

CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UniEVANGÉLICA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

SISTEMA DE ALIMENTAÇÃO AUTOMÁTICA PARA ANIMAIS

DIEGO FRANCYSKO ARANTES PACHECO
WELLINGTON ALVINO OLIVEIRA RAMOS

ANÁPOLIS
2018/02

**DIEGO FRANCYSKO ARANTES PACHECO
WELLINGTON ALVINO OLIVEIRA RAMOS**

SISTEMA DE ALIMENTAÇÃO AUTOMÁTICA PARA ANIMAIS

Projeto de pesquisa apresentado ao Curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA, sob a orientação do Prof. Me. William Pereira dos Santos Junior.

**ANÁPOLIS
2018/02**

**DIEGO FRANCYSKO ARANTES PACHECO
WELLINGTON ALVINO OLIVEIRA RAMOS**

SISTEMA DE ALIMENTAÇÃO AUTOMÁTICA PARA ANIMAIS

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Aprovado(a) pela banca examinadora em 12 de Dezembro, de 2018, composta por:



William Pereira dos Santos Júnior
Presidente da Banca



Walquíria Fernandes Marins
Prof(a). Convidado(a)



Luciana Nishi
Prof(a). Convidado(a)

Resumo

Essa pesquisa apresenta o estudo e a elaboração de um sistema *web* para a alimentação automática de animais de estimação ou *pets*. A aplicação tem o objetivo permitir que a reposição da ração ocorra a distância, de forma que ela esteja ligada a um dispositivo que controle o despejo da ração, para assim, diminuir o problema de pessoas que possuem animais de estimação e não conseguem alimentá-los por certo tempo de ausência ou em caso de viagens. Para entender as necessidades e verificar a viabilidade, foi realizada uma pesquisa estatística de mercado para animais de estimação no Brasil, a pesquisa revelou que a população desses animais girava em torno de 30 milhões de cães e 11 milhões de gatos (ASSOFAUNA, 2011 apud SEBRAE, 2013).

Palavras-chave: *pet*; distância; mercado; alimentação.

Abstract

This research presents the study and the elaboration of a web system for automatic feeding of pets or pets. The purpose of the application is to allow feed replacement to occur at a distance, so that it is connected to a device that controls the feed dump, to reduce the problem of people who have pets and can not feed them for a certain time of absence or in case of travel. In order to understand the needs and verify the viability, a statistical survey of the market for pets in Brazil was carried out, the research revealed that the population of these animals revolved around 30 million dogs and 11 million cats (ASSOFAUNA, 2011 apud SEBRAE, 2013).

Keywords: pet; distance; market; feed.

Lista de Abreviaturas e Siglas

3D	3 Dimensões
ANFALPET	Associação Nacional De Fabricantes De Alimentos Para Pequenos Animais
API	<i>Application Programming Interface</i>
ASSOFAUNA	Associação Dos Revendedores De Produtos, Prestadores De Serviço E Defesa Destinados Ao Uso Animal.
CSS	<i>Cascading Style Sheets</i>
EPROM	<i>Erasable and Programmable Read Only Memory</i>
HLT	<i>High-Level Time</i>
HTML	<i>HyperText Markup Language</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	<i>Integrated Development Environment</i>
JNI	<i>Java Native Interface</i>
JVM	<i>Java Virtual Machine</i>
MDF	Material Compensado de Madeira
OTP	<i>One Time Programmable</i>
PIC	<i>Peripheral Interface Controller</i>
UML	<i>Unified Modeling Language</i>
USB	<i>Universal Serial Bus</i>
XML	<i>Extensible Markup Language</i>

Lista de Ilustrações

Figura 1 – Placa de arduino	13
Figura 2 – Microcontrolador.....	15
Figura 3 – Micro servo motor	17
Figura 4 – Movimento servo motor.....	18
Figura 5 – Sensor Ultrassônico.....	19
Figura 6 – Base de Compensado em MDF.....	22
Figura 7 – Esquema para construção do dispositivo	22
Figura 8 – Página Sistema de Alimentação Automática para Animais arduino IDE.....	24
Figura 9 – Importação da Biblioteca RXTXcomm.	25
Figura 10 – Porta para comunicação serial.....	26
Figura 11 – Página sistema localhost.	26
Figura 12 – Página Sistema de Alimentação Automática para Animais.....	27
Figura 13 – Recipiente cheio	28
Figura 14 – Posição do servo motor e arduino	29
Figura 15 – Posição do sensor de distância e recipiente.....	30

Sumário

1. Introdução.....	9
1.1. Objetivo Geral	10
1.2. Objetivos Específicos	10
1.3. Justificativa.....	11
2. Fundamentação teórica.....	12
2.1. Arduino.....	12
2.2. Microcontroladores.....	13
2.3. Sensores	15
2.4. Servo Motor.....	15
2.5. Sensor Ultrassônico	17
2.6. Linguagem Java.....	18
2.7. Linguagem C	18
2.8. Linguagem C++.....	19
3. Desenvolvimento.....	21
3.1. Tecnologias.....	21
3.2. Construção da base MDF	21
3.3. Montagem do aparelho	21
4. Metodologia.....	23
5. Implementação	24
7. Resultados	29
8. Considerações Finais	31
Referências Bibliográficas	32
Apêndice A: Diagrama Caso de Uso	35
Apêndice B: Visão lógica.....	37
Apêndice C: Repositório com códigos de fonte.....	38

1. Introdução

O mercado *pet* no Brasil anda na contramão em relação a diversas áreas da economia brasileira, setores de produtos e serviços para animais, conseguiram passar por momentos difíceis praticamente de forma ilesa. Foi estimado que 25 bilhões de reais tivessem movimentado a economia no ano de 2017, um aumento de 7% em relação ao ano anterior (INFOMONEY, 2017).

A sociedade atual passa por constantes mudanças, dentro da estrutura familiar, aquelas famílias numerosas já não existem mais, o aumento do número de casais sem filhos ou número reduzido, solteiros morando sós, muitas dessas pessoas optam por criar animais de pequeno e médio porte, os chamados pets em seus lares. No Brasil a média de filhos por família caiu em 2003, era de 1,78, já em 2013, era de 1,59 filhos por família isso representa uma queda de 10,7% em 10 anos (MINISTÉRIO DO DESENVOLVIMENTO SOCIAL, 2013).

Dados coletados em 2013 mostram que a cada 100 famílias, 44 criam, por exemplo, cachorros e só 36 têm crianças até doze anos de idade. Segundo o IBGE (Instituto Brasileiro de Geografia e Estatística), as famílias brasileiras cuidam de 52 milhões de cães contra 45 milhões de crianças (IBGE, 2013).

Dentro do cenário acima surgem alguns problemas, o tempo de ausência do dono em sua residência por um longo tempo ou realização de viagens são obstáculos encontrados para cuidar da alimentação desses animais, como permitir que a alimentação dos animais de estimação pudesse ser controlada a distância?

No decorrer do trabalho algumas atividades foram realizadas, a pesquisa bibliográfica para atingir o conhecimento teórico necessário para entender o mercado brasileiro referente aos animais de estimação, funcionamento da plataforma arduino e de alguns de seus componentes, o levantamento de requisitos mostrando as funcionalidades do sistema através do diagrama de caso de uso, Apêndice A.

Durante o desenvolvimento do projeto, foi realizada a construção de um protótipo do dispositivo de alimentação feito de madeira, acoplado a plataforma arduino que foi utilizada para integrar a parte lógica e física do produto. Para a arquitetura do sistema foi escolhido o padrão *MVC (Model View Controller)*, Apêndice B. A programação foi baseada em C++ para a comunicação entre o dispositivo arduino e a programação *web* utilizando Java (código fonte no apêndice D).

1.1. Objetivo Geral

O trabalho propõe o desenvolvimento de um sistema integrado a uma infraestrutura capaz de controlar o despejo de ração animal, a partir de um comando dado em tempo real ou agendamento através da *internet*.

1.2. Objetivos Específicos

- Pesquisar sobre os componentes da plataforma arduíno.
- Desenvolver o sistema utilizando Java para a parte web e a linguagem C para comunicação com arduíno.
- Desenvolver um protótipo físico adequado para integrar a parte física com a parte lógica do dispositivo.
- Usar a plataforma arduíno para obter a comunicação entre o sistema e a estrutura física do produto.
- Apresentar os resultados obtidos.

1.3. Justificativa

A criação do projeto surgiu com a ideia de criar de um produto de baixo custo e eficiente que possibilite a alimentação do *pet* de forma dinâmica, onde o criador economiza esforços e ganha comodidade através da possibilidade de alimentar seu *pet* à distância. Visto que o mercado para esse segmento vem sempre em crescimento.

Nos últimos anos houve um crescimento no mercado mundial para animais domésticos, alguns fatores demonstram esse avanço, o aumento do mercado de rações, produtos e serviços para animais de médio e pequeno porte. Dentro do cenário brasileiro, as indústrias de produtos para a saúde animal têm investido cada vez mais em pesquisas e inovações, demonstrando um grande potencial de crescimento. Segundo dados do Euromonitor Internacional, no período entre 2005 e 2010, esse setor teve um crescimento anual de 11,9%. Na América Latina houve um crescimento de 7,6% em 2005 para 10,2% em 2010 (ANFALPET, 2011 apud SEBRAE, 2013).

