

Sistema Automatizado de Controle de Estufas para Cultivo de Hortaliças

Douglas Guilherme Fernandes¹, Evandro Preuss², Teresinha Leticia da Silva²

¹Curso de Bacharelado em Sistemas de Informação,

²Departamento de Tecnologia da Informação

Universidade Federal de Santa Maria (UFSM)

Campus Frederico Westphalen - RS

douglasfr20@gmail.com,

evandro.preuss@ufsm.br, leticiasilva.ufsm@gmail.com

Resumo. *Em tempos de ampliação tecnológica no setor da agricultura encontra-se a carência opções de sistemas automatizados pensados para pequenos agricultores. Esses sistemas precisam apresentar uma garantia da qualidade do produto cultivado, de forma mais simples e eficiente, além de apresentar um baixo custo de desenvolvimento, conservando a competitividade de mercado do produtor. O presente trabalho apresentar uma solução automatizada para o controle da temperatura, luminosidade e umidade do solo em estufas, voltado para pequenos e médios produtores rurais de hortaliças. O resultado do trabalho é a implementação de um protótipo automatizado para estufas, baseado na plataforma Android e Arduino, operado através de um aplicativo instalado em um smartphone com sistema operacional Android, proporcionando vantagens como acesso aos dados da plantação, controle e monitoramento em tempo real da estufa.*

Palavras-chave: *Arduino. Android. Controle de Estufa automatizado.*

Abstract. *At a time of technological expansion in the agricultural sector is the lack of automated systems options designed for small farmers. These systems need to provide a guarantee of the quality cultivated product, in a simpler and efficient way, besides presenting a low development cost, preserving the market competitiveness of the producer. The present work aims to present an automated solution for the control of soil temperature, luminosity and humidity in greenhouses, aimed at small and medium - sized rural producers of vegetables. The result of the work is the implementation of an automated prototype for greenhouses, based on the Android platform and Arduino, operated through an application installed on a smartphone with Android operating system, providing advantages such as access to planting data, control and monitoring in real time the greenhouse.*

Keywords: *Arduino. Android. Automated Greenhouse Control.*

1. Introdução

O clima no Brasil é bastante propício para o cultivo de diversos tipos de hortaliças, pois existem gêneros que são típicos de temperatura tropical como pimenta, pepino, coentro e feijão-vagem entre outros, enquanto que existem também as de temperatura amena, como por exemplo a alface, rúcula, ervilha, morango e cebola entre outras. Por outro lado, o Brasil como um todo tem muitas variações climáticas, com altas temperaturas e amplitudes térmicas, frio extremo, temporais de granizo, períodos de seca, grande quantidade de pragas e doenças nas lavouras. Isso torna-se uma das dificuldades para o cultivo de hortaliças a céu aberto, outro problema é o controle sobre os recursos utilizados para o cuidado da plantação como água, energia elétrica e gastos com mão de obra para

o cuidado permanente da plantação, isso faz com que vários agricultores adotem o cultivo em ambientes protegidos como a estufa para diminuir os riscos, redução da perda dos recursos e uma diminuição de mão de obra.

As estufas são estruturas fechadas que permitem o controle da temperatura, umidade, luminosidade e controle eficiente de pragas e doenças, podendo se adaptar a diversos tipos de cultivo, permitindo o controle do microclima adequado durante todas as estações do ano. A utilização de estufas tornou-se fundamental para o abastecimento seguro dos produtos alimentares no mundo. Atualmente encontram-se nas propriedades rurais estufas designadas à produção de diversos tipos de hortaliças, onde são utilizados procedimentos de irrigação e umidificação, mas que exigem uma grande atenção e mão de obra para manter os níveis adequados, pois praticamente não há um registro e controle dos dados adequados.

O uso de sistemas automatizados em estufas vem crescendo ano após ano, pois os produtores buscam cada vez mais tecnologias que possam auxiliar no dia a dia, para diminuir a possibilidade de erros comuns no cuidado das plantações, aumento da produção e também para baixar os custos com contratação de mão de obra. Processos de automação garantem um controle total do ambiente, proporcionando maior desenvolvimento da produtividade e qualidade dos produtos com maior eficiência e uso coerente dos recursos.

Desta forma, é fundamental que se desenvolva um sistema de informação que automatize essas tarefas, como o controle de umidade, de temperatura e de luminosidade e registre os dados coletados através dos sensores para futuras tomadas de decisão. Através do uso desse tipo de sistema é possível ter inúmeras vantagens, como a diminuição da intervenção manual, reduzindo a chance de erros primários, monitoramento dos dados 24 horas por dia, redução da admissão de mão de obra, maximização da produtividade e qualidade dos produtos. Uma vez que há a automação dentro da estufa, a plantação utiliza aquilo de que carece na dose e na hora certa.

Neste sentido, o presente trabalho tem por objetivo o desenvolvimento de um protótipo automatizado para controle de dados de uma estufa para cultivo de hortaliças, tais como a temperatura, umidade do solo e luminosidade em estufas, utilizando as plataformas Arduino e Android e armazenando os dados das variáveis de controle do microclima numa base de dados para futuras comparações e auxílio nas tomadas de decisão.

2. Referencial Teórico

Esta seção apresenta um breve referencial teórico sobre as áreas envolvidas neste trabalho, destacando o Arduino, características de sensores e a plataforma Android.

2.1 Arduino

O Arduino pode ser definido como uma plataforma de prototipagem eletrônica de código aberto, composto por um microcontrolador, com uma estrutura de entradas e saídas acopladas que podem ser conectados a outros circuitos e sensores. São processos digitais interligados a sensores e atuadores, permitindo a construção de sistemas que possam captar valores da realidade e consequentemente responder ações físicas programadas (FONSECA; BEPPU, 2010).

Um microcontrolador pode ser definido como um *chip* eletrônico, com uma inteligência programável, utilizado no controle de processos lógicos. Toda a lógica de execução é elaborada em forma de programa e escrita no microcontrolador, sendo

executada toda vez que o componente é energizado (SOUZA, 2009). De forma simples, pode-se conectar dispositivos de entrada e saída às portas de comunicação do Arduino, programando a plataforma de maneira a realizar a atividade desejada.

A plataforma Arduino é usada para desenvolver objetos comunicativos independentes, também pode ser conectado a um computador e à Internet com o propósito de buscar e enviar dados e atuando sobre eles. Como por exemplo, ele pode mandar um lote de dados obtidos de sensores para um *site*, dados estes que podem ser mostrados em forma de gráfico (MCROBERTS, 2011). Com a plataforma Arduino pode-se conectar a variados tipos de periféricos, como *displays*, botões, sensores, módulos *Ethernet*, entre outros, qualquer equipamento que forneça dados ou possa ser monitorado pode ser utilizado.

A plataforma Arduino dispõe de diferentes versões, onde basicamente o que muda são as quantidades de portas de comunicação disponíveis, sendo que a versão que foi empregada no protótipo proposto neste trabalho é chamada Arduino Mega, que possui recursos bem interessantes para prototipagem e projetos mais robustos, contém 54 portas de comunicação de entrada e saída digitais, 16 portas analógicas e 4 portas de comunicação serial. A versão faz uso de um microcontrolador ATMEGA2560, um cristal oscilador e um regulador linear de 5v. A placa conta com conectores associados às entradas e saídas do microcontrolador, possibilitando a conexão dos periféricos, possui também uma porta USB (Universal Serial Bus), um *chip* programado como conversor USB para serial, que proporciona conectá-lo a um computador para mandar e obter dados (ARDUINO, 2017a).

A figura 01 mostra a visão superior da plataforma do Arduino Mega, bem como seus recursos com um breve resumo sobre eles.

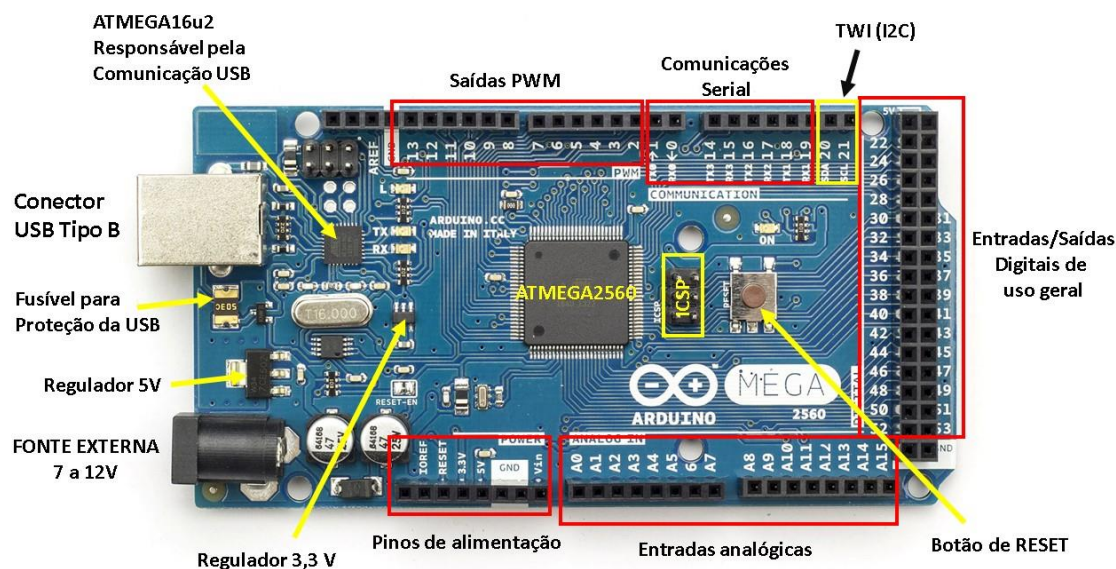


Figura 01 - Placa Arduino Mega

Fonte: <https://www.embarcados.com.br/arduino-uno/>

O *firmware* do Arduino é programado através da IDE (*Integrated Development Environmen*) do Arduino, um *software open source* onde é feita toda a programação do código que será executado pelo equipamento, a linguagem de programação é baseada nas linguagens C e C++ (MCROBERTS, 2011). O *software* pode ser baixado no portal oficial do Arduino, encontram-se disponíveis versões para Windows, MAC, e Linux, no portal também é capaz de ter acesso ao código básico de programação (ARDUINO, 2017b).

Na figura 02 é apresentado a tela principal da IDE do Arduino, e um resumo sobre suas funções.

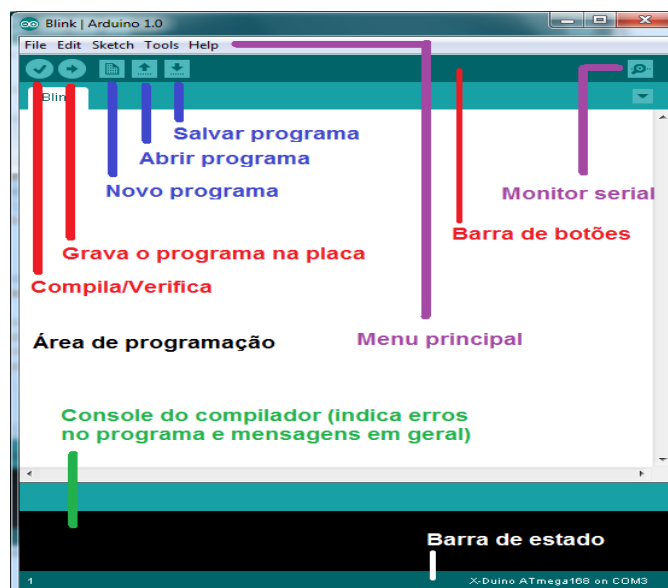


Figura 02 - IDE Arduino

Fonte: <http://projetoardroid.blogspot.com.br/p/arduino.html>

2.2 Sensores

No estudo da automação é necessário definir condições para o sistema obter valores de variáveis físicas do ambiente a ser controlado e esta é a atividade dos sensores (WENDLING, 2010). Sensores têm a função de informar ao circuito eletrônico um fato que ocorra fora, sobre o qual ele deve agir, ou a partir de um fato que ele deva gerenciar a uma determinada ação (WENDLING, 2010).

Para a plataforma Arduino, os sensores são responsáveis por captar as informações do ambiente e as transformar em sinais elétricos que podem ser interpretados pelo Arduino. A informação pode ser lida através de mais variados tipos de sensores como o sensor de temperatura, sensor de umidade e sensor de luminosidade entre outros. Os sensores que foram utilizadas nesse sistema são descritos nas seções a seguir.

