



UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA - UESB  
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS - DCET  
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

LUCAS DIAS AMARAL

**LaraApp:  
APLICATIVO MÓVEL PARA ENSINO DE PROGRAMAÇÃO UTILIZANDO  
LABORATÓRIO REMOTO**

**Vitória da Conquista - BA  
Agosto de 2019**

LUCAS DIAS AMARAL

**LaraApp:  
APLICATIVO MÓVEL PARA ENSINO DE PROGRAMAÇÃO UTILIZANDO  
LABORATÓRIO REMOTO**

Trabalho de Conclusão de Curso  
submetido à Universidade Estadual do  
Sudoeste da Bahia – UESB, como parte dos  
requisitos necessários para a obtenção do  
Grau de Bacharel em Ciência da  
Computação.

Orientadora: Prof.<sup>a</sup> Dra. Maísa Soares  
dos Santos Lopes

**Vitória da Conquista - BA**

**Agosto de 2019**

## RESUMO

O presente trabalho apresenta o desenvolvimento de um aplicativo para dispositivos móveis voltado para o suporte no ensino e aprendizagem de programação, por meio do qual é possível manipular um robô móvel remoto através de algoritmos e visualizar a execução dos códigos em tempo real. A Experimentação Remota Móvel é um conceito que vem crescendo como ferramenta pedagógica de ensino, capaz de auxiliar professores na transmissão de conteúdo e encurtar a distância entre disciplinas complexas e a realidade do aluno, que vivencia uma era tecnológica. O processo de software utilizado durante o levantamento de requisitos e desenvolvimento do aplicativo foi o SCRUM Solo. Na implementação foram utilizados HTML, CSS3, JavaScript e a framework Apache Cordova. Por fim, um método de avaliação heurística de usabilidade foi utilizado na avaliação da aplicação, que apresentou alta usabilidade.

Palavras-chave: Laboratório Remoto; Aprendizagem Móvel; Desenvolvimento Híbrido; Open-source; Usabilidade Mobile.

## **ABSTRACT**

This paper presents the development of an application for mobile devices focused on support in programming teaching and learning, through which it is possible to manipulate a remote mobile robot through algorithms and visualize the execution of codes in real time. Mobile Remote Experimentation is a growing concept as a pedagogical teaching tool, capable of assisting teachers in the transmission of content and shortening the distance between complex subjects and the reality of the student, who experiences a technological age. The software process used during the requirements gathering and application development was SCRUM Solo. In the implementation were used HTML, CSS3, JavaScript and the Apache Cordova framework. Finally, a heuristic usability evaluation method was used to evaluate the application, which presented high usability.

Keywords: Remote laboratory; Mobile learning; Hybrid development; Open-source; Mobile Usability.

## **AGRADECIMENTOS**

Aos meus pais, pelo suporte incondicional, que mesmo nos momentos mais difíceis, nunca desistiram de me apoiar.

A minha avó, pelo sorriso e benção a cada despedida.

Ao amigo e professor Willian Silva, pelos ensinamentos durante a fase pré-vestibular.

Ao meu amigo Deli, e aos professores Alexsandra Andrade e Roque Mendes, por terem me ajudado com uma moradia nos momentos que não pude arcar com os custos de aluguel.

Aos amigos que fiz no curso, com quem dividi os momentos felizes e as dificuldades da universidade e da vida. E aos familiares desses amigos, que me trataram como da família.

A professora Máisa Soares, por mostrar-se sempre disponível e paciente enquanto orientadora deste trabalho.

*“Frequentemente a imaginação nos transporta a mundos que nunca existiram, mas sem ela não vamos a parte alguma. ”*  
*(Carl Sagan)*

## **LISTA DE SIGLAS**

AVA - Ambiente Virtual de Aprendizagem

CSS3 - Cascading Style Sheets

DSR - Design Science Research

HTML5 - HyperText Markup Language

INEP - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

LARA - Laboratório Remoto em AVA

MRE - Mobile Remote Experimentation

PDA - Personal Digital Assistant

TI - Tecnologia da Informação

UML - Unified Modeling Language

## **LISTA DE TABELAS**

Tabela 1 – Resultados da avaliação do LaraApp pelo checklist.....37

Tabela 2 – Problemas, correções propostas e prioridade para correção .....38



## LISTA DE FIGURAS

Figura 1 - Interação com experimento remoto a partir do aplicativo.....	19
Figura 2 - Interface do aplicativo e caminhos lógicos.....	21
Figura 3 - Interface com possibilidade do jogo.....	22
Figura 4 - Diagrama de casos de uso.....	24
Figura 5 - Protótipos das telas.....	26
Figura 6 - Screenshot da tela de programação.....	29
Figura 7 - Screenshot da barra lateral.....	30
Figura 8 - Screenshot da tela de códigos do usuário.....	31
Figura 9 - Screenshot da tela de reservar sessão.....	32

# SUMÁRIO

<b>INTRODUÇÃO</b>	11
<b>2. EMBASAMENTO TEÓRICO</b>	14
2.1 Aprendizagem Móvel (Mobile Learning)	14
2.2 Experimentação Online	14
2.2.1 Experimentação Virtual (Laboratório Virtual)	15
2.2.2 Experimentação Remota	15
2.2.3 Experimentação Remota Móvel	17
<b>3. TRABALHOS RELACIONADOS</b>	19
3.1 Desenvolvimento de aplicação para interfaceamento com experimentos remotos por smarthphones	19
3.2 Aplicação da Robótica no Ensino-Aprendizagem de Lógica de Programação para Crianças	20
3.3 Aplicativo Lúdico-Pedagógico para Ensino de Programação e Robótica Educacional	21
<b>4. LaraApp</b>	24
4.1 Implementação do LaraApp	27
4.2 Apresentação do LaraApp	28
<b>5. VALIDAÇÃO DO LaraApp</b>	33
5.1 Avaliação Heurística	33
5.2 Avaliação Heurística aplicada ao LaraApp	34
5.3 Resultado da Avaliação	36
<b>6. CONCLUSÃO</b>	40
<b>7. REFERÊNCIAS BIBLIOGRÁFICAS</b>	41

## INTRODUÇÃO

A experimentação é de grande importância para despertar o interesse de alunos pela ciência e facilitar a compreensão de seus fenômenos, uma vez que, sua utilização encurta a distância entre a teoria e a realidade vivida por cada um, favorecendo desse modo, a absorção dos conceitos científicos. Ainda assim, são poucas as escolas e universidades que podem contar com laboratórios convencionais e infraestrutura para comportar experimentos com qualidade, em virtude do alto custo para construção e manutenção dos mesmos.

Dados do último Censo Escolar do Inep (2018) apontam que, enquanto o acesso à internet é uma realidade em 95,1% das escolas de ensino médio, o laboratório de ciências, é encontrado em apenas 44,1% delas. Com relação às universidades, em 2018, a verba relativa a investimento, aquisição de bens, equipamentos e obras de expansão e reestruturação das instituições sofreu cortes, chegando a, aproximadamente, 10% daquela destinada às instituições em 2014 (SANTOS e SOARES, 2017). Como consequência da falta de recursos, as universidades se veem sujeitas a fechar seus laboratórios, que por vezes, se encontram sucateados por conta da falta de recursos.

A Experimentação Online desponta como uma alternativa de baixo custo perante a escassez de laboratórios convencionais nas escolas e universidades. Seu conceito compreende os laboratórios virtuais e remotos, favorecido por ferramentas online baseadas em realidade virtual, realidade aumentada, dispositivos sensoriais, vídeos ao vivo, vídeos interativos e jogos que promovem a imersão do usuário em ambientes virtuais (RESTIVO e CARDOSO, 2013). Tal concepção visa recriar experiências reais, seja por meio de simulações ou acesso remoto, de modo que, o aluno possa visualizar a aplicação da teoria e usufruir dos benefícios da experimentação.

O desenvolvimento da Experimentação Online é muito significativo, visto que, matérias que costumam ser obstáculos para professores e alunos, como é o caso das disciplinas introdutórias que envolvem algoritmos e programação em cursos superiores e técnicos, devido a sua complexidade, demandam por suporte experimental, uma vez que, exigem o desenvolvimento de estratégias de solução de problemas com base lógico-matemática, que para muitos são altamente abstratas e

distantes do “cotidiano” (RAABE e SILVA, 2005). Os sistemas visuais, como é o caso da experimentação e dos simuladores, apresentam um grande potencial como ferramenta de ensino da programação, pois, algoritmos computacionais podem ser pouco claros quando apresentados em formato textual (GOMES 2010).