Segundo a Associação Nacional dos Fabricantes de Produtos para Animais de Estimação (ANFALPET, 2011 apud SEBRAE, 2013), em 2009, o Brasil possuía pelo menos 100 mil lojas de produtos direcionados aos animais de estimação, desse total, 40 mil eram de pet shops, que são as lojas especializadas em oferecer produtos e serviços para animais de pequeno e médio porte.

De acordo com a Associação dos Revendedores de Produtos, Prestadores de Serviço e Defesa Destinados ao Uso Animal (ASSOFAUNA, 2011 apud SEBRAE, 2013), calculava-se que a população desses animais girava em torno de 30 milhões de cães, 11 milhões de gatos, 4 milhões de pássaros e 500 mil aquários em todo o país no ano de 2011.

O Brasil em 2011 ocupava o 2º lugar em número de cães no mundo, só perdendo para os Estados Unidos. Essa pesquisa mostrou, por exemplo, 52% dos domicílios da classe A possuem animais de estimação. Esse percentual cai para 47% na classe B e 36% na classe C. Isso significa que mais da metade da população que possui condições financeiras favoráveis, criam um animal de estimação e consideram o animal como um integrante da família.

2. Fundamentação teórica

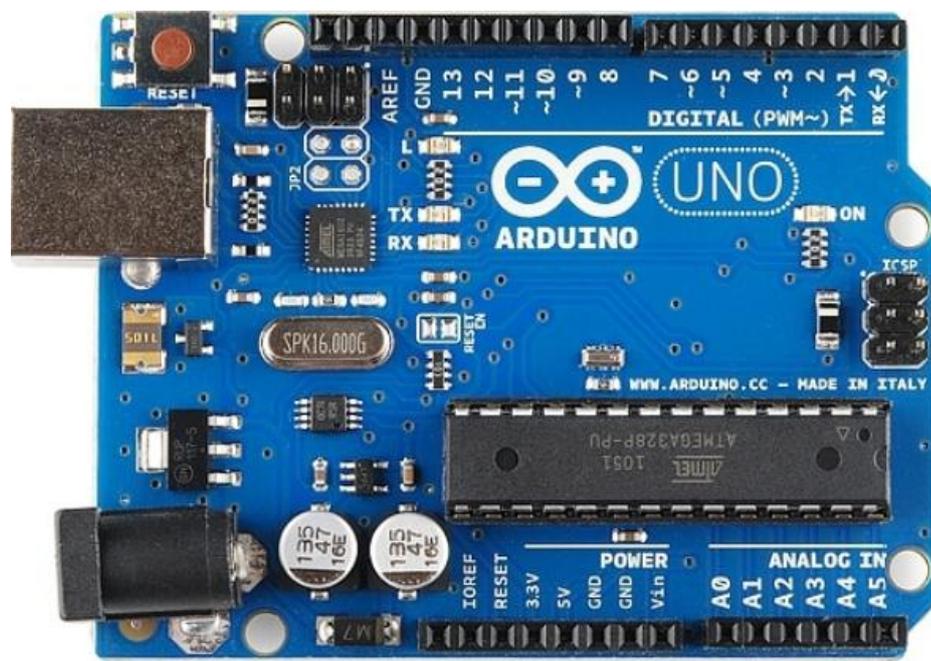
Nesta seção serão mostradas informações referentes aos componentes eletrônicos como os microcontroladores, arduino, sensores, servo motor e sensor ultrasônico. Também serão apresentados os conceitos básicos das linguagens de programação Java e C/C++.

2.1. Arduino

A plataforma de desenvolvimento arduino como mostra a Figura 1, possui diversas características que a tornaram uma ótima solução (ROBOCORE, 2005): Algumas delas são:

- Regulador de tensão de entrada, podendo utilizar fontes de alimentação que possuam tensão entre 6 a 20 volts;
- Saída de alimentação de 3,3 e 5 volts para ligar outros componentes ao *hardware* sem ter necessidade de utilizar outra fonte de alimentação;
- Conectores em todas as portas de entrada e saída do microcontrolador facilitando a integração com outros componentes, como por exemplo, sensores de movimento;
- Ambiente de desenvolvimento de algoritmos próprio e upload através da porta *USB* (*Universal Serial Bus*) de qualquer computador.

Figura 1 – Placa Arduino.



Fonte: LABORATÓRIO DE GARAGEM (2012).

Segundo informações do site, o arduino (ARDUINO UNO, 2012) “é uma plataforma *open-source* de prototipagem eletrônica baseada na flexibilidade, *hardware software* fácil de usar”. Foi desenvolvida para ser totalmente autônoma e necessita apenas de uma fonte de alimentação o funcionamento.

2.2. Microcontroladores

Os microcontroladores são pequenos e apresenta a melhor relação custo/benefício, estão presentes em quase tudo que envolve a eletrônica. São dispositivos que possuem internamente todos os componentes necessários para seu funcionamento autônomo (SILVA, 2009 apud MARCHESAN, 2012, p.17). Dito de outra forma,

Os microcontroladores são computadores de propósito específicos. Eles possuem tamanho reduzido, baixo custo e baixo consumo de energia. Devido a esses fatores há diversos segmentos, que os utilizam, tais como a indústria automobilística, de telecomunicações, de brinquedos, de eletrodomésticos, de eletroeletrônicos, bélica [...]. (SILVA, 2009, p.17 apud MARCHESAN 2012, p.17)

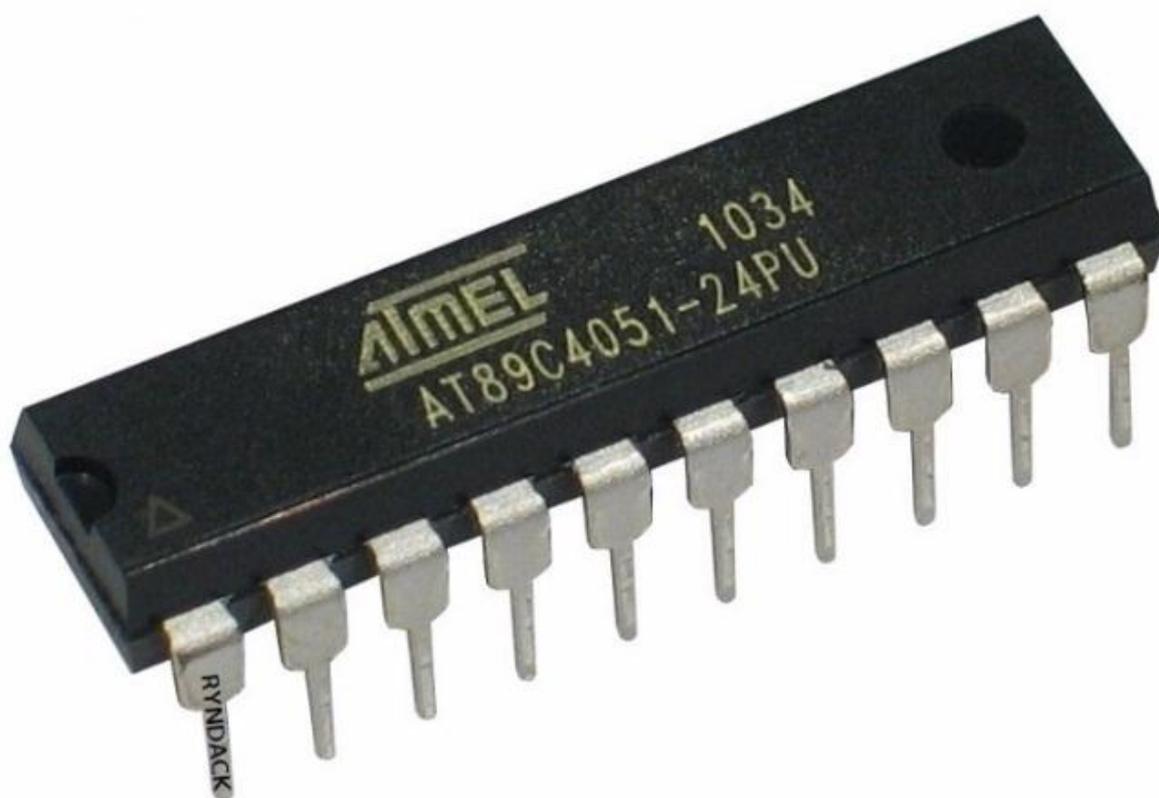
Para o funcionamento autônomo os microcontroladores incorporam vários recursos em uma única pastilha, como: memória, portas de entrada e saída, processador e diversas outras peças, todas montadas em um pequeno circuito integrado. Veja a descrição completa dos componentes dos microcontroladores feita por Martins (2005, p. 16 apud MARCHESAN, 2012, p.17):

Tipicamente, um microcontrolador caracteriza-se por incorporar no mesmo encapsulamento um microprocessador (com a finalidade de interpretar as instruções de programa e processar dados), memória de programa (com a finalidade de armazenar de maneira permanente as instruções do programa), memória de dados (com a finalidade de memorizar os valores associados com as variáveis definidas no programa), uma série de pinos de entrada/saída (com a finalidade de realizar a comunicação do microcontrolador com o meio externo) e vários periféricos (tais como temporizadores, controladores de interrupção, temporizadores cão de guarda (*WatchDog Timers* – WDTs), comunicação serial, geradores de modulação por largura de pulso ou de PWM (*Pulse Width Modulation*), conversores analógico/digital etc.), fazendo com que o hardware final fique extremamente complexo.

