

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Lucas Ferreira da Silva

**IRRIGAÇÃO E MONITORAMENTO DE HORTA ESCOLAR COMO TEMA
DE INICIAÇÃO À COMPUTAÇÃO NA EDUCAÇÃO BÁSICA**

Santa Maria, RS
2018

Lucas Ferreira da Silva

**IRRIGAÇÃO E MONITORAMENTO DE HORTA ESCOLAR COMO TEMA DE INICIAÇÃO
À COMPUTAÇÃO NA EDUCAÇÃO BÁSICA**

Trabalho Final de Graduação apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

ORIENTADORA: Prof.^a Andrea Schwertner Charão

©2018

Todos os direitos autorais reservados a Lucas Ferreira da Silva. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

End. Eletr.: lferreira@inf.ufsm.br

Lucas Ferreira da Silva

**IRRIGAÇÃO E MONITORAMENTO DE HORTA ESCOLAR COMO TEMA DE INICIAÇÃO
À COMPUTAÇÃO NA EDUCAÇÃO BÁSICA**

Trabalho Final de Graduação apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

Aprovado em 3 de dezembro de 2018:

Andrea Schwertner Charão, Dra. (UFSM)
(Orientadora)

Benhur de Oliveira Stein, Dr. (UFSM)

João Vicente Ferreira Lima, Dr. (UFSM)

Santa Maria, RS
2018

AGRADECIMENTOS

Agradeço primeiramente a meus pais, Sérgio e Ester, e a meu irmão Joel. Obrigado pelo amor, carinho e apoio incondicionais que sempre me proporcionaram durante toda a minha vida.

Agradeço aos meus colegas, hoje meus amigos, pela parceria de todos os momentos, seja nas aulas, nas maratonas de programação, nas viagens, nas jantares, nos momentos de felicidade e também nos momentos de dificuldade. Obrigado Ana, Cassiano, Iago, João, Otávio, Pedro, Rafael e Vinícius, a amizade de vocês é, sem dúvidas, um dos melhores frutos que levarei desta caminhada.

Agradeço a Universidade Federal de Santa Maria pela estrutura, pelas oportunidades e, principalmente, pelos programas de assistência estudantil oferecidos, os quais foram decisivos para a minha permanência e conclusão em um curso de graduação.

Agradeço a escola Marista Santa Marta, pela disponibilização do ambiente e dos alunos para que fosse possível o desenvolvimento deste trabalho. Agradecimento especial ao professor Leandro, por sua disposição e entusiasmo na realização da proposta.

Agradeço também, os professores e tutores do Clube de Computação, pelo auxílio nas atividades desenvolvidas durante as visitas à escola Marista Santa Marta.

Por fim, não menos importante, agradeço à professora Andrea, por ter acreditado em meu potencial para a realização deste trabalho. Sua orientação e auxílio foram certamente fundamentais para que, apesar de alguns contratemplos, eu conseguisse concluir esta etapa de minha graduação.

Science is but a perversion of itself unless it has as its ultimate goal the betterment of humanity.

(Nikola Tesla)

RESUMO

IRRIGAÇÃO E MONITORAMENTO DE HORTA ESCOLAR COMO TEMA DE INICIAÇÃO À COMPUTAÇÃO NA EDUCAÇÃO BÁSICA

AUTOR: Lucas Ferreira da Silva

ORIENTADORA: Andrea Schwertner Charão

Nos tempos atuais, com os constantes avanços tecnológicos, cada vez fica mais evidente a necessidade e a importância de se saber programar. Escolas são levadas a iniciar digitalmente seus alunos desde cedo, porém, muitas vezes, essa introdução fica apenas no âmbito de ensinar como as crianças e jovens podem usufruir dos recursos tecnológicos, não os incentivando a serem os modificadores ou até mesmo os criadores de tais recursos. Este cenário é decorrente, na maioria das vezes, da dificuldade de se abordar os conteúdos relacionados a programação dentro da sala de aula, pois ela ainda é vista como algo muito complexo até mesmo pelos professores e, para contornar esse problema, são necessárias estratégias didáticas mais eficientes e dinâmicas. Pensando nisso, no presente trabalho, pretende-se explorar o desenvolvimento de um sistema de irrigação e monitoramento de uma horta escolar como tema para o ensino de programação em escolas de educação básica, com o intuito de unir a teoria e a prática em uma atividade que possibilite uma maior dinamicidade no processo de ensino. Para a concretização da iniciativa, foi realizada uma busca e seleção sistemáticas dos elementos necessários para se trabalhar com crianças e jovens, desde recursos de hardware para que o projeto de automação seja viável enquanto estratégia didática, como também ferramentas de software que possibilitem que os alunos programem os componentes do sistema em questão. Estes recursos estão organizados e disponíveis em diversos formatos e meios, visando seu reaproveitamento em diferentes tempos e espaços.

Palavras-chave: Ensino de programação. Automação. Horta escolar.

ABSTRACT

IRRIGATION AND MONITORING OF A SCHOOL GARDEN AS A THEME OF INITIATION TO COMPUTING IN BASIC EDUCATION

AUTHOR: Lucas Ferreira da Silva
ADVISOR: Andrea Schwertner Charão

In current times, with the constant technological advances, it becomes increasingly evident the need and the importance of knowing how to programming. Schools are led to digitally start their students increasingly early, but often this introduction is only in the context of teaching how children and young people can enjoy the technological resources, not encouraging them to be the modifiers or even the creators of such resources. This scenario is due, in the most times, to the difficulty of approaching content related to programming within the classroom, because this theme it is still seen as something very complex even by teachers and, to overcome this problem, more effective and dynamic didactic strategies are needed. In this sense, the present work intends to explore the development of a system of irrigation and monitoring of a school garden as a theme for the teaching of programming in schools of basic education, with the intention of uniting theory and practice in an activity that allows a greater dynamism in the teaching process. In order to implement the initiative, was carried out a systematic search and a methodical selection of the elements necessary to work with children and young people, since hardware resources in order to make the automation project feasible as a didactic strategy, as well as software tools that enable students to program the components of the system in question. These resources are organized and available in different formats and means, aiming at reusing them in different times and spaces.

Keywords: Programming Teaching. Automation. School garden.

LISTA DE FIGURAS

Figura 2.1 – Estrutura de programa utilizando a plataforma Ardublock.	14
Figura 2.2 – Placa de prototipagem de hardware Arduino UNO.	16
Figura 2.3 – Placa de prototipagem de hardware ESPduino.	16
Figura 3.1 – Diagrama da estrutura arquitetural do projeto de irrigação e monitoramento de uma horta escolar.	23
Figura 3.2 – Diagrama da estrutura da implementação do Ardublock. Nele estão esquematizados os aspectos mais relevantes da implementação do Ardublock	25
Figura 3.3 – Diagrama que descreve o funcionamento e a tradução dos blocos visuais para código em C/C++	26
Figura 3.4 – Exemplo de definição de bloco com a tag <BlockGenus> no arquivo “ardublock.xml”.	27
Figura 3.5 – Imagem do bloco de requisições HTTP do tipo POST para números.	28
Figura 3.6 – Imagem do bloco de requisições HTTP do tipo POST para textos.	28
Figura 3.7 – Imagem do bloco de requisições HTTP do tipo GET.	28
Figura 3.8 – Diagrama ilustrativo de como são utilizadas as bibliotecas escritas em C++ no Arduino IDE e como o Ardublock atua para utilizá-las	30
Figura 3.9 – Tela onde que a lógica dos aplicativos pode ser criada utilizando a linguagem visual de blocos.	34
Figura 3.10 – Aplicativo de exemplo criado utilizando a plataforma Thunkable.	34
Figura 4.1 – Processo de instalação do sistema de irrigação e monitoramento da horta na primeira visita a escola.	37
Figura 4.2 – Local destinado a instalação do projeto. Por se localizar próximo a uma residência, o acesso a água ficou facilitado.	38
Figura 4.3 – Plaquinha disposta no canteiro.	39
Figura 4.4 – Grupo 2 realizando as anotações.	40
Figura 4.5 – Cartaz simbolizando o “banco de dados” onde os valores das plaquinhas foram colocados.	40
Figura 4.6 – Escala avaliativa usada na atividade.	41
Figura 4.7 – Gráfico que representa a porcentagem de respostas referente a satisfação com a atividade desenvolvida.	42
Figura 4.8 – Gráfico que representa a porcentagem de respostas referente a motivação com a atividade desenvolvida.	42

LISTA DE TABELAS

Tabela 3.1 – Especificações de hardware das placas de prototipagem.....	19
Tabela 3.2 – Arquivos modificados para a adição de novos blocos.....	29
Tabela 3.3 – Arquivos adicionados para criação dos blocos.....	29

LISTA DE ABREVIATURAS E SIGLAS

<i>HTTP</i>	<i>HyperText Transfer Protocol</i>
<i>JSON</i>	<i>JavaScript Object Notation</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>REST</i>	<i>Representational State Transfer</i>
<i>XML</i>	<i>Extensible Markup Language</i>
<i>SRAM</i>	<i>Static Random Access Memory</i>
<i>DRAM</i>	<i>Dynamic Random Access Memory</i>
<i>E/S</i>	<i>Entrada/Saída de dados</i>
<i>GPIO</i>	<i>General Purpose Input/Output</i>
<i>PWM</i>	<i>Pulse Width Modulation</i>
<i>I2C</i>	<i>Inter-Integrated Circuit</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVO	12
1.1.1	Objetivos Específicos	12
1.2	JUSTIFICATIVA	13
2	REVISÃO DA LITERATURA	14
2.1	PROGRAMAÇÃO VISUAL ATRAVÉS DE BLOCOS	14
2.2	PLATAFORMAS DE PROTOTIPAGEM NO ENSINO DE PROGRAMAÇÃO	15
2.2.1	Arduino	15
2.2.2	ESPduino	16
3	DESENVOLVIMENTO	18
3.1	ESCOLHA DOS MATERIAIS E RECURSOS	18
3.1.1	Hardware	18
3.1.2	Ambiente de programação	21
3.2	VISÃO GERAL DO PROJETO	22
3.3	VERSÃO PERSONALIZADA DO ARDUBLOCK	24
3.3.1	Estrutura de implementação do Ardublock	24
3.3.2	Adição de novos blocos	26
3.3.3	Ajuste das bibliotecas de I/O para o ESPduino	30
3.4	COMUNICAÇÃO COM UM SERVIDOR EXTERNO	31
3.5	PROPOSTA DE APLICATIVO PARA MONITORAR A HORTA	32
3.6	CRIAÇÃO DE MATERIAL DE APOIO PARA O PÚBLICO ALVO	35
4	RESULTADOS E DISCUSSÃO	37
4.1	IMPLANTAÇÃO DO PROJETO EM UMA ESCOLA	37
4.2	REALIZAÇÃO DE ATIVIDADE DESPLUGADA	39
4.3	AVALIAÇÃO DA ACEITAÇÃO DOS ALUNOS	41
5	CONSIDERAÇÕES FINAIS	44
	REFERÊNCIAS BIBLIOGRÁFICAS	45

1 INTRODUÇÃO

A tecnologia vem revolucionando a humanidade: a cada dia passa, uma nova solução digital surge para facilitar a vida das pessoas. Inovações tecnológicas, para os mais variados fins, surgem a todo momento, seja como um novo aplicativo para o smartphone ou até mesmo na forma de alguma ferramenta para solução de algum problema cotidiano.

Todavia, para um indivíduo acompanhar essa evolução de velocidade exponencial, já não basta mais ser apenas um bom usuário, é necessário compreender como essa realidade funciona e saber manuseá-la a fim de obter o melhor que ela tem a oferecer. Desta forma, saber programação tem se tornado indispensável, pois é a partir dela que passa a ser possível moldar os recursos digitais a favor dos seus usuários.

Neste contexto, há um crescente incentivo para que escolas não se atenham apenas na introdução dos seus alunos ao meio digital, mas sim os preparem para serem parte dos construtores dessa realidade. Para isso, é necessário que o pensamento computacional seja incorporado ao currículo das escolas e os preceitos da lógica de programação sejam apresentados desde cedo, fazendo com que esse conhecimento seja construído junto ao desenvolvimento escolar do aluno, pois, como é reforçado por (MARTINEZ; GOMEZ; BENOTTI, 2015), mesmo alunos de faixas etárias mais baixas (5 e 6 anos de idade) são capazes de aprender os fundamentos básicos da lógica computacional e aplicá-la em algum contexto.