2.2.1 Sensor de Temperatura LM35

O LM35, apresentado na figura 03, é um circuito integrado, para medição de temperatura em graus centígrados, é um termômetro preciso e sensível, tem uma voltagem de saída analógica, sua faixa de medição é de -55C a +150C com uma precisão de mais ou menos 0,5C. O sensor não requer uma calibração externa para oferecer os dados com precisão, mas para poder ser interpretado pelo Arduino requer uma conversão através de um cálculo matemático, visto que o Arduino interpreta apenas dados inteiros dentro de 0 a 1023 devido à conversão analógico-digital de 10 bits (MCROBERTS, 2015).

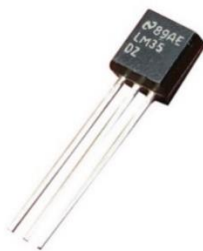


Figura 03 – Sensor de temperatura LM35

Fonte: <http://blog.vidadesilicio.com.br/arduino/basico/lm35-medindo-temperatura-com-arduino/>

2.2.2 Sensor de Umidade do solo FC-28

O sensor de umidade do solo FC-28 (figura 04) pode ser usado para detectar as variações de umidade. Utiliza duas sondas para passar corrente através do solo, e faz leitura da resistência para obter o nível de umidade ideal. Pode ser usado na areia, terra ou até mesmo diretamente na água, com alimentação de 3.3V ou 5V e sinal da tensão de saída de 0V a 4.2V (MCROBERTS, 2011).

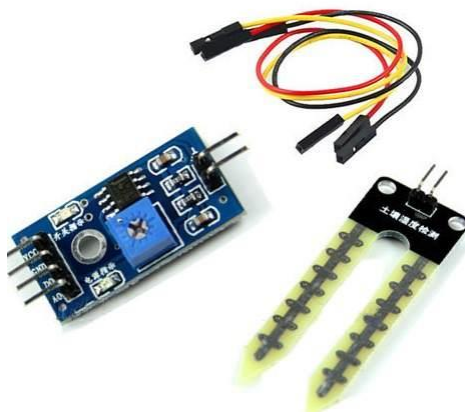


Figura 04 – Sensor de Umidade do solo FC-28

Fonte: http://www.tato.ind.br/product_images/f/853/Umidade__66906_zoom.jpg

4.2.3 Sensor de Luminosidade LDR

O sensor de luminosidade LDR (*Light Dependent Resistor*), apresentado na figura 05, é um componente cuja resistência varia de acordo com a intensidade da luz. Quanto maior for a intensidade da luz sobre ele, menor a resistência e quando menor for a intensidade da luz maior sua resistência (MCROBERTS, 2015).



Figura 05 – Sensor de luminosidade LDR

Fonte: http://www.huinfinito.com.br/542-thickbox_default/fotocelula-ldr-10mm.jpg

2.3 Plataforma Android

Android representa um conjunto de *softwares* para dispositivos móveis como sistema operacional, *middleware* e aplicativos essenciais. O Android SDK (*Software Development Kit*) oferece as ferramentas e APIs (*Application Programming Interfaces*) obrigatórias para começar a desenvolver aplicações que rodam em produtos com Android (GOOGLE, 2017).

Esta plataforma de *software* para dispositivos móveis é baseado no *kernel* do Linux, de uma forma geral, o Android é um máquina virtual Java sendo executada sobre o *kernel* (esqueleto principal que liga as aplicações ao processamento real do sistema, gerenciador de recursos) Linux, oferecendo apoio para desenvolvedores de programas Java por meio de um agregado de bibliotecas e serviços (PRADO, 2017). Possui uma arquitetura de quatro camadas apresentada na Figura 06:

- a) **Aplicações:** Nesta camada ficam as aplicações (desenvolvidas em Java) para Android (PRADO, 2017).
- b) **Framework de Aplicações:** Esta camada é encarregada pela *interface* com as aplicações Android, prevê um conjunto de bibliotecas para ter acesso a diversos recursos do dispositivo como localização GPS (*Global Positioning System*), armazenamento no cartão SD (*Secure Digital*), telefonia e interface gráfica. O *framework* já vem pré-instalado com o Android (LECHETA, 2016).
- c) **Bibliotecas e Serviços:** Esta camada destina-se as principais bibliotecas empregadas pelo Android, dentre elas a *OpenGI/ES* para trabalhar com gráficos e a *SQLite* que possibilita manipular a base de dados. Situa-se também a *Dalvik*, que é a JVM (*Java Virtual Machine*) para executar o conteúdo Java, a grande maioria das bibliotecas foi feita na linguagem C++ (PRADO, 2017).
- d) **Linux Kernel:** Nesta camada contém programas de gerenciamento da memória, configurações de segurança, o *software* de comando de energia e muitos drivers de *hardware* (LECHETA, 2016).

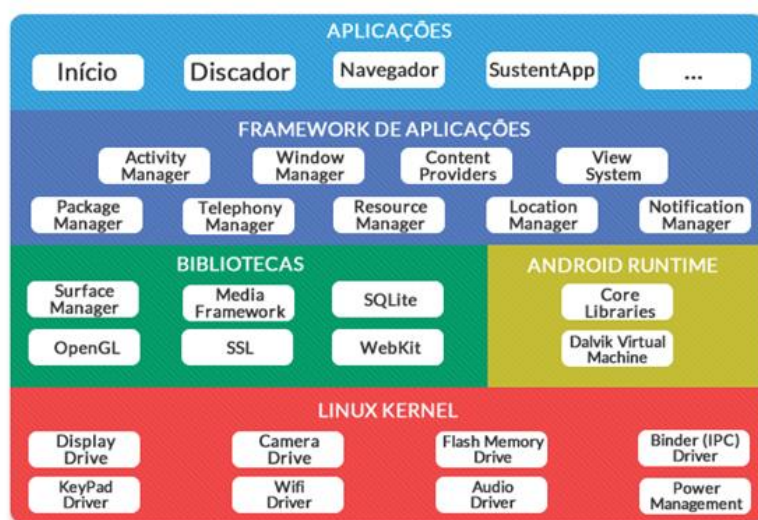


Figura 06 – Camadas da Arquitetura Android

Fonte: <http://quase1nerd.com/wp-content/uploads/2016/07/arquitetura-android-quase1nerd.png>

A JVM é executada em cima do *kernel* do Linux existente no SO (Sistema Operacional) Android chamado de Dalvik. A Dalvik não emprega *bytecode* (estágio intermediário entre código fonte e a aplicação final) Java, e sim *dexcode*, com isso a

Google criou o instrumento chamado “dx”, que transforma Java *bytecodes* (.class) em *dexcodes* (.dex). Nas versões atuais, o Android possui uma execução de JIT (Just-in-time), que compila *dexcodes* para arquitetura-alvo em tempo de execução, facilitando os processos e tornando-os mais rápidos (PRADO, 2017).

Atualmente encontram-se sete versões do Android acessíveis, saber qual é a versão com maior uso faz com que mais usuários podem baixar e utilizar sua aplicação. No site oficial do Android existe um gráfico que mostra as versões disponíveis e qual a porcentagem de usuários (GOOGLE, 2017). Para o desenvolvimento de uma aplicação, a plataforma Android permite uma grande variedade de opções, que incluem elementos de tela, acesso a uma base de dados interna no dispositivo (SQLite) e acesso à Internet para as mais variadas interações, como por exemplo acesso aos webservices, hospedados num servidor web, que podem fazer a troca de informações da aplicação Android com uma base de dados MySql, por exemplo.

As classes Activity e View, são encarregadas pela reprodução gráfica de uma aplicação Android. A classe android.app.Activity apresenta uma tela composta por elementos visuais, os que são representados pela classe android.view.View, estas duas classes andam juntas, *activity* define que existe uma tela, controla seu estado e a passagem de parâmetros de uma tela para outra, controla também os métodos que serão invocados quando uma ação for feita, mas a *activity* necessita exibir elementos visuais na tela, este é o trabalho da classe *View*, que tem a finalidade de desenhar algo na tela (LECHETA, 2016).

3. Estado da Arte

Esta seção apresenta alguns trabalhos relacionados a esta proposta, desenvolvidos na área de Automação em Estufas baseadas na Plataforma Arduino e Android. Ao final da seção são identificadas e apresentadas semelhanças e características com relação ao trabalho apresentado.

3.1 Sistema Automatizado para Controle de Umidade e Temperatura em Cultura de Morangos Aplicados aos Pequenos Produtores

Santos (2014) apresenta um sistema de irrigação automatizado utilizando-se de um sistema microcontrolado Arduino, a custo acessível e aplicado a pequenos produtores rurais. Toda a comunicação do projeto é realizada através da radiofrequência.

A proposta do trabalho foi a de desenvolver um *hardware* eletrônico capaz de realizar os comandos de controle da irrigação. O sistema proposto realiza a comunicação da plantação e a casa de máquinas através de radiofrequência, basicamente na casa de máquinas onde ficará o *hardware* com o Arduino, receberá as informações da plantação via radiofrequência. O protótipo também conta com uma bomba d’água, que será ativada ou desativada dependendo das informações vindas da plantação e sua comunicação também será através de radiofrequência. Outra função deste protótipo é a de abrir e fechar a estufa, para aumentar a claridade, utilizando um servomotor.

Neste trabalho o autor escolheu para desenvolver seu protótipo com os seguintes materiais: Arduino Nano, sensor de temperatura, sensor de umidade do solo, *display* LCD, módulo de relé, módulo Link de radiofrequência, sensor de luminosidade, Servomotor e bomba de aspersão. A figura 07 apresenta o esquema das entradas do protótipo.



Figura 07 – Esquema das entradas do protótipo

Fonte: Santos (2014)

O autor desenvolveu um software em linguagem C, para a utilização do controle do Arduino, neste programa encontram-se as definições dos valores de temperatura e umidade, os parâmetros para os atuadores serem ligados e desligado, a programação do *display* LCD, que faz o papel da interface humano computador, através do *display* o usuário saberá os dados de temperatura, umidade e status da bomba, os valores dos parâmetros para os sensores foram coletados por meio de um técnico da Embrapa que auxiliou no desenvolvimento e ajuste dos dados como temperatura e umidade ideal para a plantação de morangos. Também construiu um protótipo simulando uma plantação, onde é possível visualizar toda a estrutura pretendia pelo projeto e suas funcionalidades. O circuito desenvolvido é capaz de realizar as medições de umidade e temperatura, a figura 08 apresenta o protótipo do projeto.

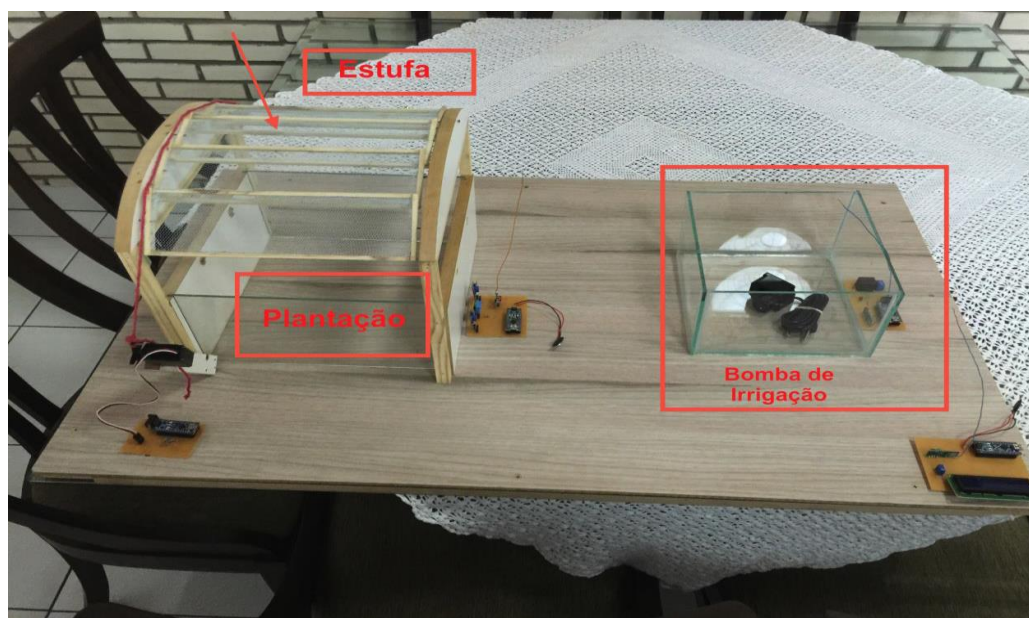


Figura 08 –Protótipo do projeto

Fonte: Santos (2014)

A autora conclui que o projeto atendeu todas as necessidades propostas de criar um modelo de sistema de irrigação capaz de ser acionado e/ou desligado de acordo com o cultivo escolhido. Ainda assim o projeto tem uma grande quantidade a ser melhorado, como a utilização de sensores com maior precisão, um desenvolvimento de um programa

para *Desktop* onde o agricultor possa acionar e/ou desligar a irrigação via computador, instalação de câmeras em toda a plantação, possibilitando assim que o agricultor acompanhe seu cultivo e veja o que está sendo feito nele e uma irrigação com a reutilização da água da chuva.