Martins (2005, p. 15 apud MARCHESAN, 2012, p.18) afirma ainda que existem vários tipos de microcontroladores. Segundo ele, o que os diferencia são: a velocidade do processamento; a quantidade de memória interna disponível para armazenar dados (memória de dados) e para armazenar as instruções de programas (memória de programa); “a quantidade de pinos de [...] [entrada e saída], a forma de alimentação, os tipos e as quantidades de periféricos, a arquitetura e o conjunto de instruções disponibilizado nos circuitos internos”. Um exemplo de microcontrolador, talvez o mais popular, é o da série

PIC (Peripheral Interface Controller), fabricado pela empresa *Microchip Technology Inc.*, que se popularizou pelo bom plano de marketing “baseado na disseminação de uma ferramenta de auxílio à construção de programas – o *MPLab IDE (Integrated Development Environment)*. [...] [e, por possuírem] uma linguagem *Assembly* menos complexa em relação às disponibilizadas por outros fabricantes” (MARTINS, 2005, p. 15 apud MARCHESAN, 2012, p.18). Os microcontroladores da série *PIC* têm algumas opções de memória de programas, como o *OTP (One Time Programmable)* e o *EPROM (Erasable and Programmable Read Only Memory)*. Ademais, têm “opções de baixa tensão e inúmeros tipos de circuito osciladores, assim como várias opções de encapsulamento” (MARTINS, 2005, p. 17 apud MARCHESAN, 2012, p.18). Outro microcontrolador como mostra figura 1, fortemente citado na literatura e que tem ampla aceitação no mercado é o da família *AVR* (acrônimo sem significado), fabricado pela empresa de nome *Atmel*.

Figura 2 – Microcontrolador.



Fonte: LABORATÓRIO DE GARAGEM (2012).

Este microcontrolador apresenta “bom desempenho frente ao número de instruções executadas por ciclo de *clock*.” (LIMA, SCHWARZ, 2009, p. 94 apud MARCHESAN, 2012, p.18). Além disso, Lima e Schwarz (2009, p. 94 apud MARCHESAN, 2012, p.18)

afirmam que a Atmel fornece um programa gratuito, chamado AVR Studio, para a programação desses microcontroladores, que podem ser programadas tanto em linguagem *Assembly*, quanto em linguagem C (empregando-se o compilador C gratuito). A plataforma arduino (Seção 3.1), caracteriza-se por utilizar um microcontrolador da família AVR. “O que antes necessitava de conhecimentos técnicos específicos de eletrônica e programação, agora se tornou extremamente simples e até intuitivo.” (CARVALHO, 2011, p. 34 apud MARCHESAN, 2012, p.18). Além disso, essa plataforma facilita o uso de microcontroladores. Com ela, podem-se monitorar sensores, pode-se comunicar com computadores e celulares e, inclusive, pode-se controlar algumas funções como ligar e desligar lâmpadas (através do controle de relés que funcionam como interruptores), abertura de fechaduras elétricas, leitura de sensores, etc.

Adicionado a tudo isso, tem-se que,

O arduino oferece uma interface de *hardware* proporcionando todo o circuito necessário para funcionamento do microcontrolador e uma interface e ambiente de desenvolvimento em software para programação. Por ser uma plataforma de código aberto (*open-source*) há uma grande comunidade de desenvolvedores do mundo inteiro que publicam bibliotecas já com toda a programação pronta para se usar, com funções específicas, como, por exemplo, o controle de servo motores ou leitura de sensores analógicos. (CARVALHO, 2011, p. 34 apud MARCHESAN, 2012, p.19).

2.3. Sensores

Sensores são dispositivos que mudam de estado conforme a interação com o ambiente. Seu hardware pode ser composto por diversos componentes eletrônicos ou por apenas um componente. Estes dispositivos podem interagir com diversos tipos de grandezas físicas, tais como temperatura, movimento, pressão, entre outras, convertendo essas grandezas em sinais elétricos analógicos ou digitais. Os que possuem saídas analógicas tem seu valor oscilando conforme a variação da grandeza de entrada. Já os com saída digital, geralmente digital binário, assumem apenas dois valores bem definidos (MARCHESAN, 2012).

2.4. Servo Motor

O servo motor (Figura 3), é um sistema com propriedades de controle de posição, velocidade e força por meio de tensão, corrente ou frequência, e variam conforme sua estrutura. Funcionam também com estator e rotor (FERRAZ, 2008 apud KRETZSCHMAR, 2012, p.22).

Figura 3 – Micro servo motor.



Fonte: (USINAINFO, 2016).

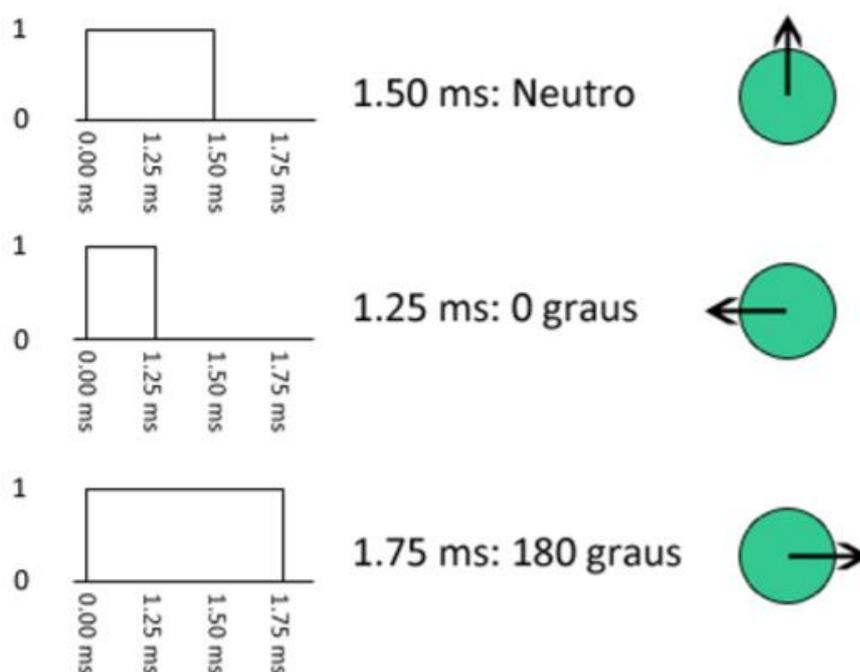
O eixo do servo (rotor), pode ser posicionado para posições angulares específicas, enviando ao servo um sinal elétrico. Enquanto o sinal se mantiver na saída, o eixo se manterá na posição. À medida que forem feitas as alterações do sinal, a posição angular do eixo é alterada (SEATTLE ROBOTICS SOCIETY, 2012 apud KRETZSCHMAR, 2012, p.22).

Os servos normalmente podem se movimentar entre 0° e 90° graus ou 0° e 180° graus. Para a alimentação desses tipos de atuadores mecânicos, é utilizada uma tensão de 4,5 a 6,5 volts, e sua corrente varia de acordo com cada modelo (ESTG, 2011 apud KRETZSCHMAR, 2012, p.22).

Para comunicar o ângulo, os servos da Figura 4 possuem um fio de controle, o ângulo é determinado pela duração de um pulso enviado. Isso é chamado de modulação de pulso codificado, o servo espera um pulso a cada 20 milissegundos, o comprimento do pulso irá determinar até que ponto o motor gira. Um pulso de 1,5 milissegundos, por exemplo, vai fazer o motor girar para a posição de 90° graus (também designada como posição neutra). Se o pulso é menor do que 1,5 milissegundos, então o motor vai levar o

eixo para cerca de 0° grau. Se o pulso é maior do que 1,5 milissegundos, o eixo gira mais 180° graus (SEATTLE ROBOTICS SOCIETY, 2012 apud KRETZSCHMAR, 2012, p.22).

Figura 4 – Movimento servo motor.



Fonte: (TRITON ROBOS, 2018).

O servo motor possui muitas aplicações, que vão desde o setor de robótica de pequeno porte até grandes máquinas industriais. Cada servo possui aplicabilidade e particularidade específica que devem ser analisadas na hora da escolha para um projeto (CORRÊA, 2009 apud KRETZSCHMAR, 2012, p.23).

Para pequenos projetos, o uso de servos motores pode tornar-se indispensável, pois em relação a outros motores, os servos são de fácil controle com o arduino, além de uma boa precisão oferecida.

2.5. Sensor Ultrassônico

O sensor ultrassônico como mostra na Figura 5, possibilita medir a distância de um objeto que se encontra a sua frente, onde um pulso ultrassônico é emitido pelo, se reflete sobre o objeto e retorna para o sonar e com base no tempo decorrido entre a emissão e recepção do pulso, é possível determinar distância do objeto.

Figura 5 – Sensor Ultrassônico – Modelo HC-SR04.



Fonte: LABORATÓRIO DE GARAGEM (2012).

O pulso é emitido de forma automática pelo sonar a uma frequência de 40 Khz e quando é recebido de volta, ele calcula *HLT* (*High-Level Time*), vezes a velocidade do som, onde o resultado da multiplicação é dividido por dois, onde se retorna o resultado da distância em que o objeto se encontra (FERRAZ, 2008 apud SMIDT, 2013, p.22).

2.6. Linguagem Java

Java é uma linguagem de programação orientada a objetos, multiplataforma, e funciona em qualquer sistema operacional, aonde se tenha um interpretador instalado. Esse Interpretador denominado Máquina Virtual Java (JVM), é um programa que converte o código Java em comandos que o sistema operacional possa executar (PERKOVIC, 2016).