Considerando as dificuldades enfrentadas por alunos iniciantes em cursos de graduação que envolvem computação e informática, o LARA - Laboratório Remoto em Ambiente Virtual de Aprendizagem, disponibiliza experimentos remotos de robótica online, com o intuito de auxiliar o ensino de programação (LOPES et al., 2016). Utilizando uma IDE online, usuários do LARA podem programar o LARA Remote Robot (L1R2), e em seguida, assistir em tempo real a execução dos comandos programados. As principais funções do L1R2 são seguir linha e desviar de obstáculos, entretanto, o robô possui sensores e atuadores que ampliam as possibilidades durante a experimentação.

Como pode ser visto, os avanços tecnológicos vêm provocando profundas mudanças no âmbito dos processos de ensino e aprendizagem. Um fator que tem contribuído para isto é a ascensão dos dispositivos móveis, sobretudo dos smartphones, que independente de algumas limitações, apresentam vantagens como: acessibilidade, conectividade, comodidade e portabilidade. A integração entre essas vantagens e o conceito de Experimentação Online deram origem a Aprendizagem Móvel, concepção que estabelece novos parâmetros com relação à inclusão digital no ensino e aprendizagem.

Da mesma maneira que abriu portas para novas possibilidades pedagógicas, o uso frequente dos smartphones elevou o nível de exigência dos seus usuários com relação a interface, tanto que cerca de 60% desses usuários tendem a abandonar uma página se ela não oferecer uma boa experiência mobile (CAELUM, 2017).

Em razão de suas particularidades, o desenvolvimento de aplicações mobile demandam por metodologias de avaliação de usabilidade específicas, conseqüentemente, as heurísticas tradicionais de usabilidade para desktop costumam ser adaptadas para que tenham aplicabilidade em dispositivos móveis.

O presente trabalho tem como objetivo desenvolver um aplicativo (app) para dispositivos móveis com boa usabilidade, que permita programar o robô remoto do LARA. Por meio da ferramenta, o usuário pode criar um programa utilizando estruturas sequenciais, de seleção e de repetição que será executado pelo robô.

Em busca do objetivo geral proposto, pretende-se alcançar os objetivos específicos a seguir:

- Levantar requisitos para um aplicativo móvel de um laboratório remoto voltado a aprendizagem de programação;
- Projetar interface gráfica do aplicativo considerando características de usabilidade para dispositivos móveis;
- Implementar o app com base nos requisitos levantados;
- Validar a usabilidade do app.

A metodologia de pesquisa utilizada foi a Design Science Research (DSR), cuja finalidade é elaborar artefatos para solucionar necessidades identificadas em domínios de aplicação específicos, para esse fim, é aplicado o conhecimento científico (HORITA et al., 2018). A DSR determina o conjunto sistemático de etapas a seguir, para que o artefato desenvolvido seja relevante e inovador em sua área: identificar o problema, definir o objetivo, projetar e desenvolver o artefato, demonstrar a aplicabilidade, avaliar o artefato, comunicar os resultados.

## 2. EMBASAMENTO TEÓRICO

Este capítulo abrange os conceitos de Aprendizagem Móvel e Experimentação Remota (Experimentação virtual, remota e remota móvel), em razão de sua importância como base para o desenvolvimento do presente trabalho.

### 2.1 Aprendizagem Móvel (Mobile Learning)

A Aprendizagem móvel (M-learning) é a aprendizagem mediante o uso de dispositivos móveis do tipo telefones móveis básicos, tablets, PDAs, reprodutores MP3, memórias USB, dispositivos de leitura eletrônica e Smartphones (DA SILVA et al., 2013). A ideia se baseia em prover conteúdos educacionais, disponibilizando-os por intermédio de dispositivos móveis.

Este tipo de aplicação concede ao aluno a oportunidade de realizar atividades educacionais a qualquer hora e local, assegurando, de tal forma, maior flexibilidade no acesso aos conteúdos, simulações ou experimentos reais manipulados à distância. Portanto, considerando a baixa infraestrutura das escolas em termos tecnológicos, abre-se um caminho ainda maior para a inclusão digital de estudantes e professores nas escolas públicas brasileiras (NICOLETE et al., 2016).

Por meio da Aprendizagem Móvel, é possível aproximar métodos de ensino e aprendizagem da realidade do aluno, já que este vivencia uma era tecnológica. No entanto, os dispositivos móveis por si só não se constituem em ferramentas educativas úteis, tornando-se imprescindível a pesquisa sobre o uso pedagógico destes (DA SILVA et al., 2013).

### 2.2 Experimentação Online

A experimentação Online compreende os laboratórios virtuais e remotos, favorecido por ferramentas online baseadas em realidade virtual, realidade aumentada, dispositivos sensoriais, vídeos ao vivo, vídeos interativos e jogos que promovem a imersão do usuário em ambientes virtuais (RESTIVO e CARDOSO, 2013). Tal concepção visa recriar experiências reais, seja por meio de simulações ou acesso remoto. Sua característica predominante é a disponibilidade de acesso remotamente a qualquer hora através da Internet.

As subseções a seguir apresentam os conceitos de Experimentação virtual e Experimentação remota, ferramentas pedagógicas que compõem a Experimentação online.

### 2.2.1 Experimentação Virtual (Laboratório Virtual)

Os laboratórios virtuais apresentam resultados preestabelecidos e observações simuladas por animações ou previamente gravadas, estes laboratórios são disponibilizados via páginas web. A experimentação virtual oferece os seguintes benefícios (ZORICA et al., 2003):

- Eficácia na explicação de conceitos teóricos;
- Possibilidade de condução de experimento passo a passo;
- Meio de interação;
- Oferece ferramentas flexíveis e de fácil uso;
- É uma alternativa de baixo custo;
- Não há restrições de tempo ou físicas.

Apesar de oferecerem uma experiência pertinente para absorção de conceitos quando bem projetados, simuladores pecam no que diz respeito à previsão de resultados, motivo que afasta a experimentação virtual dos fatores naturais da experimentação real (DA SILVA et al., 2013).

As simulações auxiliam bastante na assimilação de teorias, mas não podem substituir um laboratório convencional (EXEL et al., 2000). A fim de evitar possíveis erros de cálculo ou equívocos gerados por mal entendimento da teoria, o laboratório virtual proporciona ao aluno a oportunidade de experimentar ao máximo, de modo a ganhar experiência para o futuro contato com o experimento real.

### 2.2.2 Experimentação Remota

Os conceitos de Experimentação Remota são relativamente novos, porém, seu número cresceu exponencialmente junto ao recente progresso tecnológico e disponibilidade de ferramentas computacionais para o seu desenvolvimento. Sua aplicação permite ao usuário alterar parâmetros de controle, executar experiências, observar resultados em tempo real (capturados por câmeras de vídeo) e baixar dados através de uma interface web (CASINI et al., 2003). Por meio deste conceito,

instituições de ensino que possuem laboratórios convencionais, podem disponibilizar seus experimentos remotamente, permitindo desta forma que qualquer estudante com acesso à internet possa manipular os mesmos.

Diante do alcance global da internet atualmente e do alto custo e complexidade que envolve a construção e manutenção de laboratórios convencionais, o conceito de Laboratório Remoto tem ganhado força como ferramenta de ensino. Abaixo são listadas algumas vantagens que os Laboratórios Remotos podem oferecer aos seus usuários em comparação aos laboratórios virtuais e convencionais (ZORICA et al., 2003):

- Acesso remoto ao laboratório;
- Realizar experimentos em equipamentos reais;
- Colaborar (realizar e discutir experimentos em grupo);
- Aprender por tentativa e erro;
- Realizar análises em dados experimentais reais;
- Flexibilidade na escolha de horário e local para a realização de experimentos.

Na prática, existem situações nas quais o emprego dos laboratórios convencionais torna-se inconveniente e/ou dispendioso. Um experimento convencional que permita o manejo por apenas um indivíduo por vez, em uma aula para um grande número de pessoas, pode custar demasiado tempo para que todos os estudantes efetuem o experimento, impossibilitando que o mesmo seja aplicado de forma concreta durante o período de duração da aula. Em contrapartida, o acesso ao experimento remoto, não exige que toda uma turma seja movida até um laboratório (EXEL et al., 2000), se o experimento for realizado por um grupo remotamente, de forma colaborativa, todos os integrantes podem observar e discutir os resultados, enquanto um deles efetua a experimentação.

Os Laboratórios Remotos levam o estudante a vivenciar uma experiência direta com o processo real (CASINI et al., 2003), fator importante para o aprendizado das ciências e engenharias, sendo sua única desvantagem, o fato de proporcionar não mais que uma presença virtual no laboratório (ZORICA et al., 2003).