No entanto, um dos maiores desafios ao se trabalhar com crianças e jovens é acerca da didática utilizada para ensinar os conceitos relacionados a programação, já que esse tema ainda é visto como algo complexo até mesmo para os professores, cuja experiência na abordagem desse tema, é fator influenciador, seja de forma positiva ou até mesmo negativa, no processo de aprendizagem do aluno, reafirma (FAGIN; MERKLE, 2003). Para contornar esse obstáculo, é importante que se adotem didáticas mais dinâmicas, que se aproximem do dia a dia dos alunos, mostrando a aplicabilidade e a utilidade do que lhes está sendo ensinado.

Dentre as várias metodologias de ensino de programação, ultimamente uma abordagem que vem ganhando força é a do movimento maker, em que os alunos são instigados a pôr a mão na massa durante o processo de aprendizagem, objetivando a criação de um ambiente educacional mais colaborativo e que explore a criatividade e a autonomia das crianças e jovens. Segundo (BLIKSTEIN, 2003), as possibilidades desse método estão associadas com teorias pedagógicas de Seymour Papert e Paulo Freire, em que a escola se torna mais conectada com a realidade do jovem e com os problemas que ele enfrenta no cotidiano, o que contribui para o seu aprendizado.

O movimento maker, conforme (SABIN, 2017), tem sido visto como uma das mais promissoras alternativas para a mudança do sistema educacional atual, já que suas carac-

terísticas o tornam de fácil integração com várias disciplinas e influencia significativamente na compreensão dos mais variados conteúdos, pois faz com que o aluno interaja e busque o entendimento mais profundo de como e o porquê da necessidade de se aprender determinada matéria.

Desta forma, recursos como a plataforma Arduino, Lego, impressoras 3D, entre outros, passaram a ser grandes aliados na concretização da educação maker, pois possibilitam a criação, prototipação, montagem, programação e reprogramação dos diversos projetos desenvolvidos pelos alunos que, com esses recursos palpáveis, podem transformar sua imaginação em algo físico, além de contribuir para que os mesmos identifiquem problemas relevantes ao meio ao qual estão inseridos e busquem soluções colaborativas para essas necessidades.

1.1 OBJETIVO

O presente trabalho possui como principal objetivo apresentar uma proposta de ensino de programação para alunos do ensino médio, utilizando ferramentas de aprendizado de programação e/ou robótica para tornar tal tarefa mais dinâmica e atrativa para os alunos. O contexto das atividades a serem realizadas será o projeto da automatização da irrigação e monitoramento de uma horta escolar.

1.1.1 Objetivos Específicos

- Realizar uma busca referente as alternativas de hardware necessárias para a implementação do sistema de irrigação e monitoramento da horta escolar, levando em consideração aspectos como custo, disponibilidade, eficiência e quantidade de recursos;
- Investigar ferramentas que abstraíam a complexidade da programação dos sensores e atuadores do projeto, visando viabilizar o sistema de automação em questão para ser utilizado também para fins didáticos;
- Explorar o cenário de uma horta escolar enquanto ambiente passível de automação, tanto no que tange a coleta de dados que podem ser utilizados posteriormente, quanto no âmbito de automatização de atividades de manejo, como a irrigação.
- Obter um sistema que possibilite o monitoramento e coleta de alguns dados da horta em questão, possibilitando assim, criar uma base de dados que permita inferir informações importantes do ambiente e sua influência nas plantas, bem como propiciar o

uso de tais informações no âmbito escolar de forma multidisciplinar;

- Instigar os alunos a propor melhorias para o projeto de irrigação e monitoramento da horta escolar a partir dos dados que vão sendo coletados pelas soluções que os próprios alunos vão desenvolvendo no decorrer das atividades.

1.2 JUSTIFICATIVA

Atualmente saber programação já é quase que indispensável para acompanhar e usufruir das novas facilidades que surgem diariamente. Assim, os conceitos relativos à lógica computacional estão sendo introduzidos para as crianças e jovens cada vez mais cedo, principalmente em países desenvolvidos. Porém, a programação muitas vezes pode ser vista como um conteúdo demasiadamente complexo, de modo que professores enfrentam o desafio de escolher a abordagem a ser adotada para que os alunos não vejam a programação como algo complicado, descontextualizado e, por sua vez, maçante. Uma abordagem que promova o aprendizado significativo é desejável e pode fazer toda diferença no processo de aprendizagem das crianças e dos jovens.

Neste contexto, uma das alternativas baseia-se na transmissão dos preceitos de programação de uma forma lúdica e ao mesmo tempo palpável, em que recursos como placas de prototipagem, como a plataforma Arduino e ESPduino, são os instrumentos principais para aliar a teoria com a prática. Isso possibilita que se explore não somente conhecimentos relacionados à programação, mas também à integração desses com o ambiente no qual a escola e o aluno estão inseridos. Além disso, conforme (LITTS et al., 2017), trabalhar com esses recursos pode ser inicialmente um tanto desafiador para os alunos, porém, com uma abordagem de ensino colaborativa onde todos constroem o conhecimento e aprendem juntos, é mais fácil de sanar as possíveis diferenças de aprendizado que podem surgir entre os alunos e, por consequência, a turma progride junto.

Um desses ambientes é a horta escolar, que possui um grande potencial interdisciplinar que, muitas vezes, não é suficientemente explorado e acaba sendo esquecido. Desta forma, aliar os recursos de sensores e atuadores das plataformas Arduino e ESPduino, com o ensino de programação, ambos aplicados no contexto de uma horta escolar, pode revelar uma estratégia didática que promove a interdisciplinaridade, coletividade e criatividade de uma forma dinâmica e aplicada no cotidiano do aluno.

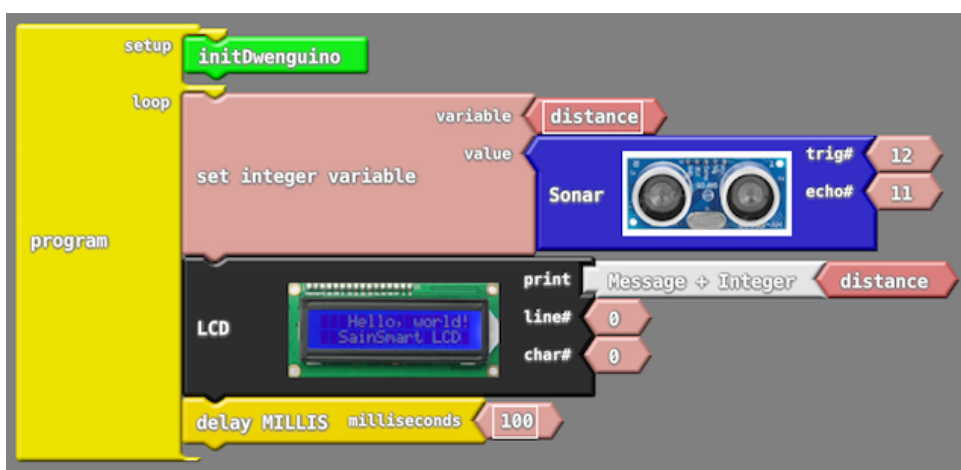
2 REVISÃO DA LITERATURA

Neste capítulo é apresentada uma breve descrição de algumas abordagens e ferramentas utilizadas para o desenvolvimento do trabalho. A seção 2.1 trata da utilização da programação visual através de blocos como um recurso para abstrair a complexidade dos comandos de uma linguagem de programação e suas potencialidades enquanto facilitador didático. Na seção 2.2, são apresentadas algumas opções de placas de prototipagem de hardware e como recursos desse tipo são empregados no ensino de programação.

2.1 PROGRAMAÇÃO VISUAL ATRAVÉS DE BLOCOS

A programação visual pode ser vista como uma forma de abstração da parte léxica e/ou sintática dos comandos de uma linguagem de programação convencional, transformando as complexas sentenças em recursos visuais que consigam expressar a mesma lógica da linguagem só que de uma forma mais clara. Dentre os recursos visuais geralmente utilizados nesse tipo de abordagem estão os blocos de encaixe, que transformam cada parte do código em um bloco que pode ser ligado a outros, possibilitando assim que a lógica do algoritmo possa ser construída como se fosse um quebra-cabeças.

Figura 2.1 – Estrutura de programa utilizando a plataforma Ardublock.



Fonte: Adaptado de Ardublock (2018).

O recurso de programação visual utilizado neste trabalho é o de programação visual através de blocos, utilizando a plataforma Ardublock (ARDUBLOCK, 2018), uma aplicação desenvolvida por meio da linguagem Java e que objetiva abstrair de forma visual os códigos da linguagem de programação utilizada por placas de prototipagem como Arduino e ESPduino.

O Ardublock pode ser facilmente instalado na IDE do Arduino, sendo necessário apenas o download dos arquivos (disponíveis no site do desenvolvedor) e posteriormente incorporar na IDE como uma nova ferramenta. Sua utilização é bastante simples, muito semelhante a outros recursos mais populares como por exemplo o Blockly, além de possuir abstração para quase a totalidade das funções da linguagem utilizada pelo Arduino e seus similares.

2.2 PLATAFORMAS DE PROTOTIPAGEM NO ENSINO DE PROGRAMAÇÃO

As plataformas de prototipagem de hardware surgem como ferramentas facilitadoras para o projeto, prototipagem e aprendizado dos preceitos de eletrônica e microcontroladores. Tais plataformas permitem que com uma mesma placa de prototipagem, sejam realizados vários projetos distintos, de maneira simples e com um baixo investimento.

Dada as facilidades propiciadas por essas plataformas para a confecção e aprendizado de novos projetos de hardware, elas acabaram se popularizando como alternativas para o ensino de conceitos básicos tanto de eletrônica quanto de programação. Para o ensino de programação, a maior contribuição desses recursos está na dinamicidade que proporcionam, pois possibilitam que o aluno possa manusear, montar, programar e reprogramar tais artefatos, tornando visível e palpável a lógica empregada pelo aluno, fazendo com que o mesmo assimile de forma mais efetiva os conceitos em questão.

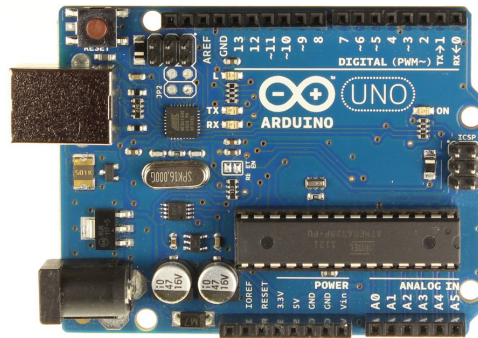
No presente projeto, as plataformas Arduino e ESPduino foram selecionadas como as alternativas a serem utilizadas como os recursos auxiliares no ensino de programação.

2.2.1 Arduino

Arduino é uma plataforma para prototipagem e projeto de eletrônica e hardware em geral que foi criada em 2005, por um grupo de pesquisadores formado por: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. A plataforma segue o conceito de hardware e software livre, possibilitando que qualquer pessoa possa montar, modificar e personalizar a plataforma, partindo da especificação de hardware base do projeto (ARDUINO, 2018).

A placa é composta por um microcontrolador Atmel AVR com suporte para entrada e saída, e pode ser conectada a um computador via USB e reprogramada por meio de uma linguagem de programação baseada em C/C++. Após programado o microcontrolador pode ser utilizado de forma independente para os mais diversos fins, além de possibilitar a integração com variados módulos, conhecidos como shields, que estendem as

Figura 2.2 – Placa de prototipagem de hardware Arduino UNO.



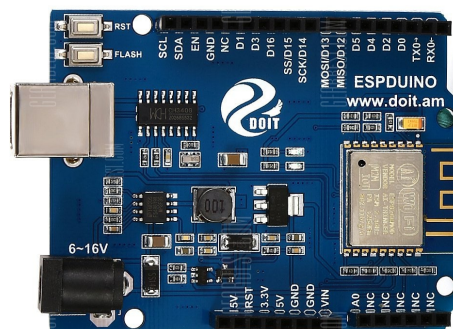
Fonte: Adaptado de Arduino (2018).

funcionalidades da placa com recursos como rede e sensores.

2.2.2 ESPduino

O ESPduino (Shenzhen Doctors of Intelligence and Technology Co. Ltd, 2018), assim como o Arduino, é uma placa de prototipagem para projetos em eletrônica, porém com o diferencial de possuir wifi integrado. Essa placa é desenvolvida pela empresa chinesa Doit (DOIT, 2018) e possui como microcontrolador o ESP8266, desenvolvido pelo fabricante chinês Espressif, que possibilita capacidade de comunicação por Wi-Fi no seu chip por meio de conexões TCP/IP e utilizando um conjunto de instruções da linguagem de comandos Hayes.