O Java utiliza um conceito diferente, ao invés de gerar um código binário diferente para cada plataforma, é gerado um binário que pode ser executado em qualquer plataforma, dentro de uma máquina virtual. Este código binário "universal" é chamado de *bytecode*. Como Java é uma linguagem que não se prende a uma plataforma, ou a um sistema operacional, Java é utilizado em vários ambientes de desenvolvimento. Qualquer dispositivo que tenha uma JVM instalada pode rodar programas escritos na linguagem Java.

2.7. Linguagem C

A linguagem C foi criada por Dennis Ritchie, em 1972, no centro de Pesquisas da Bell *Laboratories*. Sua primeira utilização importante foi a reescrita do Sistema Operacional Unix, que até então era escrito em *assembly*. Em meados de 1970 o Unix saiu do laboratório para ser liberado para as universidades. Foi o suficiente para que o sucesso da linguagem atingisse proporções tais que, por volta de 1980, já existiam várias versões

de compiladores C oferecidas por várias empresas, não sendo mais restritas apenas ao ambiente Unix, porém compatíveis com vários outros sistemas operacionais (KERNIGHAN, 1986 apud UNICAMP, 2006, P.3).

O C é uma linguagem de propósito geral, sendo adequada à programação estruturada. A linguagem C pertence a uma família de linguagens cujas características são: portabilidade, modularidade, compilação separada, recursos de baixo nível, geração de código eficiente, confiabilidade, regularidade, simplicidade e facilidade de uso.

2.8. Linguagem C++

O C++ é uma linguagem de programação de nível médio, baseada na linguagem C. O desenvolvimento da linguagem começou na década de 80, por Bjarne Stroustrup. O objetivo era melhorar uma versão do núcleo Unix. Para desenvolver a linguagem, foram acrescentados elementos de outras linguagens de vários níveis, na tentativa de criar uma linguagem com elementos novos, sem trazer problemas para a programação. No início do desenvolvimento, a linguagem usava um pré-processador, mais Stroustrup criou um compilador próprio, com novas características.

A primeira versão oficial do C++ foi lançada em 1985, juntamente com o livro *“The C++ Programming Language”*, que esclareceu vários pontos sobre o C++. Em 1989 foi lançada uma segunda versão da linguagem, com acréscimo das características: Herança múltipla, classes abstratas, métodos estáticos, métodos constantes e membros protegidos, incrementando também um suporte de orientação a objeto (PACIEVITCH, 2018).

O C++ possui uma enorme variedade de códigos, pois além de seus códigos, pode contar com vários da linguagem C. Esta variedade possibilita a programação em alto e baixo nível.

Alguns fatos sobre o C++:

- O C++ é uma linguagem criada para ser tão eficiente quanto o C, porém com novas funções;
- É uma linguagem que suporta múltiplos paradigmas;
- A linguagem dá liberdade para o programador escolher as opções, mesmo sendo a opção errada;
- Muitos códigos podem ser transferidos para C facilmente, pois o C++ foi criado para ter compatibilidade com o C;

- A linguagem não tem privilégios para alguns grupos de programadores, os comandos são feitos para todas as especialidades de programadores;

Alguns programas conhecidos feitos em C++, ou partes dos seus códigos são:
Adobe Photoshop, MySQL, Mozilla Firefox, Internet Explorer, Microsoft Windows, etc.

3. Desenvolvimento

Nas seguintes seções são descritas as etapas aplicadas no desenvolvimento e implementação do sistema, utilização componentes para a montagem do aparelho e construção da base de alimentação em MDF (Material Compensado de Madeira).

3.1. Tecnologias

Para o desenvolvimento do sistema proposto foram usadas as seguintes tecnologias:

- **Arduino** – Placa de arduino.
- **Astah Community** – Software para modelagem *UML (Unified Modeling Language)*.
- **C/C++** – Para interface *IDE* Arduino.
- **Eclipse** – Ambiente de desenvolvimento de software em linguagem Java.
- **JAVA** – Linguagem de programação para manipular e direcionar dados do sistema e parte web.
- **UML** – Linguagem que representa o sistema de forma simples a fim de facilitar a compreensão da sua pré-implementação.

3.2. Construção da base MDF

Para a montagem da base do protótipo o material utilizado foi o MDF (Figura 6), onde foi feito por encomenda em uma loja de materiais artísticos, e modelado de acordo com as dimensões do dispositivo arduino. Alguns detalhes de suas características a seguir:

- Protótipo é apenas de encaixe, dispensando uso de cola e parafusos, facilitando assim sua montagem;
- Peso de 300 gramas;
- Recipiente na parte inferior;
- 32 cm de altura e 18 cm de comprimento.
- A distância da tampa de alimentação até o recipiente é de 6 cm.

3.3. Montagem do aparelho

Para o processo de montagem foram necessários os seguintes aparatos:

- Arduino Uno (Figura 7);
- Sensor Ultrassônico (Figura 7);
- Micro servo motor (Figura 7);

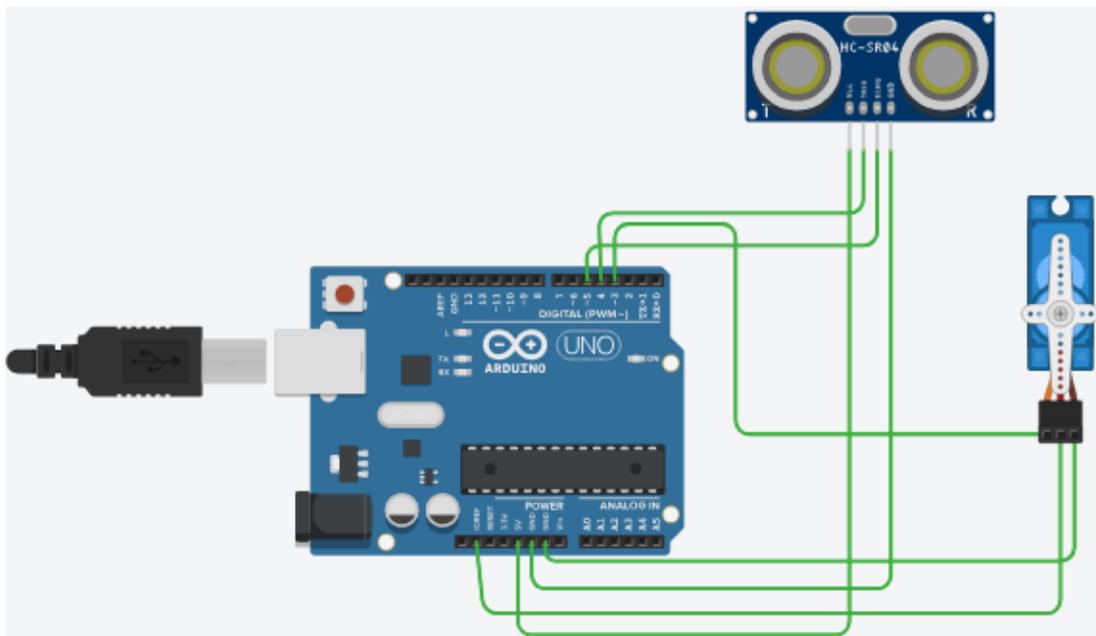
- Base de compensado em MDF, mostrado na Figura 6.
- Recipiente (Figura 6);

Figura 6- Base de Compensado em MDF.



Fonte: Autores da pesquisa.

Figura 7 - Esquema para construção do dispositivo.



Fonte: Autores da pesquisa.

4. Metodologia

Foram realizadas pesquisas bibliográficas mostrando o crescimento do mercado brasileiro em relação a animais domésticos.

Falar sobre o futuro conduz um pensamento com várias incertezas, sendo assim à compreensão de todo ambiente que envolve a organização é fundamental o uso de ferramentas como pesquisas e análises que abordem o mundo real. Estar preparado e ser flexível possibilitam um cenário mais preciso sobre diversas hipóteses de acontecimentos futuros. Segundo Aaker (2004, p. 94).

O trabalho ainda conta com uma pesquisa, que contém os conceitos necessários para o desenvolvimento do produto referente à microcontroladores, arduino, sensores e servo motor.

Para atender as necessidades do produto foi realizada a fase de levantamento de requisitos, onde as informações foram coletadas para obter as funcionalidades do produto. Segundo Sommerville (2011), o levantamento de requisitos, compreende na identificação de características e funcionalidades fundamentais para atender as expectativas dos usuários de um determinado sistema. Para a modelagem do projeto foi feito o Diagrama de Caso de Uso e uma breve descrição de suas funcionalidades.

Após serem obtidas as funcionalidades, o projeto contou com a construção do mecanismo de alimentação animal, que foi integrado a plataforma arduino por comunicação serial, para que pudesse ocorrer a parte de despejo da ração.

A implementação foi realizada com a utilização das tecnologias, linguagens C para o arduino, Java para a programação web e comunicação serial com o dispositivo arduino que foi acoplado ao mecanismo de alimentação.

5. Implementação

Para a codificação do sistema *web* foi utilizada a linguagem JAVA, onde foi escolhido o ambiente de desenvolvimento Eclipse, e usado por possuir boa integração com bibliotecas externas, no qual foi necessário para alcançar o resultado proposto. Em conjunto com a interface *IDE (Integrated Development Environment)* do arduino (Figura 8), foi aplicado uso da biblioteca RXTXcomm no Eclipse para tarefas relacionadas a comunicação serial do Arduino com a linguagem a JAVA.