3.2 Desenvolvimento de um Irrigador Microcontrolado Integrado a um Sistema de Supervisão e Controle

Dutra, Lang e Waromby (2013) apresenta um protótipo de pequeno porte de um sistema de irrigação baseado no microcontrolador Arduino, comandado por meio de dispositivos móveis que possuam a tecnologia wireless, utilizando um sistema de supervisão e controle. Os autores desenvolveram um sistema que realiza de forma automática a irrigação de acordo com a temperatura e umidade do ar. O sistema conta com um reservatório de água, que tem seus limites mínimo e máximo controlado de forma automática e seu enchimento é controlado por uma válvula solenoide.

O protótipo possui dois modos de funcionamento: automático e manual. Operando de forma manual, o operador pode acionar as saídas do sistema; no modo automático o equipamento identifica a temperatura e a umidade do ar por meio de sensores. Através dos dados dos sensores é que será definido o tempo da irrigação, a vantagem de ter dois modos de operação é que o usuário poderá escolher se ele quer controlar seu sistema de forma manual, ou seja, fazer uma ação com o programa a qualquer momento, independentemente dos valores dos sensores e atuadores ou deixar de forma automática que o próprio programa toma as decisões. Os autores utilizaram um material de apoio da Embrapa para fazer a análise dos dados dos sensores para determinar o tempo da irrigação. A figura 09 apresenta o protótipo do projeto.



Figura 09 –Protótipo do projeto

Fonte: Lang (2013)

A operação e o monitoramento do sistema ocorrem através de um sistema de operação e controle desenvolvido no ScadaBR (www.scadabr.com.br). O sistema pode

ser acessado a partir de um desktop, *notebook*, *tablet* ou *smartphone*. O sistema de supervisão e controle é executado em um computador que fica conectado ao controlador Arduino através da porta USB e a comunicação ocorre de forma serial. Os comandos inseridos pelo operador são registrados em variáveis e enviadas ao controlador, que processa e ativa as saídas de acordo com o código de programação, enviando ao sistema de supervisão e controle o valor atual das saídas, para que possa ser exibido ao operador.

Os autores fizeram uso do software ScadaBR para fazer controle de seu projeto, que executa a partir de um servidor de aplicação. O computador servidor foi conectado ao um roteador *wireless*, desta forma os demais dispositivos que queiram acessar o sistema devem estar conectados no mesmo roteador. ScadaBR é um software livre, gratuito e de código-fonte aberto, para desenvolvimento de aplicações de automação, aquisição de dados e controle supervisão. A figura 10 apresenta a interface do ScadaBR.



Figura 10 –Interface do ScadaBR

Fonte: Lang (2013)

Após o protótipo ser finalizado os autores destacam que encontraram dificuldades como, no desenvolvimento do código, pois a linguagem utilizada pelo controlador não era familiar a equipe, outro obstáculo foi na utilização dos sensores, pois o material fornecido pelos fabricantes não possuía detalhamento de suas programações, com o término do projeto atingiu-se os objetivos propostos. Os autores destacam que, em um trabalho futuro, sugere-se a implantação de sensores industriais, que possuem maior precisão e durabilidade, o que aumentaria o grau de confiabilidade do projeto.

3.3 Sistema Automatizado para Irrigação de Estufas

Madalosso (2014) apresenta um projeto de automatização do processo de irrigação de estufas, em sistema de malha fechada, utilizando um microcontrolador e sensores de umidade. No projeto foi usado um microcontrolador para acionar válvulas solenoides para o fornecimento de água, a informação da quantidade de água suficiente viria de sensores de umidade do solo e a comunicação do sistema com o usuário é através de uma interface de rede a um servidor gratuito, o qual armazena os dados em uma base de dados e o usuário acessa a esta base através de uma interface *web*.

O autor fez uso de um *kit* Stellaris LaunchPad LM4F120, desenvolvido pela Texas Instruments, que é uma plataforma para desenvolvimento em microcontroladores baseado na tecnologia ARM Cortex - M4F, também desenvolvida pela Texas Instruments. Este kit contém vários componentes para desenvolvimento de sistemas, utilizando um microcontrolador LM4F120H5QR. A figura 11 apresenta o *kit* Stellaris LaunchPad.

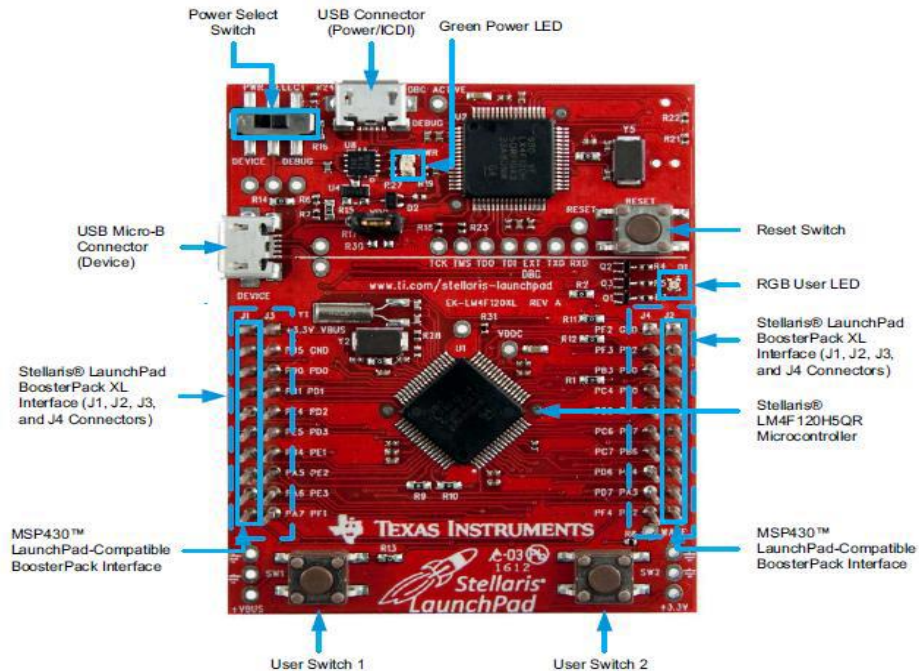


Figura 11 – Stellaris LaunchPad

Fonte: Malalosso (2014)

Para o sistema de irrigação o autor utilizou 4 válvulas solenoides, também fez uso de um protótipo para testar este projeto, o autor montou um sistema para distribuição de água, sendo alimentado por uma torneira e tendo 4 válvulas solenoides como saída; cada válvula foi atribuída a um vaso, representando uma estufa e em cada vaso foi atribuído um sensor de umidade do solo, o microcontrolador foi ligado a interface de rede para se comunicar com o servidor. A figura 12 apresenta um esquema da montagem do protótipo.

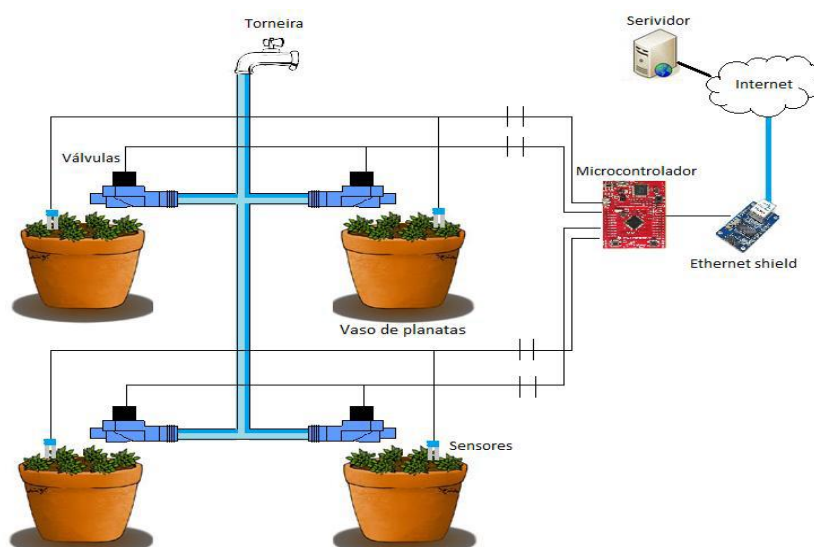


Figura 12 – Esquema de montagem do protótipo

Fonte: Malalosso (2014)

O autor também desenvolveu uma página web utilizando a linguagem PHP (*Hypertext Preprocessor*), capaz de receber as leituras dos sensores de umidade, a página web foi hospedada em um servidor gratuito. A página foi criada para que o microcontrolador enviasse dados dos sensores para o servidor, facilitou a comunicação com o banco de dados, para a realização das comparações definindo se o valor lido para cada sensor estava de acordo com o valor adequado, todas as páginas criadas para a interface entre os usuários do sistema e o banco de dados possibilitou realizar cadastros e acompanhamento do sistema com mais facilidade. A figura 13 apresenta a interface web do sistema desenvolvido.

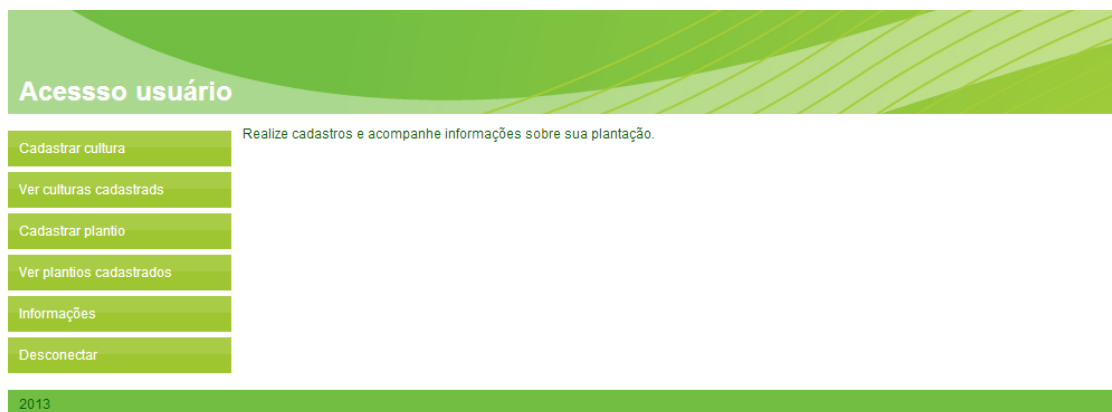


Figura 13 – Página web do sistema

Fonte: Malalosso (2014)

O autor destaca que foi possível verificar com o protótipo construído que o sistema atendeu os objetivos definidos, a implementação do protótipo utilizou diversas áreas abrangidas pelo curso de Engenharia de Computação (programação, banco de dados, redes, eletrônica e microcontroladores). Outra observação foi que em um trabalho futuro, para que o sistema não fique dependendo da Internet para seu funcionamento, uma opção seria fazer com que as informações referentes aos níveis de umidade sejam gravados no próprio microcontrolador e no banco de dados, ou seja, as informações cadastradas pelo usuário seriam gravadas primeiramente no servidor e depois passadas para o microcontrolador.