Figura 2.3 – Placa de prototipagem de hardware ESPduino.



Fonte: Adaptado de DOIT (2018).

Quando lançado no mercado, o ESP8266 não fez muito sucesso por não possuir documentação em inglês, porém, em outubro de 2014, a Espressif disponibilizou um kit de desenvolvimento de software (SDK) que permitiu que o chip fosse programado diretamente, sem a necessidade de um microcontrolador intermediário como vinha sendo feito até então. Desta forma, começaram a serem exploradas as potencialidades do ESP8266 e

percebeu-se que sua capacidade era equivalente ou até mesmo superior a do microcontrolador Atmel presente nos Arduinos aos quais ele era utilizado como um simples módulo de wifi e, assim, surgiram projetos para a utilização do ESP8266 como um microcontrolador independente com os mesmos recursos dos Arduinos comuns.

Dentre os projetos de uso do ESP8266, surgiu o ESPduino, que se baseia na estrutura da placa Arduino convencional, porém com o ESP8266 como microcontrolador ao invés do Atmel AVR, além do recurso de wifi já integrado a placa e a possibilidade de ser programado na linguagem de programação e IDE utilizados com o Arduino. Sendo assim, o ESPduino pode ser utilizado da mesma forma que um Arduino convencional, possibilitando inclusive , a adição módulos de expansão de recursos.

3 DESENVOLVIMENTO

Neste capítulo serão explicadas quais atividades foram desenvolvidas no decorrer do trabalho a fim de se alcançar o objetivo proposto. Primeiramente será abordado o processo de escolha dos materiais mais adequados a serem utilizados no projeto (seção 3.1). Em seguida, é apresentada uma visão geral do projeto de irrigação e monitoramento (seção 3.2), juntamente com as adaptações nos recursos de software para tornar o projeto viável (seção 3.3). Além disso, é descrita como é realizada a comunicação do sistema de irrigação com um servidor externo (seção 3.4), a fim de possibilitar que os dados coletados da horta estejam acessíveis para o monitoramento remoto, o que oportuniza a criação de ferramentas para realizar esse monitoramento, como na proposta de criação de aplicativo descrita na seção 3.5.

Por fim, na seção 3.6, é relatado como todo o material de apoio desenvolvido foi disponibilizado para a replicação do projeto.

3.1 ESCOLHA DOS MATERIAIS E RECURSOS

Para a concretização do projeto pretendido no presente trabalho, em que se propõe a integração de ferramentas de hardware e software para fins didáticos, faz-se necessária uma pesquisa em busca de recursos que possibilitem a implementação de uma solução funcional e ao mesmo tempo replicável. Assim, para a escolha dos materiais de hardware e software, as opções disponíveis foram sistematicamente avaliadas, considerando os principais critérios que geralmente definem a viabilidade de uma proposta do tipo. Nas seções seguintes, são apresentadas as escolhas dos recursos de hardware e de software utilizados neste trabalho, além da descrição de quais critérios que influenciaram tais escolhas.

3.1.1 Hardware

Na escolha das plataformas de prototipagem foram consideradas duas opções: a plataforma Arduino e a plataforma ESPduino, ambas amplamente utilizadas atualmente em projetos de automação e para o ensino de eletrônica e robótica.

Enquanto placas de prototipagem, tanto Arduino quanto ESPduino se equiparam em várias características, como por exemplo, a disposição das conexões na placa e a forma como ambas podem ser programadas. Entretanto, ao se fazer uma avaliação mais aprofundada, comparando as especificações de cada placa, é possível perceber algumas

particularidades que, dependendo do projeto, podem classificar uma placa como mais adequada que outra.

Analisando a tabela de especificações (Tabela 3.1), no quesito de frequência de *clock* o ESPduino apresenta características superiores ao Arduino, sendo que a frequência máxima de *clock* do microcontrolador do ESPduino é em torno de 8 vezes maior que a do microcontrolador do Arduino. No que se refere às características de memória, o ESPduino também supera o Arduino, possuindo memória flash de 4MB e SRAM de 64KB, contra os 32KB de memória flash e 2KB SRAM do Arduino. Tais características fazem do ESPduino uma melhor opção para projetos que necessitem melhor desempenho a nível do processamento das instruções do programa, além de possibilitar o armazenamento de programas maiores e utilização de um maior número de variáveis em cada programa.

Tabela 3.1 – Especificações de hardware das placas de prototipagem.

Especificação	Arduino UNO R3	ESPduino
Frequência Máxima de Clock	20MHz	160MHz
Microcontrolador	ATmega328	ESP8266
Memória Flash	32KB	4MB
SRAM	2 KB	64 KB SRAM / 96KB DRAM
USB	Sim	Sim
Conector para fonte de energia externo	Sim	Sim
Pinos	32	26
GPIO	14	13 (9 E/S Digital)
E/S Analógica	6	1
WiFi	Não	Sim
UART	1	1
E/S PWM	6	9
I2C	Sim	Sim

Tratando-se da “pinagem” das placas, o Arduino possui melhores especificações: 14 GPIOs e 6 portas analógicas, de um total de 32 pinos ativos. Já o ESPduino possui um total de apenas 26 pinos ativos, dos quais 13 são GPIOs (9 destinadas a E/S digital) e apenas 1 porta analógica. Desta forma, o Arduino tende a ser uma melhor opção para projetos que demandem um maior número de pinos, sem que seja necessário o acoplamento de componentes como multiplexadores de portas ou módulos de expansão de número de portas da placa de prototipagem.

Em termos de comunicação com a internet, o ESPduino possui o diferencial de ter a tecnologia wifi nativamente integrada a placa, sem a necessidade de um módulo adicional para fornecer acesso a internet, como é necessário no Arduino, que não possui tal recurso.

Além da avaliação comparativa dos recursos de cada placa, segundo suas especificações, alguns outros critérios também são importantes de serem considerados na escolha de uma das placas de prototipagem para a realização do projeto proposto no trabalho. Tais critérios são pontuados e descritos a seguir:

- **Disponibilidade no mercado:** a plataforma Arduino é mundialmente difundida, sendo de fácil aquisição em qualquer país e comercializado por várias lojas especializadas em comércio eletrônico. Trata-se de uma plataforma mais antiga, desenvolvida por italianos, que ganhou fama na época de seu lançamento por possibilitar projetos de baixo custo. Em contrapartida, o ESPduino não é uma plataforma tão difundida quanto o Arduino, já que o mesmo é relativamente novo. Trata-se de um projeto de origem chinesa e, no Brasil sua disponibilidade no mercado ainda não é tão grande, porém está em ascensão.
- **Facilidade de utilização:** o Arduino e o ESPduino possuem um grau de facilidade de utilização equiparável, tanto a nível de recursos de hardware quanto para programação. O ESPduino foi projetado visando aproveitar as facilidades que o Arduino possibilita, propiciando a compatibilidade e disposição da “pinagem”, além da possibilidade de ambas plataformas serem programadas utilizando o Arduino IDE. Porém, ao se utilizar o Arduino IDE para a programação, é necessária a instalação de alguns complementos e bibliotecas adicionais para a utilização como o ESPduino. Quanto a utilização de programação visual por blocos, mais especificamente com a plataforma Ardublock, os blocos existentes para Arduino são compatíveis com o ESPduino, porém, ao se utilizar recursos de comunicação com a internet via wifi, não há blocos genéricos que atendam a variedade de possíveis módulos do Arduino para esse recurso e, conseqüentemente, também não há blocos especializados para comunicação via wifi com o ESPduino.
- **Custo:** no que tange ao custo de aquisição, podem haver divergências dependendo da versão e do microcontrolador de cada placa de prototipação. No presente trabalho, utilizou-se para comparação uma placa Arduino Uno R3 com microcontrolador ATMEL ATmega328 e um ESPduino com microcontrolador ESP8266, pois essas placas possuem configurações semelhantes, o que torna a comparação mais justa. Assim, dentre as duas placas comparadas, a variação de preço é pequena: em média, o valor é praticamente o mesmo. Entretanto, as novas placas ESPduino estão sendo fabricadas utilizando microcontroladores mais novos como o ESP32, os quais possuem desempenho bastante superior aos microcontroladores ATmega dos Arduinos. Seu preço no mercado brasileiro é um pouco acima do preço das placas Arduino com ATmega328, equivalente ao preço das placas com ATmega2560, versão mais robusta dos microcontroladores do Arduino.

Ao analisar todos os critérios pontuados, concluiu-se que o ESPduino é, para o presente projeto, uma melhor opção se comparado ao Arduino, sendo que ambos se equivalem quanto ao preço e usabilidade, porém o ESPduino conta com recursos que se enquadram melhor na proposta deste trabalho.

Mesmo com a restrição de número de pinos, principalmente das portas analógicas, a utilização do ESPduino ainda é viável para o trabalho, já que o sistema de irrigação e monitoramento proposto não demanda tantas conexões com módulos de expansão ou sensores. De fato, o maior obstáculo seria na leitura dos dados provenientes dos sensores de umidade do solo que, para uma maior precisão, utilizam as portas analógicas. Contudo, esse problema pode ser contornado com a utilização das portas digitais, pois os sensores de umidade do solo possibilitam que os dados provenientes de uma leitura sejam em formato digital, com o porém de apenas ser detectado se o solo está seco ou úmido. A definição de qual valor lido pelo sensor distingue um solo úmido de um solo seco, é realizada a partir de um *trimpot* de ajuste presente no próprio sensor, que possibilita a calibração dos níveis lógicos que serão enviados para a porta digital da placa de prototipagem.

Outro fator que influenciou na escolha do ESPduino é o recurso de wifi embutido na placa nativamente, pois elimina a necessidade de um módulo adicional para prover a comunicação via wifi e também torna a montagem do circuito mais simplificada, já que módulos para essa funcionalidade geralmente necessitam da adição de resistores e reguladores de tensão ao projeto. Além do mais, esse diferencial do ESPduino torna o projeto um pouco mais barato, porque não é necessária a preocupação com a compra de um módulo de expansão para comunicação, o que se somaria ao valor do projeto.

3.1.2 Ambiente de programação

Ao se tratar de um projeto que será utilizado no ensino de programação como um recurso didático, há que se considerar formas de abstração da complexidade dos códigos da programação das placas de prototipagem. Quando usadas linguagens de programação textuais como C/C++ (programação do Arduino e ESPduino), alunos recém iniciados na lógica de programação costumam ter dificuldades com expressões e estruturas particulares da linguagem. Em muitos casos, essa dificuldade se torna um obstáculo no aprendizado de programação.

Uma alternativa neste cenário é o uso de uma linguagens de programação visual, que possibilita que os algoritmos sejam descritos sob a forma de blocos que se encaixam, permitindo que a lógica do algoritmo seja construída como se fosse uma espécie de quebra-cabeças.

Contudo, para o uso na programação de placas de prototipagem, não há muitas opções de plugins e bibliotecas para programação visual disponíveis, sendo que todas derivam da plataforma Ardublock. Desta forma, como optou-se pelo uso do ESPduino para o desenvolvimento do projeto proposto, foram pesquisadas plataformas derivadas do Ardublock que fornecessem suporte de programação visual para o microcontrolador ESP8266. Assim, foi encontrado um projeto disponível em um repositório do usuário “karlTH” do

GitHub (KARLTH, 2018), que se propunha a ser uma solução derivada do Ardublock para o ESP8266.

O projeto disponível no referido repositório conta com blocos para quase todos os recursos necessários para a proposta deste trabalho. Porém, alguns recursos de comunicação via HTTP foram implementados por karlTH utilizando como base as bibliotecas de IO para os microcontroladores ESP8266 modificados pela empresa Adafruit (ADAFRUIT... , 2018), tornando o uso do Ardublock modificado por karlTH incompatível em alguns recursos com o ESPduino. Também constatou-se a ausência de alguns blocos responsáveis pelas instruções de requisições HTTP para um servidor externo ao ESPduino. Essa funcionalidade é de grande importância para monitoramento dos dados coletados da horta escolar, pois os mesmos serão posteriormente enviados, por meio de requisições HTTP do tipo POST, para uma API RESTful onde serão armazenados, para posteriormente serem disponibilizados para consulta via requisições HTTP do tipo GET.