O RXTXcomm é uma biblioteca Java utilizada para criar aplicações que necessitam de comunicação com as portas serial/paralela do computador A RXTXcomm utiliza libs que são bibliotecas nativas do sistema através da *API (Application Programming Interface) JNI (Java Native Interface)* para interagir com as portas serial/paralela (DEV MEDIA, 2015).

Figura 8 - Página Sistema de Alimentação Automática para Animais arduino IDE.



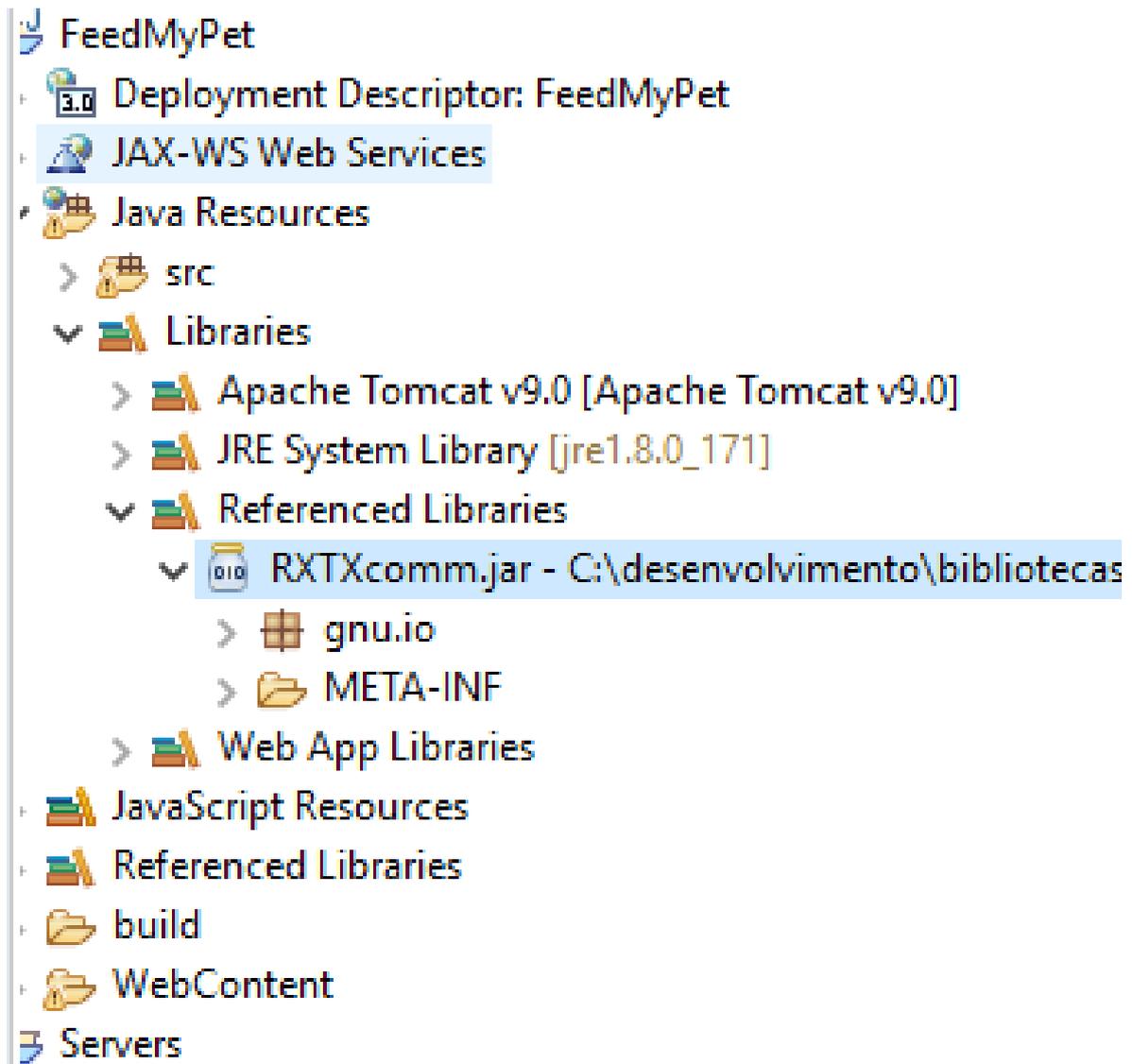
```
FeedMyPet | Arduino 1.8.7
Arquivo Editar Sketch Ferramentas Ajuda
FeedMyPet
void loop()
{
  if (Serial.available() > 0) { //verifica se existem dados na serial
    op = Serial.read(); //lê dados na serial
    switch (op) {
      case 1:
        while (true) {
          float cmMsec;
          long microsec = ultrasonic.timing();
          //Le as informacoes do sensor, em centimetros
          cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
          //Serial.print("Distancia em cm: "); //escreve valores obtidos na serial
          //Serial.println(cmMsec);
          delay(300);

          if (cmMsec > 9) {
            for (i = 0; i < 40; i++) {
              //abertura e fechamento do motor em 40 graus e 0 graus
              motor.write(i);
              delay(30);
            }
            motor.write(0);
          }
        }
      }
    }
  }
```

Fonte: Autores da pesquisa.

A primeira etapa executada foi à importação da biblioteca necessária para comunicação serial com o arduino (código fonte no Apêndice C) e também a biblioteca para o servidor local Apache Tomcat v 9.0, como mostrado na Figura 9.

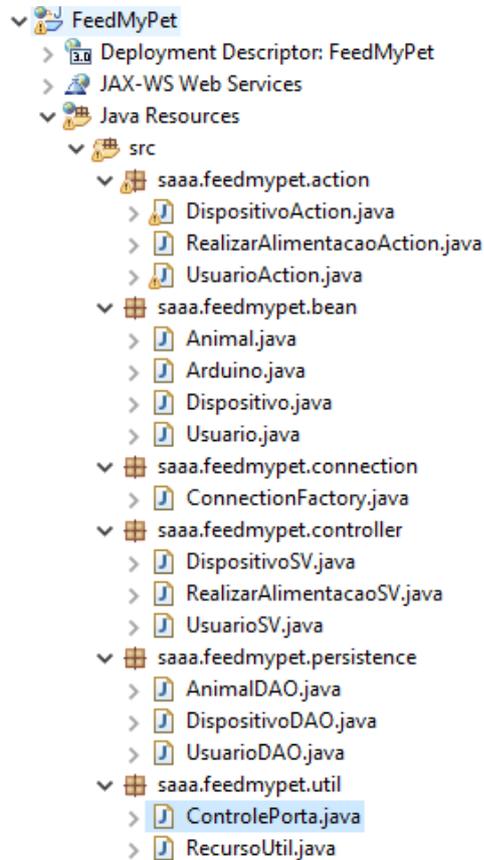
Figura 9 - Importação da Biblioteca RXTXcomm.



Fonte: Autores da pesquisa.

Em seguida foi necessário fazer configurações que indiquem ao compilador Eclipse, o caminho da porta para comunicação serial que devem ser usados pelo Sistema de Alimentação Automática para animais, onde também partirá o comando para a alimentação via serial, através da página *web localhost*, (Figura 10).

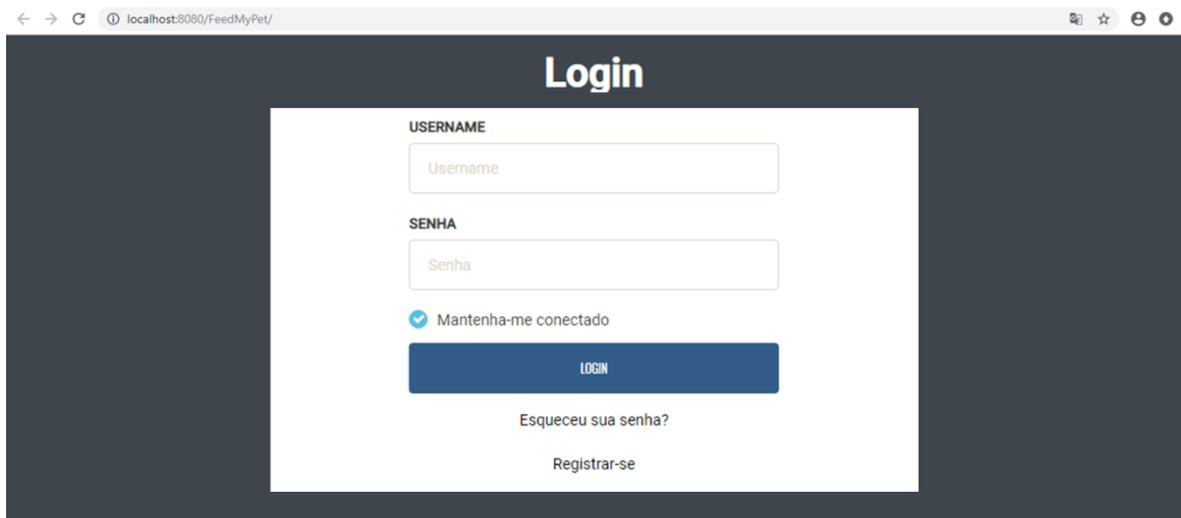
Figura 10 - Porta para comunicação serial.



Fonte: Autores da pesquisa.

