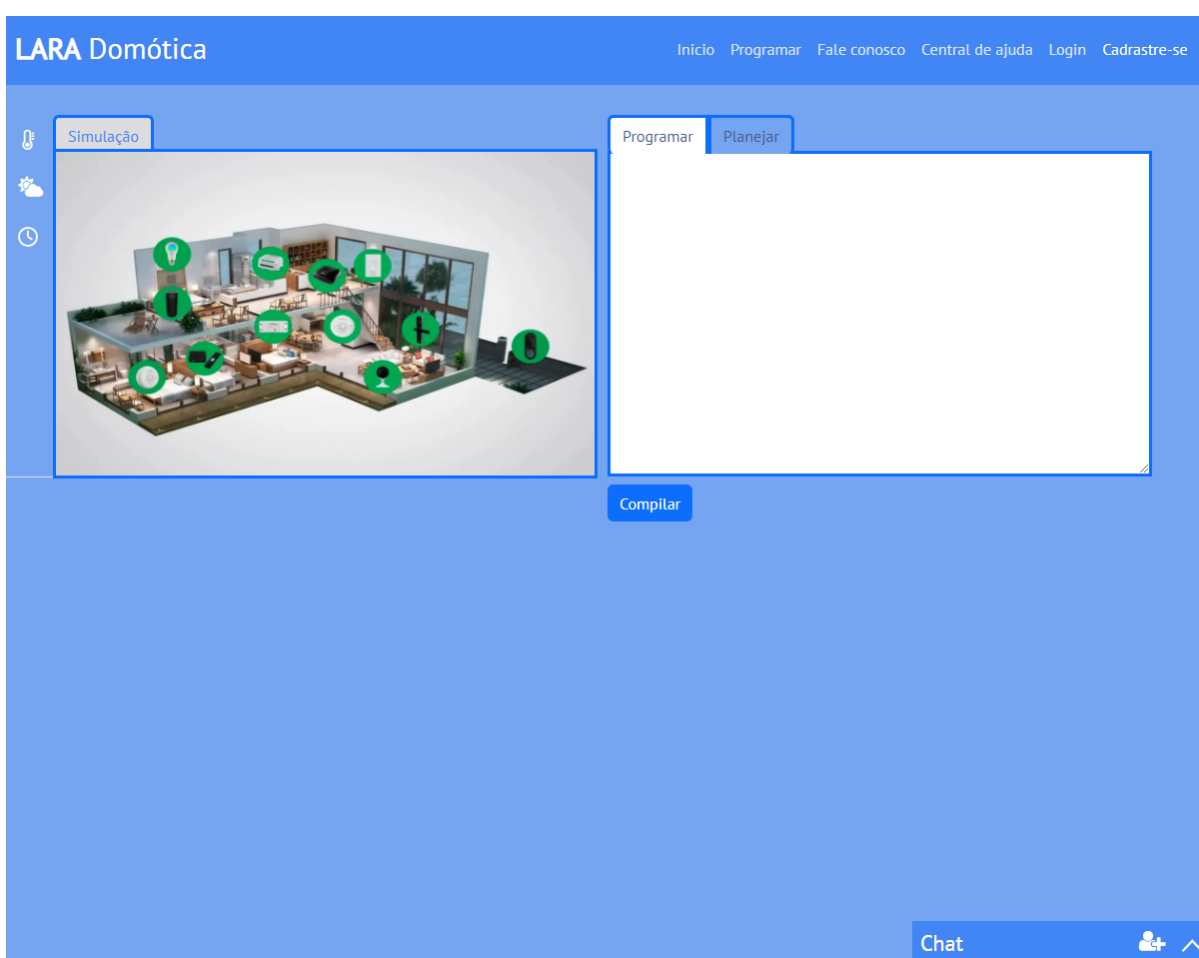
Fonte: De autoria própria criado no Bootstrap Studio<sup>4</sup>

Essa tela é a primeira que o usuário verá ao acessar o site. Nela é possível observar uma breve apresentação do sistema. Ao clicar no botão "Começar", o

<sup>4</sup> Disponível em: <https://bootstrapstudio.io/>

usuário é redirecionado para a página “Programar” onde é possível criar seus experimentos. Ao clicar no botão “Ajude-me”, o usuário é redirecionado para a página “Central de ajuda” onde se encontram tutoriais que auxiliam novos usuários a utilizar esta ferramenta. Logo abaixo, em cada área intitulada, existe um link que, ao ser clicado, leva o usuário ao texto completo que explica melhor a funcionalidade especificada pelo título e como utilizá-la.

Figura 17 -Tela de programação e simulação



Fonte: De autoria própria criado no Bootstrap Studio

Nessa página o programador finalmente poderá desenvolver seu código. Na área “Simulação”, o programador poderá modificar o ambiente no qual deseja executar o seu código. Na área “Planejar”, o programador pode definir o passo a passo do seu código, a lógica a ser seguida e a forma como seu algoritmo funciona.



Já na área “Programar”, o usuário pode criar seu código e compilar ao clicar no botão “Compilar”.

O chat, localizado no canto inferior direito, pode ser usado por usuários logados no sistema. Caso a conta do usuário seja do tipo aluno e este esteja realizando uma atividade, não é possível adicionar novos usuários ao chat. Ao criar uma atividade, o professor sempre fará parte do chat de cada grupo definido na atividade. Será possível adicionar novas pessoas ao chat somente quando a atividade desenvolvida pelo usuário não for avaliativa.

Através do menu lateral esquerdo é possível definir como o ambiente da simulação irá se comportar. Podem ser modificadas características como clima, luminosidade e horário. Cada uma dessas mudanças, dependendo do modo como o programador cria seu código, pode exercer influência na ativação de sensores. Cada aba do menu permite escolher opções pré-definidas ou customizar essas variáveis (Apêndice a).

Na página “Fale conosco” (Figura 18), o usuário poderá enviar comentários, dúvidas, reclamações ou erros do sistema caso tenha encontrado algum problema ao utilizá-lo. É necessário manter o sistema com uma usabilidade agradável e corrigir os erros encontrados constantemente prezando pela boa experiência do usuário.

Figura 18 -Tela “Fale conosco”



A imagem mostra a interface de usuário de uma página web. No topo, há uma barra azul com o texto "LARA Domótica" à esquerda e links de navegação "Início", "Programar", "Fale conosco", "Central de ajuda", "Login" e "Cadastre-se" à direita. O conteúdo principal é um formulário branco centralizado com o título "Fale conosco". O formulário possui três campos de entrada: "Nome", "Email" e "Mensagem" (uma área de texto maior). Abaixo dos campos, há um botão azul com o texto "Enviar". Na base da página, há uma barra azul com o texto "Web design", "Development" e "Hosting" em uma fonte pequena.

Fonte: De autoria própria criado no Bootstrap Studio

Na área “Central de ajuda” (Figura 19), são apresentados tutoriais que guiam o aluno ou usuário novato pela ferramenta, ensinando as funcionalidades mais importantes que este disponibiliza. Cada card apresenta as etapas e como realizá-las para obter o resultado desejado.

Figura 19- Tela “Central de ajuda”

**LARA Domótica** Início Programar Fale conosco Central de ajuda Login Cadastre-se

**Programar dispositivos**  
 Programe um ou mais dispositivos  
 Nullam id dolor id nibh ultricies vehicula ut id elit. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus.  
[Ler mais →](#)

**Atividades**  
 Faça atividades e seja avaliado  
 Nullam id dolor id nibh ultricies vehicula ut id elit. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus.  
[Ler mais →](#)

**Variáveis de ambiente**  
 Modifique o ambiente em cada situação  
 Nullam id dolor id nibh ultricies vehicula ut id elit. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus.  
[Ler mais →](#)

**Colaboração nas atividades**  
 Realize atividades em grupo  
 Nullam id dolor id nibh ultricies vehicula ut id elit. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus.  
[Ler mais →](#)

**Ativar sensores**  
 Saiba como ativar cada sensor  
 Nullam id dolor id nibh ultricies vehicula ut id elit. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus.  
[Ler mais →](#)

Fonte: De autoria própria criado no Bootstrap Studio

Na área de login (Figura 20), é necessário inserir o nome de usuário e a senha. Caso o usuário tenha esquecido a senha é possível redefini-la.

Figura 20 - Tela de login

Clique aqui'." data-bbox="142 212 895 462"/>

Fonte: De autoria própria criado no Bootstrap Studio

Para fazer o cadastro (Figura 21), é necessário nome de usuário, email e senha. Todo novo usuário criado é por padrão definido como tipo aluno. Caso um professor queira modificar o tipo de sua conta deve entrar em contato com o administrador para que este a modifique.

Figura 21 - Tela de cadastro

Entrar'." data-bbox="185 657 850 892"/>

Fonte: De autoria própria criado no Bootstrap Studio

Logado em uma conta do tipo professor, o usuário poderá adicionar tarefas, ter o controle do histórico de tarefas, adicionar novas turmas e observar o progresso dos alunos (Figura 22).