3.2 VISÃO GERAL DO PROJETO

O projeto proposto pelo presente trabalho depende da integração de vários recursos, tanto de software quanto de hardware. O esquema da Figura 3.1 apresenta, de forma sintetizada, como o projeto está arquiteturalmente organizado, possibilitando que se tenha uma visão geral do que se está sendo proposto.

No esquema da Figura 3.1 está representada a horta escolar na qual o projeto está instalado e realizando a irrigação e monitoramento. Nela estão dispostos tanto os sensores quanto os atuadores.

Os sensores desempenham o papel de coletar os dados da horta, permitindo que informações do ambiente, como umidade do solo, luminosidade e temperatura sejam coletadas em forma de números para que possam ser enviados ao ESPduino. Ao se tratar de um projeto de irrigação e monitoramento de uma horta, um dos sensores a ser usado seria o de umidade do solo (higrômetro), pois com ele é possível saber se o solo precisa ou não ser irrigado, algo trivial de se controlar em um sistema desse tipo. A interação dos sensores com o ambiente em que estão situados é tida como direta, já que para se obter os dados é necessário o contato direto com a fonte dessa informação.

Ainda na horta, se situam os atuadores, eles são responsáveis por realizar as ações que modificarão o estado e/ou as condições da horta, como por exemplo, acionar a válvula que fornece água para as mangueiras de irrigação, modificando assim a umidade do solo. Os atuadores, assim como os sensores, trocam informações com o ESPduino, porém os atuadores são controlados pelas instruções enviadas pela placa de prototipagem. No caso desse projeto, o comportamento dos atuadores depende dos dados que os sensores estão enviando ao ESPduino que, após a aplicação de uma lógica sobre o que está sendo

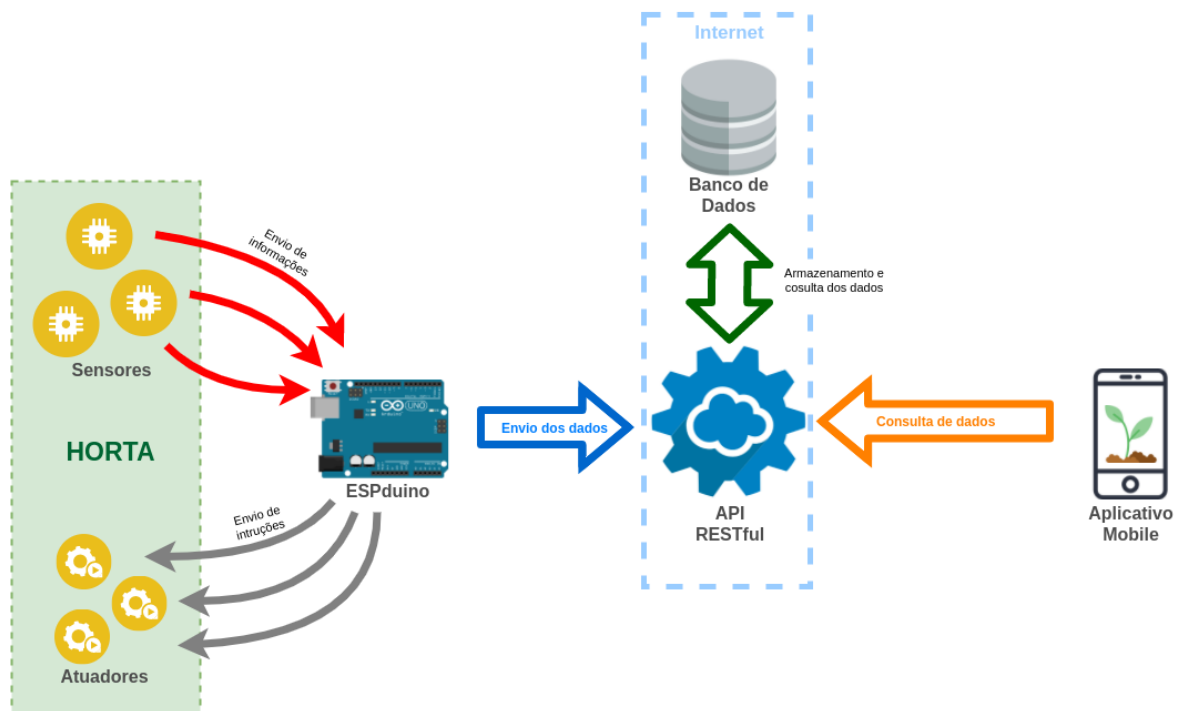


Figura 3.1 – Diagrama da estrutura arquitetural do projeto de irrigação e monitoramento de uma horta escolar.

coletado pelos sensores, é capaz de decidir quando ativar ou desativar um atuador.

No esquema, é possível também perceber que o ESPduino desempenha um papel de fundamental importância para o projeto, pois é nele que se concentra toda a lógica que define as interações com a horta, seja no que se refere ao sistema de irrigação ou até mesmo na parte de monitoramento. Essa lógica é definida pela programação do microcontrolador do ESPduino utilizando uma linguagem para tal (geralmente C/C++), porém, como mencionado em seções anteriores, a proposta desse projeto é justamente abstrair possíveis obstáculos para o aprendizado de programação e, por isso, a ideia é que o ESPduino possa ser programado utilizando uma linguagem de programação visual, como a da plataforma ArduBlock.

Com o conjunto dos recursos ESPduino, sensores e atuadores, viabiliza-se a parte de irrigação do projeto, sendo esses, os componentes suficientes para se ter um sistema automatizado, mesmo que simples, para o controle de umidade do solo da horta. Entretanto, a proposta do projeto é mais ampla, fornecendo também a possibilidade de se monitorar a horta escolar em questão.

No que tange ao monitoramento, faz-se necessário que os dados que são coletados dos sensores e as instruções enviadas aos atuadores, estejam acessíveis de outros ambientes além da própria horta. Assim, a estratégia utilizada no projeto baseia-se no envio dos dados que estão concentrados no ESPduino para uma base de dados externa, para que, ao acessá-la, outros dispositivos consigam visualizar as informações provenientes da

horta.

Como é possível observar no esquema, o ESPduino faz o envio dos dados para uma base de dados por meio do acesso a uma API RESTful que possibilita uma interface comum a dispositivos distintos e seu acesso é realizado usando requisições HTTP (GET, POST, UPDATE e etc). Portanto, para o ESPduino tornar os dados dos sensores e atuadores persistentes, basta a realização de uma requisição HTTP para a API, enviando tais dados. O funcionamento mais detalhado sobre essa API RESTful será explicado mais adiante na seção 3.4.

Por fim, ao passo que os dados coletados da horta já estão registrados em uma base de dados e acessíveis por vários dispositivos, torna-se possível a criação de várias formas de se inferir informações pertinentes ao monitoramento das condições desse ambiente. Uma dessas opções é relatada na seção 3.5, em que é proposta a criação de um aplicativo para o monitoramento dos dados provenientes da horta.

3.3 VERSÃO PERSONALIZADA DO ARDUBLOCK

Visto que as opções de versões do Ardublock disponíveis, não atendem com totalidade todas as funcionalidades do ESPduino, as quais se pretende explorar com este trabalho e, sendo o Ardublock o recurso de programação visual que se aproxima mais da proposta do uso do projeto como tema para ensino de programação, a solução encontrada para a falta de alguns recursos foi então a modificação do Ardublock para se obter uma versão personalizada e que disponha de todos os requisitos.

Assim, a implementação de “karlTH” que, mesmo não sendo totalmente completa, é a modificação do Ardublock que melhor atende as necessidades do trabalho, serviu como base para o incremento dos novos recursos como: melhorias nos blocos de requisição HTTP, criação de blocos novos e ajustes em algumas bibliotecas.

3.3.1 Estrutura de implementação do Ardublock

O Ardublock é todo desenvolvido utilizando a linguagem de programação Java e, por ser um projeto de médio porte, possui uma estrutura robusta, fazendo que seu entendimento não seja tão trivial à primeira vista, sendo necessária uma investigação um pouco mais profunda relativa a importância de cada classe, arquivo e diretório que a constitui.

Desta forma, antes de dar início ao processo de implementação da versão personalizada do Ardublock, foi necessário um estudo de sua estrutura de implementação para fornecer uma compreensão sobre seu funcionamento e de como é realizado o processo de adição e modificação de recursos do projeto. Assim, esse estudo resultou no diagrama

da (Figura 3.2), que apresenta de uma forma simplificada e esquematizada a estruturação do projeto como um todo.

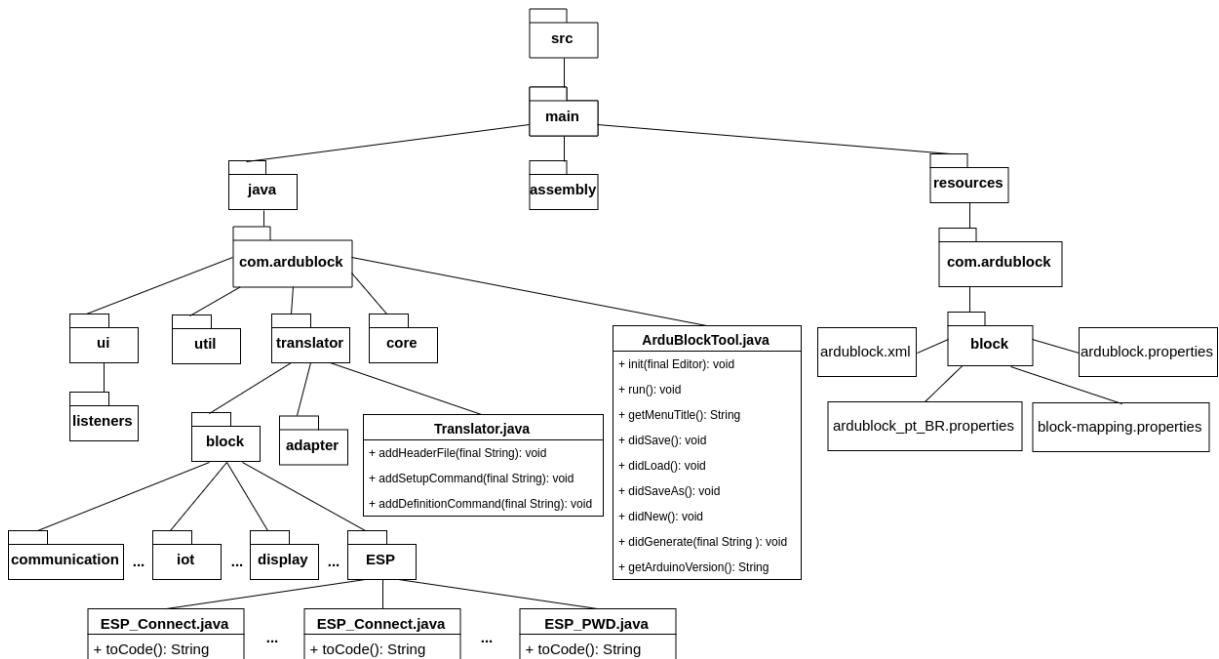


Figura 3.2 – Diagrama da estrutura da implementação do ArduBlock. Nele estão esquematizados os aspectos mais relevantes da implementação do ArduBlock

Conforme o diagrama da (Figura 3.2), o diretório “src” é onde se localizam os detalhes mais importantes da implementação e da estrutura do projeto, dentro dele se localizam os diretórios “java” (destinado às classes java da implementação) e “resources” (destinado a recursos como: arquivos de definições, tradução, mapeamento e etc).

Dentro do diretório “java”, se encontram outros diretórios, sendo que cada um agrupa classes de acordo com seu objetivo, como por exemplo: o diretório “ui” que agrupa as classes da interface gráfica. Dentre esses subdiretórios de “java”, está o subdiretório “translator”, que agrupa as classes que são responsáveis pela tradução do bloco visual para o código em linguagem C/C++ do ESPduino. Além disso, ainda no subdiretório “translator”, mais especificamente no diretório “block”, se encontram as classes que definem cada bloco e/ou funcionalidade de bloco.

No diretório “resources”, existe o subdiretório “block”, que agrupa arquivos de diferentes tipos. Os arquivos xml são responsáveis por conter a descrição da forma e demais características visuais dos blocos e interface. Já os arquivos de propriedades, de extensão “.properties”, são responsáveis por fazer a ligação entre os identificadores presentes nos arquivos xml com as suas funcionalidades.