3.4 Comparativo

O Quadro 1 apresenta um estudo comparativo entre os trabalhos apresentados e a solução proposta neste artigo, onde pode-se notar a utilização da plataforma Arduino, como predominante no uso de projetos de automação dos trabalhos estudados, ficando a cargo dele fazer a comunicação e controle entre os atuadores, sensores e periféricos. Outro dado importante na análise do Quadro 1, é que a preocupação maior é com o controle dos atuadores e se faz muito pouco uso de uma implementação de um aplicativo para controle do sistema e armazenamento dos dados.

Se o projeto proporciona um aplicativo para controle e gestão dos dados, isso possibilita uma praticidade, mobilidade e facilidade em acompanhar os dados e se essa aplicação for a partir de um *smartphone* pode-se ter acesso ao sistema a partir de qualquer lugar. Observou-se que nenhum dos outros programas são dotados de uma função que traz um retorno para o usuário ou um *feedback* em forma de relatório para que o usuário, consiga fazer uma gestão dos dados gravados numa base de dados.

Quadro 1 – Estudo Comparativo

Características	Trabalho de Santos (2014)	Trabalho de Dutra, Lang e Waromby (2013)	Trabalho de Madalosso (2014)	Solução Desenvolvida
Controle de Luminosidade da estufa	Sim	Não	Não	Sim
Controle de temperatura da estufa	Não	Não	Não	Sim
Controle de Umidade do solo da estufa	Sim	Sim	Sim	Sim
Implementação de um Timer automático	Não	Não	Não	Sim
Implementação de um Web Service	Não	Não	Não	Sim
Tipo do Microcontrolador ou tipo do Arduino	Arduino Nano	Arduino Uno	Microcontrolador Stellaris LM4F120	Arduino Mega
Implementação de uma base de dados	Não	Não	Sim	Sim
Implementação de relatório para gestão dos dados	Não	Não	Não	Sim
Cadastro, Edição e Exclusão de itens do Sistema	Não	Não	Sim	Sim
Listagem de todos os itens cadastrados na base de dados	Não	Sim	Sim	Sim
Status do sistema	Sim	Sim	Sim	Sim
Implementação de um Aplicativo Móvel para controle	Não	Não	Não	Sim

Comparando os trabalhos estudados com a solução desenvolvida, verificou-se que todos controlam a irrigação da estufa, uma vez que controlar a irrigação de forma automática traz inúmeras vantagens e benefícios a produção dentro da estufa, proporcionando aumento da eficiência da produção, baixo custo de manutenção, mão de obra e maior gestão dos recursos hídricos. Porém apenas um dos trabalhos armazena os dados numa base de dados e nenhum dos três permite o acesso aos dados e controle através de um aplicativo para *smartphone* ou *tablet*.

4. Solução Implementada

A solução para automação de estufas para o cultivo de hortaliças implementada neste trabalho proporciona aos usuários um controle maior sobre a temperatura, umidade do solo e luminosidade dentro de uma estufa. Para tal, foi desenvolvido um sistema de automação baseado em Arduino e controlado por um aplicativo Android, que armazenam os dados em uma base de dados MySQL, acompanhado de um protótipo do mesmo para demonstrar o seu funcionamento.

Foi criado um plano de automação que permite o controle das variáveis climáticas dentro da estufa, através de um aplicativo Android instalado no celular do usuário, no qual basta cadastrar os sensores e atuadores e realizar a conexão dos sensores e atuadores nas respectivas portas de entrada e saída do Arduino. O aplicativo Android acessa os dados de uma base MySQL através de um *webservice*, cujas informações foram coletadas através das leituras dos sensores conectados em um micro controlador Arduino, a comunicação do Arduino com a base de dados é feita de forma direta através da biblioteca

MySQL própria para o controlador. Com isso, o usuário poderá fazer uma análise verificando todas as medições durante toda uma plantação, podendo assim comparar os dados entre as diferentes plantações e ajustar os parâmetros para obter uma melhor produtividade e um menor custo.

Para o desenvolvimento deste trabalho, foram realizadas as seguintes atividades:

- Levantamento bibliográfico, buscando matérias que permitam conceituar Sistemas de Automação em estufas, bem como trabalho correlacionados.
- Análise dos métodos que melhor podem ser empregados para o tipo certo de automatização da temperatura, umidade do solo e luminosidade.
- Estudo das tecnologias Arduino e Android para o desenvolvimento do protótipo.
- Modelagem do Sistema Automatizado para o controle de temperatura, umidade do solo e luminosidade em Estufas, com uma modelagem da base de dados.
- Implementação de um Protótipo Automatizado para o controle de temperatura, umidade do solo e luminosidade em Estufas, visando validar a proposta, com o objetivo de fazer as leituras dos sensores e acionar os atuadores na estufa.
- Implementação de um *webservice* para fornecer os dados e interagir com o aplicativo Android.
- Implementação de um aplicativo Android para acompanhar e ajustar os parâmetros de temperatura, umidade do solo e luminosidade nas estufas, além de outros dados relacionados à produção de hortaliças nas estufas.
- Implementação de um *software* para o controlador Arduino Mega, capaz de enviar e receber informações dos sensores e atuadores e realizar a comunicação com a base de dados MySQL.
- Teste do protótipo e discussão dos resultados e métodos empregados.

4.1 Plano de Automação

Durante os estudos que embasaram o referencial teórico e o estado da arte deste projeto, se viu a grande quantidade e variedade de sensores, atuadores e capacidade que a Plataforma Arduino disponibiliza. Estudos estes que proporcionaram o entendimento da melhor forma possível para a aplicação destes recursos a este protótipo de automação em estufas.

Também com os estudos se notou que com o uso integrado de um aplicativo *mobile*, a um sistema automatizado utilizando a Plataforma Arduino, se tem inúmeras vantagens tais como: proporcionar máximo controle dos dados uma vez que pode se acessar os dados da estufa a qualquer hora e lugar, possibilitando uma mobilidade maior para o usuário do sistema. Utilizando estas tecnologias da plataforma Arduino somado a plataforma Android e a comunicação com *webservices* e armazenamento dos dados numa base MySQL, se tem um sistema automatizado para estufas mais robusto e seguro.

Fundamentados nos estudos feitos, foi estabelecido um plano de automação para estufas, evidenciando uma diversidade de ações e opções para monitoramento, controle que podem ser desenvolvidos, utilizando uma aplicação *mobile*, sensores, atuadores e uma placa *Arduino* a qual será o centralizador das ações. Neste plano salientam as seguintes funções que podem ser feitas de forma automática, sem intervenção humana.

- a) **Controle da Umidade do solo:** é responsável por verificar a umidade do solo da estufa, as culturas cultivadas dentro de estufas são muito sensíveis ao déficit hídrico do solo, com isso a irrigação do solo é uma prática indispensável. Sempre que o solo estiver seco o sistema ativar a bomba de irrigação e irá desligá-la assim que os sensores identificarem que o solo está úmido o suficiente, o sistema irá agir conforme as configurações dos parâmetros que o usuário cadastrou no sistema, a umidade mínima e a máxima do solo, toda vez que for feita uma nova leitura do solo o programa verifica se houve uma variação de umidade do solo, se tiver ocorrido um aumento ou diminuição da umidade a aplicação armazenará os dados na base, para posterior acesso do usuário.
- b) **Controle da Temperatura:** por meio de um sensor de temperatura, instalado dentro da estufa, o sistema receberá os valores da medição do sensor, com estes dados o sistema determinará se terá a necessidade de ligar o sistema de ventilação para baixar a temperatura dentro da estufa, mantendo uma temperatura adequada para o cultivo das culturas de forma automática. O controle da temperatura funciona 24 horas por dia, assim se houver uma noite muito quente podendo prejudicar a plantação, o sistema funcionará o durante a noite para evitar este tipo de problema, quem informa ao programa a temperatura ideal para a plantação é o operador na hora que ele realiza o cadastro de um novo atuador.
- c) **Controle de Luminosidade:** por meio de um sensor de luminosidade, colocado dentro da estufa, será medida a intensidade da luz. Se a intensidade da luz não estiver adequada à cultura cultivada e dependendo do turno do dia ligará uma luz, ou seja, se estiver com baixa claridade durante um dia, como por exemplo um dia nublado o sistema ligará uma luz *led* branca, proporcionando assim uma luz artificial dentro da estufa compensando a insuficiência de luz natural para o cultivo. E durante a noite se for determinado pelo usuário do sistema poderá ter a iluminação noturna de luz negra (ultravioleta), em função *timer*, esta função possibilita a programação de horários de ativação e desativação de forma independente de sensores. O uso de luz ultravioleta é uma novidade no Brasil em cultivo de hortaliças em estufas, porém em outros países como Canadá e Japão utilizam com a finalidade de aumentar a incidência da radiação fotossintético nas plantas, proporcionando aumento da produção.

Com estas possibilidades de controle, pode-se comprovar a grande utilidade deste projeto, que poderá atender de uma forma geral os produtores rurais de hortaliça proporcionando-os uma série de vantagens. No protótipo que foi implementado para validar a proposta serão automatizadas todas as tarefas constantes no plano de automação.

Como o sistema todo é baseado no cadastro, a partir de um dispositivo móvel, e como cada cultura necessita de diferentes parâmetros de umidade, luminosidade e temperatura, cabe ao produtor ou ao técnico agrícola a definição dos parâmetros mínimos e máximos para cada um dos sensores, de acordo com a cultura.

4.2 Modelagem do Sistema de Automação

Os sistemas de automação, em sua maioria são centralizados, ou seja, um dispositivo gerencia todo o sistema e transporta todos os comandos entre as interfaces de controle e os controladores de dispositivos (GOME; TAVARES, 2013). O projeto mostrado neste artigo também opera desta maneira. Nele contém a interface de controle, desenvolvida na forma de uma aplicação *mobile* na Plataforma Android, controladores de dispositivos construídos com a utilização de *kits* Arduino, uma base de dados (MySQL) onde ficara armazenado todas as informações da estufa e as leituras dos sensores e um *Web Service*

para fazer a comunicação da Aplicação Android e o bando de dados. A Figura 14, demonstra em uma arquitetura de alto nível e como funcionará a comunicação entre os dispositivos do sistema.

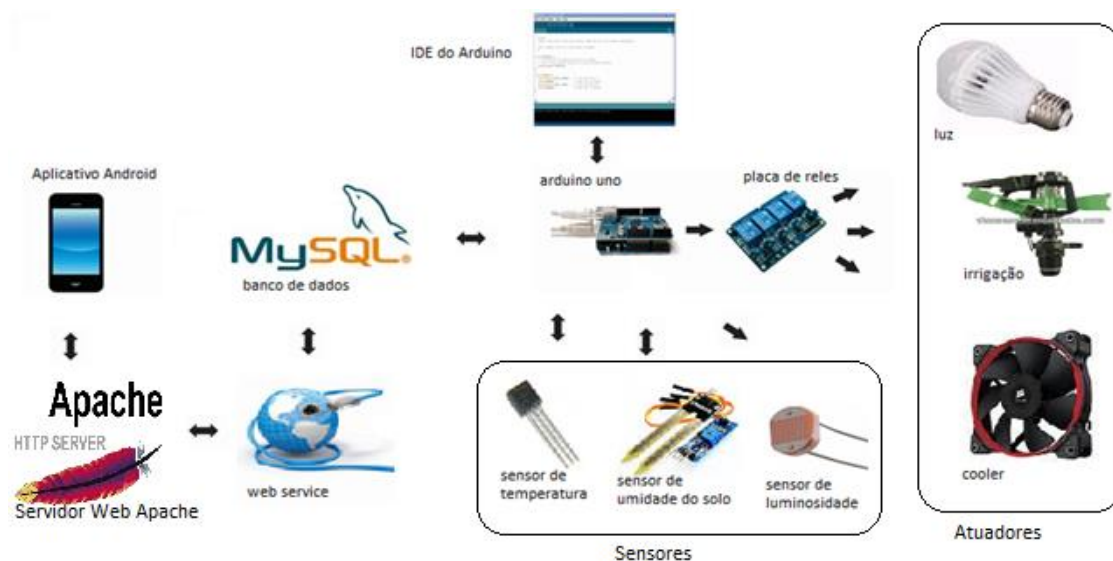


Figura 14 – Projeto do sistema

Fonte: Autor

Como mostra a Figura 14, toda as solicitações são feitas e processadas no Arduino, cabendo a ele gerenciar toda a comunicação com os equipamentos a serem acionados. No aplicativo Android, será possível adicionar uma nova cultura, juntamente com os seus sensores e atuadores vinculados a esta cultura. Após o usuário tem de informar os valores que ele deseja para cada sensor, como por exemplo a temperatura ideal que ele precisa, seu máximo e mínimo, cujos valores serão os parâmetros para o Arduino gerenciar.