Como não é possível realizar mais de uma experiência simultaneamente sobre a mesma infraestrutura, os recursos físicos devem ser geridos para que não haja conflitos durante a experimentação (COSTA, 2005). A gestão de acesso pode ser feita através do método de controle síncrono ou assíncrono.



O Controle Síncrono requer que um sistema de gestão de reserva seja implementado. Dessa forma, o acesso de um ou de um grupo de usuários ao experimento precisa ser precedido de uma reserva do dia e hora pretendidos.

Por outro lado, o Controle Assíncrono exige que seja feito um sistema de gestão de acesso baseado em fila de espera. Com esse método o usuário terá acesso contínuo ao experimento remoto, podendo enviar comandos de controle sempre que desejar, porém, em casos onde há mais de um usuário utilizando o sistema simultaneamente, comandos enviados poderão ficar na fila de espera, antes de serem executados.

### 2.2.3 Experimentação Remota Móvel

Assim como ocorrido com a evolução dos computadores pessoais, os avanços tecnológicos dos recursos inerentes aos dispositivos móveis (acesso à internet, armazenamento interno, touchscreen, acelerômetro, entre outros), consequentemente viabilizaram o empregado destes aparelhos no suporte ao ensino e aprendizado das ciências e engenharias. Isso se deve sobretudo, ao aumento substancial do número de smartphones pelo mundo nos últimos anos, tanto é que hoje, no Brasil, existem mais aparelhos como estes em atividade que habitantes propriamente ditos.

A Experimentação Remota Móvel (Mobile Remote Experimentation, MRE), herda os conceitos e características dos Laboratórios Remotos, sendo assim, destinam-se a disponibilizar acesso a experimentos reais por meio de dispositivos móveis, como smartphones e tablets. Experimentos reais acessados remotamente por intermédio de dispositivos móveis, proporcionam aos estudantes uma nova maneira de interagir com as disciplinas de forma simples e agradável, em qualquer lugar e a qualquer momento (DA SILVA et al., 2013).

A popularidade dos smartphones, tende a ampliar a perspectiva de acesso aos Laboratórios Remotos. Atualmente, estes aparelhos com alto poder de processamento, leves e fáceis de transportar e munidos de diversas alternativas de acesso à internet (3g, 4g, Wi-Fi, entre outras), são realidade para a maioria dos estudantes por conta dos preços acessíveis. Suas principais vantagens são a disponibilidade de acesso e a notável portabilidade, essas características, aliadas às tecnologias de redes sem fio (wireless), permitem ao estudante realizar experimentos

sem qualquer restrição temporal e/ou geográfica (casa, universidade, hotéis, transporte público, etc.) (COSTA, 2005), o mesmo não se aplica aos computadores pessoais, em razão de sua portabilidade limitada.

Apesar das vantagens mencionadas, algumas características inerentes aos dispositivos móveis podem limitar o desenvolvimento de aplicações para MRE. Dentre elas estão: tamanho reduzido da tela, internet por vezes limitada, ausência de teclado amigável, bateria com pouca autonomia, memória reduzida, entre outras (COSTA, 2005). Estas características estabelecem os dispositivos móveis, preferencialmente, como auxiliares ou alternativa de expansão para os Laboratórios Remotos, visto que, os computadores pessoais, ainda dispõem de qualidades significativas no que diz respeito a experimentação a distância.

### 3. TRABALHOS RELACIONADOS

A presente seção apresenta alguns trabalhos relacionados que abrangem o tema Aprendizagem Móvel e o desenvolvimento de seus aplicativos adaptados para o Ensino-Aprendizagem.

#### 3.1 Desenvolvimento de Aplicação para Interfaceamento com Experimentos Remotos por Smartphones

Este trabalho, desenvolvido pelo Laboratório de Experimentação Remota da Universidade Federal de Santa Catarina, apresenta o REx-Mobile, um aplicativo para uso de experimentos remotos através de dispositivos móveis.

O aplicativo implementado com base nas ferramentas de programação HTML5, jQuery e jQuery Mobile, assegura compatibilidade com diversos sistemas operacionais móveis por um baixo custo. Sua interface, exibida na Figura 1, possibilita o acesso a experimentos como o painel elétrico CA (experimento da física sobre correntes elétricas alternadas), além de oferecer material didático e atividades que possibilitam ao aluno absorver o conteúdo e posteriormente observar seus efeitos reais através de seu dispositivo móvel.

Figura 1 - Interação com experimento remoto a partir do aplicativo



Fonte: Rochadel et al. (2012).

Como pode ser visto na Figura 1, o controle dos experimentos se dá por meio de botões postos na tela, valendo-se dos recursos do touchscreen, característicos dos smartphones atuais. Já as imagens da experimentação, são transmitidas por streaming de vídeo e exibidas em tempo real, dessa forma, o usuário visualiza, de imediato, o efeito de cada ação que realiza.

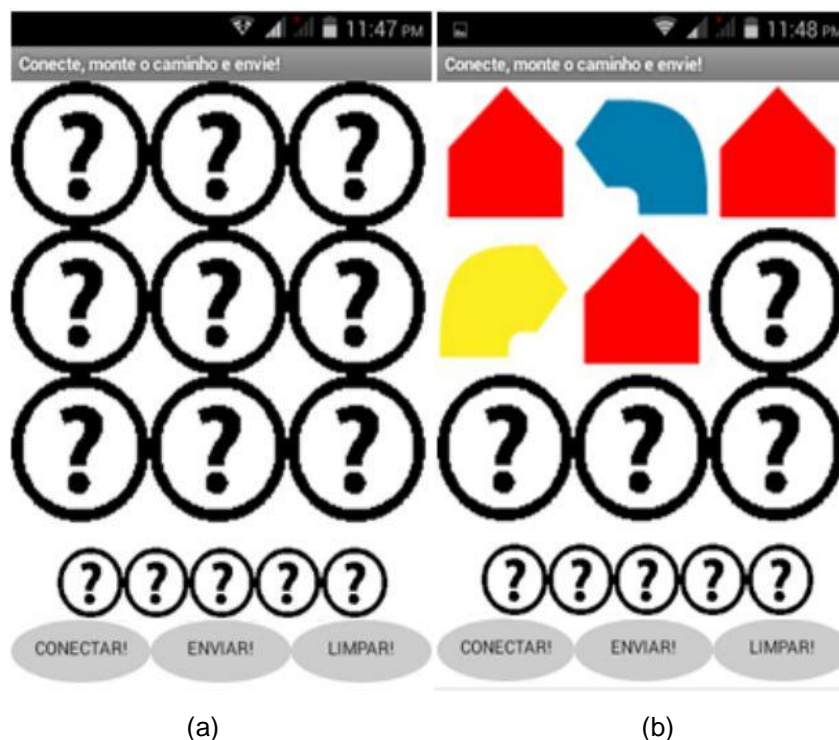
Com o intuito de oferecer uma experiência agradável para o seu usuário, o REx-Mobile procura simplicidade no que diz respeito a usabilidade, uma vez que, por conta de suas limitações, a interação com os smartphones pode ser desconfortável, principalmente por falta de alguns recursos presentes exclusivamente nos computadores pessoais.

### 3.2 Aplicação da Robótica no Ensino-Aprendizagem de Lógica de Programação para Crianças

Este trabalho apresenta o desenvolvimento de uma ferramenta capaz de auxiliar no processo de ensino e aprendizagem de lógica de programação para crianças de 5 a 7 anos. Para isso, o projeto conta com um aplicativo para dispositivos móveis, desenvolvido para plataforma android, e um artefato robótico com rodas, baseado na plataforma arduino e que pode ser controlado remotamente, devido a seu módulo bluetooth de comunicação e controle.

O aplicativo, implementado com uso da ferramenta MIT App Inventor, apresenta uma interface de concepção lúdica, na qual o usuário monta caminhos lógicos, que indicam a forma como o robô deve se mover, com o propósito de realizar uma determinada tarefa.

Figura 2 - Interface do aplicativo e caminhos lógicos



Fonte: Calegari (2015).

A Figura 2a exibe a tela principal do aplicativo, onde o usuário pode pressionar os campos com sinais de interrogação, até alcançar a sequência de comandos desejada. A Figura 2b, por sua vez, mostra a tela principal após a adição de alguns comandos de locomoção para o artefato robótico.

Além dos comandos de movimento, o aplicativo possui um comando de desvio de fluxo, através do qual é possível criar desvios nos caminhos lógicos e laços de repetição.

A ludicidade presente na interface do aplicativo tem como finalidade desenvolver a criatividade e o raciocínio lógico das crianças, de uma maneira divertida e interativa, e favorecendo o uso da imaginação (CALEGARI, 2015).