3.3.2 Adição de novos blocos

Após se obter uma visão global da estrutura e funcionamento do Ardublock, seguiu-se de fato para a etapa de implementação das modificações no projeto Ardublock para a então criação de sua versão personalizada. Assim, dada a falta de alguns blocos e a ausência de alguns recursos em blocos já existentes na versão desenvolvida por “karlTH”, foi necessária a criação e modificação de alguns blocos para o Ardublock.

Contudo, o processo de criação e modificação dos blocos não é tão trivial, pois as definições que descrevem um bloco e suas funcionalidades são compostas pela inter-relação de várias classes e arquivos que, juntos, formam o componente visual e funcional o qual o usuário posteriormente irá interagir. Desta forma, por meio de uma investigação um pouco mais aprofundada no projeto Ardublock, foram levantados todos os arquivos e classes que compõem o componente “bloco” e que necessitam modificação ou criação para se ter um novo bloco. Como resultado, obteve-se o diagrama esquemático da (Figura 3.3), que descreve quais classes dependem de quais, além da relação entre os arquivos de definição com as classes.

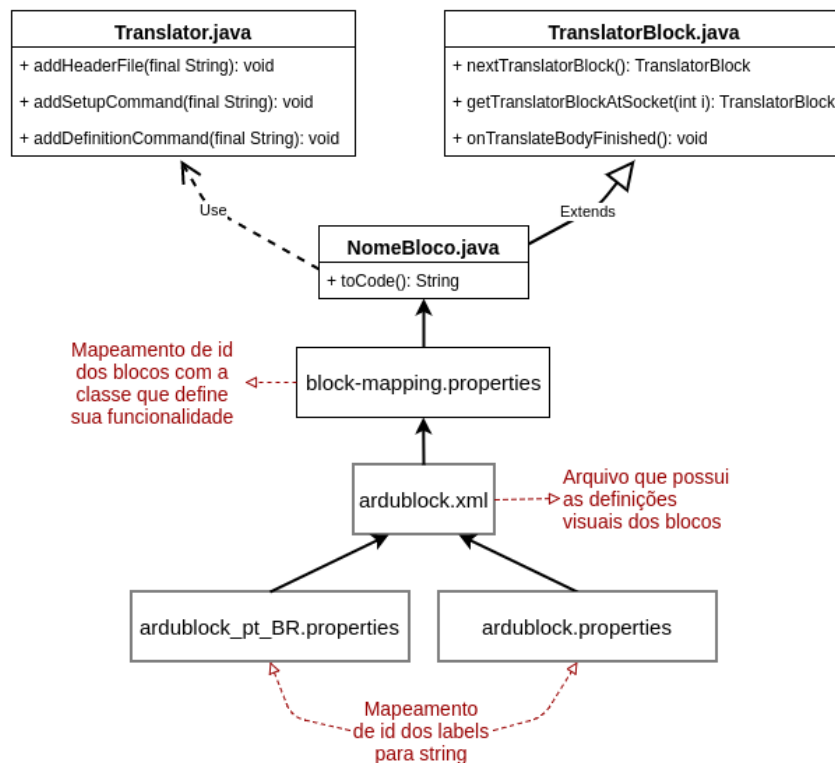


Figura 3.3 – Diagrama que descreve o funcionamento e a tradução dos blocos visuais para código em C/C++

Conforme pode ser visto na Figura 3.3, para a criação ou modificação dos blocos é necessária a manipulação de vários elementos, dentre classes e arquivos, para modificar tanto parâmetros visuais quanto funcionais.

Analisando o diagrama, percebe-se que cada bloco possui uma classe java nome-

ada de acordo com o bloco em questão, um exemplo é a classe “NomeBloco.java” da figura. Essa classe é responsável por conter as “strings” que serão parte do código em C/C++ que o bloco irá gerar, além de definir em qual posição no sketch que elas serão inseridas, ou seja, é nela que se define qual será o código resultante da tradução do bloco visual para código C/C++ do sketch. Essa tradução é abstraída pelas classes “TranslatorBlock.java” e “Translator” as quais tem por objetivo:

- **TranslatorBlock:** classe da qual “NomeBloco.java” é estendida e que possui a função de colocar o código gerado pelo bloco no sketch a ser compilado no ESPduino;
- **Translator:** classe usada para realizar manipulações prévias ou encapsular funções constantemente usadas no ajuste e montagem do código C/C++ a partir das strings descritas na classe “NomeBloco.java”;

A parte visual do bloco é definida pelo arquivo “ardublock.xml”, no qual são descritas todas as características do componente visual, como: cor, conexões, descrição, forma, tipo, entre outras características. Cada bloco é descrito dentro de uma tag “<BlockGenus>” do *schema* desse arquivo xml a qual é identificada com o parâmetro “name”, conforme exemplificado na Figura 3.4.

```
<BlockGenus name="NomeBloco" kind="command" color="0 255 0" initlabel="bg.NomeBloco">
  <BlockConnectors>
    <BlockConnector connector-type="string" connector-kind="socket" label="bc.con1">
      <DefaultArg genus-name="glue_msg" label="conexao 1" />
    </BlockConnector>
    <BlockConnector connector-type="string" connector-kind="socket" label="bc.con2">
      <DefaultArg genus-name="glue_msg" label="conexao 2" />
    </BlockConnector>
  </BlockConnectors>
</BlockGenus>
```

Figura 3.4 – Exemplo de definição de bloco com a tag <BlockGenus> no arquivo “ardublock.xml”.

O arquivo “ardublock.xml” possui também, além da descrição visual de todos os blocos, a descrição de todos os demais componentes visuais que constituem a interface gráfica do Ardublock.

Ainda relacionado com as definições visuais do bloco, está o mapeamento dos textos apresentados nos blocos com os “labels” de sua descrição no arquivo xml. Para tal, é usado o arquivo “ardublock.properties”, que relaciona o identificador do “label” com o texto que será mostrado. Entretanto, os textos dos “labels” podem ser adaptáveis conforme a língua definida no computador do usuário do Ardublock, bastando a criação de um arquivo semelhante ao “ardublock.properties” porém com os textos traduzidos para a língua desejada e identificando o novo arquivo com o código da língua da tradução. O diagrama expõe um exemplo de arquivo de tradução dos “labels” para o português do Brasil, o qual foi nomeado indicando o código da língua: “ardublock_pt_BR.properties”.

Com as definições visuais dos blocos definidas, é necessário então que sejam associadas as funcionalidades do bloco com sua descrição visual. Essa ligação é realizada através do arquivo “block-mapping.properties”, em que são relacionados cada “name” de bloco, descrito no arquivo “ardublock.xml”, com a sua respectiva classe java.

Como citado anteriormente, a versão do Ardublock utilizada como base para as modificações, não dispunha de blocos para a manipulação de requisições HTTP e os que possuem um objetivo semelhante não atendem as necessidades propostas pelo trabalho. Dessa forma, foram criados três novos blocos para a plataforma Ardublock: um bloco para requisições HTTP do tipo POST de valores numéricos (Figura 3.5), um bloco também para requisições HTTP do tipo POST, porém para envio de textos (Figura 3.6), e um terceiro bloco para as requisições HTTP do tipo GET (Figura 3.7).

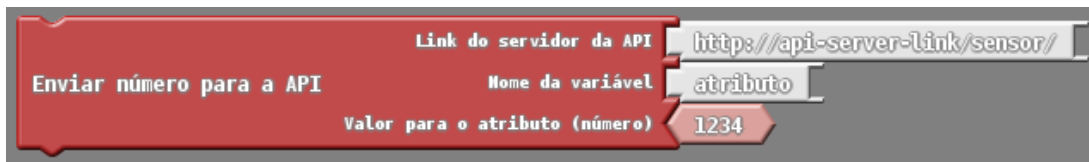


Figura 3.5 – Imagem do bloco de requisições HTTP do tipo POST para números.

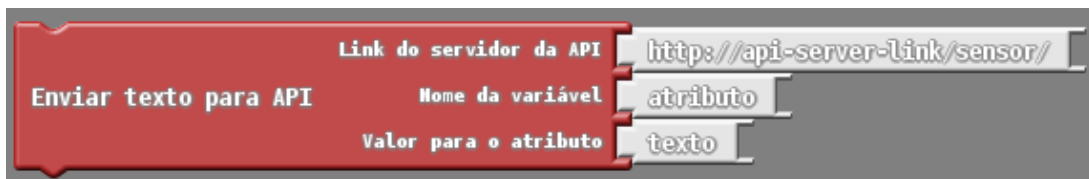


Figura 3.6 – Imagem do bloco de requisições HTTP do tipo POST para textos.

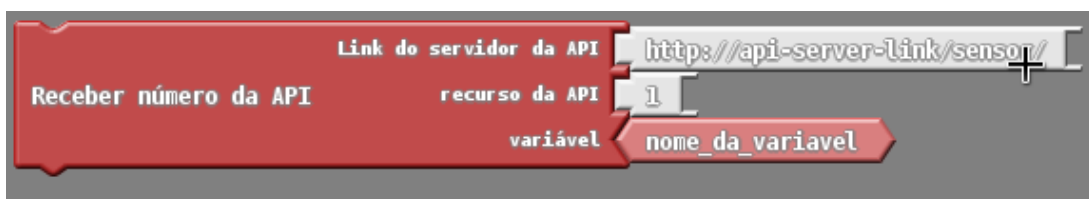


Figura 3.7 – Imagem do bloco de requisições HTTP do tipo GET.

- **Bloco de requisição HTTP POST (número):** Bloco responsável por fornecer a funcionalidade de se efetuar uma requisição HTTP do tipo POST para o envio de um valor numérico para uma API RESTful. Seus parâmetros são: o link de acesso ao recurso da API, o nome do atributo do objeto do recurso da API e o valor que será registrado para tal atributo.
- **Bloco de requisição HTTP POST (texto):** Analogamente ao bloco anterior, este bloco é responsável por realizar uma requisição do tipo POST para o recurso de uma

API RESTful, porém, seu diferencial é que ele se destina ao envio de valores textuais para o atributo do recurso da API.

- **Bloco de requisição HTTP GET:** Bloco que permite que requisições HTTP do tipo GET sejam realizadas, permitindo que se obtenha um valor de atributo do recurso registrado na API RESTful. Seus parâmetros são: o link de acesso ao recurso da API, o identificador do recurso a ser consultado e, por último, a variável que receberá o valor proveniente da consulta ao recurso da API.

Por se tratar de consultas a uma API RESTful, o corpo das requisições é construído sob o *schema* dos arquivos JSON, entretanto, a parte de serialização dos dados a serem enviados por meio das requisições está abstraída do usuário dos blocos, evitando que se agregue uma complexidade de formatação de conteúdo que é desnecessária para o objetivo da proposta do presente trabalho.

Conforme já descrito, o trabalho de adição de um novo bloco ou modificação de algum recurso na plataforma Ardublock, não é uma tarefa trivial, sendo que envolve a manipulação de vários arquivos. Dessa forma, todo o trabalho de criação dos novos blocos citados, foi mensurado utilizando como métrica o número de linhas, tanto para os arquivos que foram modificados, quanto para os novos arquivos cuja criação foi necessária. A Tabela 3.2 refere-se aos arquivos que foram modificados e a Tabela 3.3 refere-se aos novos arquivos adicionados ao projeto Ardublock:

Tabela 3.2 – Arquivos modificados para a adição de novos blocos.

Arquivo modificado	Tamanho em linhas	Linhas modificadas	Linhas adicionadas
block-mapping.properties	762	0	4
ardublock_pt_BR.properties	649	2	9
ardublock.properties	1545	0	9
ardublock.xml	9248	6	61

Tabela 3.3 – Arquivos adicionados para criação dos blocos.

Arquivo adicionado	Tamanho em linhas
API_Receive_Number.java	25
API_Send_Number.java	26
API_Send_String.java	26

Outro fator que impacta na manipulação dos arquivos do projeto Ardublock é o tamanho total em linhas dos mesmos, pois em arquivos como o “ardublock.xml”, que possui um total de 9248 linhas, a adição ou a modificação de qualquer recurso acaba se tornando uma tarefa complicada, já que cada trecho de código tem de ser adicionado em um local distinto, obedecendo o *schema* adotado pelo arquivo.

3.3.3 Ajuste das bibliotecas de I/O para o ESPduino

Como citado anteriormente, a versão do Ardublock modificado por karlTH, utiliza bibliotecas de I/O de fabricantes de versões modificadas do microcontrolador ESP8266 e isso acarreta no surgimento de incompatibilidades com alguns recursos do ESPduino. Tais incompatibilidades foram identificadas através dos erros do processo de compilação dos sketches (Figura 3.9), que faziam uso de alguns blocos com funcionalidades de conexão com a internet, os quais traduzem o bloco para um código que utiliza as bibliotecas não compatíveis com o microcontrolador do ESPduino.