Após o procedimento feito na Figura 10, é possível carregar a página do sistema *web* através do Eclipse, (Figura 11).

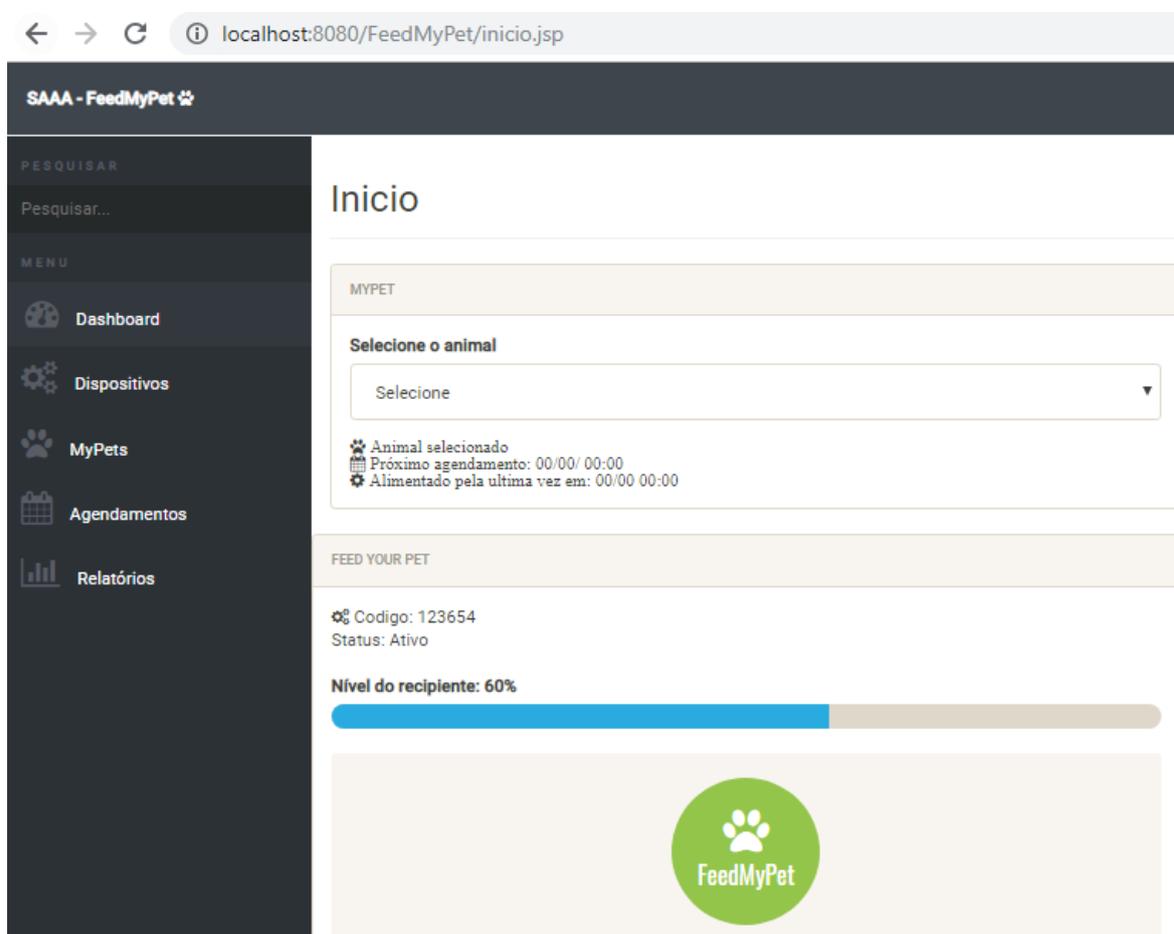
Figura 11 - Página sistema localhost.



Fonte: Autores da pesquisa.

A autenticação de usuário não foi desenvolvida, assim como outras funcionalidades do sistema que serão feitas no futuro (Manter Agendamento de Alimentação, Manter Usuário, Controlar Nível de Ração e Gerar Relatório) o foco ficou por conta de programar a funcionalidade “Alimentar Animal”, pois, por conta da complexidade do sistema em geral, ficaria inviável por questões de tempo, o desenvolvimento de demais funcionalidades desta pesquisa. Portanto, na tela de autenticação, a entrada no sistema é feita de forma direta, clicando no botão de *Login*, será feito o redirecionamento para a tela principal do sistema mostrada da Figura 12.

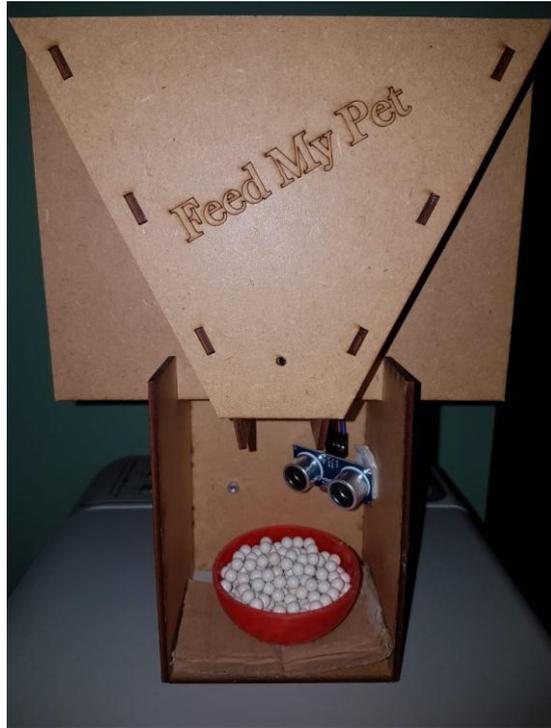
Figura 12 - Página Sistema de Alimentação Automática para Animais.



Fonte: Autores da pesquisa.

Na Figura 12, ao clicar no botão “*FeedMyPet*”, será realizada a comunicação serial com o Arduino, onde é enviada a instrução para realizar a alimentação, a ração que está colocada na parte superior do aparelho será liberada, e começará a encher o recipiente que está embaixo até que atinja o *status* cem por cento (100%) cheio, mostrado na (Figura 13).

Figura 13 – Recipiente cheio.



Fonte: Autores da pesquisa.

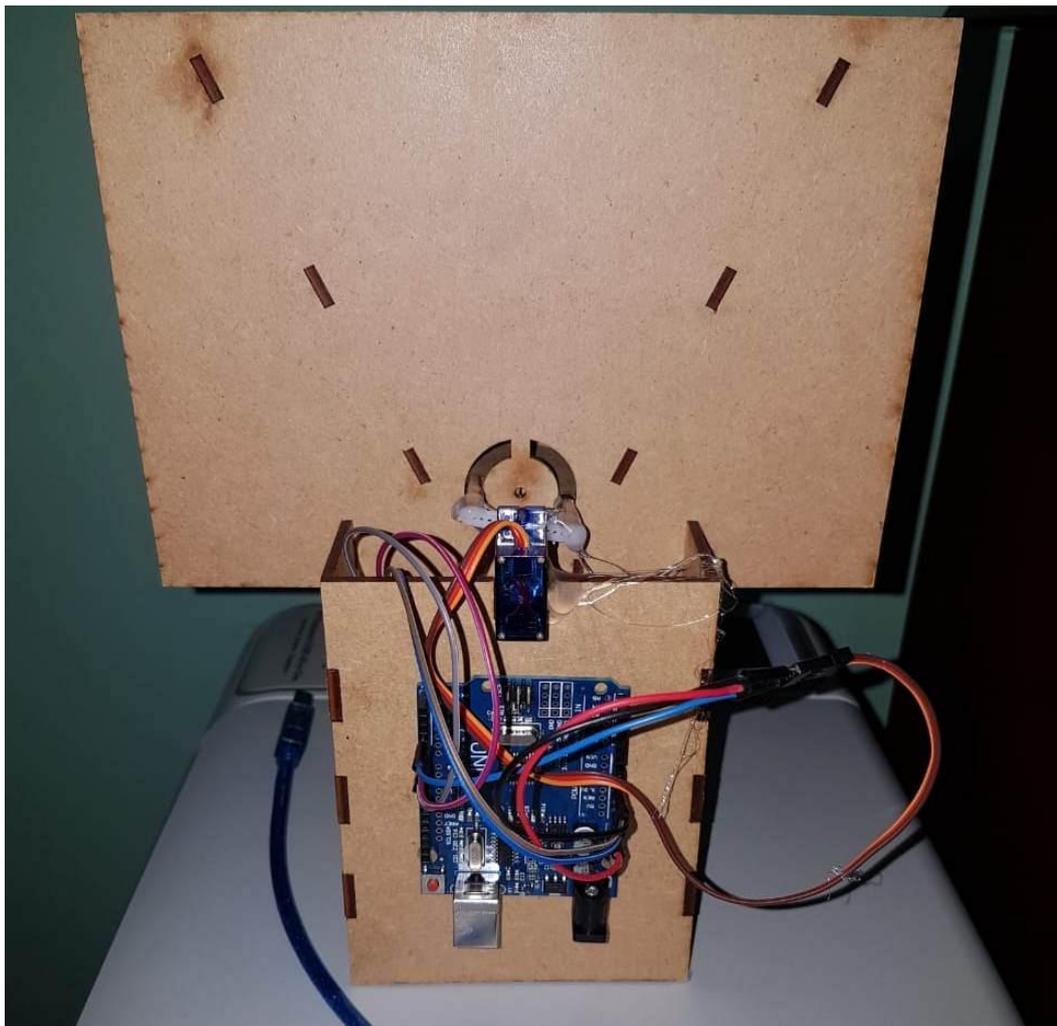
7. Resultados

Nos tópicos a seguir são mostrados os resultados obtidos no final do projeto, referentes ao funcionamento do mecanismo, que é composto pelo arduíno e seus componentes (servo motor, microcontroladores e sensores) mais a base de MDF (Figura 7), descrição de como operou o sistema web e a comunicação serial entre dispositivo e computador.