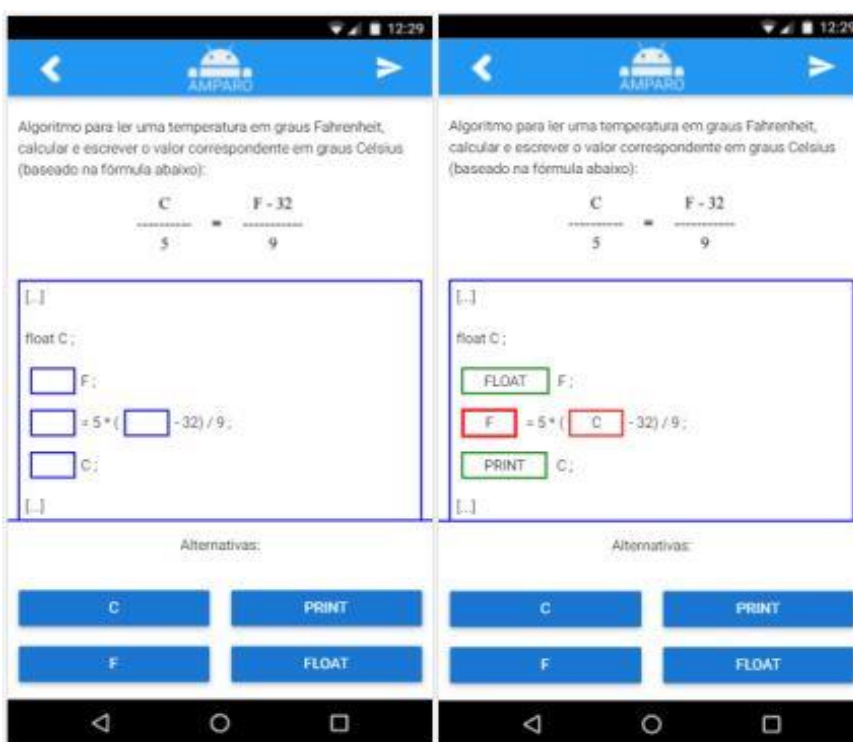
### 3.3 Aplicativo Lúdico-Pedagógico para Ensino de Programação e Robótica Educacional

Este trabalho foi desenvolvido no Instituto federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS), campus de Canoas. Seu propósito é desenvolver um aplicativo de caráter lúdico-pedagógico, com o intuito de influenciar positivamente no ensino de programação (BARDINI et al., 2017).

Além do aplicativo, implementado para sistemas Android, com uso de HTML, Javascript e da framework Cordova, a ferramenta conta também com um robô Arduino, apto a executar instruções recebidas via bluetooth, enviadas do aplicativo.

O aplicativo traz uma interface em formato de jogo, na qual o usuário deve preencher lacunas em funções de lógicas de programação, de acordo com as alternativas disponíveis na tela (BARDINI et al., 2017). A Figura 3 ilustra a tela com os comandos e lacunas antes e depois de serem preenchidas.

Figura 3 – Interface com possibilidade do jogo



Fonte: Bardini et al. (2017).

Como visto na Figura 3, ao iniciar o jogo, o usuário se vê diante de um algoritmo com algumas lacunas a serem preenchidas. Ao passo que o usuário preenche uma lacuna, a mesma é realçada de verde caso a resposta esteja correta, ou vermelho caso esteja incorreta. Nos casos em que o algoritmo tem relação com o robô arduino, o usuário pode enviar o código para o dispositivo robótico executar, porém, isso só é permitido quando todas as lacunas são preenchidas corretamente.

A interface em formato de jogo, juntamente com o conceito de robótica educacional, foram englobados na ferramenta com o propósito de atrair iniciantes na área de Tecnologia da Informação (TI) e oferecer suporte no ensino da mesma, visto

que, nos dias de hoje, existe uma grande demanda por profissionais capazes de resolver problemas que exigem um alto nível de raciocínio lógico.

## 4. LaraApp

Para desenvolvimento do LaraApp, foi utilizada a metodologia do Scrum Solo, processo de software iterativo e incremental proposto por Pagotto et al. (2016) com o objetivo de suprir a ausência de um processo para desenvolvedores solo.

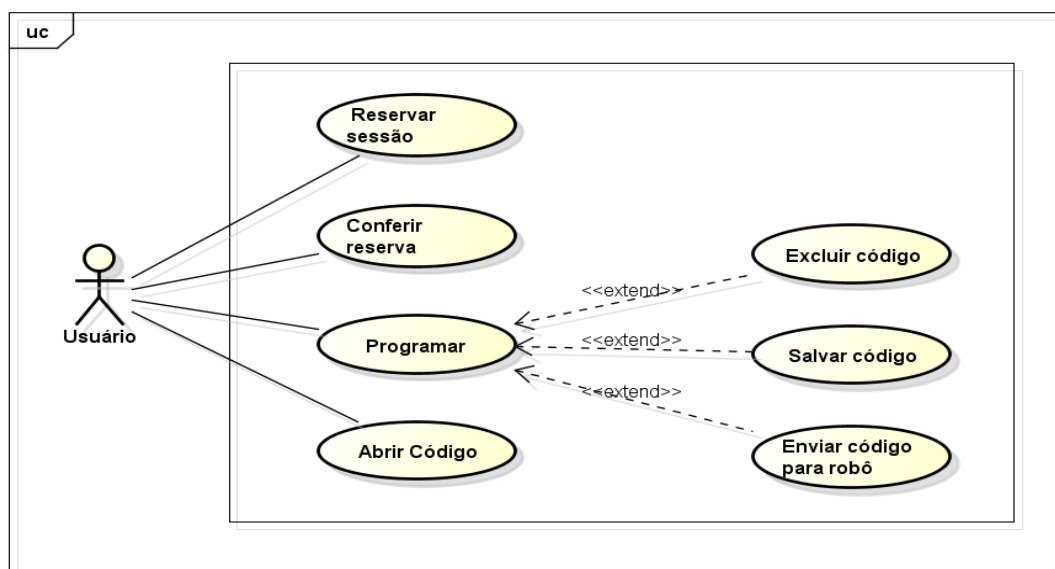
Esse processo recomenda sprints de uma semana de duração e sem reuniões diárias. Protótipos do software atualizados devem ser entregues no final de cada sprint pelo desenvolvedor, e se necessário, reuniões de orientação com o grupo de validação podem ser combinadas.

O Scrum Solo propõe quatro atores no processo de desenvolvimento, sendo eles: o proprietário do produto ou grupo de usuários alvo do software; o desenvolvedor individual, responsável por executar o processo e criar o produto; o orientador, conhecedor do processo, e que possui uma vasta visão a respeito do produto em desenvolvimento; e o grupo de validação, prováveis usuários do produto final, devem participar da validação do produto.

O levantamento de requisitos se baseou na análise das funcionalidades do LARA, da API do robô e em entrevistas com alunos e professores. A modelagem foi realizada utilizando a Linguagem de Modelagem Unificada (UML).

A Figura 4 apresenta o diagrama de casos de uso do LaraApp, onde o usuário interage com a aplicação, mediante possíveis ações indicadas nas elipses.

Figura 4 - Diagrama de casos de uso



Fonte: Autoria própria.



Ao abrir o aplicativo, o usuário tem acesso a tela de programação, na qual pode-se combinar as estruturas e comandos de programação para gerar códigos, salvar algoritmos e abrir códigos salvos anteriormente. Caso o usuário tenha interesse em visualizar a execução dos seus códigos pelo robô do LARA, uma reserva deve ser feita pelo mesmo, com isso, durante o tempo de reserva, o usuário poderá enviar códigos e observar o comportamento do robô em tempo real. Além disso, o usuário pode verificar quanto tempo de reserva lhe resta.

Ainda durante a fase de modelagem, com o intuito de obter um modelo de interface que melhor se adequasse ao objetivo do trabalho, protótipos das telas foram criados levando em conta os requisitos do projeto.

Em virtude de ser uma ferramenta de apoio a iniciantes em programação, o LaraApp foi planejado com o propósito de possuir uma interface simples e intuitiva. Desta forma, evita-se que o aluno se desprenda do objetivo principal da ferramenta enquanto resolve problemas relativos ao manuseio da interface.

A fim de atender aos requisitos, alguns conceitos e técnicas de usabilidade de interface foram empregadas no processo de modelagem das telas, mais especificamente a Lei de Fitts, as Thumb Zones e as microinterações.