Figura 22 - Tela “Área professor” na aba “Nova tarefa”

A interface de usuário da LARA Domótica apresenta uma barra de navegação superior com o nome do sistema e links para Início, Programar, Fale conosco, Central de ajuda, Área professor e Sair. Abaixo, há uma barra de abas com 'Nova tarefa' selecionada. O formulário principal, intitulado 'Adicionar tarefa', possui os seguintes elementos:

- Nome da tarefa:** Campo de texto com o placeholder 'Defina um nome para sua tarefa'.
- Descrição:** Área de texto grande com o placeholder 'Defina como a sua tarefa deve ser feita e o resultado final esperado'.
- ID da turma:** Campo de texto com o placeholder 'Turma que fará essa atividade'.
- Nota máxima:** Campo de texto com o placeholder 'Nota máxima da atividade'.
- Data de início:** Campo de data com o formato 'dd/mm/aaaa' e ícone de calendário.
- Data do fim:** Campo de data com o formato 'dd/mm/aaaa' e ícone de calendário.
- Em grupo:** Checkbox desativada.
- Material de apoio:** Botão 'Escolher arquivos' e o texto 'Nenhum arquivo escolhido'.

Um botão azul 'Publicar' está localizado no canto inferior direito do formulário.

Fonte: De autoria própria criado no Bootstrap Studio

Na aba “Nova tarefa” (Figura 22), o professor poderá criar uma nova atividade avaliativa definindo o nome, descrição da tarefa e as datas de início e fim. Além disso, é necessário informar o identificador da turma para que a atividade seja aplicada numa turma específica. Também podem ser adicionados materiais que ajudem os alunos na resolução do problema proposto. Caso a atividade seja em grupo é necessário marcar a *checkbox* “Em grupo” para revelar a área “Definir grupos” na qual é possível estabelecer a quantidade de alunos por grupo e como esses alunos serão divididos entre si.

Ao definir os grupos é possível adicionar e remover alunos de cada agrupamento, utilizando o botão de edição, respeitando o limite máximo do número de alunos definido no campo “Número de alunos” (Figura 23).

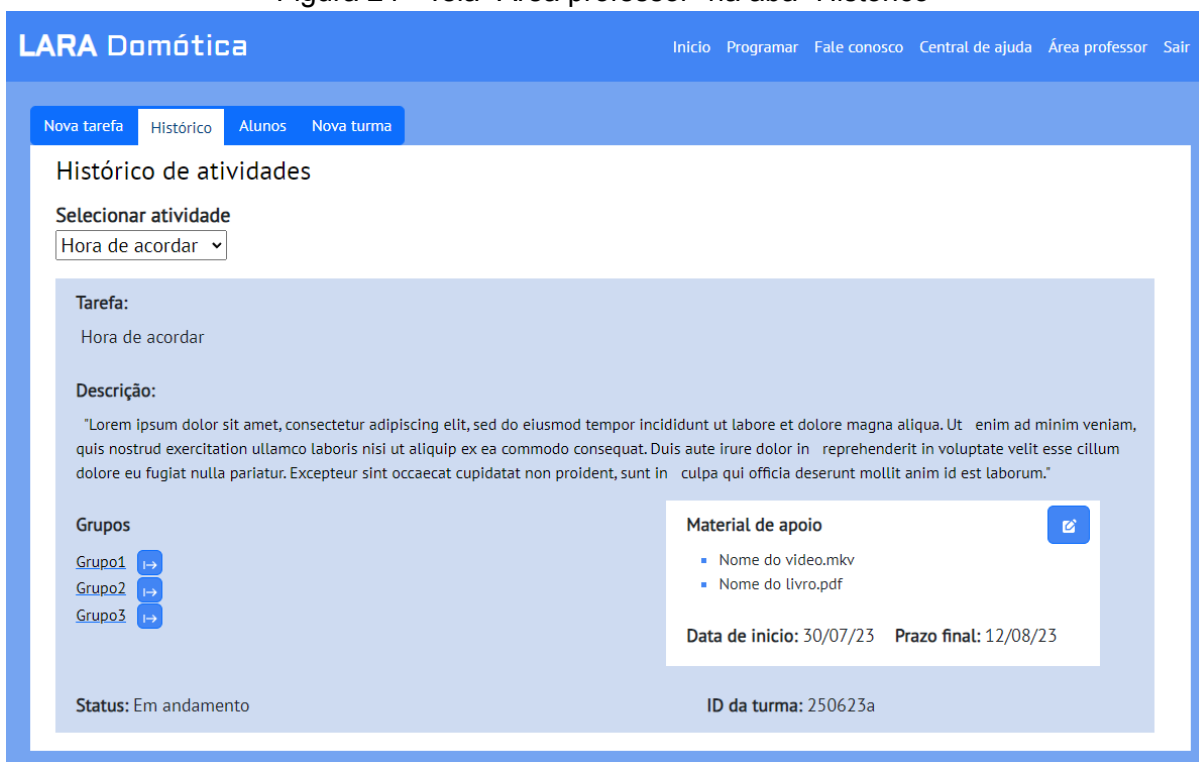
Figura 23 - Tela “Área professor” na aba “Nova tarefa” criando uma atividade em grupo

The screenshot displays a web interface for creating a group activity. At the top, there is a checked checkbox labeled "Em grupo". Below it, the "Material de apoio" section contains a button "Escolher arquivos" and the text "Nenhum arquivo escolhido". The main section is titled "Definir grupos" and includes a "Numero de alunos" input field with the value "4". A "Usuário" input field with the placeholder "Digite um usuário aluno válido" and an "Adicionar" button are also present. A pink error message states "Ainda existem alunos nessa turma que nao fazem parte de nenhum grupo." Below this, there are two lists of student names, each with a share icon and a plus icon to its left. The first list contains: 1. Joao.P, 2. Pedro.H, 3. Andressa.L, 4. Julia.M. The second list contains: 1. Marcelo.G, 2. Fernanda.L, 3. Maria.C, 4. Milena.D. A "Publicar" button is located at the bottom right of the interface.

Fonte: De autoria própria criado no Bootstrap Studio

Através do “Histórico de atividades”, o professor terá controle sobre o andamento dos alunos ao realizar as tarefas criadas por ele. Essa área tem o objetivo de manter o professor atualizado no progresso dos alunos enquanto eles desenvolvem as soluções propostas. Ao clicar em “grupo1”, por exemplo, o professor será redirecionado para a tela de programação onde poderá acompanhar a resolução de sua atividade por todo o grupo e também poderá participar de discussões no chat para auxiliar os alunos em caso de dúvida (Figura 24).

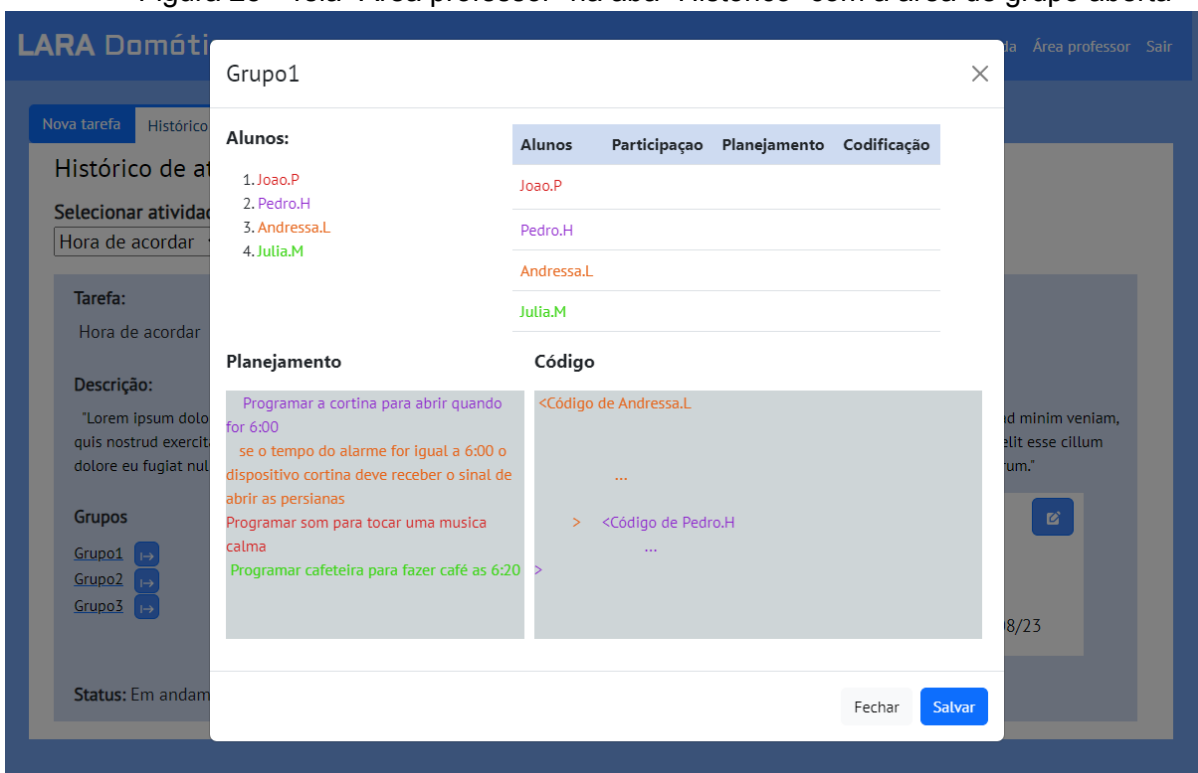
Figura 24 - Tela “Área professor” na aba “Histórico”



Fonte: De autoria própria criado no Bootstrap Studio

O botão com uma seta posicionado ao lado de cada grupo faz surgir a tela do grupo (Figura 25), na qual é possível observar o desenvolvimento do código de maneira simplificada e permite que o professor já avalie o trabalho. As cores dos nomes de cada aluno são as mesmas utilizadas no código para representar quais as contribuições de cada aluno em atividades em grupo, assim facilitando o processo de avaliação individual.