O funcionamento ocorreu da seguinte forma:

- Para utilização do servo motor de forma correta, seu posicionamento foi feito na parte superior traseira do MDF (Figura 14);
- O arduino fixado na parte inferior traseira (Figura 14);
- A placa arduino aqueceu bastante, porém não afetou o seu funcionamento;

Figura 14 – Posição do servo motor e arduíno.



Fonte: Autores da pesquisa.

- O sensor de distância (Figura 15) ficou posicionado no interior da base de MDF e acima do recipiente;
- O recipiente de ração foi posicionado na base de MDF (Figura 15).

Figura 15 – Posição do sensor de distância e recipiente.



Fonte: Autores da pesquisa.

A *interface web* utilizou *HTML (HyperText Markup Language)* e *Java* e a estilização das páginas feitas em *CSS (Cascading Style Sheets)*. O funcionamento *web* ocorreu com o botão *Feed* sendo acionado, assim foi mandada uma instrução para o despejo de ração no recipiente, para que o funcionamento ocorresse foi necessária à comunicação serial entre arduíno e sistema *web* (código fonte no Apêndice C).

A plataforma arduíno mostrou-se eficiente e supriu as necessidades propostas, apresentando uma boa velocidade de resposta. A comunicação entre o computador e o dispositivo, é feita via *USB* através de um cabo serial com a utilização da biblioteca “*RXTXcomm*” da linguagem *Java* que possibilitou a realização da comunicação através da porta serial (Figura 8).

8. Considerações Finais

Durante o desenvolvimento do projeto foi construído um dispositivo para melhorar e automatizar assim a alimentação animal. Além disso, possibilitou conhecer o mercado brasileiro referente à área de animais de estimação e de ferramentas disponíveis que fizeram parte do desenvolvimento do projeto (servo motor, microcontroladores e sensores). Foram utilizados conhecimentos adquiridos durante o curso de Engenharia de Computação, que vão desde a parte de modelagem ao desenvolvimento do sistema.

A função de alimentação do sistema foi desenvolvida, e a comunicação entre sistema *web* e dispositivo físico ocorreu de forma esperada. Entretanto, com o encerramento do projeto, foi possível perceber que o sistema necessita de melhorias e outras funcionalidades pendentes: 1- Manter o Agendamento de Alimentação, uma vez que a alimentação foi feita de forma direta em tempo real; 2- Manter Usuário, não foi desenvolvido, assim a entrada no sistema ocorreu de forma direta; 3- Controlar o Nível da Ração, desse modo só foi possível manter o recipiente de ração cheio; 4- Gerar Relatório, onde será mostrado ao usuário informações referente aos agendamentos e a quantidade de ração que foi gasta.

Para continuação do projeto no futuro, será necessário conseguir realizar a comunicação entre sistema *web* e dispositivo de forma independente, sem a necessidade de ligação direta através de cabo de rede. A base de MDF não foi feita de forma que acoplasse o arduino em sua base, desse modo o dispositivo ficou solto, uma base feita em uma impressora 3D (3 Dimensões) resolveria este problema. Ao final do trabalho foi obtido um *software web* e um protótipo funcional que poderá ser aproveitado em projetos futuros.

Referências Bibliográficas

AAKER, D. A; DAY G. S; KUMAR, V. **Pesquisa de marketing**. 2. Ed. São Paulo: Atlas, 2004.

ARDUINO UNO. **Arduino**. [S.l.]. Disponível em: <<http://arduino.cc/en/Main/ArduinoBoardUno>>. Acesso em: 28 abr. 2018.

ASSOCIAÇÃO DOS REVENDEDORES DE PRODUTOS, PRESTADORES DE SERVIÇO E DEFESA DESTINADOS AO USO ANIMAL (ASSOFAUNA). **Apresenta informações especializadas em 165 animais domésticos**, 2011. Disponível em: <<http://www.petbr.com.br>> Acessos em: 21 abr. 2018;

ASSOCIAÇÃO NACIONAL DE FABRICANTES DE ALIMENTOS PARA PEQUENOS ANIMAIS (ANFAL). PET. **Apresenta informações sobre o mercado de rações para animais**, 2011. Disponível em: <<http://www.anfalpet.org.br>>. Acessos em: 17 mai. 2018.

CARVALHO, M. F. P. R. **Automação e controle residencial via internet utilizando arduino**. In: SEMANA DE EXTENSÃO, 1. Rio de Janeiro. Anais... Rio de Janeiro: [online], 2011. Disponível em: <http://portal.cefetrj.br/files/extensao/outros-livro_sem_ext_2011.pdf#page=34>. Acesso em: 20 mai. 2018.

CORRÊA, M. A. **Sensores**. MECATRONICA ATUAL. 2012. Disponível em: <<http://mecatronicaatual.com.br/-secoes/leitura/567>>. Acesso em: 19 set. 2018

CORRÊA, M. A. **Servo acionamento e servomotores**. MECATRONICA ATUAL. 2012. Disponível em: <<http://mecatronicaatual.com.br/secoes/leitura/494>>. Acesso em: 15 mai. 2018.

ARIAS, J. **Lares brasileiros já têm mais animais que crianças**. EL PAÍS. 2015. Disponível em: <https://brasil.elpais.com/brasil/2015/06/09/opinion/1433885904_043289.html>. Acesso em: 16 dez. 2018.

DEVMEDIA. **Raspberry Pi e a biblioteca Java RxTx**. DEVMEDIA. 2015. Disponível em: <<https://www.devmedia.com.br/raspberry-pi-e-a-biblioteca-java-rxtx/32722>>. Acesso em: 17 dez. 2018.

ESTG. Escola Superior de Tecnologia e Gestão de Portugal. **Servo-motores**. 2011.

FERRAZ, M. **Servo Acionamento**. São Paulo – SP, SENAI, 2008.

BOM DIA BRASIL. **Levantamento aponta que brasileiras estão tendo cada vez menos filhos**. G1. 2015. Disponível em: < <http://g1.globo.com/bom-dia-brasil/noticia/2015/03/levantamento-aponta-que-brasileiras-estao-tendo-cada-vez-menos-filhos.html>>. Acesso em: 16 dez. 2018.

REVISTA LIDE. **Sem crise: mercado de pets no Brasil é o terceiro do mundo em faturamento**. INFOMONEY. 2018. Disponível em: <<https://www.infomoney.com.br/negocios/canal-do-empresario/noticia/7375940/sem-crise-mercado-pets-brasil-terceiro-mundo-faturamento/>>. Acesso em: 24 out. 2018.

KERNIGHAN, B W. ; RITCH, D M. C: **A Linguagem de Programação**. 1. Ed Rio de Janeiro, Campus, 1986.

KRETZSCHAMAR, C. **Robô quadrúpede utilizando arduíno**. Itajaí, p. 22-23, 2012.

LABORATÓRIO DE GARAGEM. **Tutorial Ultrarrápido de Arduíno**, 2012. Disponível em: < <http://labdegaragem.com/profiles/blogs/tutorial-ultra-rapido-de-arduino>>. Acesso em: 20 mai. 2018.

LIMA, C. B; SCHWARZ, L. **Kit didático para trabalho com os microcontroladores AVR – Kit ATMEGA++**. Revista ilha digital, [online], v. 1, p. 93- 99, 2009. Disponível em: <<http://ilhadigital.florianopolis.ifsc.edu.br/index.php/ilhadigital-/article/view/7/7>>. Acesso em: 20 mai. 2018.

MARCHESAN, M. **Sistema de monitoramento residencial utilizando a plataforma arduíno**. Santa Maria, p. 17-19, 2012.

MARTINS, N. A. **Sistemas Microcontrolados: Uma Abordagem com o Microcontrolador PIC 16F84**. São Paulo: LIVRARIA CULTURA. 2005. Disponível em: <<http://www.livrariacultura.com.br-/imagem/capitulo/3173664.pdf>>. Acesso em: 20 mai. 2018.

PACIEVITCH, Y. **C++**. INFOESCOLA. 2018. Disponível em: <<https://www.infoescola.com/informatica/cpp/>>. Acesso em: 16 dez. 2018.

PERKOVIC, L. **Introdução à computação usando Java**: um foco no desenvolvimento de aplicações. 1. Ed. Rio de Janeiro: LTC, 2016.

ROBOCORE. **Kit Iniciante para Arduino**. ROBOCORE. 2005. Disponível em: <<https://edisciplinas.usp.br/mod/resource/view.php?id=1539137>>. Acesso em: 16 dez. 2018.

SEATTLE ROBOTICS SOCIETY. **What's a servo**. 2012. Disponível em: <<http://www.seattlerobotics.org/guide/servos.html>>. Acesso em: 15 mai. 2018.

SEBRAE. **Estudo de tendências de mercado**. Salvador, 2. Ed, p. 71, 2013.