A Lei de Fitts é um cálculo capaz de prever quanto tempo um usuário levaria para escolher uma opção de um submenu após clicar em um item do menu principal (CAELUM, 2017). O ideal é que opções mais comuns de serem escolhidas sejam alcançadas com o mínimo possível de tempo.

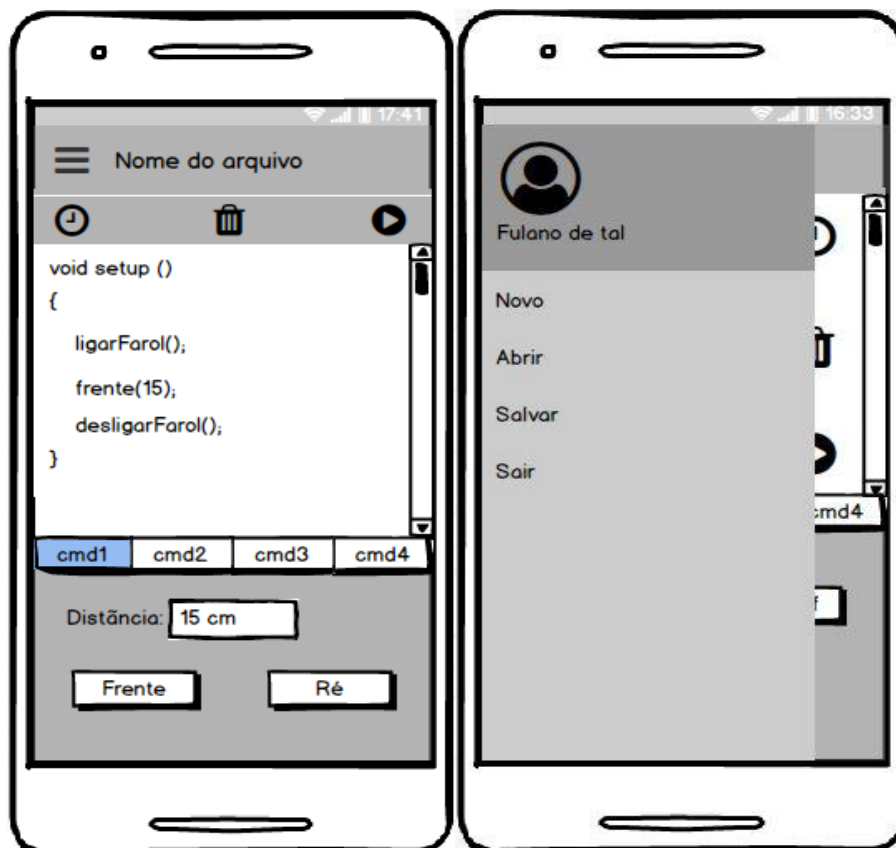
Já o termo Thumb Zone se refere a área mais confortável para toque com uma mão só. Pesquisas apontam que 49% dos usuários manuseiam smartphones com apenas uma das mãos, utilizando o polegar para tocar na tela (CAELUM, 2017). Baseado na Thumb Zone é possível posicionar elementos interativos em locais agradáveis para o usuário.

Por fim, as microinterações são casos particulares de interação entre o usuário e a interface, com o propósito de completar uma tarefa específica. A proposta é identificar essa interação e criar um feedback para ela. Tendo como exemplo a situação em que um botão ao ser pressionado muda de cor ou se torna inativo para indicar que a ação foi efetuada. Outro exemplo seriam as mensagens de feedback com relação a erros cometidos pelo usuário, como o preenchimento incorreto de formulários. O ideal é que as microinterações sejam claras, rápidas e pareçam

naturais, na maioria das vezes passando despercebido aos olhos dos usuários. Sua finalidade é melhorar a experiência de uso da aplicação sem a necessidade de criar novas funcionalidade.

A Figura 5 exibe os protótipos da tela de programação e da barra lateral do aplicativo.

Figura 5 - Protótipos das telas



Fonte: Autoria própria.

A fim de exigir o mínimo possível de entradas a partir do teclado, o sistema com abas (cmd1, cmd2, cmd3 e cmd4) e botões, apresentado na Figura 5, foi planejado com o propósito de evitar uma experiência exaustiva, por conta do teclado pouco amigável característico dos smartphones. Apesar dessa limitação, a aplicação permite inúmeras combinações, que podem ser exploradas pelo usuário durante a criação de algoritmos.

#### 4.1 Implementação do LaraApp

A criação de aplicativos para dispositivos móveis é realizável basicamente mediante desenvolvimento nativo ou híbrido. O desenvolvimento nativo se dá através do uso de uma linguagem específica para a plataforma na qual se deseja desenvolver o app, sendo que Android e IOs são as plataformas dominantes nos smartphones atualmente. No caso do Android, os apps nativos são implementados em Java, enquanto no IOs, utiliza-se Objective-C.

O desenvolvimento nativo é recomendado em casos onde um alto processamento de dados é necessário e a performance é indispensável, uma vez que, aplicativos nativos não exigem qualquer interpretação de código enquanto executa, além do mais, esse método de desenvolvimento possibilita o acesso a recursos do dispositivo (câmera, GPS, acelerômetro, agenda, entre outros), o que garante maior imersão ao usuário (SILVA e SANTOS, 2014).

Apesar das vantagens citadas, em razão da incompatibilidade das plataformas, se o desenvolvedor pretende alcançar uma grande variedade de usuários, o mesmo app precisa ser implementado uma vez para cada plataforma desejada, o que demanda maior tempo e custo.

Por outro lado, o desenvolvimento híbrido tem como principal vantagem a portabilidade do código, pois, bastam poucos ajustes para o mesmo aplicativo ser lançado para diversas plataformas. Esta flexibilidade é possível por conta de um recurso existente em todas as plataformas móveis, o navegador Web.

Os aplicativos híbridos, desenvolvidos com HTML5, CSS3 e JavaScript, interagem com a interface Javascript de código nativo da plataforma para abrir uma instância do navegador que contém a interface do aplicativo, a WebView (SILVA e SANTOS, 2014). As ferramentas de desenvolvimento (frameworks) empregam esta técnica com o intuito de oferecer além de flexibilidade, o acesso a recursos nativos, no entanto, o desenvolvimento híbrido não é recomendado para projetos que demandam alta performance, robustez e execução de funções avançadas.

Em razão de não exigir alto poder de processamento e visando a portabilidade com menos tempo de trabalho, o LaraApp foi implementado utilizando desenvolvimento híbrido. Na escolha da ferramenta de implementação optou-se pelo Apache Cordova, framework de código aberto que permite o uso de tecnologias

padrão da web (HTML5, CSS3 e JavaScript) para desenvolvimento em diversas plataformas móveis (CORDOVA, 2019).

O HTML5 define a estrutura do aplicativo, tais como abas, inputs e botões. Enquanto o CSS3 modifica seu estilo, dando cores e detalhes à estrutura. Por fim, o JavaScript descreve como a estrutura deve se comportar, executando tarefas dinamicamente, tendo como exemplo a ativação de uma função quando um botão é acionado com um click.

O armazenamento dos códigos do usuário foi feito por meio do SQLite, uma biblioteca de código aberto que implementa um banco de dados SQL incorporado na aplicação, sem um processo de servidor separado e nenhuma configuração (SQLITE, 2019). O SQLite é indicado para sistemas de pequeno porte, em que poucos dados são utilizados. Sua grande vantagem se deve a simplicidade e praticidade em implementar e administrar dados.

O desenvolvimento aplicando tecnologias padrão da web, incorporado aos dispositivos móveis com o uso do Apache Cordova, alcançou um resultado muito próximo do que foi projetado a princípio.

#### 4.2 Apresentação do LaraApp

Ao logar no LaraApp o usuário se depara com a tela de programação. Nessa tela, como pode ser observado na Figura 6, se encontram as mais importantes funções do aplicativo. O usuário tem acesso livre a todas essas funções para criação e gestão de seus códigos, com exceção da função de envio de códigos para o robô, a qual pode ser acessada somente no período reservado pelo usuário.

Figura 6 - Screenshot da tela de programação



Fonte: Autoria própria.