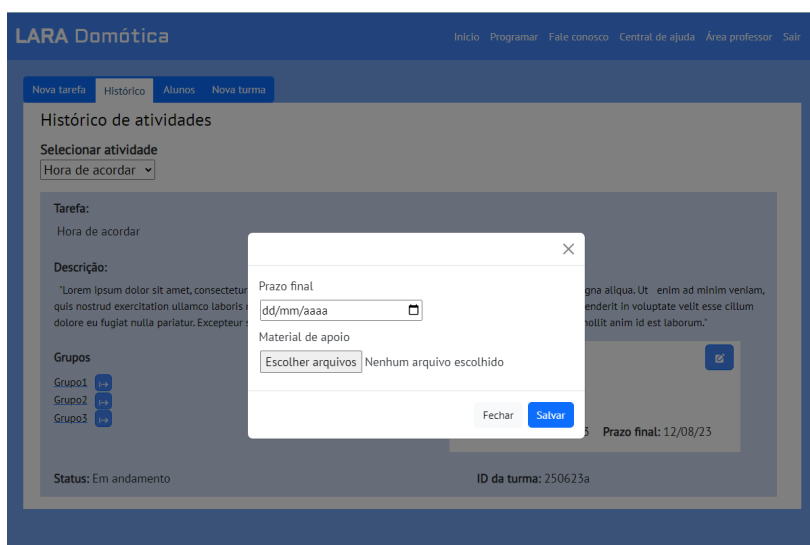
Figura 25 - Tela “Área professor” na aba “Histórico” com a área de grupo aberta



Fonte: De autoria própria criado no Bootstrap Studio

A aba “Histórico” também permite mudanças tanto em prazo da atividade como adições de materiais educativos que venham a auxiliar os alunos no desenvolvimento. Ao clicar no botão de edição, uma área aparece permitindo que o prazo final e os materiais de apoio possam ser modificados (Figura 26).

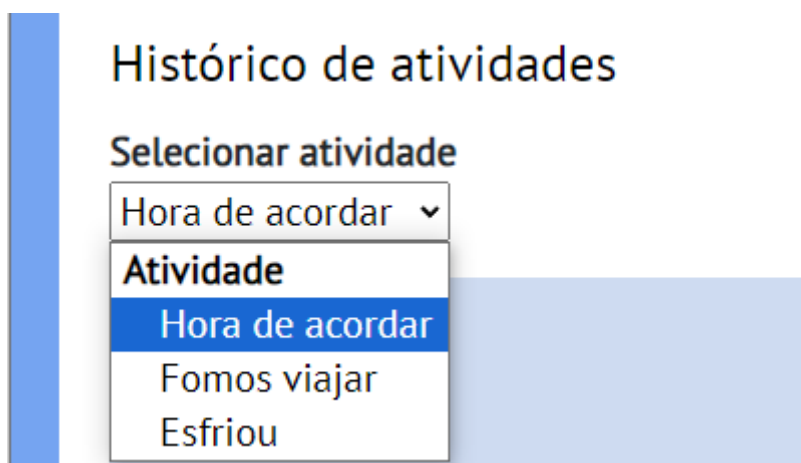
Figura 26 - Tela “Área professor” na aba “Histórico” com a área de edição aberta



Fonte: De autoria própria criado no Bootstrap Studio

Todas as atividades criadas pelo professor são adicionadas ao seletor da aba “Histórico” (Figura 27).

Figura 27 - Seletor de atividades criadas



Fonte: De autoria própria criado no Bootstrap Studio

Na aba “Alunos” (Figura 28), é possível observar o progresso de todos os alunos de uma turma por atividade. Em uma tabela são apresentados os alunos, o grupo que fizeram parte e suas notas em cada categoria. O botão de edição permite que o professor modifique as notas de cada aluno. Em seguida é só salvar .



Figura 28 - Tela “Área professor” na aba “Alunos”

**LARA Domótica** Início Programar Fale conosco Central de ajuda Área professor Sair

**Alunos**

Selecionar turma: 250623

Selecionar atividade: Hora de acordar

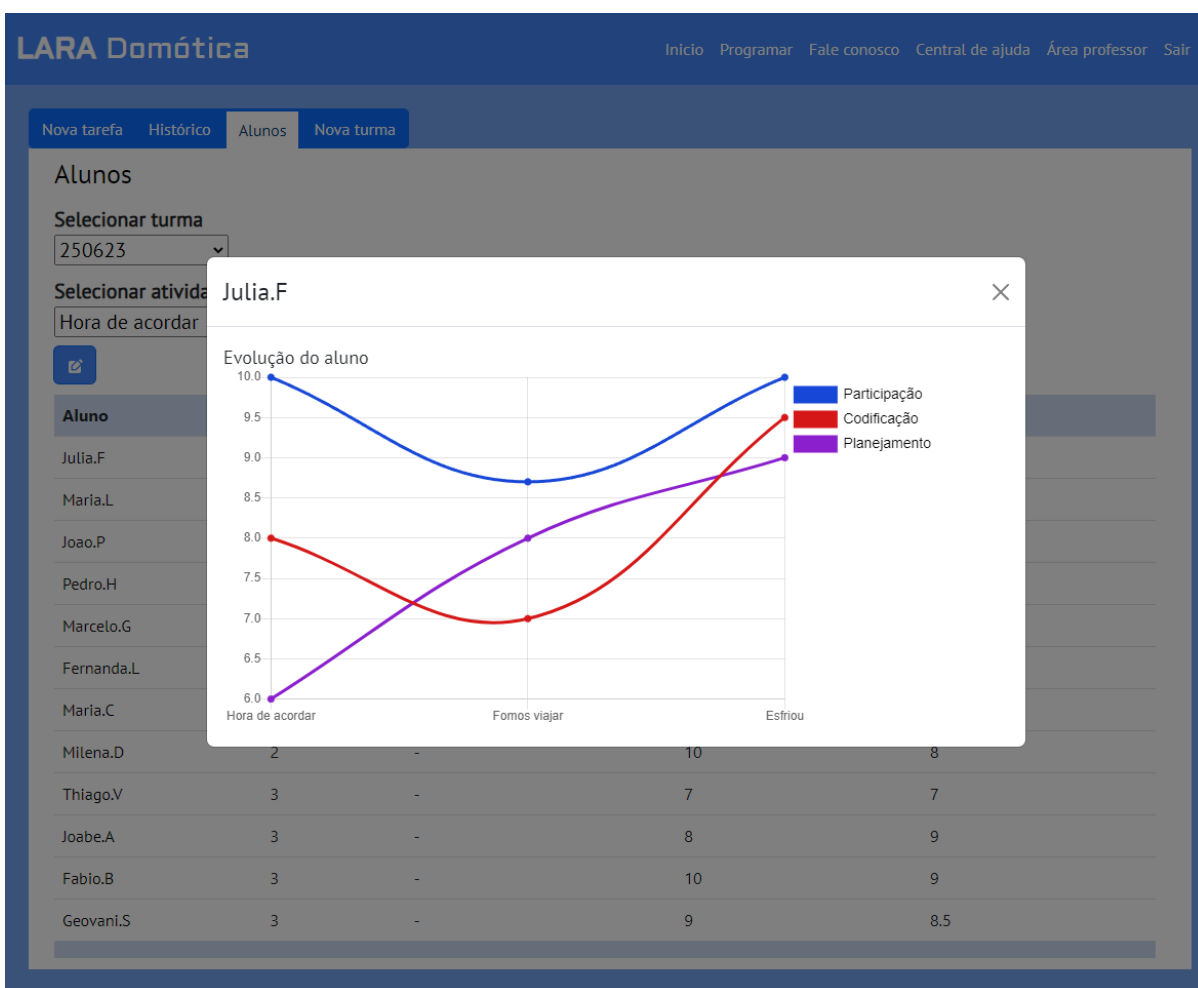
Aluno	Grupo	Planejamento	Participação	Codificação
Julia.F	1	-	10	8
Maria.L	1	-	8	8
Joao.P	1	-	6	7
Pedro.H	1	-	8	9
Marcelo.G	2	-	8	7
Fernanda.L	2	-	10	8
Maria.C	2	-	7.5	7.5
Milena.D	2	-	10	8
Thiago.V	3	-	7	7
Joabe.A	3	-	8	9
Fabio.B	3	-	10	9
Geovani.S	3	-	9	8.5

**Salvar**

Fonte: De autoria própria criado no Bootstrap Studio

O professor poderá selecionar um aluno para visualizar o seu progresso através de um gráfico que apresenta as notas desse aluno em cada atividade proposta por ele separado pelas categorias de participação, codificação e planejamento (Figura 29). O objetivo desse gráfico é deixar claro visualmente a evolução (ou decaimento) do aprendizado do aluno para que o educador possa adaptar ou manter sua estratégia de ensino.