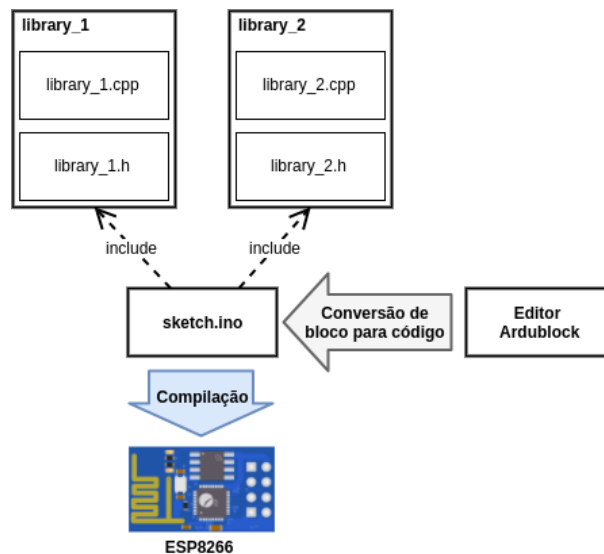


Figura 3.8 – Diagrama ilustrativo de como são utilizadas as bibliotecas escritas em C++ no Arduino IDE e como o Ardublock atua para utilizá-las

Constatado o problema de incompatibilidade, primeiramente foi necessário identificar todos os blocos que geravam código que referenciava as bibliotecas problemáticas, além de identificar também se nesse código gerado não estavam sendo usadas referências que levavam a uma dependência em um segundo nível às bibliotecas problemáticas. Nesse processo de busca, foi possível perceber que nem todas as funções das bibliotecas problemáticas geravam erros de compilação, já que alguns blocos que as referenciavam funcionavam sem ocasionar erros.

Dessa forma, a solução do problema baseou-se na identificação de todas as referências a bibliotecas que possuíam funções que ocasionavam os erros de compilação, para que então fossem ajustadas, de modo a substituir as chamadas de funções de I/O das implementações da fabricante Adafruit, que se mostraram incompatíveis, por chamadas a funções de I/O compatíveis com o ESPduino.

Entretanto, a solução acabou convergindo ao ajuste de uma única biblioteca, a “Dui-noedu_Esp8266.h”, sendo essa utilizada para encapsular as funções chamadas pelos códigos provenientes dos blocos que faziam uso da comunicação. Além disso, ela era responsável por ocasionar a maioria das dependências de segunda ordem na hierarquia de

importação dos arquivos de bibliotecas problemáticas. As modificações feitas foram basicamente a alteração das referências às bibliotecas problemáticas para uma nova biblioteca auxiliar criada, a “HttpApi_Esp8266.h”, que ficou responsável de encapsular as funções de comunicação, ao menos para as funções que envolvem requisições HTTP (principais fontes dos erros de compilação).

Nessa nova biblioteca, foram usadas apenas chamadas de I/O e recursos do protocolo HTTP nativos do microcontrolador ESP8266, desta forma, acredita-se que quase totalidade dos microcontroladores que usam o projeto do ESP8266 como base para sua fabricação, tendem a funcionar sem problemas, mesmo que sem algumas otimizações ou abstrações de funções.

3.4 COMUNICAÇÃO COM UM SERVIDOR EXTERNO

Ao descrever a arquitetura organizacional do projeto, foi mencionado que a comunicação do ESPduino com uma base de dados online é requisito fundamental para que se possibilite no projeto, além de um sistema de irrigação, também um sistema para o monitoramento dos dados coletados da horta.

Com isso em mente, ainda na fase de planejamento das tecnologias e materiais necessários para a concretização do projeto, foram avaliadas várias alternativas referentes a essa necessidade de comunicação e armazenamento de dados remotamente ao ESPduino.

Em um primeiro momento, a utilização de algum serviço disponível na internet que se propunha a resolver esse problema parecia a melhor opção para a ideia do projeto, já que não haveria a necessidade desenvolver algo do tipo partindo-se do zero. Entretanto, visando o levantamento das melhores opções e que melhor se adequavam a proposta do presente trabalho, foi possível perceber que as soluções e serviços já disponíveis, na verdade, acrescentariam dificuldades no projeto, sendo que na maioria deles é necessária a criação de uma conta, configuração, estruturação dos recursos, não há versões em português, entre outras coisas que variam de opção para opção e que acabariam fugindo da ideia do projeto.

Por esses motivos, a alternativa encontrada acabou sendo a implementação de um serviço simples e que atenda os requisitos de comunicação e armazenamento já citados. Além do mais, seguindo a ideia central do projeto, tal serviço visa ser apenas um recurso para auxiliar na didática do ensino de programação, focando apenas no entendimento de como é realizada a comunicação do ESPduino com uma base de dados remota.

A solução então desenvolvida, baseia-se no conceito de API (*Application Programming Interface*), que refere-se ao conjunto de rotinas e padrões de programação que possibilitam uma interface comum para o acesso das funcionalidades de um sistema. Contudo,

ao se tratar de uma solução disponível online e acessível por não só, softwares distintos, mas também por dispositivos distintos, o conceito de API que se aplica é o das API do tipo REST (*REpresentational State Transfer*), em que os recursos disponibilizados devem ser coerentes com as restrições e propriedades do protocolo HTTP, ou seja, o acesso aos recursos de uma API REST devem ser possíveis com qualquer consulta que atenda as características do protocolo HTTP.

Assim, a implementação de uma API REST se enquadra nas necessidades de comunicação do projeto, visto que o problema em questão era fazer com que dois meios distintos (ESPduino e uma base de dados *online*) trocassem informações e com uma API REST essa comunicação é feita de maneira simples apenas usando as requisições básicas do protocolo HTTP, as quais, quase que a totalidade dos dispositivos com acesso a internet, possui nativamente implementado.

Como a solução necessária não precisava ser robusta, tendo que apenas possuir o básico para possibilitar a comunicação desejada, optou-se pela utilização do framework de desenvolvimento web Django para sua criação, escolha essa motivada pela facilidade e rapidez de criação de um sistema *web* com acesso a banco de dados. Além disso, ao se optar pela utilização do framework Django a implementação de uma API REST torna-se bastante facilitada por um outro framework que atua sobre o Django, o Django REST Framework que possibilita a criação de uma API REST de maneira simples. Pelo fato das APIs criadas com o Django REST Framework seguirem todas as restrições estabelecidas pelo protocolo HTTP, elas podem ser consideradas APIs RESTful.

Ao passo que a parte de implementação foi concluída, foi necessária hospedagem dessa API RESTful em algum serviço para que ela ficasse acessível *online*. O melhor serviço então encontrado foi a plataforma Heroku, que possibilita a hospedagem de sistemas desenvolvidos em diversas linguagens de programação e frameworks, dentre eles o Django.

Estando a API disponível *online* e acessível por dispositivos distintos, o problema da comunicação entre ESPduino e a base de dados foi resolvido. Na utilização com o público alvo, na atividade que será explicada no capítulo 4, não foi necessário nenhum tipo de cadastro, configuração ou qualquer outra ação para a utilização, mostrando que essa API RESTful implementada, mesmo que extremamente simples, cumpriu o propósito a que se destinava no projeto de irrigação e monitoramento.

3.5 PROPOSTA DE APLICATIVO PARA MONITORAR A HORTA

Nos tempos atuais, os dispositivos móveis já são parte da vida dos jovens, sendo que já é uma cena bastante comum ver uma criança ou adolescente com um smartphone ou tablet em mãos. Segundo apontado pela pesquisa TIC Domicílios 2017 (VALENTE,

2018), os dispositivos móveis já são o principal meio de acesso à internet pelos brasileiros. Como consequência, a presença dessas tecnologias no cotidiano das pessoas já não é mais apenas como um recurso facilitador para a comunicação e já se tornou um meio fortemente presente também nas interações sociais e de trabalho.

Seguindo a proposta do presente trabalho, cujo intuito é explorar um ambiente pertencente ao cotidiano dos alunos como um cenário de aplicação prática dos conhecimentos de programação, percebeu-se que os dispositivos móveis são mais um exemplo que reforça a presença da programação na vida desses jovens. Assim, surgiu a ideia de propor a criação de um aplicativo como parte do projeto de irrigação e monitoramento da horta escolar.

Propõe-se que os próprios alunos criem seus aplicativos para monitorar a horta, possibilitando o uso da criatividade na hora de definir o que monitorar e como monitorar, além de instigá-los a observarem a associação do estado do ambiente da horta com os dados coletados pelos sensores.

Para tornar essa proposta possível foram levantadas as opções de plataformas destinadas a criação de aplicativos e que possam ser utilizadas como recurso de ensino de programação. Dentre as opções, destacou-se a plataforma Thunkable (THUNKABLE, 2018), uma ferramenta para o desenvolvimento de aplicativos tanto para o sistema operacional Android, quanto para o IOS.

O Thunkable utiliza-se também do recurso de programação visual por blocos (Figura) e, diferente de seu antecessor o MIT App Inventor (INVENTOR, 2018), traz consigo blocos que possibilitam a criação tanto de aplicativos simples para o aprendizado do básico, até mesmo aplicativos que englobam funcionalidades mais profissionais. A interface da plataforma conta com dois ambientes para o usuário, o primeiro refere-se a tela de “Design” e é destinado a criação do *layout* do aplicativo (disposição dos componentes e estilização), já na tela de “Blocos” é onde que toda lógica do aplicativo pode ser descrita por meio da programação por blocos.

Dessa forma, para demonstrar uma das possíveis aplicações e uso dos dados coletados da horta, foi desenvolvido um exemplo de aplicativo que acessa as informações armazenadas na base de dados online e as usa para mostrar algumas informações sobre monitoramento do sistema, conforme a Figura 3.10. Esse aplicativo utiliza os dados coletados no sistema implementado na escola Marista Santa Marta, cuja implantação está descrita mais a fundo na seção 4.1.

A ideia central do desenvolvimento desse aplicativo é fornecer um exemplo que possa ser usado como base para novas ideias para o monitoramento da horta. Portanto, em aspectos de como e quais informações são apresentadas não são de tanta relevância nesse caso, tendo em vista que o aplicativo é um exemplo base.

Esse aplicativo foi desenvolvido utilizando a estratégia da programação visual com blocos por meio da plataforma Thunkable. Assim, foram explorados principalmente os

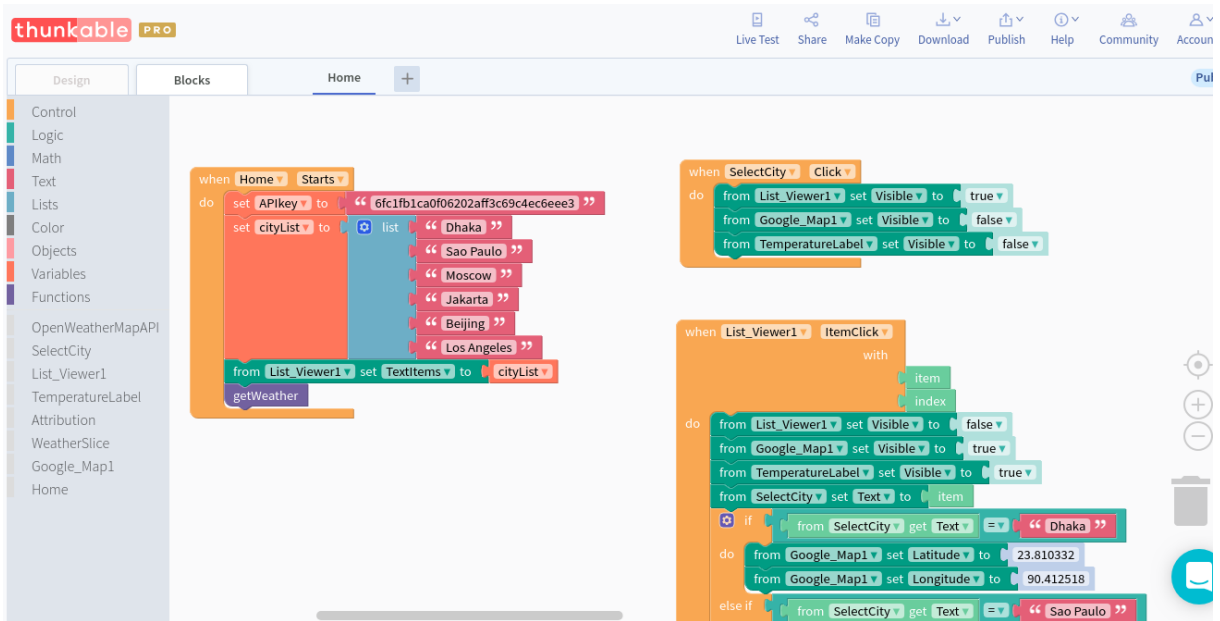


Figura 3.9 – Tela onde que a lógica dos aplicativos pode ser criada utilizando a linguagem visual de blocos.