Os dados cadastrados são enviados através de um *Web Service*, para uma base de dados, e o Arduino acessará esta base de dados de forma direta para uma maior velocidade na troca de informações, fazendo a leitura dos valores cadastrado para cada sensor, e também irá gravar os dados coletados dos sensores. A comunicação do sistema é feita por meio de uma rede local, através do *Ethernet Shield* conectada ao Arduino.

O banco de dados deste projeto possui um fator muito importante, pois é por meio deste que o sistema irá gerenciar todos os dados do projeto. Nele serão gravados os dados do controlador Arduino, a cultura que será cultivada na estufa, os sensores que fazem as leituras, os atuadores que fazem a ação de ligar e desligar a tarefa automatizada. A figura 15 apresenta a modelagem do banco de dados do projeto, bem como seus respectivos relacionamentos.

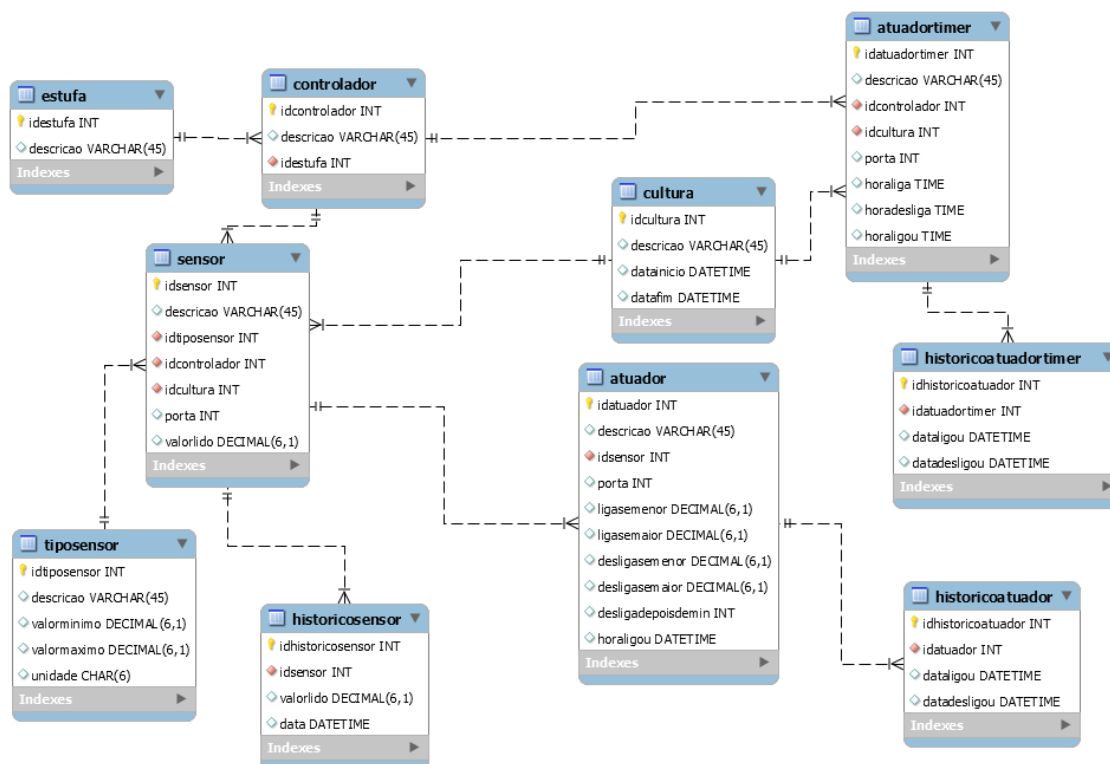


Figura 15 – Modelagem da base de dados

Fonte: Autor

No sistema são cadastrados as estufas que fazem parte do sistema de automação cada estufa está vinculada a um controlador, cada controlador é um Arduino que pode ter um número variável de portas, nas quais serão conectados os sensores e atuadores. O sistema exige pelo menos um controlador Arduino e não há limites de controladores no sistema, sendo que cada Arduino terá um programa instalado que o identifique com o identificador do respectivo controlado. Um único controlador pode ser usado para várias estufas, mas também podem ser usados vários controladores para uma única estufa.

Dentro das estufas podem ter uma ou mais culturas plantadas as quais podem ser controladas de maneira separada, para cada cultura é cadastrada uma descrição para a mesma e a data de início e fim do seu cultivo, através dos valores da data inicial e final é que o sistema saberá o momento de começar seu funcionamento e seu término, o programa só irá funcionar dentro das datas determinadas pelo usuário.

A partir dos cadastros de controlador e cultura, são cadastrados os sensores, cada sensor é vinculado a uma cultura, controlador e a um tipo do sensor, o valor do tipo do sensor é a característica do sensor pois no programa existem três tipos de sensores diferentes um do outro, sendo que cada um foi cadastrado conforme sua característica e valor. Outro campo é o identificador de qual porta o mesmo está conectado. Para cada sensor instalado, um ou mais atuadores podem ser cadastrados, de acordo com os valores dos parâmetros cadastrados, ligará ou desligará um componente conectado na porta que o mesmo está conectado, este atuador poderá ligar ou desligar o dispositivo se o valor do sensor for maior ou menor que o estabelecido, cada atuador está vinculado a uma porta do controlador esta que por sua vez ira emitir o sinal para os componentes ligarem e desligarem.

Ainda poderá ser cadastrado um atuador em função *timer*, vinculado a um controlador (Arduino) que ligará ou desligará um dispositivo, de acordo com um tempo pré-programado, ou seja, independe de sensores. Isso permite, por exemplo, apagar ou

acender uma luz nos tempos pré-programados ou irrigar por 5 minutos, todos os dias a plantação.

Toda a mediação dos sensores e toda a vez que os atuadores forem ativados ou desativados serão gravados na base de dados, como por exemplo, a tabela `historicosensor` guarda o valor lido de cada sensor e a data e hora que ligou e desligou; a tabela `historicoatuador` armazena os horários que foram ativados e desativados cada um dos atuadores vinculados aos sensores e a tabela `históricocoatuadortimer` grava as datas e horas de funcionamento do atuador baseado no *timer*. Estas três tabelas são incrementadas de forma automática por meio de um *trigger* do próprio MySQL, que usa uma função `UPDATE` que grava no banco toda a vez que o último valor lido for diferente do valor anterior. Com isso pode-se fornecer um *feedback* para o usuário, permitindo a ele saber toda uma variação do clima dentro da estufa, bem como os tempos de uso de cada atuador, podendo assim fazer comparações entre as diversas plantações, e podendo ajustar os parâmetros para obter uma melhor produtividade e um menor custo.

A função do aplicativo para o projeto é simplesmente cadastrar na base de dados uma nova cultura plantada na estufa, juntamente com todas as informações de cada sensor e atuador. As informações de cada sensor e atuador poderá ser alterada a qualquer momento pelo usuário. Outra funcionalidade do aplicativo é saber em tempo real o *status* do programa, como a temperatura da estufa, umidade do solo, luminosidade dentro da estufa e quais atuadores estão ligados e quais estão desligados, obtendo instantaneamente uma visão geral do funcionamento do sistema. A aplicação *mobile* tem um recurso para fazer uma pesquisa dos dados de cada sensor, ou seja, poderá fazer uma busca de todos os valores medidos de cada sensor, podendo-se saber toda uma variação de temperatura dentro da estufa de um determinado tempo, bem como seus valores mínimos e máximos.

O *webservice* por sua vez é responsável pela comunicação do dispositivo, *mobile* com o banco de dados, o tipo de *webservice* escolhido para este projeto foi baseado no servidor *web* Apache com implementação em PHP, devido à sua facilidade de implementação, robustez e exigência de poucos recursos de hardware, além de ser um dos servidores *web* mais utilizado no mundo.

4.3 Desenvolvimento do Aplicativo Android

A aplicação *mobile* foi criada para ser bastante simples e intuitiva ao usuário. Ao iniciar o aplicativo é mostrada uma tela de boas-vindas ao usuário, com um menu lateral deslizante, onde se encontram todas as opções do sistema como, cadastrar, listar, status, relatório, sobre e sair. A figura 16 ilustra a tela inicial e o menu lateral com todas as opções do aplicativo.

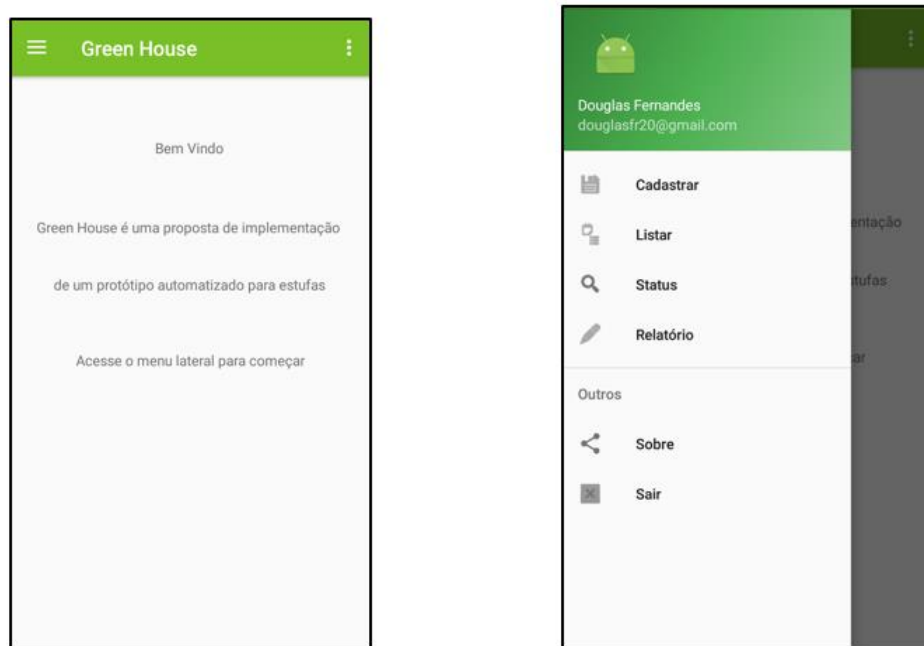


Figura 16 – Tela Inicial e Menu Lateral

Fonte: Autor

A opção de cadastrar permite fazer um novo cadastro no sistema. No total são seis tipos de cadastros: cadastrar estufa, cultura, controlador, sensor, atuador e atuador autônomo. Ao clicar no ícone cadastrar o usuário é redirecionado para a tela de cadastro, que contém quatro botões: um para inserir uma nova estufa e outro para inserir uma nova cultura e outro para voltar e outro para avançar. Nesta tela o usuário pode começar fazendo o cadastro de novas estufas e culturas, que são a parte inicial de cadastro. Ao clicar nos botões de cadastro, será trocada a tela para a tela de cadastro subsequente ao seu botão.

A tela de cadastro de estufa contém uma caixa de entrada de texto para ser informada a descrição da estufa, ou seja, seu nome, e também contém botões de opções que se repetem em todas as telas de cadastro do aplicativo. Nestas opções se tem: *home*, voltar, salvar e avançar. No botão *home*, o aplicativo volta para a tela de boas-vindas; no botão voltar ele volta para a tela anterior; no botão salvar ele salva o cadastro no banco de dados. Antes dele salvar é feita uma validação dos dados informados se tiver alguma divergência dos dados será informado ao usuário e por fim o botão de avançar, este permite ao utilizador navegar dentre todas as telas de cadastro do aplicativo.

Na tela de cadastro da cultura se pede as informações da cultura plantada, sua descrição, a data de início e data do término da plantação, pois é através dos valores da data inicial e final que o sistema controlará a estufa, ou seja são estes dois valores que determinaram o ponto inicial para o controlador começar a fazer a automação e o ponto final que encerrará o controle automatizado sobre a plantação. A figura 17 apresenta as três primeiras telas do cadastro onde é possível fazer um novo cadastro de estufa e cultura.