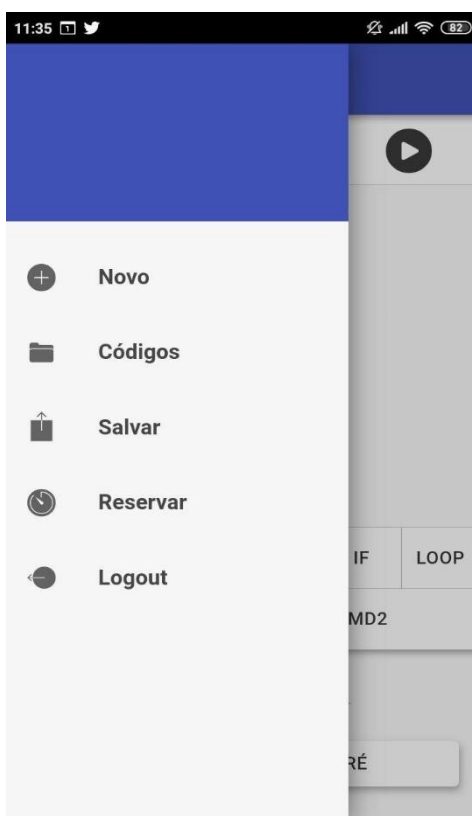
Na tela de programação, o usuário pode navegar por abas que contém comandos de programação do robô: MOV - movimento do robô frente, ré, direita, esquerda; FAROL - ligar e desligar farol; VAR - declaração variável; OP - operações matemáticas de soma, subtração, divisão, multiplicação; IF - comando de seleção; e LOOP - comando de repetição.

O usuário deve combinar os comandos e estruturas para gerar seu código, o qual é exibido na área de texto. Por exemplo, para gerar o código exposto na Figura 6, seriam necessários os seguintes passos: na aba VAR, declarar uma variável “cont” de valor igual a 0; na aba LOOP, adicionar um “while” com condição igual a “cont < 7”; na aba CMD1 contida na aba MOV, adicionar o comando para que o robô se mova 14 cm para frente; na aba CMD2 contida na aba MOV, adicionar o comando para que o robô se mova 7 cm para direita; e finalmente, na aba OP, criar a operação onde “cont” receba o valor “cont+1”.

A barra superior com ícones, permite ao usuário conferir sua reserva, apagar o código contido na área de texto e enviar algoritmo para o robô.

Linhas de código também podem ser apagadas individualmente, para isso, basta que o usuário toque na linha que pretende excluir e confirme a ação. Entretanto, em casos onde apagar uma linha específica causaria um erro no código, a função é vetada e um alerta é mostrado ao usuário.

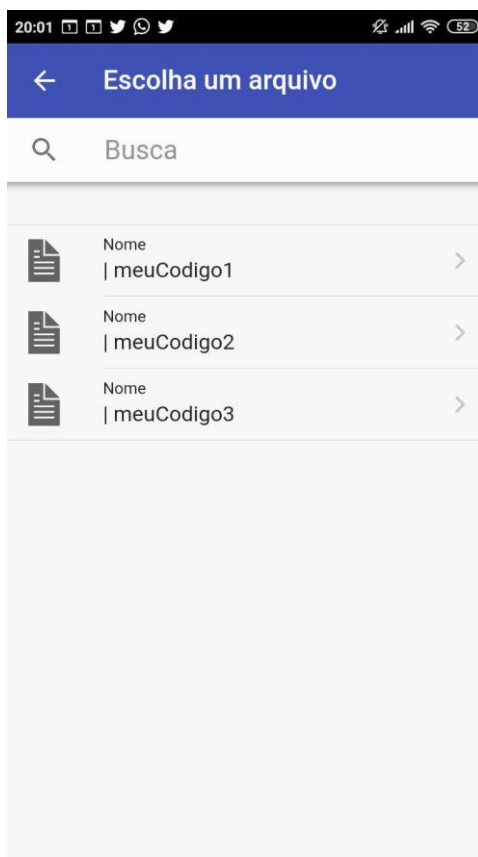
Figura 7 - Screenshot da barra lateral



Fonte: Autoria própria.

Na barra lateral do aplicativo, exibida na Figura 7, o usuário pode: criar um novo código, escolhendo a opção “Novo”; gerir códigos salvos, escolhendo a opção “Códigos”, que abre a tela de códigos do usuário (Figura 8); salvar códigos, escolhendo a opção “Salvar”; reservar uma sessão para ter acesso ao robô do LARA, escolhendo a opção “Reservar”, que leva para a tela de reserva (Figura 9); e sair da aplicação (deslogar), escolhendo a opção “Logout”.

Figura 8 - Screenshot da tela de códigos do usuário



Fonte: Autoria própria.

A tela de códigos do usuário, mostrada na Figura 8, permite que códigos salvos sejam abertos na tela de programação ou excluídos da aplicação.

Figura 9 - Screenshot da tela de reservar sessão



The screenshot shows a mobile application interface for reserving a session. At the top, there is a blue header bar with a white back arrow on the left and the text 'Reservar sessão' in white. Below the header, the main content area is light gray. It contains two text input fields: the first is labeled 'Data' and the second is labeled 'Hora de início'. Below these fields is a prominent blue button with the white text 'CADASTRAR SESSÃO'. The top status bar of the phone is visible, showing the time 14:01, signal strength, Wi-Fi, and battery icons.

Fonte: Autoria própria.

A Figura 9 apresenta a tela de reserva de sessão, para fazer uma reserva, basta o usuário selecionar a data e hora que pretende ter acesso ao robô do LARA.



## 5. VALIDAÇÃO DO LaraApp

Com o grande aumento do número de smartphones por todo o mundo, aplicações que antes eram desenvolvidas apenas para computadores pessoais, passaram a ser desenvolvidas também para os dispositivos móveis. O número de aplicativos produzidos para os dispositivos em questão cresceu a ponto de hoje existirem lojas online específicas para venda de apps, com milhares de exemplares de diversos tipos, como rede social, compras e games. Diante deste cenário, surge a necessidade de se avaliar a usabilidade das interfaces de sistemas desenvolvidos para os smartphones, afinal, suas aplicações prometem realizar diversos tipos de tarefas, apesar de suas limitações com relação aos computadores tradicionais.

As características inerentes aos telefones touchscreen (mobilidade, tamanho reduzido da tela, internet por vezes limitada, bateria com pouca autonomia, memória reduzida, entre outras), demandam por um conjunto de princípios e diretrizes de usabilidade diferentes dos desenvolvidos para desktop. No entanto, adaptar heurísticas tradicionais para que tenham aplicabilidade em smartphones, não é uma tarefa simples, uma vez que, os métodos tradicionais de avaliação de usabilidade não consideraram as particularidades destes dispositivos (KRONE, 2013).

A presente avaliação de usabilidade contou com um conjunto de princípios e diretrizes adaptados a partir das heurísticas de Nielsen (1994) para dispositivos touchscreen. Nas seções seguintes, será apresentado o método de inspeção empregado no teste de interface do LaraApp, o conjunto de heurísticas adotado e o resultado da avaliação.

### 5.1 Avaliação Heurística

A avaliação heurística é um método de avaliação analítico. Também conhecidos como avaliação por inspeção, estes são métodos nos quais avaliadores inspecionam ou examinam aspectos de uma interface de usuário relacionados a usabilidade (PRATES e BARBOSA, 2003). Geralmente utilizados quando os usuários não são facilmente encontrados, o tempo disponível para a avaliação não é muito grande ou ainda quando os recursos disponíveis são escassos (FEIJÓ et al., 2013), os métodos de avaliação analíticos em sua essência visam encontrar falhas de usabilidade em interfaces que podem prejudicar a experiência de utilização do sistema

sendo testado. Os problemas encontrados passam por uma análise, de onde devem surgir recomendação de melhorias a serem implementadas e sua ordem de prioridade com base na gravidade do problema e seu custo de correção.

O método de avaliação heurística trata-se da avaliação de especialistas com base em um conjunto de heurísticas de usabilidade, as quais são definidas como melhores práticas por profissionais experientes e especialistas em IHC, ao longo de diversos anos de trabalho nesta área (PRATES e BARBOSA, 2003).

Diferente da maioria dos métodos de avaliação, este exige pouco tempo e investimentos em sua aplicação, no entanto, por possuir especialistas envolvidos em sua aplicação, apresenta resultados confiáveis e eficientes (FEIJÓ et al., 2013).

Em sua aplicação recomenda-se de 3 a 5 avaliadores especialistas. Cada avaliador manuseia o sistema individualmente, simulando o que seria feito por usuários comuns, anotando cada problema no sistema que foge do recomendado pelo conjunto de heurísticas de usabilidade utilizado como base da avaliação.

## 5.2 Avaliação Heurística aplicada ao LaraApp

Para avaliar o LaraApp, optou-se pela aplicação da avaliação heurística de usabilidade, por ser um teste eficaz e exigir poucos recursos. O processo contou com três avaliadores especialistas, cada um deles com conhecimento tanto no que diz respeito ao domínio da aplicação, quanto para análise de interfaces com base em padrões de usabilidade predefinidos.