Fonte: Adaptado de Thunkable (2018).

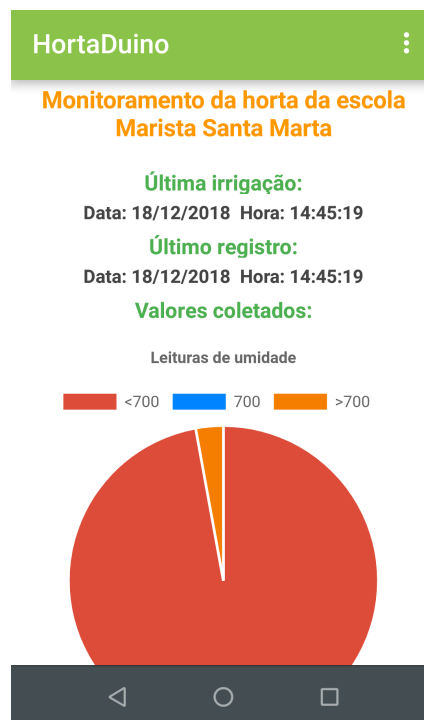


Figura 3.10 – Aplicativo de exemplo criado utilizando a plataforma Thunkable.

blocos relacionados ao acesso e manipulação dos dados de uma consulta a uma API RESTful, como por exemplo o bloco que efetua uma requisição HTTP do tipo GET e os blocos responsáveis por permitir a extração dos valores de um texto em formato JSON (formato do texto retornado pela requisição get a API RESTful). Além disso, buscou-se a demonstração de como os dados podem ser apresentados de uma forma um pouco mais

visual por meio de gráficos e como isso pode ser feito utilizando os blocos da plataforma Thinkable.

3.6 CRIAÇÃO DE MATERIAL DE APOIO PARA O PÚBLICO ALVO

Durante o desenvolvimento deste trabalho foi realizado um esforço para reunir elementos para que o projeto de irrigação e monitoramento de uma horta escolar seja viável enquanto estratégia didática. Além disso, houve um planejamento da melhor forma de integrar os recursos de ensino de programação disponíveis atualmente, desde os componentes de hardware educacional utilizados na automação da irrigação, até os elementos de programação visual por blocos utilizados para a criação da lógica do sistema.

Essa preocupação na seleção dos recursos e no planejamento da melhor forma de uní-los, motiva-se pelo fato de que um projeto desse tipo, quando bem pensado, possui um grande potencial interdisciplinar e que, munido das ferramentas certas, fornece uma infinidade de possibilidades para criação de atividades que explorem as aplicações práticas da programação no cotidiano dos alunos, contribuindo para a desmistificação do tema.

Desta forma, todo o processo de desenvolvimento da proposta de um sistema de irrigação e monitoramento de uma horta escolar, como um cenário para o ensino de programação, foi detalhadamente documentado, a fim de se disponibilizar um material de apoio para que a ideia do projeto seja replicada como um aporte didático para o ensino de programação.

A documentação aborda um passo a passo do que é necessário e como fazer cada parte do projeto de irrigação e monitoramento, sendo que cada etapa está disposta em uma ordem que facilite o progresso no desenvolvimento do projeto de forma incremental. Inicialmente, é apresentada uma listagem dos materiais necessários para a replicação do projeto, juntamente com sugestões de onde encontrar tais materiais e também um custo médio de cada item. Em seguida, é explicado como instalar as ferramentas de softwares necessárias e como adicionar recursos complementares como bibliotecas e *plugins*.

Após a explicação do que é necessário, a documentação traz a descrição, também detalhada, de como implementar o sistema. De início na explicação de “como fazer?”, é descrito o funcionamento do sistema, dando uma visão geral de como as coisas estão interligadas e interagem entre si. Em sequência, o restante das explicações são apresentadas em pequenas seções, chamadas de “projetos”, que possuem o objetivo de demonstrar que o sistema todo é constituído da junção de vários pequenos “projetos”. Assim, nos “projetos” iniciais, é abordado como os componentes de hardware podem ser conectados e programados para que o sistema de irrigação automatizada vá tomando forma. Já nos demais “projetos”, que são voltados a integração de tudo para se obter um sistema também monitorável remotamente, é explicado como é o processo de envio dos dados para

uma base de dados *online*, como acessar tais dados e como criar um aplicativo para a visualização das informações monitoradas.

Esse material de apoio desenvolvido, está disponibilizado em um repositório da plataforma GitHub e acessível pelo link: <https://github.com/hortaduino>.

4 RESULTADOS E DISCUSSÃO

Este capítulo destina-se a apresentar os resultados da implantação do projeto proposto pelo presente trabalho no cenário real de uma escola, além da aplicação ao público alvo a que o mesmo se destina. Na seção 4.1 é descrito como se deu o processo de implantação no cenário da horta escolar, em seguida (seção 4.2), explica-se como foi planejada e desenvolvida uma das atividades realizadas e por fim, na seção 4.3 como foi realizado o processo de avaliação da aceitação do projeto com os alunos participantes da atividade.

4.1 IMPLANTAÇÃO DO PROJETO EM UMA ESCOLA

A implantação do projeto de irrigação e monitoramento foi realizado na Escola Marista Santa Marta, localizada no município de Santa Maria - RS, que ao saber da iniciativa se mostrou interessada em colaborar. Foram realizadas duas idas até a escola: a primeira destinada a instalação dos materiais para a demonstração, e a segunda com o objetivo de demonstrar a proposta do projeto.

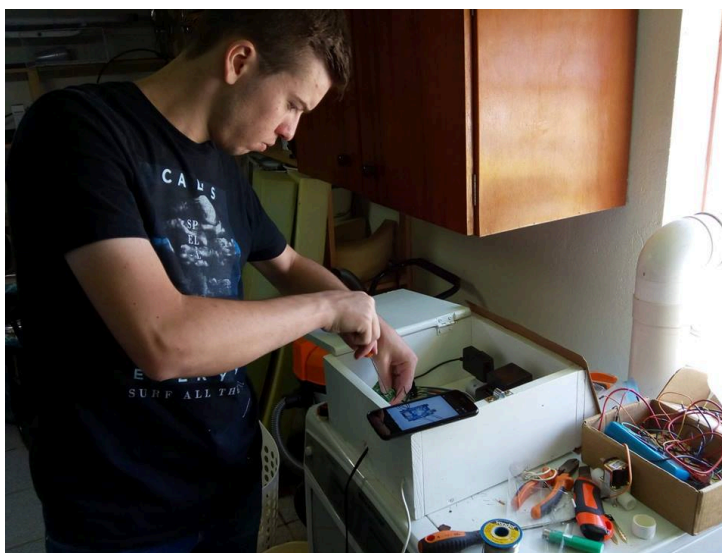


Figura 4.1 – Processo de instalação do sistema de irrigação e monitoramento da horta na primeira visita a escola.

A primeira visita a escola, realizada durante uma manhã, foi reservada para a realização da implantação do sistema de irrigação e monitoramento (Figura 4.1) e durou cerca de 4 horas. O ambiente disponibilizado, um canteiro destinado ao cultivo de flores (Figura 4.2), mostrou-se um bom local para instalação do sistema, pois além de atender a ideia de ser algo semelhante a uma horta, também facilitou a instalação de alguns componentes

por estar próximo aos recursos que facilitariam a comunicação e a instalação do sistema de irrigação. Nesse primeiro contato com a comunidade e ambiente da escola, não foram envolvidos os alunos, pois não haveria tempo hábil para uma explicação minuciosa do processo, tendo em vista que foi uma atividade realizada em período letivo.



Figura 4.2 – Local destinado a instalação do projeto. Por se localizar próximo a uma residência, o acesso a água ficou facilitado.

Na segunda ida até a escola, que teve como objetivo transmitir a ideia da proposta, bem como demonstrar as possibilidades que ela permite e, tanto alunos (39 participantes na faixa etária de 12 a 14 anos) quanto professores (3 participantes) foram envolvidos. Esse segundo encontro durou 2 horas e foi dividido em 4 momentos: apresentação da proposta, demonstração do sistema instalado, realização de uma dinâmica desplugada (explicada na seção 4.2) e avaliação da aceitação da proposta (seção 4.3).

A apresentação feita nessa segunda visita, teve como objetivo demonstrar aos alunos que a computação está presente em seus cotidianos de várias maneiras e que, por meio da programação, é possível que eles transformem o ambiente a sua volta, como por exemplo, a horta escolar. Ao fim da apresentação, percebeu-se bastante empolgação com o projeto e, inclusive, alguns alunos realizaram questionamentos referentes a melhorias que poderiam ser implantadas no projeto de irrigação e monitoramento, como por exemplo, o reaproveitamento da água da chuva para a irrigação.

Logo em seguida, os alunos, divididos em pequenos grupos, conheceram os detalhes do projeto de irrigação e monitoramento implantado no canteiro de flores da escola. Para cada grupo, foi realizada a explicação de como todos os componentes do projeto interagem e constituem o sistema, demonstrando como tudo funciona na prática.

4.2 REALIZAÇÃO DE ATIVIDADE DESPLUGADA

As atividades desplugadas são dinâmicas de ensino de programação sem o uso de computadores, nelas são trabalhados os conceitos da lógica e estrutura dos algoritmos de uma forma concreta e dinâmica, pois os próprios participantes passam a ser parte do programa criado. Conforme ressalta (BEZERRA, 2014), com esse tipo de atividade os alunos são levados a resolver um determinado problema e, conforme vai ocorrendo a interação com a dinâmica em busca da resolução desse problema, o aluno é exposto aos conceitos relacionados a computação e os usa de alguma forma para encontrar a solução.

Usando essa ideia, de o aluno construir o conhecimento sendo parte dele, a atividade desplugada na escola teve como objetivo, transmitir os conceitos que englobam o funcionamento do sistema irrigação e monitoramento da horta, desde a coleta de dados pelos sensores até o armazenamento dos valores coletados em uma base de dados *online*. Sendo assim, na atividade, foi solicitado que os 39 alunos se reunissem em 3 grupos: Grupo 1, Grupo 2 e Grupo 3.

Os integrantes do Grupo 1 representaram os sensores e cada um dos alunos tinha como tarefa coletar plaquinhas, contendo um número em cada e dispostas no canteiro que o sistema de irrigação e monitoramento estava instalado. Ao coletar uma plaquinha, cada integrante do grupo deveria então levá-la até o Grupo 2.



Figura 4.3 – Plaquinha disposta no canteiro.

Os alunos do Grupo 2, representando o ESPduino, recebiam as plaquinhas entregues pelos membros do Grupo 1 e logo em seguida deveriam aplicar um teste lógico no valor recebido. Esse teste baseava-se somente na ideia de verificar se o valor da plaqui-

nha era maior ou menor que “500”, simulando assim o teste realizado pelo ESPduino para verificar se o solo está seco ou úmido. Por fim, o grupo deveria anotar o valor da plaquinha classificando-o conforme o resultado do teste, se o valor fosse menor que “500” o valor deveria ser anotado junto aos demais valores menores que “500”, caso o valor fosse maior que “500”, então deveriam anotá-lo junto aos demais valores maiores que “500”. Após realizarem as anotações alguém do grupo então encaminhava a plaquinha para o Grupo 3.



Figura 4.4 – Grupo 2 realizando as anotações.

O Grupo 3, simbolizando a base dados *online*, recebiam a plaquinha vinda do Grupo 2 e a “armazenavam” em um cartaz, representando o banco de dados.