Figura 29 - Tela “Área professor” na aba “Alunos” ao clicar no nome de um aluno



Fonte: De autoria própria criado no Bootstrap Studio

Na aba “Nova turma” (Figura 30), podem ser formadas novas turmas. Para isso, o professor deve inserir um nome de usuário válido no campo e pressionar o botão “Adicionar”. Assim os usuários adicionados são colocados no final da lista. Para remover um usuário da lista o professor deve clicar no botão que se encontra acima da lista e clicar nos alunos que deseja excluir da turma.

Figura 30 - Tela “Área professor” na aba “Nova turma”

LARA Domótica

Início Programar Fale conosco Central de ajuda Área professor Sair

Nova tarefa Histórico Alunos Nova turma

### Criar turma

ID da turma: 250623a

Usuário Digite um usuário aluno válido Adicionar

- 1. Joao.P
- 2. Pedro.H
- 3. Andressa.L
- 4. Julia.M
- 5. Marcelo.G
- 6. Milena.D
- 7. Maria.C
- 8. Fernanda.L
- 9. Thiago.V
- 10. Joabe.A
- 11. Fabio.B
- 12. Geovani.S

Criar turma

Fonte: De autoria própria criado no Bootstrap Studio

A área do aluno contém todas as atividades nas quais o aluno foi adicionado, sua descrição e prazo final de entrega (Figura 31). Além disso, é possível visualizar suas notas, seus colegas de grupo (se for uma atividade grupal), e os materiais de apoio disponíveis por atividade. Ao clicar no nome da tarefa o aluno é direcionado para a área de programação onde poderá conversar com os colegas de grupo e professor através do chat e desenvolver a resolução da atividade. Caso seja uma atividade individual o aluno só terá contato com o professor. Lembrando que todo esse processo de resolução pode ser acompanhado pelo professor.

Figura 31 - Tela “Área do aluno”

**LARA Domótica** Início Programar Fale conosco Central de ajuda Área do aluno Sair

### Histórico de atividades

Selecionar atividade  
 Hora de acordar ▾

**Tarefa:**  
[Hora de acordar](#)

**Descrição:**  
 "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

<b>Seu grupo:</b>	<ul style="list-style-type: none"> <li>▪ Pedro.H (você)</li> <li>▪ Joao.P</li> <li>▪ Maria.L</li> <li>▪ Julia.F</li> </ul>	<b>Sua nota:</b>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="padding: 5px;">Planejamento</th> <th style="padding: 5px;">Participação</th> <th style="padding: 5px;">Codificação</th> </tr> </thead> <tbody> <tr> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">6</td> <td style="text-align: center; padding: 5px;">9</td> </tr> </tbody> </table>	Planejamento	Participação	Codificação	3	6	9
Planejamento	Participação	Codificação							
3	6	9							

**Material de apoio:**

- Nome do video.mkv
- Nome do livro.pdf

**Status:** Finalizada **Data de inicio:** 30/07/23 **Prazo final:** 12/08/23

Fonte: De autoria própria criado no Bootstrap Studio

## 4.AVALIAÇÃO

Nessa seção serão apresentados o método avaliativo usado para validar a usabilidade do sistema, os colaboradores envolvidos e os resultados obtidos. Com base nos resultados foram apresentadas as mudanças feitas no sistema visando a melhora da interface.

### 4.1.Heurísticas de Nielsen

Para avaliar a interface foram utilizadas as 10 heurísticas de usabilidade de Nielsen (1994) que são um conjunto de diretrizes que podem ser aplicadas para avaliar uma interface quanto a experiência de uso. Essas diretrizes são como guias para designers e desenvolvedores que buscam melhorar a experiência do usuário e identificar problemas na interface do seu projeto. Em seu livro “*Usability Engineering*”, Nielsen explica como o processo avaliativo acontece:

Durante a sessão de avaliação, o avaliador percorre várias vezes a interface e inspeciona os vários elementos de diálogo e os compara com uma lista de princípios de usabilidade reconhecidos. Essas heurísticas são regras gerais que parecem descrever propriedades comuns de interfaces utilizáveis. Além da lista de verificação de heurísticas gerais a serem consideradas para todos os elementos de diálogo, o avaliador obviamente também pode considerar quaisquer princípios ou resultados de usabilidade adicionais que venham à mente e que possam ser relevantes para qualquer elemento de diálogo específico. (1994, tradução nossa)

Abaixo são descritas as 10 heurísticas de Nielsen:

- I. Visibilidade do sistema: Diz respeito à capacidade da interface em manter o usuário ciente da sua localização no sistema. A informação de onde o usuário se encontra, para onde ele pode ir e de onde ele veio devem estar claras na tela. Além disso, mensagens de erro e informações sobre carregamentos devem ser expostos para o usuário.
- II. Compatibilidade entre o sistema e o mundo real: Essa regra diz que é necessário que o sistema utilize uma linguagem que facilite a compreensão das informações, tanto em relação ao texto como aos ícones escolhidos. A comunicação do sistema deve ser clara e de fácil

entendimento para que se alinhe às expectativas dos conhecimentos prévios do usuário.

- III. Controle e liberdade para o usuário: Caso o usuário realize uma ação e se arrependa, é necessário que o sistema disponha de um modo que possibilite ao usuário revertê-la. Com isso, este se sentirá mais à vontade e confiante ao utilizar o sistema, sem medo de cometer erros graves, mesmo que seja uma ferramenta nova para ele.
- IV. Consistência e padronização: Pensando na melhor adaptação do usuário ao sistema se faz necessário a padronização das telas. Um design constante das telas facilita a navegação e a compreensão da aplicação.
- V. Prevenção de erros: A interface deve proteger o usuário de erros, como equívocos ou não entendimento de alguma funcionalidade. Para isso, o sistema deve dispor de mecanismos de confirmação e orientações claras.
- VI. Reconhecimento em vez de memorização: O sistema deve ser composto de elementos reconhecíveis que não dependam da memorização do usuário para utilizá-lo.
- VII. Eficiência e flexibilidade de uso: Para usuários mais experientes devem existir atalhos que permitam maior eficiência e rapidez no uso do sistema. Esses atalhos não devem comprometer a usabilidade de novos usuários.
- VIII. Estética e design minimalista: Para que a ferramenta transmita informações de maneira eficiente é necessário que o design seja minimalista, evitando elementos desnecessários que possam interferir na usabilidade do sistema.
- IX. Ajuda e documentação: É importante que exista uma documentação do sistema, mas a interface deve ser intuitiva para que o usuário não precise recorrer a esses recursos.
- X. Reconhecimento, diagnóstico e recuperação de erros: A interface deve informar ao usuário de maneira clara e direta, caso este cometa um erro, como corrigi-lo.

#### 4.1.1. Grau de severidade

Para analisar o sistema, caso exista falhas de usabilidade, Nielsen define graus de severidade para cada diretriz não satisfeita. Essas variam de 0 a 4 de acordo com a tabela abaixo:

Tabela 02

<b>Nível de severidade</b>	<b>Descrição</b>
0.Sem importância	Não afeta o uso da interface
1.Cosmético	Não há necessidade de solução imediata
2.Simples	Pode ser solucionado
3.Grave	Deve ser solucionado
4.Catastrófico	Muito importante ser solucionado

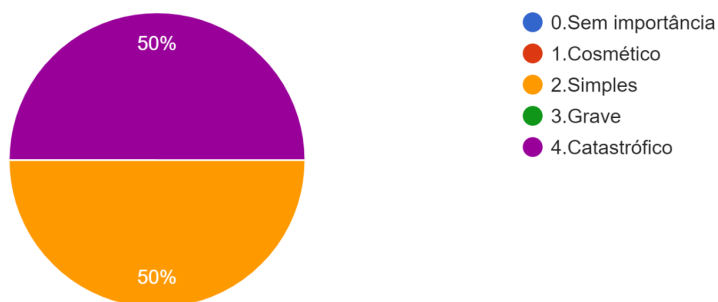
#### 4.1.2. Avaliadores

Os avaliadores são os responsáveis por utilizar o sistema com o objetivo de avaliá-lo utilizando as heurísticas de Nielsen. Para isso, caso encontrem erros de usabilidade no sistema, eles devem classificá-lo em alguma das heurísticas e definir um nível de severidade para o problema (descritas na tabela 2).