O conjunto de heurísticas de usabilidade tomado como base para essa avaliação foi proposto pelo Grupo de Qualidade do Software da Universidade Federal de Santa Catarina (GQS/INSCoD/UFSC). Desenvolvido a partir das dez heurísticas de Nielsen (1994), o instrumento disponibiliza um checklist (lista de verificação) com 48 questões, gerado por meio da adaptação dos princípios de usabilidade tradicionais, para as interfaces de celulares *touchscreen*. O checklist funciona como medidor da qualidade da interface sendo avaliada. Após o preenchimento do questionário por parte do avaliador, o resultado é exibido, no qual a classificação de qualidade da interface é revelada, juntamente com uma breve descrição, apontando os princípios de usabilidade que foram violados, ou não, pela aplicação. O checklist para avaliação da usabilidade de aplicativos para celulares *touchscreen* (MATCh) pode ser acessada em: <http://match.inf.ufsc.br:90>.

Os possíveis níveis de classificação de usabilidade dados pela própria ferramenta de avaliação são:

- **Até 30 (Usabilidade muito baixa):** Somente iniciam as tarefas ao comando do usuário, evidenciam a necessidade de inserção de dados, possuem botões e links com área clicável do tamanho dos mesmos, evitam abreviaturas, além disso, são consistentes, utilizam o mesmo idioma em seus textos, apresentam os links de forma consistente entre as telas e funções semelhantes de forma similar.
- **30 - 40 (Usabilidade baixa):** Além de possuir as características do nível anterior, fornecem update do status para operações mais lentas por meio de mensagens claras e concisas, mantêm o mesmo título para telas com o mesmo tipo de conteúdo, utilizam títulos de telas que descrevem adequadamente seu conteúdo, exibem apenas informações relacionadas à tarefa que está sendo realizada, apresentam ícones e informações textuais de forma padronizada com contraste suficiente em relação ao plano de fundo e imagens com cor e detalhamento favoráveis à leitura em uma tela pequena, possuem navegação consistente entre suas telas, permitem retornar à tela anterior a qualquer momento, mantêm controles que realizam a mesma função em posições semelhantes na tela, permitem que as funções mais utilizadas sejam facilmente acessadas e possuam botões com tamanho adequado ao clique.
- **40 - 50 (Usabilidade razoável):** Além de possuir as características dos níveis anteriores, dispõem as informações em uma ordem lógica e natural, apresentam as mensagens mais importantes na posição padrão dos aplicativos para a plataforma, oferecem uma navegação intuitiva e um menu esteticamente simples e claro, contêm títulos e rótulos curtos, possuem fontes, espaçamentos entrelinhas e alinhamento que favorecem a leitura, realçam conteúdos mais importantes, possuem tarefas simples de serem executadas que deixam claro qual seu próximo passo, oferecem feedback imediato e adequado sobre seu status a cada ação do usuário, evidenciam que controles e botões são clicáveis, distinguem claramente os componentes interativos selecionados, utilizam objetos (ícones) ao invés de botões, com significados compreensíveis e intuitivos e não apresentam problemas durante a interação (trava, botões que não funcionam no primeiro clique, etc.).

- **50 - 60 (Usabilidade alta):** além de possuir as características dos níveis anteriores, exibem pequenas quantidades de informações em cada tela, mantêm acessíveis menus e funções comuns do aplicativo em todas as telas, evidenciam o número de passos necessários para a realização de uma tarefa, permitem que o usuário cancele uma ação em progresso, possuem navegação de acordo com os padrões da plataforma a que se destinam e possibilitam fácil acesso de mais de um usuário no caso de aplicativos associados a cadastro de login.
- **Acima 60 (Usabilidade muito alta):** tem ainda maior probabilidade, que os níveis anteriores, de possuir todas as características descritas nos mesmos, possuindo um alto nível de usabilidade.

A presente avaliação consistiu-se dos seguintes passos:

1. Por aproximadamente uma hora, cada avaliador pôde examinar os aspectos de usabilidade da interface através do manuseio da mesma, durante essa fase, os avaliadores tiveram a liberdade de anotar problemas identificados, possibilitando recomendações visando futuras correções.
2. Nesta etapa, o checklist para avaliação da usabilidade de aplicativos para celulares touchscreen (MATCh) foi respondido pelos avaliadores, com base na experiência individual obtida durante o uso da aplicação.
3. Nesta fase, os avaliadores unificaram seus relatórios a fim de julgar e reunir todos os problemas encontrados durante a inspeção da interface.
4. Por fim, o gerente do projeto junto aos avaliadores, julgaram a gravidade, e conseqüentemente, a prioridade de correção para cada problema identificado nos estágios anteriores.

### 5.3 Resultado da Avaliação

Ao fim da avaliação, obteve-se os resultados exibidos na tabela 1.

Tabela 1 - resultados da avaliação do LaraApp pelo checklist

<b>Avaliador</b>	<b>Resultado</b>
Avaliador 1	58.6 pontos - Usabilidade alta
Avaliador 2	60.9 pontos - Usabilidade muito alta
Avaliador 3	61.7 pontos - Usabilidade muito alta

Fonte: Autoria própria.

Além dos resultados retornados pelo checklist, obteve-se as recomendações dos avaliadores sobre possíveis soluções para problemas de usabilidade na interface do LaraApp. A fim de definir a gravidade de cada problema de acordo com sua prioridade, tem-se a seguinte relação:

- 1 - Prioridade muito baixa: caso não haja tempo de sobra para correção, esse tipo de problema pode ser ignorado a princípio.
- 2 - Prioridade baixa: é desejável que seja consertado, porém recebe baixa prioridade.
- 3 - Prioridade alta: é importante que seja corrigido, se possível, antes do lançamento.
- 4 - Prioridade muito alta: é imprescindível que seja solucionado antes do lançamento do sistema.

A tabela 2 expõe as falhas identificadas no decorrer da inspeção, a gravidade de cada problema e as soluções propostas pelos avaliadores.

Tabela 2 - problemas, correções propostas e prioridade para correção

<b>Problema</b>	<b>Gravidade</b>	<b>Proposta de correção</b>
As labels dos botões “FAROL ON” e “FAROL OFF” da aba farol com nomes que combinam dois idiomas.	1	Alterar os nomes dos botões da aba farol para “LIGAR” e “DESLIGAR”.
Ícone que indica “salvar” pouco intuitivo.	1	Utilizar um ícone mais comum para “salvar”.
Complexidade para utilizar comandos mais robustos: VAR, OP, IF e WHILE.	2	Adicionar um ícone de ajuda que abre um pop-up com dicas de como usar os comandos.
Não indica em que linha no código o cursor do usuário está em cada momento.	3	Destacar a linha de código onde o cursor está, aplicando uma cor diferente nesta posição.
Dificuldade em esconder o teclado após ter digitado em um input (entrada).	3	Fazer com que o botão avançar do teclado, quando acionado, esconda o mesmo.

Fonte: Autoria própria.

Verifica-se na tabela 2, que os avaliadores identificaram cinco problemas de usabilidade no LaraApp, sendo que dois apresentam gravidade 1, um apresenta gravidade 2, e por fim, dois apresentam gravidade 3.

As falhas de usabilidade de gravidade 1 foram apontados como sendo as labels “FAROL ON” e “FAROL OFF” dos botões da aba “FAROL” e o ícone pouco comum para a ação “salvar”, localizado na barra lateral da tela de programação. Segundo os avaliadores, a combinação de idiomas das labels na aba “FAROL”, podem causar confusão, sendo assim, foi indicado que as labels fossem alteradas para “LIGAR” e “DESLIGAR”, evitando desse modo, a combinação de dois idiomas distintos. Já com

relação ao ícone “salvar”, os avaliadores sugeriram que fosse trocado por um ícone mais comum, uma vez que o ícone utilizado não é tão intuitivo.

A falha de usabilidade de gravidade 2, está relacionada a complexidade que envolve o uso das estruturas de programação (VAR, OP, IF e WHILE) por meio da aplicação, para os avaliadores, um sistema de ajuda deve ser implementado, de modo que o usuário iniciante possa observar exemplos de como utilizar os comandos mais complexos.

Por fim, as falhas de usabilidade de gravidade 3, segundo os avaliadores, foram a falta de feedback com relação a linha em que se encontra o cursor do usuário em seu código após realizar alguma ação e a dificuldade em esconder o teclado, após o uso do mesmo. Como resolução para o primeiro problema de gravidade 3, os avaliadores propuseram que a linha onde se encontra o usuário, tenha uma coloração distinta das demais, de forma a destacar essa posição, impedindo de tal maneira, que o usuário se perca enquanto constrói algoritmos mais complexos. Já o segundo problema de gravidade 3, deve ser sanado, adicionando ao botão “avançar” do teclado, a função de esconder o próprio, de modo a impedir possíveis incômodos no decorrer do processo de programação.