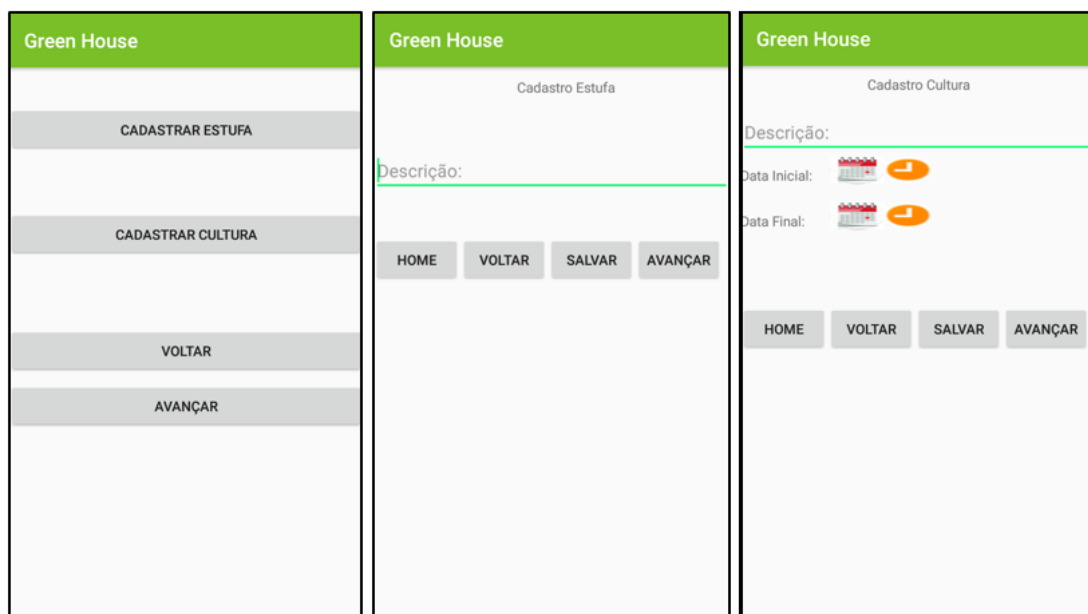


Figura 17 – Primeiras três telas de cadastro

Fonte: Autor

Após finalizado o cadastro da cultura a próxima tela de cadastro será a do controlador, onde se tem o campo de entrada de texto para a descrição do controlador e uma caixa de seleção que contém todas as estufas cadastradas no banco de dados, o usuário deve selecionar a estufa que o controlador está ligado, fazendo assim a conexão entre a estufa e seu controlador. O próximo cadastro é o sensor, onde as informações que se pede são sua descrição, a identificação do tipo do sensor, identificação do controlador (Arduino), identificação da cultura que este sensor faz parte e sua porta de comunicação com o controlador Arduino. A figura 18, mostra a tela de cadastro de controlador e sensor.

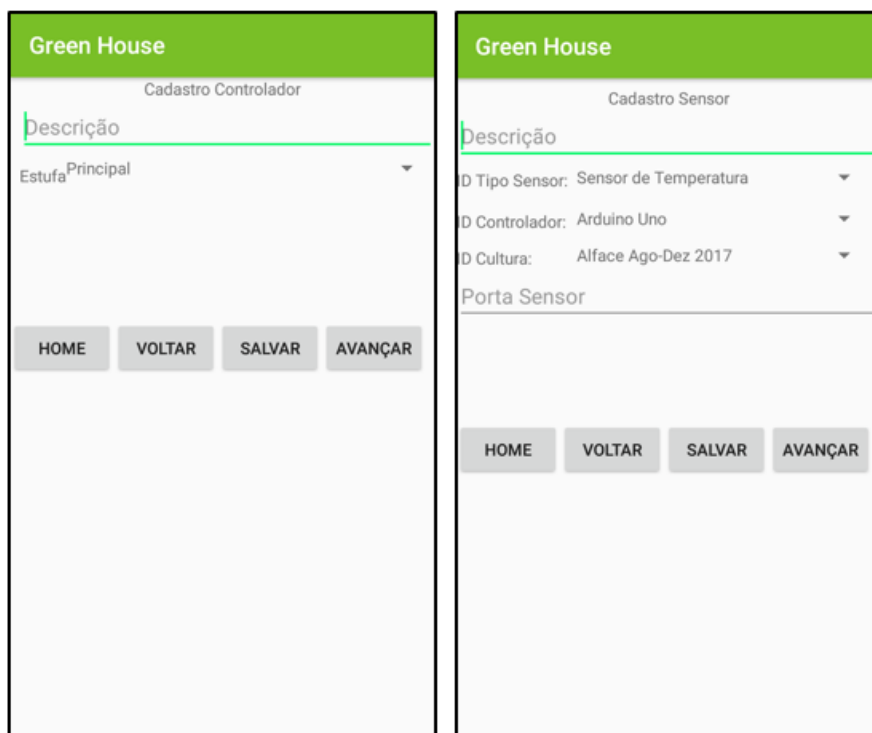


Figura 18 – Tela de cadastro controlador e sensor

Fonte: Autor

Depois de salvo o sensor, tem a tela de cadastro do atuador, que contém as seguintes entradas de informações: descrição, identificador do sensor que irá receber as informações, porta de comunicação com o Arduino, suas informações de atuação, ou seja o valor para ele ligar ou para desligar e também o valor para ele desativar, caso queira que o atuador desligue após outras informações sendo ela um valor maior para o desligamento, ou seja fazendo que tenha duas comparações para desativação de atuadores, é nesta tela que será determinado os parâmetros de atuação para cada sensor como por exemplo a temperatura ideal da estufa, como por exemplo: entre 26º graus de máxima e 22º graus de mínima, estes serão os valores para o controlador monitorar a variação de temperatura, se a temperatura for maior do que 26º graus será ligado o sistema de resfriamento e quando a temperatura estiver abaixo ou igual a 22º graus será desligado o sistema.

Por fim a tela de cadastro do atuador autônomo (baseado num *timer*), que não precisa de sensor para funcionar, e sim apenas dos horários para ele ligar e desligar, contendo sua descrição, identificador do controlador que está ligado, identificador da cultura que ele representa e sua porta de comunicação com o Arduino. Assim, o atuador autônomo só funcionará nos horários programados pelo usuário, como por exemplo ligar às 21 horas e desligar às 22 horas. Após finalizado todos os cadastros o usuário é direcionado novamente para a tela inicial. A figura 19, mostra a tela de cadastro de atuador e atuador autônomo.

The figure displays two side-by-side screenshots of a mobile application interface titled 'Green House'.

The left screenshot shows the 'Cadastro Atuador' screen. It includes a 'Descrição:' field, a 'D Sensor:' dropdown menu with 'Temperatura' selected, and a 'Porta:' input field. Below these is a section titled 'Informações Atuador' with four radio button options: 'Liga se Maior', 'Desliga se Maior', 'Liga se Menor', and 'Desliga se Menor'. Each option has a corresponding 'Valor:' input field. A note states: 'O campo abaixo só deve ser utilizado quando, a opção DESLIGA SE MAIOR, não tiver sido selecionada e se necessário.' Below this is a 'Desliga se maior:' input field. At the bottom are four buttons: 'HOME', 'VOLTAR', 'SALVAR', and 'AVANÇAR'.

The right screenshot shows the 'Cadastro Atuador Autonomo' screen. It includes a 'Descrição:' field, a 'D Controlador:' dropdown menu with 'Arduino Uno' selected, and a 'D Cultura:' dropdown menu with 'Alface Ago-Dez 2017' selected. Below these is a 'Porta:' input field. Further down are two orange clock icons for 'Hora para Ligar:' and 'Hora para Desligar:'. At the bottom are three buttons: 'HOME', 'VOLTAR', and 'SALVAR'.

Figura 19 – Tela de cadastro atuador e atuador autônomo

Fonte: Autor

Feito todos os cadastros o usuário tem a opção de visualizar todos os itens cadastrados, para fazer isso basta clicar no ícone de listar no menu lateral. Com isto será acessado a tela de listagem de todas as opções que foram cadastradas, cada opção se tem um botão correspondente, basta clicar no botão desejado que terá acesso a todos os itens cadastrado do botão selecionado. Após clicado no botão mostrará uma nova tela com todo os itens cadastrados no banco de dados, nesta tela de listagem o usuário poderá editar

qualquer item da lista, basta clicar em cima do item desejado para poder editar, após clicado será direcionado para a tela de edição do item, uma tela bem semelhante à do cadastro, o que muda é o botão de opção que fica em vez de salvar para editar e também possui todas as validações dos campos de informações. Com esta opção de editar o usuário pode modificar a qualquer momento todo o seu sistema, outra questão que muda na tela de edição é o botão deletar que permite excluir o item selecionado pelo usuário. A figura 20 apresenta tela de listar com todos os botões de lista disponível, tela de listagem das culturas cadastradas no banco de dados e tela de edição e exclusão de um item selecionado na lista de cultura.

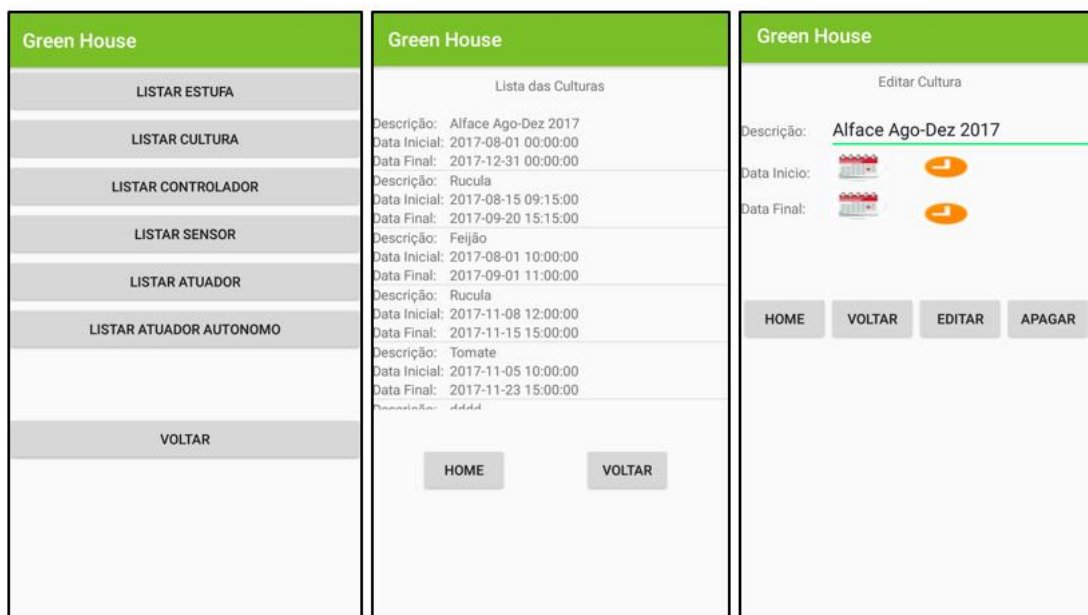


Figura 20 – Tela de opções de listagem, tela de lista de cultura e tela de edição e exclusão de uma cultura selecionada

Fonte: Autor

A opção *status* do menu lateral permite verificar a situação dos atuadores em tempo real. É necessário selecionar qual controlador que será verificado o seu status, feito isso basta clicar no botão pesquisar. O usuário saberá no exato momento da pesquisa quais atuadores estão ligados e quais estão desligados, também é informado o valor atual de cada sensor, podendo ao usuário saber a temperatura, umidade do solo e luminosidade em tempo real da estufa. Com isso ele terá uma noção de funcionamento do sistema, na tela contém também um botão voltar que volta para a tela principal. A figura 21 apresenta a tela de status dos atuadores vinculados ao controlador Arduino 01, bem como os valores dos sensores.

Green House

Status do Sistema

Controlador: Arduino 01

Descrição:

Ar Frio

Status:

Ligado

Sensor:

Temperatura

Valor Lido:

35.0°C

Descrição:

Luz

Status:

Ligado

Sensor:

Luminosidade

Valor Lido:

38.0%

Descrição:

Irrigação

Status:

Desligado

Sensor:

Umidade do Solo

Valor Lido:

62.0%

VOLTAR

PESQUISAR

Figura 21 – Tela de status dos atuadores do controlador Arduino 01

Fonte: Autor

Na tela de relatório, o usuário poderá fazer consultas no banco de dados para obter informações sobre o sensor, atuador e atuador autônomo, com este tipo de pesquisa ele poderá saber todas as informações de leituras dos sensores e todas as vezes que os atuadores foram ligados e desligados. Com isso o usuário terá acesso a toda a informação do clima entre uma plantação e outra, podendo assim fazer comparações, estudos e análise para cada vez mais poder ter plantações mais produtivas e vantajosas. A figura 22 mostra a tela relatório com as opções de relatório e a tela do histórico do sensor de temperatura selecionado entre uma data e hora específicas.