Esses avaliadores normalmente são desenvolvedores e designers. Em seu livro, Nielsen (1994) aponta que avaliadores com experiência na área de usabilidade realizam avaliações mais precisas pois são capazes de analisar melhor o sistema e assim identificar problemas que avaliadores sem experiência não encontrariam. Para a avaliação desse projeto foram convidados dois profissionais da área de tecnologia. A quantidade de avaliadores se justifica pela dificuldade para encontrar profissionais que possuam a experiência desejável na área de desenvolvimento e usabilidade. O primeiro avaliador é formado em Ciência da computação e trabalha como UX/UI designer, e o segundo também é formado em Ciência da computação e trabalha como desenvolvedor.

## 4.2. Resultado da avaliação

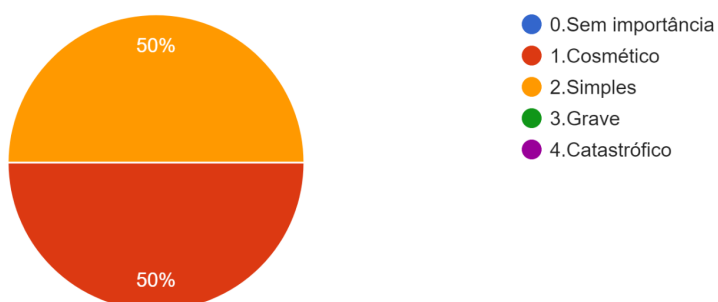
- Visibilidade do sistema:



Fonte: Pesquisa realizada pela autora

Nesse quesito o sistema obteve avaliações distintas: O primeiro avaliador apontou erros catastróficos quanto a falta da separação das atividades na área professor de acordo com o seu status (finalizada/em andamento) e com relação a corretude de cada uma delas (corrigida, em correção, não corrigida). Além disso, foi apontada a falta de um console que informe ao aluno os erros apresentados no seu processo de codificação. O segundo avaliador apontou um erro simples na clareza do sistema em informar se o usuário está logado.

- Compatibilidade do sistema com o mundo real:

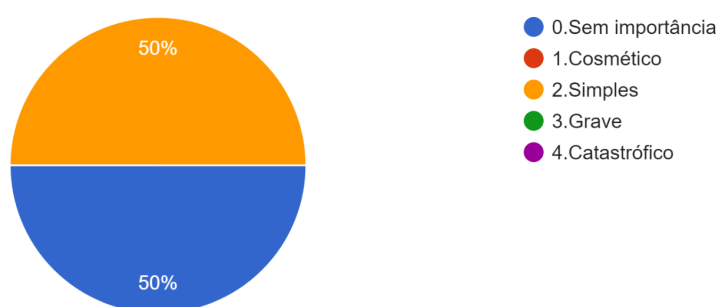


Fonte: Pesquisa realizada pela autora



O erro apontado pelo primeiro avaliador foi considerado simples e diz respeito ao termo utilizado no menu lateral da área de simulação. Para ela a palavra “clima” deve ser substituída por “tempo”, pois seu significado define melhor a funcionalidade proposta. Já o segundo avaliador sugere que as áreas “Fale conosco” e “Central de ajuda” sejam unificadas por trazerem ideias semelhantes de ajuda ao usuário. Este erro foi classificado como cosmético.

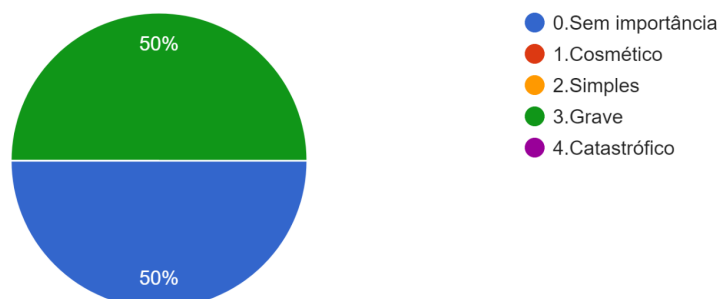
- Controle e liberdade para o usuário:



Fonte: Pesquisa realizada pela autora

Somente o primeiro avaliador encontrou um problema em relação a essa heurística. Classificado como um erro simples, o avaliador apontou que o sistema deveria permitir ao usuário professor a edição do ID da turma para que seja mais fácil de memorizá-lo.

- Consistência e padronização:

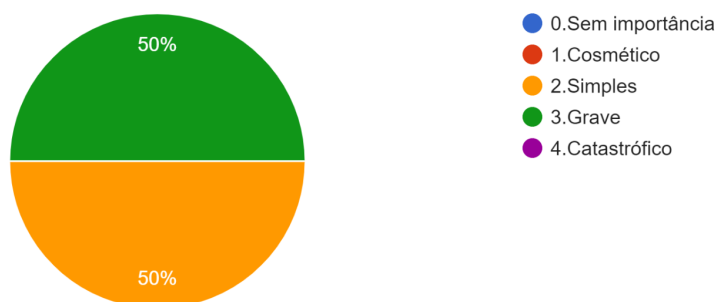


Fonte: Pesquisa realizada pela autora

De acordo com essa heurística, o primeiro avaliador apontou como um problema de severidade grave: “Em relação a linguagem usada, existem alguns

pontos que poderiam ser melhor padronizados, por exemplo, a utilização de termos no infinitivo ou seguindo a mesma conjugação verbal como nos casos de "Login" e "Cadastre-se", e "Começar" e "Ajude-me".

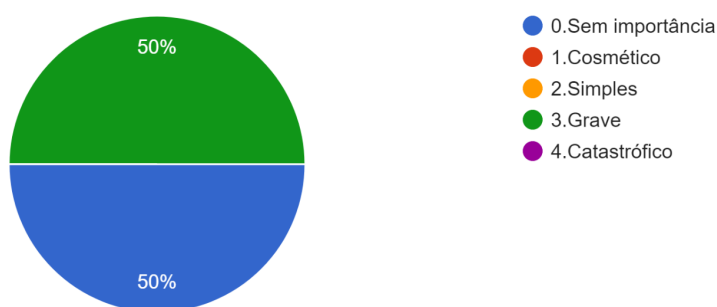
- Prevenção de erros:



Fonte: Pesquisa realizada pela autora

O primeiro avaliador relatou como grave a falta de indicação da obrigatoriedade do preenchimento de alguns campos nos formulários e a falta de um console para o aluno acompanhar o status da compilação de seu código. Já o segundo avaliador sugeriu que, na tela "Programar", a aba "Planejar" deveria vir primeiro em relação a aba "Programar". Esse último erro foi classificado como simples.

- Reconhecimento ao invés de memorização:

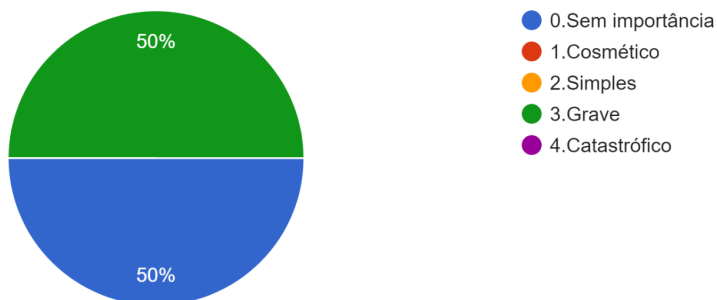


Fonte: Pesquisa realizada pela autora

O primeiro avaliador apontou que a falta de liberdade do usuário professor em não poder personalizar o ID da turma cria um problema grave: "Por conta de não ser possível alterar o ID da turma ou associar um nome mais intuitivo, ao gerenciar a

sua turma e as atividades o professor tem que fazer uma certo esforço cognitivo para lembrar o ID da turma.”

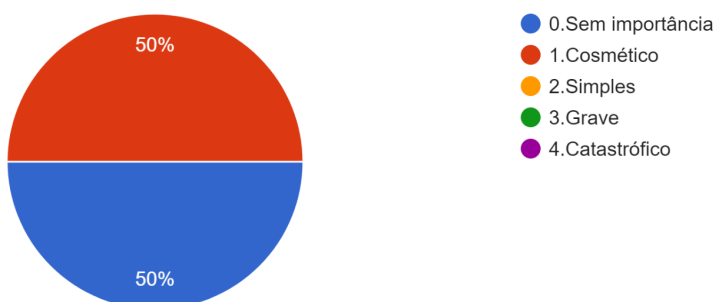
- Flexibilidade e eficiência de uso:



Fonte: Pesquisa realizada pela autora

Nessa heurística, o primeiro avaliador reforçou o problema antes citado sobre o ID da turma e sugeriu que seria interessante que o aluno fosse capaz de expandir as áreas de simulação, programação e planejamento. O primeiro erro foi classificado como grave.