Com relação aos resultados apresentados na presente sessão, nota-se que os problemas encontrados no aplicativo durante a fase de inspeção (etapa em que o avaliador examina a interface em busca de falhas), não influenciaram nas respostas do checklist a ponto de ocasionar uma má pontuação. A baixa gravidade dos problemas encontrados pode ser a razão para este fato.

## 6. CONCLUSÃO

O presente trabalho apresenta uma aplicação desenvolvida com a finalidade de facilitar o acesso ao robô remoto do LARA contribuindo desta forma para o ensino e aprendizagem de programação através do conceito de experimentação remota. Essa ferramenta é composta por um aplicativo para dispositivos móveis, utilizado para criação de algoritmos de controle de um robô remoto móvel encarregado de executar o código.

A interface do LaraApp leva em consideração as particularidades dos dispositivos móveis no que diz respeito a usabilidade e as funcionalidades do LARA. A interface de programação, baseada em abas e botões, se mostrou bastante eficaz para evitar que o usuário digite grandes quantidades de texto enquanto programa. Essa funcionalidade é importante, pois, o público alvo do aplicativo tratar-se de iniciantes em cursos que envolvem computação e informática.

Quanto ao suporte no ensino e aprendizagem de disciplinas que envolvem programação, o aplicativo mostrou ter potencial para contribuir como ferramenta de apoio aos iniciantes, além de figurar como alternativa perante a escassez de laboratórios convencionais nas escolas e universidades. Porém, até então, o desenvolvimento da aplicação se encontra em andamento, uma vez que, a conexão com o robô do LARA e o sistema de controle de reservas, estão em fase de finalização.

O LaraApp foi validado de acordo com um método de avaliação heurística de usabilidade para celulares touchscreen, na qual o aplicativo apresentou excelentes resultados. Sua alta usabilidade se deve a características como navegação simples e intuitiva, menu agradável e simples de usar, ícones com significados compreensíveis e intuitivos, ausência de travamentos durante a interação, fácil acesso a funções comuns do aplicativo, navegação padronizada para a plataforma destino, entre outras.

A respeito de trabalhos futuros, além do aprimoramento da interface, com base no feedback dos usuários e da avaliação do LaraApp como ferramenta de ensino e aprendizagem de programação, também existe a possibilidade de expansão da aplicação, mediante a integração com um laboratório virtual capaz de trabalhar com a linguagem de programação compatível com o robô do Lara.



## 7. REFERÊNCIAS BIBLIOGRÁFICAS

BARDINI, Augusto et al. Aplicativo Lúdico-Pedagógico para Ensino de Programação e Robótica Educacional. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2017. p. 222.

CAELUM. **Usabilidade em dispositivos móveis**. Disponível em: <<https://www.caelum.com.br/apostila-ux-usabilidade-mobile-web/usabilidade-mobile>>. Acesso em: 27 de jul. de 2019.

CALEGARI, Paulo Ferreira. **Aplicação da robótica no ensino-aprendizagem de lógica de programação para crianças**. 2015.

CASINI, Marco; PRATTICCHIZZO, Domenico; VICINO, Antonio. E-learning by remote laboratories: A new tool for control education. **IFAC Proceedings Volumes**, v. 36, n. 10, p. 73-78, 2003.

CORDOVA.APACHE. **Overview**. Disponível em: <<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>>. Acesso em: 22 de jul. de 2019.

COSTA, Ricardo. Tele-Experimentação Móvel (Mobile Remote Experimentation)- Considerações sobre uma área emergente no ensino à distância. **Journal of scientific activity at ISEP**, v. 15, 2005.

DA SILVA, Juarez B. et al. Uso de dispositivos móveis para acesso a Experimentos Remotos na Educação Básica. **VAEP-RITA**, v. 1, n. 2, p. 129-134, 2013.

DA SILVA, Marcelo Moro; SANTOS, Marilde Terezinha Prado. Os paradigmas de desenvolvimento de aplicativos para aparelhos celulares. **Revista TIS**, v. 3, n. 2, 2014.

DOS SANTOS LOPES, Maísa Soares et al. Web environment for programming and control of a mobile robot in a remote laboratory. **IEEE Transactions on Learning Technologies**, v. 10, n. 4, p. 526-531, 2016.

EXEL, Matija. et al. Simulation workshop and remote laboratory: Two web-based training approaches for control. In: **Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)**. IEEE, 2000. p. 3468-3472.

FEIJÓ, Valéria Casaroto; GONÇALVES, Berenice Santos; GOMEZ, Luiz Salomão Ribas. Heurística para avaliação de usabilidade em interfaces de aplicativos smartphones: Utilidade, produtividade e imersão. **Design e Tecnologia**, v. 3, n. 06, p. 33-42, 2013.

GOMES, Anabela de Jesus. **Dificuldades de aprendizagem de programação de computadores: contributos para a sua compreensão e resolução**. 2010. Tese de Doutorado.

HORITA, Flávio Eduardo Aoki; NETO, Valdemar Vicente Graciano; DOS SANTOS, Rodrigo Pereira. Design Science Research em Sistemas de Informação e Engenharia de Software: Conceitos, Aplicações e Trabalhos Futuros. In: André Luiz Satoshi Kawamoto; Ana Grasielle Dionísio Corrêa; Valéria Farinazzo Martins. (Org.). **I Jornada Latino-Americana de Atualização em Informática**, p. 191-210, 2018.

INEP. **Dados do Censo Escolar – Noventa e cinco por cento das escolas de ensino médio têm acesso à internet, mas apenas 44% têm laboratório de ciências**. Disponível em: <[http://portal.inep.gov.br/artigo/-/asset\\_publisher/B4AQV9zFY7Bv/content/dados-do-censo-escolar-noventa-e-cinco-por-cento-das-escolas-de-ensino-medio-tem-acesso-a-internet-mas-apenas-44-tem-laboratorio-de-ciencias/21206](http://portal.inep.gov.br/artigo/-/asset_publisher/B4AQV9zFY7Bv/content/dados-do-censo-escolar-noventa-e-cinco-por-cento-das-escolas-de-ensino-medio-tem-acesso-a-internet-mas-apenas-44-tem-laboratorio-de-ciencias/21206)>. Acesso em: 5 de jun. de 2019.

KRONE, Caroline. Validação de Heurísticas de Usabilidade para Celulares Touchscreen. **Florianópolis (SC): Universidade Federal de Santa Catarina, Grupo de Qualidade de Software**, 2013.

NEDIC, Zorica; MACHOTKA, Jan; NAFALSKI, Andrew. **Remote laboratories versus virtual and real laboratories**. IEEE, 2003.

NICOLETE, Priscila Cadorin et al. **Integração de tecnologia na educação: Grupo de Trabalho em Experimentação Remota Móvel (GT-MRE) um estudo de caso**. 2016.

NIELSEN, Jakob. Usability inspection methods. In: **Conference companion on Human factors in computing systems**. ACM, 1994. p. 413-414.

PAGOTTO, Tiago et al. Scrum solo: software process for individual development. In: **2016 11th Iberian Conference on Information Systems and Technologies (CISTI)**. IEEE, 2016. p. 1-6.

PRATES, Raquel Oliveira; BARBOSA, Simone Diniz Junqueira. Avaliação de interfaces de usuário—conceitos e métodos. In: **Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação, Capítulo**. 2003. p. 28.

RAABE, André Luís Alice; SILVA, JMC da. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. In: **XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil**. 2005.

RESTIVO, Maria Teresa; CARDOSO, Alberto. Exploring online experimentation. **International Journal of Online and Biomedical Engineering (iJOE)**, v. 9, n. S8, p. 4-6, 2013.

ROCHADEL, Willian; DE CARVALHO AQUINO, Ellen Larissa; SILVA, Juarez Bento. Desenvolvimento de aplicação para interfaceamento com experimentos remotos por smartphones. **RENOTE**, v. 10, n. 1, 2012.

SANTOS, Maiza; SOARES, Ingrid. **Com universidades sucateadas, Brasil cai nos ranking globais de avaliação**. Disponível em: <<https://www.correiobraziliense.com.br/app/noticia/brasil/2017/12/25/interna-brasil,649776/com-universidades-sucateadas-brasil-cai-nos-ranking-globais-de-avalia.shtml>>. Acesso em: 6 de jun. de 2019.

SQLITE. **About SQLite**. Disponível em: <<https://www.sqlite.org/about.html>>. Acesso em: 23 de jul. de 2019.