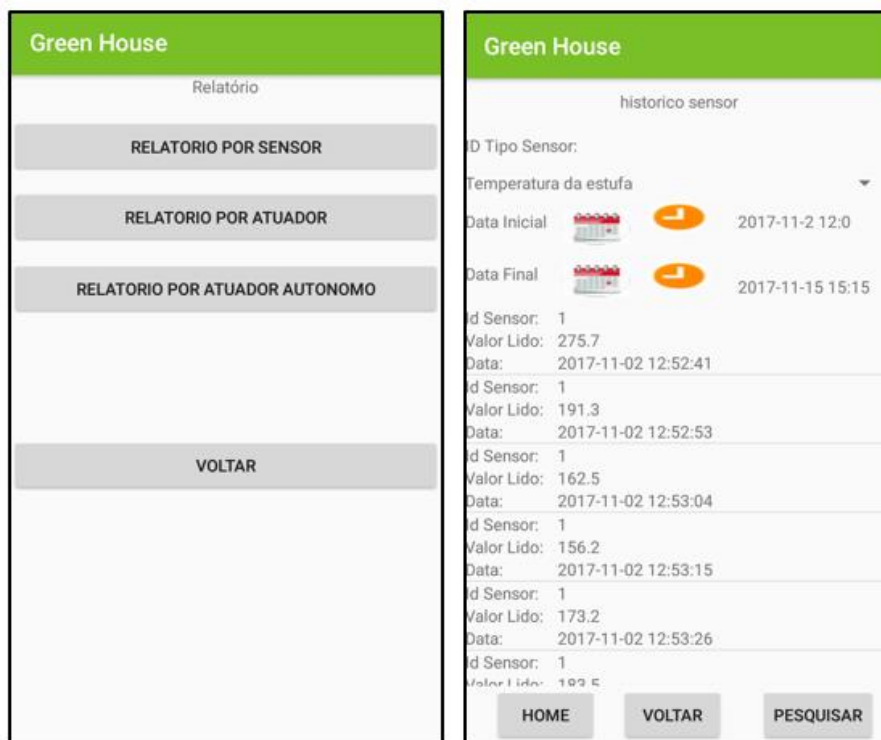


Figura 22 – Tela de relatório com suas opções e tela historio sensor

Fonte: Autor

Por fim, no aplicativo se têm as últimas duas opções no menu lateral a sobre e a sair, na sobre se tem as informações do aplicativo e a sair fecha a aplicação por inteira. Para o desenvolvimento desta aplicação *mobile* para Android, foi usado para a comunicação entre a aplicação e o banco de dados MySQL um *web service* PHP, permitindo a troca de informações de dois sistemas completamente diferentes. Na programação da aplicação foi usado uma biblioteca *Ion* desenvolvida por terceiros que facilita muito a comunicação com *web service* e também permite trabalhar com objetos do tipo *Json*.

A figura 23 apresenta um bloco de código que mostra a utilização da biblioteca *Ion* para cadastrar um sensor na base de dados, é possível ver a chamada ao *web service*, seus parâmetros sendo passados e o método de retorno da chamada ao *web service*. A figura 24 mostra o código do *web service* que é executado após o método fazer a chamada a ele, apresenta as variáveis que recebem as informações passadas via parâmetro, a conexão com o banco de dados, o SQL que é executado e seu retorno a chamada do método.

Código Java

```
Ion.with(getBaseContext()).load("http://10.1.1.200/estufa/CadSensor.php")
    //passa o valor para o obj, passando o parametro que foi digitado
    .setBodyParameter("descricao", edtCadastroSensor.getText().toString())
    .setBodyParameter("idtiposensor", tipoSensorSpinnerId)
    .setBodyParameter("idcontrolador", controladorSpinnerId)
    .setBodyParameter("idcultura", culturaSpinnerId)
    .setBodyParameter("porta", edtPortaSensor.getText().toString())
    .asJsonObject()
    .setCallback(new FutureCallback<JsonObject>() {
        @Override
        public void onCompleted(Exception e, JsonObject result) {
            //testa se deu certo a gravacao
            if (result.get("retorno").getAsString().equals("YES")) {
                //msg de sucesso
                Toast.makeText(getBaseContext(),
                    "Sensor Cadastrado com Sucesso !", Toast.LENGTH_LONG).show();
                //manda pra outra tela do cadastro
                startActivity(new Intent(getBaseContext(), CadAtuador.class));
            }
        }
    });
```

Figura 23 – Bloco de código Java, de cadastro sensor

Fonte: Autor

Código PHP

```
<?php
$descricao = $_POST["descricao"];
$idtiposensor= $_POST["idtiposensor"];
$idcontrolador = $_POST["idcontrolador"];
$idcultura= $_POST["idcultura"];
$porta = $_POST["porta"];

$conn = new mysqli("localhost", "root", "1234", "automacao");
$sql = "INSERT INTO sensor (descricao,idtiposensor,idcontrolador,
idcultura,porta) VALUES (?, ?, ?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("siiii", $descricao, $idtiposensor, $idcontrolador,
$idcultura, $porta);
if($stmt->execute()){
    $retorno = array("retorno" => "YES");
}else{
    $retorno = array("retorno" => "NO");
}
echo json_encode($retorno);

$stmt->close();
$conn->close();
?>
```

Figura 24 – Bloco de código PHP, de cadastro sensor

Fonte: Autor

4.4 Desenvolvimento do Software do Arduino

O passo inicial do programa de controle do Arduino é estabelecer a transmissão de informação entre a plataforma Arduino e a base de dados MySQL, para isso foi utilizado a biblioteca nativa do Arduino a *MySQL_Connection*, que possibilita se conectar a uma base de dados e fazer consultas, alterações e enviando diretamente os comandos SQL

(SELECT, INSERT, UPDATE e DELETE) para o Arduino, a comunicação entre a base e o Arduino é feita via interface de rede que se dá por meio de sua biblioteca SPI, que suporta o protocolo TCP/IP, juntamente com um *shield ethernet* que está acoplado na placa do controlador, com isso o *software* já pode trocar informações com a base de dados. O próximo passo do programa é ficar em *loop* fazendo uma busca na base de dados de todos os sensores que devem ser lidos e de todos os atuadores que devem ser ligados ou desligados. Para cada sensor que for lido, o Arduino verifica se o valor é igual ou diferente da última leitura feita; se o valor foi alterado ele verifica se os respectivos atuadores devem ser ligados ou desligados e atualiza a base de dados. Esse loop é executado a cada 60 segundos. A figura 25 mostra a lógica do código do software do Arduino, baseado nos parâmetros configurados na base de dados MySQL.

```
Busca na base de dados os sensores e seus valores e seus respectivos
atuadores
Para cada sensor
  Lê o valor atual
  Se o valor atual é diferente do último valor lido
    Para cada atuador vinculado ao sensor
      Verifica se precisa ligar ou desligar o atuador
      Atualiza a base de dados com o status do atuador
      Atualiza o valor da última leitura do sensor
Busca na base de dados os atuadores autônomos baseados em timer
Para cada atuador autônomo baseado em timer
  Verifica se precisa ligar ou desligar o atuador
  Atualiza a base de dados com o status do atuador
```

Figura 25 – Lógica do software Arduino

Fonte: Autor

4.5 Montagem do Protótipo

Na elaboração do projeto foi desenvolvido uma maquete em pequena escala, simulando uma estufa. Nela foram instalados os sensores de temperatura, luminosidade e umidade do solo, também contendo seus atuadores, um reservatório de água, um motor elétrico 12 volts para fazer a irrigação dentro da estufa, dois *coolers* 12 volts com o intuito de regular a temperatura no interior da estufa. Também foram instalados duas fitas de *led* uma na cor branca responsável por manter o índice de luminosidade mais elevado em períodos de necessidade e outra na cor ultravioleta para fornecer luz ultravioleta, se o usuário assim quiser para o turno da noite para aumentar o período de fotossíntese das hortaliças.

O controlador Arduino e seu esquema elétrico também foram colocados junto a maquete para facilitar sua montagem com a estufa; um roteador *wi-fi* foi instalado para fornecer a rede de comunicação entre os dispositivos; o Arduino foi conectado com o roteador através do cabo RJ45. Já a base de dados MySQL foi instalada em um notebook, que fica ligado à rede *wi-fi* fornecida pelo roteador. O aplicativo que funcionará como interface do sistema também se conecta com a *wi-fi* do projeto para se comunicar com o sistema. Foi usada uma fonte 12 volts de 5 ampères para fornecer a energia necessária para acionar a bomba de irrigação e outra fonte 12 volts de 1 ampères para os demais atuadores. Para fazer a ligação dos atuadores foi preciso um módulo relé de 4 canais para Arduino. Para fazer uma ilustração de uma cultura plantada na maquete foram colocados 7 quilos de terra e 10 mudas de alface no interior da estufa representando um cultivo de alface. A figura 26 apresenta o protótipo e a instalação do Arduino e seus componentes.



Figura 26 – Protótipo Maquete

Fonte: Autor

4.6 Testes e validação do Protótipo

Antes de iniciar a integração dos componentes, foi necessário realizar seus testes individualmente, primeiro foi testado a conectividade do aplicativo com o banco de dados, de forma que fosse possível estabelecer uma conexão e uma troca de informação

com o banco de dados. Para isso foi feito cadastro de testes em todas as tabelas do banco de dados para verificar o seu funcionamento; após foi feita a listagem dos dados cadastrados e suas edições e exclusões via aplicativo. Com os testes feitos foram necessário fazer ajustes na programação do *App*, para que os dados cadastrados não fossem inseridos de maneira errada pelo usuário, isso acarretaria em erros futuros na aplicação como um todo.

Após feitos testes do aplicativo, foram realizados teste dos componentes do Arduino de forma separada, foram feitos programas de testes individualmente a cada tipo de sensor e atuador, para verificar sua funcionalidade e a maneira que se comportam. Depois foi feita a comunicação do Arduino Mega com o banco de dados, a fim de verificar o funcionamento da troca de informação entre o Arduino e o banco de dados MySQL; foram feitos diversos ajustes para que o software do Arduino se comunique de forma adequada com o banco de dados.

Com isso finalizado o próximo passo foi de fazer a integração de todos os sistemas, como a aplicação Android, o Arduino e o protótipo criado para testar as funcionalidades do sistema desenvolvido. Após feita a integração de todas as partes do sistema (*hardware* e *software*) este foi testado para ver se iria se comportar como o esperado. Inicialmente foi cadastrado no banco de dados via aplicativo uma nova estufa com a descrição de **estufa principal**, uma nova cultura com a descrição de **alface**, a data de início com ano de 2017, mês 11, dia 01 e data de término da cultura com a data de ano 2017, mês 12, dia 20, um controlador com a descrição de **Arduino 01** e um identificador da estufa principal, três tipos de sensores como, sensores de temperatura, umidade do solo e luminosidade, três tipos de atuadores um para a temperatura com os valores de ligar se maior que 30 graus e desligar se menor que 25 graus; irrigação com os parâmetros de ligar se menor que 55% e desligar se maior que 70% e o atuador da luz com os valores de ligar se a claridade for menor que 20% e desligar se for maior que 70% e um atuador autônomo com o horário de ativação de 2 horas da madrugada e desativação às 5 horas da manhã.