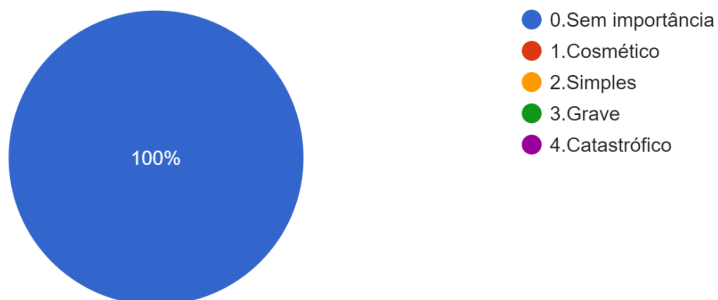
- Design estético e minimalista:



Fonte: Pesquisa realizada pela autora

Um erro cosmético foi encontrado pelo primeiro avaliador por falta de padronização de cores, espaçamentos, gráficos, tamanhos, alinhamentos e hierarquia de informações para melhorar a experiência do usuário, mas em suma a essência estética do sistema foi considerada minimalista.

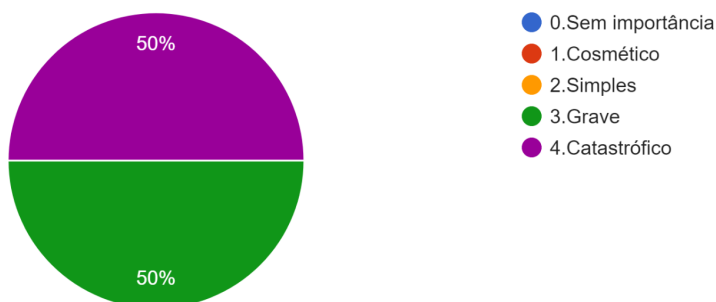
- Ajuda e documentação:



Fonte: Pesquisa realizada pela autora

Nessa categoria o sistema teve aprovação total dos participantes da avaliação. Nenhum comentário foi feito em relação a essa heurística.

- Reconhecimento, diagnóstico e recuperação de erros:



Fonte: Pesquisa realizada pela autora

Ambos avaliadores apontaram a falta de um terminal com o feedback dos erros na área de programação como um problema dessa categoria. O primeiro avaliador classificou esse problema como catastrófico e o segundo avaliador classificou como grave.

#### 4.3. Conclusões da avaliação

A avaliação foi feita com o objetivo de aperfeiçoar a interface do protótipo do sistema. Nielsen (1994) explica como o processo de prototipação é relevante no desenvolvimento do produto final:

Não se deve iniciar esforços de implementação em larga escala com base em projetos de interface iniciais de usuários. Em vez disso, a avaliação inicial de usabilidade pode ser baseada em protótipos dos sistemas finais que podem ser desenvolvidos muito mais rápido e muito mais barato, e que assim pode ser alterado muitas vezes até que uma melhor compreensão do design de interface do usuário seja alcançado (Tradução nossa).

A variedade do nível de severidade dos problemas encontrados foi grande, mostrando que, para esses avaliadores, o sistema atendeu o esperado em alguns pontos e outros ainda se fez necessário modificações. Os pontos bem avaliados apresentando erros sem importância, cosméticos ou simples foram relativos a compatibilidade do sistema com o mundo real, controle e liberdade do usuário, design estético e minimalista e ajuda e documentação. Foram encontrados erros graves e catastróficos em relação à visibilidade do sistema, consistência e padronização, prevenção de erros, reconhecimento ao invés de memorização, flexibilidade e eficiência de uso e reconhecimento, diagnóstico e recuperação de erros. A resolução dos problemas foi aplicada priorizando os erros considerados graves e catastróficos. Um resumo dos erros encontrados em cada heurística avaliada pode ser visto na tabela abaixo:

Quadro 01- Graus de severidade

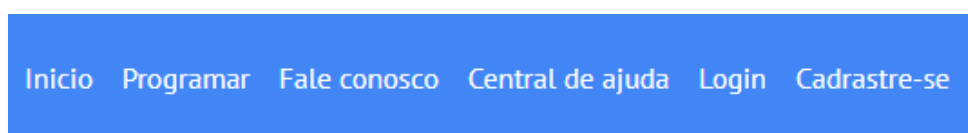
Heurísticas afetada	Grau de severidade do problema				
	Sem importância	Cosmético	Simples	Grave	Catastrófico
Visibilidade do sistema			1		1
Compatibilidade do sistema com o mundo real		1	1		
Controle e liberdade para o usuário	1		1		
Consistência e padronização	1			1	
Prevenção de erros			1	1	
Reconhecimento ao invés de memorização	1			1	

Flexibilidade e eficiência de uso	1			1	
Design estético e minimalista	1	1			
Ajuda e documentação	2				
Reconhecimento, diagnóstico e recuperação de erros				1	1

#### 4.4. Mudanças realizadas

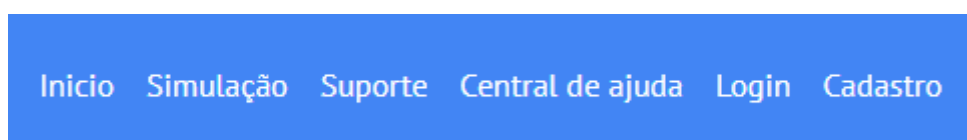
Após a avaliação da interface do sistema, mudanças foram realizadas considerando os problemas com níveis de severidade graves e catastróficos. O primeiro problema solucionado foi a falta de padronização no menu superior com relação a linguagem utilizada. Nas Figuras 32 e 33 abaixo é possível observar as mudanças nos termos empregados:

Figura 32 - Menu superior antes das modificações



Fonte: De autoria própria criado no Bootstrap Studio

Figura 33 - Menu superior depois das modificações

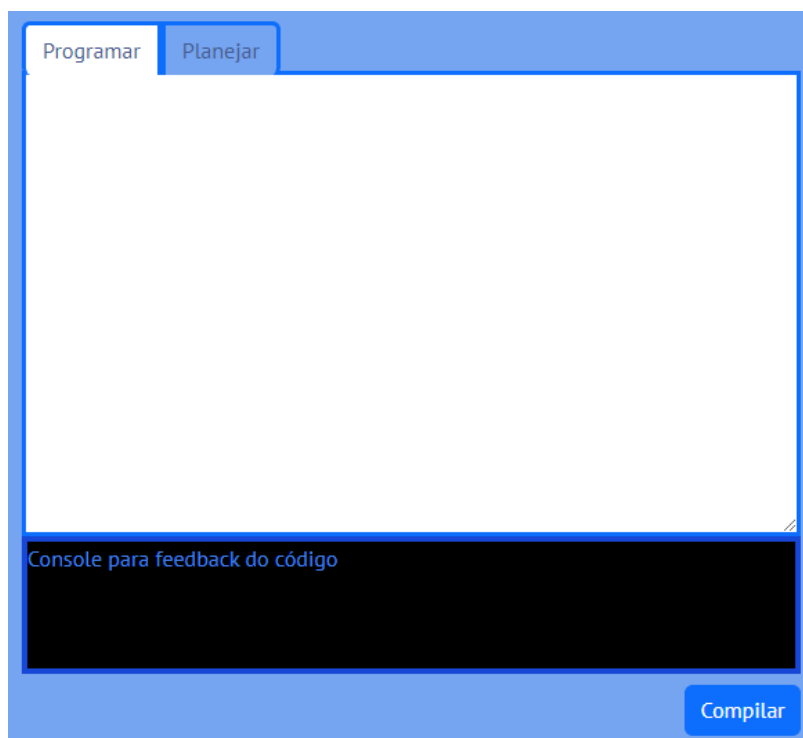


Fonte: De autoria própria criado no Bootstrap Studio

O link "Programar" foi substituído por "Simulação", o link "Fale conosco" foi substituído por "Suporte" e o link "Cadastre-se" foi substituído por "Cadastro".

A segunda mudança foi em relação ao console com o feedback do código na área de programação. Os avaliadores apontaram a falta dessa funcionalidade como problema que impossibilitaram a visibilidade do sistema, prevenção de erros e o reconhecimento, diagnóstico e recuperação de erros. Para solucionar esses problemas um console foi incrementado na área de programação (Figura 34).

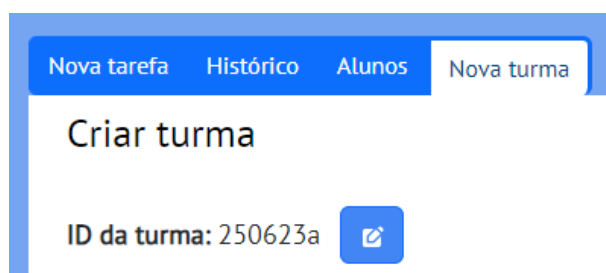
Figura 34 - Console adicionado a área de programação



Fonte: De autoria própria criado no Bootstrap Studio

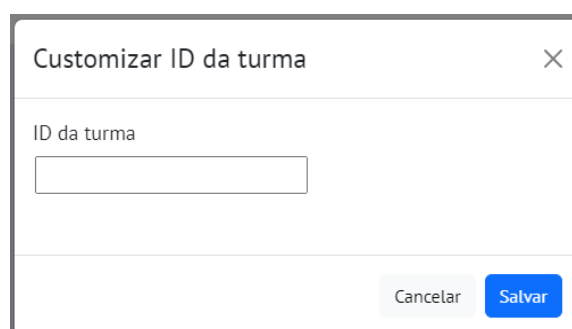
Outro problema identificado foi a falta da opção de personalizar o identificador da turma. As heurísticas afetadas por esse erro foram a flexibilidade e eficiência de uso, reconhecimento ao invés de memorização e controle e liberdade para o usuário. Como solução foi implementado um botão de edição que abre uma caixa onde é possível que o professor defina um novo identificador para sua turma. Caso o professor não queira personalizar o ID, o sistema terá uma opção já definida como padrão (Figuras 35 e 36).