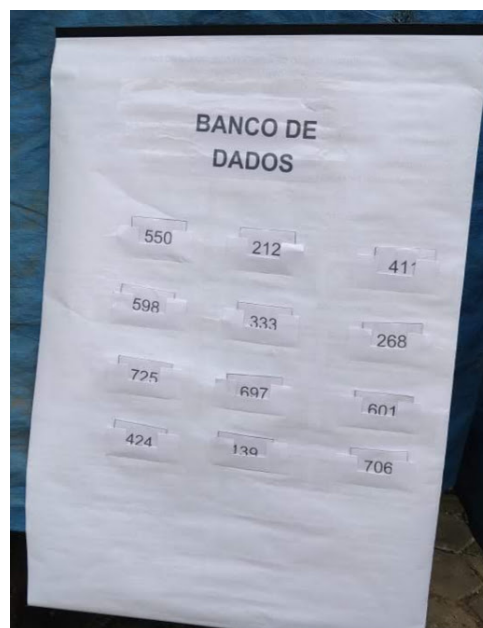


Figura 4.5 – Cartaz simbolizando o “banco de dados” onde os valores das plaquinhas foram colocados.

Por fim foi discutido então a importância de cada um dos 3 Grupo no processo, relacionando com cada etapa que envolve o funcionamento do sistema de irrigação e monitoramento da horta.

4.3 AVALIAÇÃO DA ACEITAÇÃO DOS ALUNOS

Para avaliar a aceitação dos alunos, utilizou-se uma adaptação do *Self Assessment Manikin* (BRADLEY; LANG, 1994), que é uma técnica não-verbal amplamente usada em áreas como educação e psicologia, para medir a reação afetiva de pessoas frente a estímulos. Esta técnica tem sido adaptada por outros autores (HAYASHI et al., 2016; BETTELLA; VERSCHURE, 2016) que produziram novos formatos do instrumento, mantendo seu propósito original.

Portanto, ao final da atividade desenvolvida na segunda visita a escola, cada aluno foi convidado a responder um pequeno questionário contendo três perguntas. A primeira pergunta se referia ao grau de **satisfação** do aluno com a atividade. Já a segunda, remetia ao grau de **motivação** do aluno em aprender mais sobre as atividades realizadas. Para responder ambas as perguntas, o aluno deveria marcar o rosto que mais demonstrava sua resposta para a pergunta, dentre as 5 opções apresentadas. Seguindo o conceito do *Self Assessment Manikin* (BRADLEY; LANG, 1994), cada rosto ilustrado apresentava uma reação diferente (Figura 4.6), variando de um reação muito triste (reação 1), até uma reação muito feliz (reação 5).

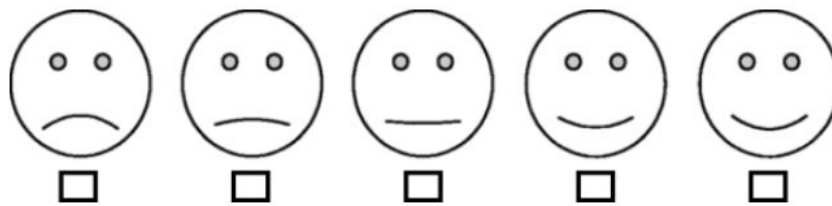


Figura 4.6 – Escala avaliativa usada na atividade.

Para a contabilização das respostas, a cada reação foi associado um número no intervalo de 1 a 5: muito triste (1), pouco triste (2), indiferente (3), pouco feliz (4) e muito feliz (5). Assim, as porcentagens das respostas obtidas dos 39 alunos, referente à primeira pergunta (satisfação) e segunda pergunta (motivação) do questionário podem ser visualizadas, respectivamente, na Figura 4.7 e Figura 4.8.

Como pode ser observado em ambos os gráficos, tanto o grau de satisfação com a atividade, quanto o grau de motivação e em aprender mais sobre as atividades, foram altos, sendo que para a pergunta sobre satisfação obteve-se, em 87,1% das respostas, grau de satisfação 4 ou 5 e, para a pergunta sobre a motivação para saber mais, um total de 79,5% de respostas de grau 4 ou 5. Tais resultados indicam que a proposta foi muito bem aceita

1) Qual o seu grau de satisfação com a atividade de hoje?

● Satisfação (1) ● Satisfação (3) ● Satisfação (4) ● Satisfação (5)

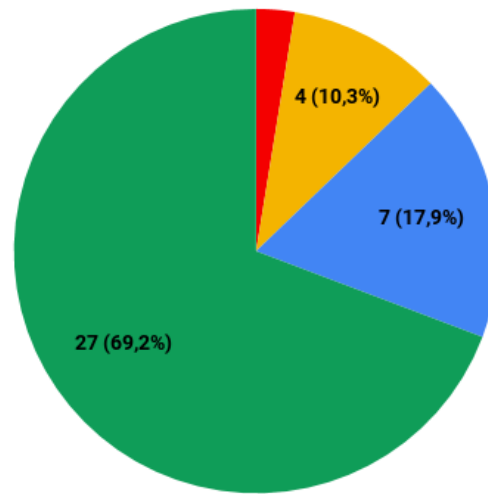


Figura 4.7 – Gráfico que representa a porcentagem de respostas referente a satisfação com a atividade desenvolvida.

2) Qual o seu grau de motivação para aprender mais sobre a atividade de hoje?

● Motivação (1) ● Motivação (3) ● Motivação (4) ● Motivação (5)

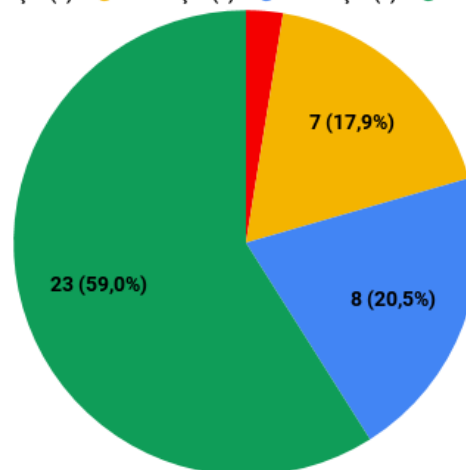


Figura 4.8 – Gráfico que representa a porcentagem de respostas referente a motivação com a atividade desenvolvida.

por parte dos alunos e, além disso, mostra que em grande parte deles há o interesse de aprender mais sobre o projeto proposto pelo trabalho, demonstrando também, que uma abordagem que consegue aliar a teoria e prática tende a despertar o interesse dos alunos.

A terceira e última pergunta do questionário destinava-se a ser um campo livre, permitindo que os alunos pudessem deixar um recado, apontar o que acharam legal, o que

não gostaram, entre outras considerações. As respostas, novamente, foram bastante positivas e em sua maioria são elogios sobre a atividade e relatos sobre o quanto foi aprendido com a atividade.

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como intuito propor um sistema de irrigação e monitoramento de horta escolar como tema para o ensino de programação, explorando o ambiente da horta escolar como um cenário real de aplicação dos conceitos de programação e computação.

Conforme relatado, houve um processo bastante cuidadoso na busca e seleção dos recursos de hardware e software para que se fosse possível o uso de uma proposta desse tipo como recurso auxiliador no ensino de programação. Desta forma, recursos como as placas de prototipagem de hardware e eletrônica juntamente com a programação visual com blocos, se mostraram bastante efetivos enquanto ferramentas didáticas, já que possibilitam explorar tanto a criatividade do aluno para “construir” uma solução, quanto dar “vida” e movimento a esses recursos físicos por meio da programação. Isso possibilita que se explore não somente conhecimentos relacionados à programação, mas também à integração desses com o ambiente no qual a escola e o aluno estão inseridos.

Parte do trabalho foi dedicado a fazer com que os recursos escolhidos pudessem ser integrados, como foi o caso das modificações necessárias na plataforma ArduBlock. Entretanto, foi possível que todas as adaptações fossem realizadas, sendo que, caso não fosse possível a utilização de algum dos recursos selecionados, possivelmente a proposta não teria sido viabilizada, o que reforça, mais uma vez, a grande importância da realização das escolhas certas dentre os recursos disponíveis e necessários para o projeto.

A implantação da proposta na Escola Marista Santa Marta, foi de grande importância, pois possibilitou que fosse colocado a prova, em um cenário real, tudo que foi argumentado durante este trabalho. A partir da avaliação das atividades desenvolvidas na escola, foi possível também validar o projeto de irrigação e monitoramento enquanto tema para a aplicação dos conhecimentos de programação em um ambiente escolar. Além disso, é possível perceber o interesse por parte dos alunos para o aperfeiçoamento do projeto de irrigação e monitoramento da horta, visto que vários alunos que participaram das atividades na escola fizeram questionamentos e sugestões de como o sistema poderia evoluir de forma a se adaptar melhor ao ambiente da escola.

Por fim, é possível concluir que o trabalho obteve êxito ao que se propôs. Ademais, espera-se que, com o auxílio do material de apoio criado e disponibilizado, seja possível que o projeto de irrigação e monitoramento de uma horta escolar possa ser replicado e expandido em outras ações que tenham como objetivo, promover formas alternativas e efetivas para o ensino de programação.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAFRUIT IO. 2018. <<https://learn.adafruit.com/adafruit-io/overview>>. Accessed: October 25, 2018.

ARDUBLOCK. **Ardublock - A Graphical Programming Language for Arduino**. 2018. <<http://blog.ardublock.com/>>. Accessed: July 31, 2018.

ARDUINO. **Arduino**. 2018. <<https://www.arduino.cc/>>. Accessed: July 31, 2018.

BETELLA, A.; VERSCHURE, P. F. M. J. The affective slider: A digital self-assessment scale for the measurement of human emotions. **PLoS One**, v. 11, n. 2, 2016.

BEZERRA, F. Bem mais que os bits da computação desplugada. In: **3º Congresso Brasileiro de Informática na Educação (CBIE 2014)**. [S.l.: s.n.], 2014.

BLIKSTEIN, P. Digital fabrication and 'making' in education: The democratization of invention. In: WALTER-HERRMANN, J.; BÜCHING, C. (Ed.). **FabLabs: Of machines, makers and inventors**. [S.l.]: Transcript Verlag, 2003.

BRADLEY, M. M.; LANG, P. J. Measuring emotion: the self-assessment manikin and the semantic differential. **Journal of behavior therapy and experimental psychiatry**, Elsevier, v. 25, n. 1, p. 49–59, 1994.

DOIT. **DOIT**. 2018. <<http://www.doit.am/>>. Accessed: July 30, 2018.

FAGIN, B.; MERKLE, L. Measuring the effectiveness of robots in teaching computer science. In: **Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education**. New York, NY, USA: ACM, 2003. (SIGCSE '03), p. 307–311. ISBN 1-58113-648-X. Disponível em: <<http://doi.acm.org/10.1145/611892.611994>>.

HAYASHI, E. C. S. et al. Exploring new formats of the self-assessment manikin in the design with children. In: **Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2016. (IHC '16), p. 27:1–27:10. ISBN 978-1-4503-5235-2.

INVENTOR, A. **Anyone Can Build Apps That Impact the World**. 2018. <<http://appinventor.mit.edu/explore/>>. Accessed: November 20, 2018.

KARLTH. **ARDUBLOCK for ESP**. 2018. <<https://github.com/karlTH/arduino-esp-ardublock>>. Accessed: October 25, 2018.

LITTS, B. K. et al. Understanding high school students' reading, remixing, and writing codeable circuits for electronic textiles. In: **Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education**. New York, NY, USA: ACM, 2017. (SIGCSE '17), p. 381–386. ISBN 978-1-4503-4698-6. Disponível em: <<http://doi.acm.org/10.1145/3017680.3017740>>.

MARTINEZ, C.; GOMEZ, M. J.; BENOTTI, L. A comparison of preschool and elementary school children learning computer science concepts through a multilanguage robot programming platform. In: **Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education**. New York, NY, USA: ACM, 2015. (ITiCSE '15), p. 159–164. ISBN 978-1-4503-3440-2. Disponível em: <<http://doi.acm.org/10.1145/2729094.2742599>>.

SABIN, C. A. **O movimento maker na Educação**. 2017. <<https://educacao.estadao.com.br/blogs/albert-sabin/o-movimento-maker-na-educacao/>>. Accessed: July 20, 2018.

Shenzhen Doctors of Intelligence and Technology Co. Ltd. **ESPduino**. 2018. <<https://github.com/SmartArduino/SZDOITWiKi/wiki/Arduino---ESPduino>>. Accessed: August 1, 2018.

THUNKABLE. **Thunkable: Build your own apps**. 2018. <<https://thinkable.com/>>. Accessed: November 20, 2018.

VALENTE, J. **Celular se torna principal forma de acesso à internet no Brasil**. 2018. <<http://agenciabrasil.ebc.com.br/geral/noticia/2018-07/celular-se-torna-principal-forma-de-acesso-internet-no-brasil>>. Accessed: November 25, 2018.