Feito estes cadastros, o protótipo estava apto para entrar em funcionamento, foi então iniciado o programa no microcontrolador, executado o servidor Apache instalado em um *notebook*. Verificou-se que na primeira leitura o sistema só coletou as informações, já na segunda leitura começou a funcionar. Primeiramente foi o atuador da temperatura que foi acionado, visto que a temperatura estava acima do informado no cadastro, sendo desligado quando o valor esteve dentro do adequado pelo sistema, em seguida foi testado o sensor de umidade do solo, que se encontrava cravado na terra no interior da maquete. Foi verificado através do aplicativo na opção de *status* que a porcentagem de umidade do solo estava em 63%, dentro do adequado para o programa, foi então feita a retirada do sensor do solo para prover uma diminuição da umidade, com o isso o programa informou uma umidade de 5% e em seguida foi acionado o atuador da irrigação, recolocando o sensor na terra o programa desligou a irrigação porque a umidade do solo voltara aos valores adequados. Outra situação de teste foi de aumentar gradativamente a luminosidade com o auxílio de uma lâmpada *led* para verificar o funcionamento do sensor luminosidade; através da tela de status do aplicativo foi possível ver a porcentagem de luz que se encontrava em tempo real na estufa era de 15%, aumentando a luminosidade acima dos 30% o atuador foi acionado como previsto e aumentando ainda mais a claridade chegando a passar dos 70% o atuador foi desligado, outro teste foi que ele fosse ligado e só desligado a partir que a claridade estivesse abaixo de 20% para isso foi colocado uma caixa na frente do sensor de luminosidade para que ele obtivesse uma claridade inferior a 20% fazendo que o atuador fosse desligado. Para testar

a função *timer* o usuário modificou os horários de acionamentos para poder verificar seu funcionamento foi colocado que o mesmo ligasse às 20 horas e desligasse às 21 horas assim permanecendo ligado por uma hora, o atuador funcionou de forma correta, foi possível verificar na opção de relatório por atuador autônomo os horário que o *timer* ligou e desligou. A figura 27 apresenta os testes realizados.

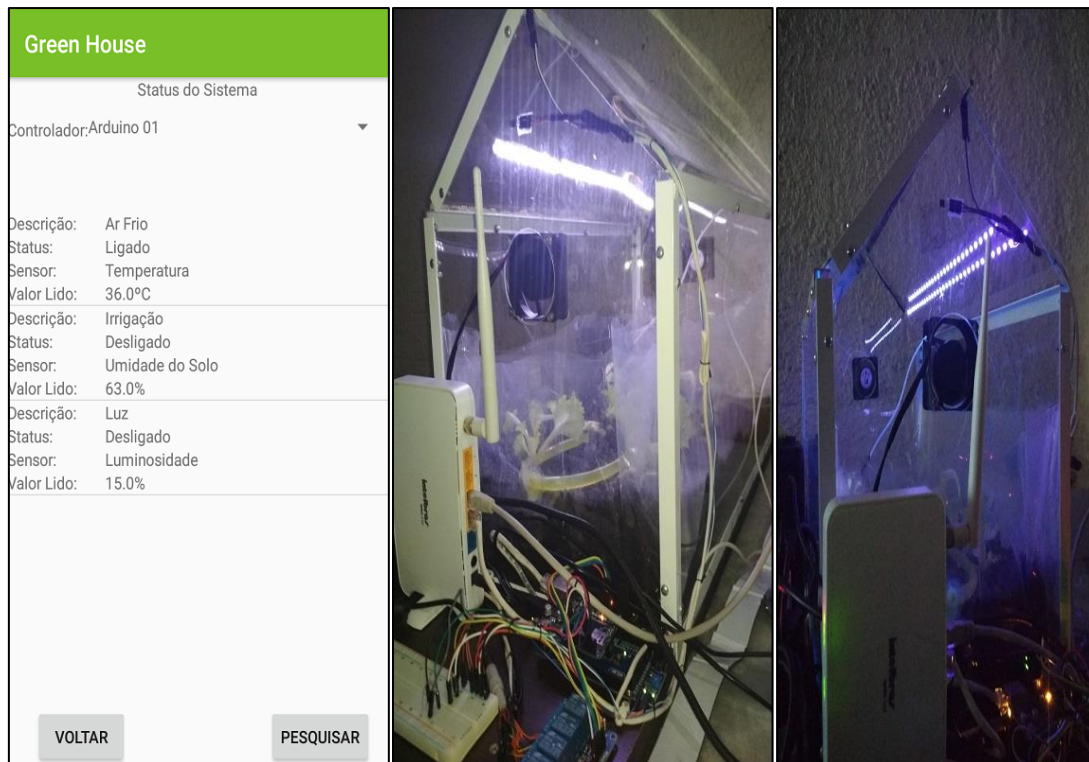


Figura 27 – Testes

Fonte: Autor

Foi verificado que em caso de falhas na conexão com a Internet, o sistema continua a fazer as leituras dos sensores, porém não tem como fazer a troca de informação com a base de dados, mantendo os atuadores ligados ou desligados de acordo com a última leitura obtida, ao ser reestabelecida a conexão o sistema é capaz de voltar a funcionar normalmente, sem a necessidade de reinicializar. Ressalta-se que como o sistema é um protótipo, não há possibilidade de fazer comparações deste com o sistema real que foi estudado, uma vez que as condições ambientais de ambos são diferentes, sendo assim ao testar o sistema esperava-se que este atendesse ao que lhe foi proposto:

- Possibilitar o usuário do sistema ter um aplicativo com sistema operacional Android, onde pudesse realizar cadastros e acompanhamento da estufa.
- Coletar informações de umidade do solo, temperatura e luminosidade dentro da estufa e gravá-los em uma base de dados para se ter um *feedback* do sistema.
- Verificar que o sistema era capaz de utilizar a base de dados, para comparar os dados coletados e interpretá-los corretamente.
- Verificar que a correta interpretação das informações refletia-se nas ativações e desativações dos atuadores corretamente, de acordo com as condições de cada sensor.

Com os testes efetuados foi possível verificar que os requisitos acima foram atendidos.

6. Conclusão

A agricultura é muito importante para o País, pois é dela que a maior parte dos pequenos agricultores obtém seu sustento. Pensando nisso, este trabalho teve como objetivo o desenvolvimento de um protótipo automatizado para estufas, que realiza o controle da umidade do solo, luminosidade e temperatura de forma automática, proporcionando aos agricultores uma forma mais fácil de cuidado com o plantio de hortaliças, um uso coerente dos recursos e uma diminuição de mão de obra.

Foram desenvolvidos dois softwares: um para o controlador do Arduino e outro para uma aplicação *mobile* em linguagem Java para Android; também foi criada uma base de dados MySQL onde são armazenadas todas as informações do sistema, desde os cadastros, leituras dos sensores, acionamentos dos atuadores e construído um protótipo de uma estufa simulando uma plantação, onde é possível visualizar toda a estrutura pretendida pelo projeto e suas funcionalidades.

Os dispositivos pertencentes ao protótipo foram testados individualmente para entender suas funcionalidades e a comunicação com o controlador, para cada item desenvolvido foi criado um código de programação de teste, quando todos os itens foram testados individualmente. Iniciou-se a integração dos equipamentos, com a elaboração dos códigos definitivos do protótipo; após o protótipo ser finalizado, foram realizados diversos testes e validações, havendo a necessidade de pequenas alterações de software e hardware, a fim de conseguir atingir os objetivos propostos.

Uma das dificuldades encontradas na elaboração do sistema, foi no desenvolvimento do código tanto do Arduino como no Android, onde foi necessário aprofundar alguns temas como o uso de Web Service na plataforma Android, como era feita a comunicação entre o aplicativo e o banco de dados MySQL, a forma correta de enviar e receber informações por meio de Web Service, como manipular os dados dentro da aplicação *mobile*, juntamente com os elementos nativos do Android.

Outra dificuldade foi na correta modelagem da base de dados, para que ele pudesse atender o projeto de uma forma bem específica não deixando passar nenhuma informação do sistema, foi preciso fazer várias modificações ao longo do desenvolvimento do protótipo, sempre visando um melhoramento da base de dados, pois a base de dados tem um fator crucial neste projeto, já que ela funciona como o cérebro do protótipo e é onde ficam armazenados todas as informações do sistema; sem ela o programa não seria tão eficiente. Na programação do microcontrolador também se fez necessárias algumas modificações para que o sistema funcionasse de forma correta e foi preciso estudar as funcionalidades de cada sensor antes de colocar todos juntos, testar várias vezes a comunicação do Arduino com o banco de dados, foi necessário estudar como o Arduino fazia a manipulação dos dados antes de serem enviados à base de dados e como ele recebia e alterava as informações coletadas pelos sensores.

Ao final foi possível analisar com o protótipo criado, que o sistema atendeu os objetivos estabelecidos, permitindo a criação de um sistema automatizado de controle de estufas para cultivo de hortaliças, baseado nos cadastros realizados a partir de um aplicativo *mobile*, que permite a interação instantânea com os sensores e atuadores e empregou diferentes temas abordados pelo curso de Sistemas de Informação, como a lógica de programação, programação para dispositivos móveis, linguagem orientada a objetos, banco de dados e redes de computadores.

Este sistema desenvolvido permite, ainda, que o mesmo seja utilizado em outros segmentos, como por exemplo a automatização de aviários, já que basta cadastrar os

sensores e atuadores e seus respectivos valores de controle, além de permitir o acompanhamento de tempos de uso e respectivos cálculos de custo com energia elétrica e poder alterar os parâmetros para obter uma maior produtividade e um menor custo.

6.1 Trabalhos Futuros

Embora o protótipo apresentado possa auxiliar o trabalho realizado pelo agricultor, nos processos de irrigação, controle de temperatura e luminosidade ele pode ser aprimorado em diferentes fatores. Como instalação de um sensor que mede a quantidade de água que se encontra no reservatório, mantendo-o sempre abastecido, a troca dos sensores de umidade do solo e temperatura por um mais robusto e com maior precisão, fazer uma melhor análise dos dados armazenados no banco, providenciando melhorias no sistema com a utilização de técnicas de inteligência artificial.

Outro ponto a ser melhorado seria a possibilidade de trocar a comunicação dos sensores com o Arduino que é feita por cabos, pela utilização de transmissão por radiofrequência. Na aplicação mobile, poderia ser feito um controle de *login* e senha no aplicativo para que só fosse acessado por pessoas cadastradas no aplicativo, fazer a hospedagem do sistema em um servidor *web* ou na nuvem, permitindo acesso ao sistema em qualquer lugar.

Referências

- ARDUINO (2017a) **Arduino Uno**. Disponível em: <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em abril, 2017.
- ARDUINO (2017b) **Arduino Development Environment**. Disponível em: <<http://arduino.cc/en/Guide/Environment>>. Acesso em abril, 2017.
- DUTRA, Lang e WAROMBY (2013), Augusto Fellipe Alves, Arthur Guilherme e Paulo Henrique. **Desenvolvimento de um Irrigador Microcontrolado Integrado a um Sistema de Supervisão e Controle**. Curitiba: Universidade Tecnológica Federal do Paraná.
- FONSECA, Erika Guimarães Pereira da; BEPPU, Mathyan Motta. **Apostila Arduino**. Niterói-RJ: Universidade Federal Fluminense Centro Tecnológico, 2010. 23 p.
- GOMES, E. L. B; TAVARES, L. A. (2013). **Uma solução com Arduino para controlar e monitorar processos industriais**. Santa Rita do Sapucaí: INATEL.
- GOOGLE. Google Code. **Projetos do Google para o Android**. Disponível em: <<https://developer.android.com/index.html>>. Acesso em abril, 2017.
- LECHETA, Ricardo. **Android Essencial**. Editora Novatec, 2016.
- MCROBERTS, Michael. **Arduino Básico**. Tradução Rafael Zanolli. São Paulo – SP: Editora Novatec, 2011.
- MADALOSSO (2014), EMANOELI. **Sistema Automatizado para Irrigação de Estufas**. Pato Branco: Universidade Tecnológica Federal do Paraná.
- MCROBERTS, Michael. **Arduino Básico**. São Paulo - SP: Editora Novatec, 2015.
- PRADO, Sérgio. **Introdução ao funcionamento interno do Android**. Disponível em: <<https://sergioprado.org/introducao-ao-funcionamento-interno-do-android/>>. Acesso em abril, 2017.
- Santos (2014), Lorena Beatriz. **Sistema Automatizado para Controle de Umidade e Temperatura em Cultura de Morangos Aplicados aos Pequenos Produtores**. Brasília: Centro Universitário de Brasília.
- SOUZA, David José de. **Desbravando o PIC – Ampliado e atualizado para PIC 16F628A**. São Paulo - SP: Editora Érica, 2009.
- WENDLING, M. (2010) **Sensores**. São Paulo: UNESP.