Figura 35 - Botão de edição do nome da turma



Fonte: De autoria própria criado no Bootstrap Studio

Figura 36 - Edição do nome da turma



Fonte: De autoria própria criado no Bootstrap Studio

A modificação seguinte teve o objetivo de corrigir a falta de indicação nos formulários sobre quais campos são obrigatórios. Essa falta atrapalhava o usuário na prevenção de erros. Para isso foram adicionados alertas em cada campo obrigatório não preenchido ao apertar o botão de enviar (Figura 37).

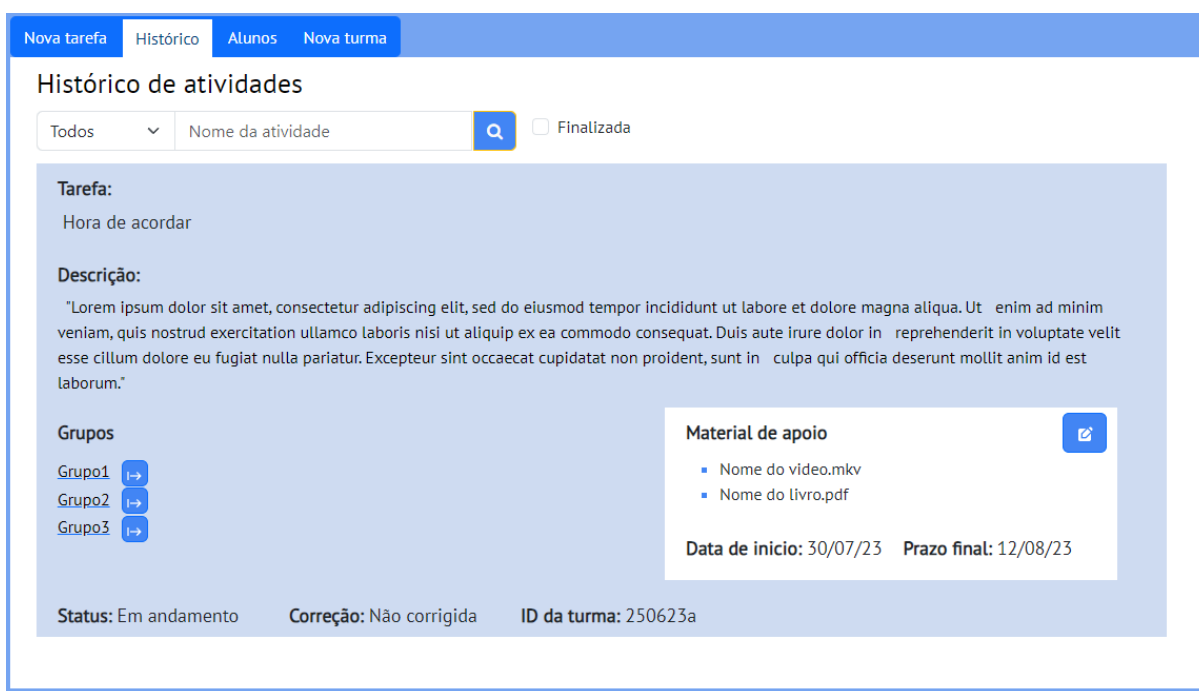
Figura 37 - Mensagens informando os campos obrigatórios

Fonte: De autoria própria criado no Bootstrap Studio



O último problema encontrado foi em relação a aba “Histórico”, na área do professor, por não permitir ao usuário filtrar as atividades em categorias como: corrigidas, não corrigidas, em correção, finalizadas e em andamento. Este problema interfere na visibilidade do sistema. Para tal foram modificados os mecanismos de busca de atividades nessa aba. O usuário poderá definir o nível de correção da atividade, pesquisar pelo nome da atividade e marcar se deseja ver atividades finalizadas. É possível fazer a busca utilizando todas essas funcionalidades ou apenas alguma delas. Se o usuário esquecer o nome da atividade, por exemplo, será possível encontrá-la utilizando outro atributo como status (em andamento ou finalizada) e correção (corrigida, em correção ou não corrigida) (Figura 38).

Figura 38 - Pesquisa de atividades



Fonte: De autoria própria criado no Bootstrap Studio

## 5.CONCLUSÃO

Os laboratórios virtuais surgem como alternativas educacionais ao padrão utilizado nas salas de aula. Estes possuem vantagens como o fato de serem financeiramente econômicos e permitirem o aprendizado a distância e de maneira lúdica. Com ele é possível utilizar equipamentos de difícil acesso de maneira segura para o aluno aonde quer que ele esteja. Além disso, com a adição de ferramentas colaborativas é possível aumentar o interesse pelo aprendizado.

O trabalho apresentado teve o objetivo de desenvolver um protótipo de laboratório virtual conceituado em Domótica para ensino de programação. A dificuldade no ensino de programação motivou a criação deste trabalho. Os alunos encontram desafios no processo de aprendizagem de programação e tendem a desistir. Além disso, os professores, por causa do grande número de alunos, não conseguem manter uma análise do nível de conhecimento de cada um.

A justificativa para o desenvolvimento deste trabalho foi a necessidade de um sistema educacional a distância que tenha ferramentas de colaboratividade e permita o aprendizado interativo. A ferramenta também pretendeu facilitar o trabalho dos professores da área de tecnologia ao disponibilizar um sistema de avaliação e acompanhamento do progresso dos alunos.

Durante o desenvolvimento do projeto foi utilizada a metodologia DSR dividida em 6 atividades definidas por Peffers (2007) para a confecção de um artefato, cuja finalidade é a demonstração do saber científico em forma de um produto que contribua na resolução do problema apresentado.

Inicialmente foram abordados conceitos necessários na implementação do protótipo e para a compreensão deste trabalho. Como forma de analisar o cenário atual foram selecionados trabalhos relacionados ao tema do protótipo proposto. Através da análise desses trabalhos e da idealização do produto foi feito o levantamento de requisitos necessários para o desenvolvimento deste, visando certas características desejáveis. Cada trabalho relacionado tinha uma proposta igualmente relevante e os pontos mais importantes na visão do autor foram mantidos na construção deste trabalho.

A prototipação se baseou em sua proposta de ensinar programação utilizando uma simulação de Domótica contida em um laboratório virtual. A motivação do projeto vem do fato de que programação é considerada uma matéria de assimilação

difícil e o método de ensino passivo não é considerado ideal. O sistema proposto vem como uma forma de transformar a obtenção das habilidades necessárias de uma maneira mais ativa e lúdica.

Ao definir a área de interação do laboratório virtual foi priorizada uma interface minimalista e simples de ser utilizada buscando a melhor experiência do usuário e a eficácia do produto. Com isso o sistema foi planejado, visando características como facilidade de uso, disponibilidade web, colaboração, atratividade, entre outros. Então foram criadas as regras de negócio, fluxo de interações e as telas do sistema.

Por fim, uma avaliação de usabilidade foi realizada utilizando as 10 Heurísticas Nielsen (1994), que tem o objetivo de encontrar problemas que possam interferir na usabilidade do sistema. A partir dos resultados encontrados foram aplicadas mudanças na interface para corrigir os problemas mais graves.

Uma mudança realizada foi a adição de um console na área de programação que informa ao aluno os erros de código. A falta desse elemento causava problemas com relação a flexibilidade e eficiência de uso, reconhecimento ao invés de memorização e controle e liberdade para o usuário.

Outro problema encontrado foi com relação aos termos utilizados no menu superior que interferiram na consistência e padronização da interface. A sua solução se deu de maneira simples ao modificar os termos utilizados.

O sistema também limitava o usuário professor ao não permitir a edição do nome da turma, afetando a flexibilidade e eficiência de uso, reconhecimento ao invés de memorização e controle e liberdade para o usuário. Essa funcionalidade foi adicionada.

A falta de indicação da obrigatoriedade de alguns campos foi identificado como um problema que impedia o usuário de se prevenir de erros. Avisos em campos obrigatórios não preenchidos foram adicionados.

Uma última mudança foi realizada no intuito de facilitar a busca por atividades no histórico do professor. Antes da correção, o método de busca prejudica a visibilidade do sistema. A solução encontrada foi aplicar mais filtros no processo de busca.

A importância da avaliação na construção do protótipo se dá na possibilidade de aprimorar os pontos fracos do sistema de modo que consiga atingir os objetivos

esperados. O protótipo de laboratório desenvolvido priorizou a colaboratividade, o aprendizado interativo, a facilidade de uso e o auxílio aos professores da área.

É necessário refletir sobre as possibilidades em relação a trabalhos futuros. O desenvolvimento do sistema através do protótipo apresentado, seus testes e aplicação em universidades pode ser uma oportunidade de facilitar o ensino de programação. Em última análise, o enriquecimento do saber nessa área demanda um compromisso com a busca por inovações e adaptações buscando sempre o melhor cenário educacional possível, todos os quais são aspectos fundamentais a serem considerados em trabalhos futuros.

## REFERÊNCIAS:

Como funciona um sensor de movimento?. **Positivo Casa inteligente**, 2022.

Disponível em:

<https://blog.positivocasainteligente.com.br/sensor-de-movimento-como-funciona/>.

Acesso em: 2 de abr. de 2023

Como funcionam os sensores de movimento ativo e passivo e onde podem ser aplicados. **Segurança eletrônica**. Disponível

em:<https://revistasegurancaeletronica.com.br/como-funcionam-os-sensores-de-movimento-ativo-e-passivo-e-onde-podem-ser-aplicados/>. Acesso em: 12 de mar. de 2023.

KETTERL, M. et al. Tema 2: Open Roberta - A Web Based Approach to Visually Program Real Educational Robots. **Tidsskriftet Læring og Medier (LOM)**, v. 8, n. 14, 25 dez. 2015. Disponível em: <https://tidsskrift.dk/lom/article/view/22183/20134>. Acesso em: 13 de abr. de 2023.

Detector de fumaça: O que é e como funciona?. **Dimensão incêndio**, 2019.

Disponível em: <https://dimensaoincendio.com.br/detector-de-fumaca/>. Acesso em: 3 ago. 2023.

BEZERRA, A. et al. **Residential automated irrigation system**. In: CONGRESSO INTERNACIONAL DE ENGENHARIA MECÂNICA, 25., 2019, Uberlândia. Anais eletrônicos[...]. Disponível em:

<https://repositorio.ufrn.br/bitstream/123456789/43014/4/Final%20manuscrit%20Residencial%20automated%20irrigation%20system.%20COBEM.pdf>. Acesso em: 3 ago. 2023.

SIROTHEAU, S. et al. **LabPy: Laboratório virtual de ensino em Python**. Anais dos Workshops do VII Congresso Brasileiro de Informática na Educação (CBIE 2018). Anais[...]. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC), 2018.

TÓFOLI, R. J. **Casa inteligente – Sistema de automação residencial**. TCC (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas)-Instituto Municipal de Ensino Superior de Assis-IMESA e Fundação Educacional do Município de Assis - FEMA. Município de Assis, p.74.2014.

GOKHALE, A. A. **Collaborative learning enhances critical thinking**. Journal of technology education, v. 7, n. 1, 1995.

NASCIMENTO, D. R.; FETTERMANN, D. Quais os fatores que interferem na aceitação das tecnologias da casa inteligente pelo usuário: uma revisão. **Produto & Produção**, v. 22, n. 3, 2021.

MONTEIRO, Alanna; LEAL, Vivianne; GOMES, Reinaldo. Requisitos Não Funcionais no Desenvolvimento de Software Educacional: Um Estudo Utilizando Grounded Theory. **Nuevas Ideas en Informática Educativa TISE 2013**, Fortaleza, p.

759-762,2013. Disponível em: <https://www.tise.cl/volumen9/TISE2013/759-762.pdf>. Acesso em: 15 jun.2023

ANDRÉ, B. P. O lugar da didática no ambiente virtual de aprendizagem. **Revista brasileira de ensino de ciência e tecnologia**, v. 7, n. 3, 2014.

KENSKI, Vani Moreira. **Tecnologias e ensino presencial e a distância**. 4.ed. Campinas: Editora Papyrus, 2003. p.157

JENKINS, Tony. **On the difficulty of learning to program**. Proceedings of the 3rd Annual LTSN\_ICS Conference, Loughborough University, United Kingdom, 53-58

PIMENTEL, Mariano; FILIPPO, Denise; SANTORO, Flávia Maria. **Design Science Research: fazendo pesquisas científicas rigorosas atreladas ao desenvolvimento de artefatos computacionais projetados para a educação**. In: JAQUES, Patrícia Augustin; PIMENTEL, Mariano; SIQUEIRA, Sean; BITTENCOURT, Ig. (Org.) Metodologia de Pesquisa Científica em Informática na Educação: Concepção de Pesquisa. Porto Alegre: SBC, 2020. (Série Metodologia de Pesquisa em Informática na Educação, v. 1) Disponível em: <https://metodologia.ceie-br.org/livro-1/>. Acesso em: 18 jun.2023

A RELAÇÃO TEORIA E PRÁTICA NO PROCESSO DE FORMAÇÃO DOCENTE. **Revista de Pesquisa Interdisciplinar**, Cajazeiras, n. 2, suplementar, p. 332- 340, set. de 2017

KEEKER, Kevin. **Improving Web Site Usability and Appeal**.1997. p 13. Disponível em: <http://jepelet.free.fr/studies/MBA/design/s4/lectures/Improving%20Web%20Site%20Usability%20and%20Appeal.pdf>. Acesso em : 10 jul. 2023

MOROZOV, Mikhail; TANAKOV, Andrey; GERASIMOV, Alexey; BYSTROV, Dmitry; CVIRCO, Eduard (2004). **Virtual Chemistry Laboratory for School Education**. In: Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT'04), Finland

BALIEIRO, Renan. **LABORATÓRIO DE PROGRAMAÇÃO EM PYTHON: LABPY**. Orientador:Eloi Luiz Favero.2018.86 f.TCC(Graduação)- Ciências da Computação, Universidade Federal do Pará, Belém. 2018. Disponível em: [https://bdm.ufpa.br:8443/jspui/bitstream/prefix/2518/1/TCC\\_LaboratorioProgramacaoPython.pdf](https://bdm.ufpa.br:8443/jspui/bitstream/prefix/2518/1/TCC_LaboratorioProgramacaoPython.pdf).Acesso em:20 abril 2023

NIELSEN,J.**Usability Engineering**.San Diego:Elsevier Science & Technology Books,1994.

NIELSEN, Jakob. 10 Usability Heuristics for User Interface Design. Disponível em: <https://www.nngroup.com/articles/ten-usability-heuristics/#poster>. Acesso em: 28 jun.2023

GOMES, Anabela; AREIAS, Cristiana; HENRIQUES, Joana; MENDES, António. Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. **Revista portuguesa de pedagogia**, n. 42-2, p. 161-179, 2008. Disponível em: [https://impactum-journals.uc.pt/rppedagogia/article/view/1647-8614\\_42-2\\_9/690](https://impactum-journals.uc.pt/rppedagogia/article/view/1647-8614_42-2_9/690). Acesso em: 12 jul.2023

SILVA, J. B. DA, et al.. Laboratórios Remotos como Alternativa para Atividades Práticas em Cursos na Modalidade EaD. **EaD em Foco**, v. 10, n. 2, e810, 2020.

Hoss, Alessandro, Marcelo da S. Hounsell & André B. Leal (2009), **Virbot4u: Um simulador de robô usando x3d**, I Simpósio de Computação Aplicada, Passo Fundo - RS: SBC pp. 1–15.

Jonassen, David (1996), O uso de novas tecnologias na educação a distância e a aprendizagem construtivista, **Em Aberto**, Brasília, ano 16, n.70, abr/jun.1996.

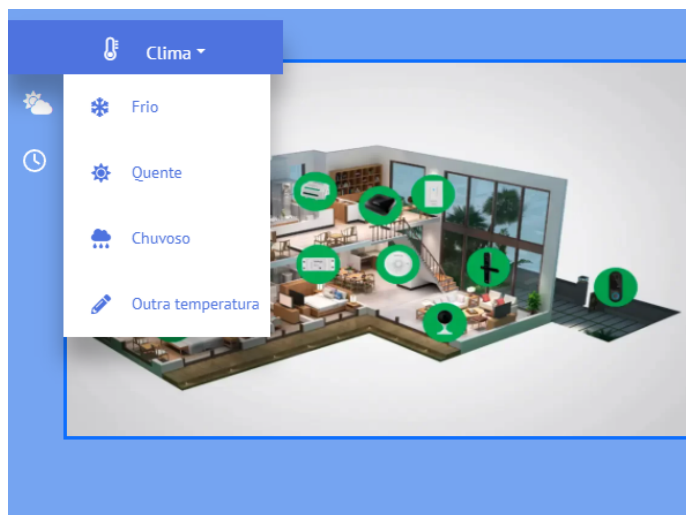
BRAGA, Marilluci. Realidade Virtual e Educação. **Revista de Biologia e Ciências da Terra**, Paraíba, v. 1, n. 1, 2001. Disponível em: <https://www.redalyc.org/articulo.oa?id=50010104>

DAM, N. V. **The E-Learning Fieldbook: Implementation Lessons and Case Studies from Companies that are Making e-Learning Work: McGraw-Hill**. 2004. 337 p.

MOORE, M. G. Three Types of Interaction. **American Journal of Distance Education**, v.3, n.2, Fev. 1989.

## APÊNDICE A - TELA DE PROGRAMAÇÃO E SIMULAÇÃO EVIDENCIANDO O MENU LATERAL

### Opções de clima



### Opções de luminosidade



### Opções de horário