SILVA, D. F. **Sistema de comunicação Bluetooth utilizando microcontrolador**. 2009. 17 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia da Computação) – Curso de Engenharia da Computação, Escola Politécnica de Pernambuco – Universidade de Pernambuco.

SMIDT, A. C. G. **Implantação de uma plataforma robótica controlado remotamente utilizando arduíno**. São Carlos, p. 20, 2013.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TRITON ROBOS. **Entendendo os servomotores**, TRITON ROBOS. 2018. Disponível em: <<http://www.tritonrobos.com.br/Servo.htm>>. Acesso em: 22 mai. 2018.

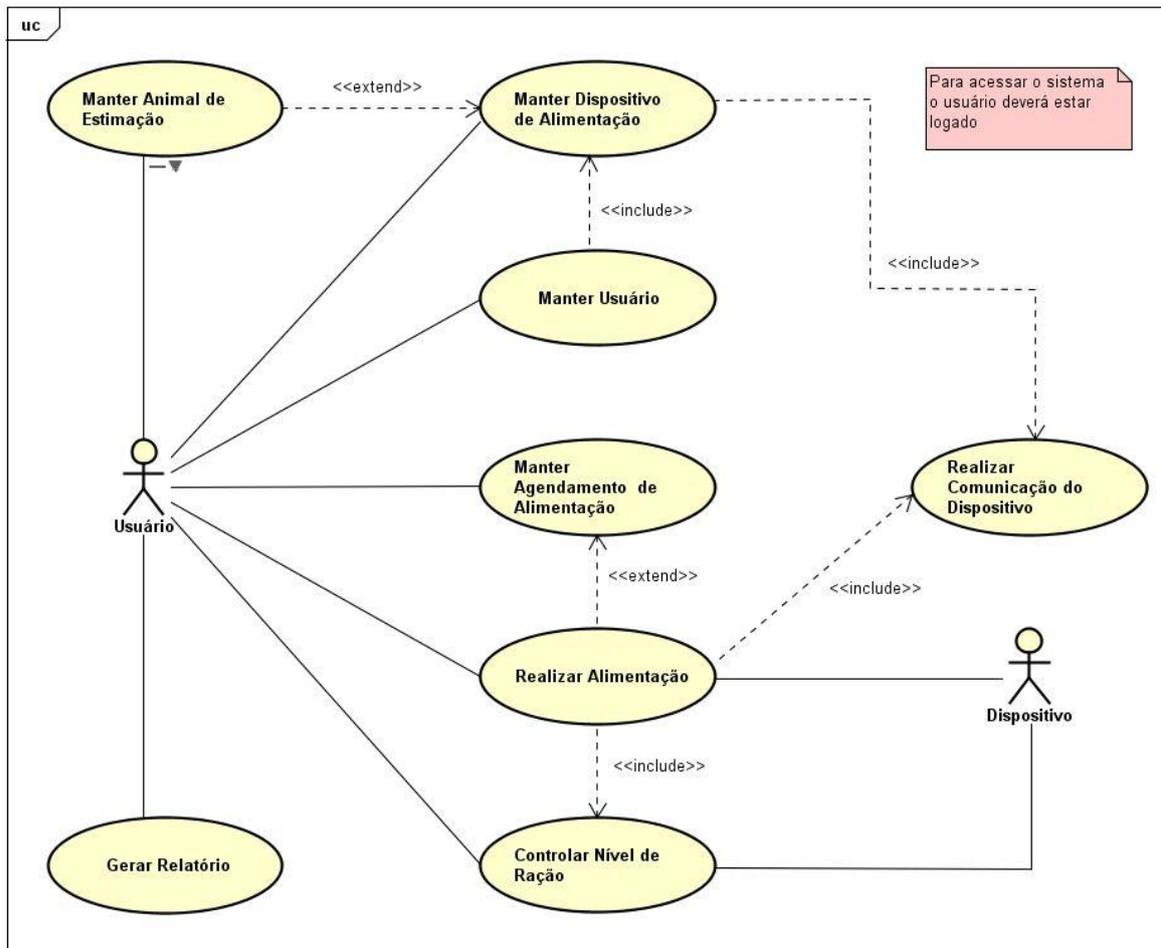
UNICAMP. **Introdução a linguagem C**. Campinas, p. 3, 2006.

USINAINFO. **Aplicação e funcionamento**. USINAINFO. 2016. Disponível em: <<http://blog.usinainfo.com.br/controle-de-posicao-do-micro-servo-motor-180/>>. Acesso em: 21 mai. 2018.

Apêndice A: Diagrama Caso de Uso

A imagem a seguir refere-se ao diagrama de caso de uso e a descrição das funcionalidades do sistema.

Imagem 1 – Diagrama de caso de uso.



Fonte: Autores da Pesquisa.

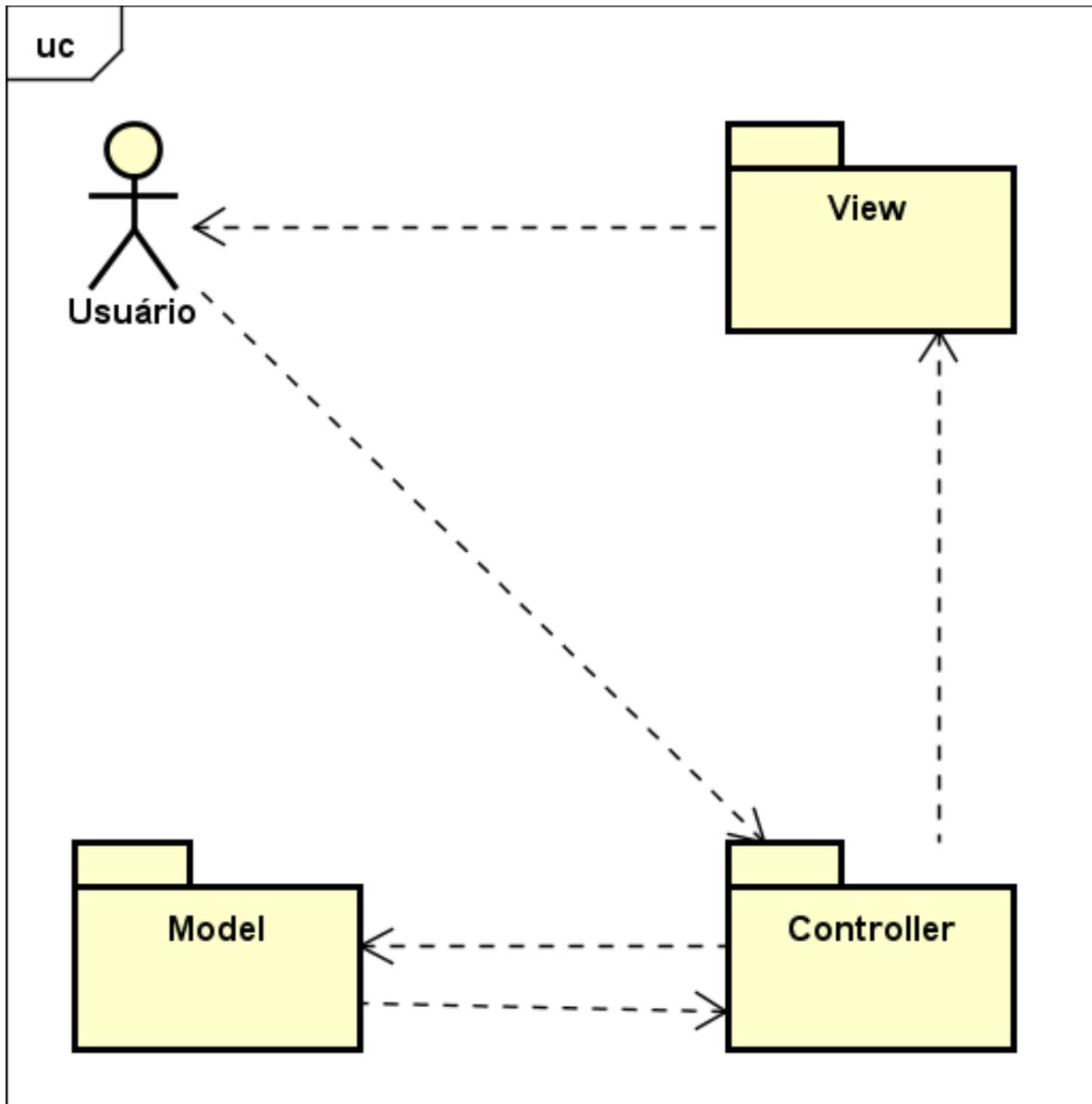
As funcionalidades do sistema:

- UC. 001 – Manter Usuário: o usuário é responsável por fazer seu cadastro no sistema, como dados pessoais, e-mail, por visualizar seus dados, alterá-los e excluí-los os mesmos;
- UC. 002 – Manter Agendamento de Alimentação: o usuário determinará em que horários o sistema irá dispensar ração no recipiente do animal;
- UC. 003 – Manter Animal de Estimação: o usuário é responsável por fazer cadastro de seu animal no sistema, como nome animal, idade, peso, raça;
- UC. 004 – Gerar Relatório: o usuário pode gerar relatórios dos agendamentos e quantidade de ração despejada para o animal;

- UC. 005 – Manter Dispositivo de Alimentação: o usuário é responsável por manter o dispositivo que irá alimentar o animal, para que possa ter o funcionamento correto;
- UC. 006 – Realizar Alimentação: o usuário tem a opção de alimentar o animal manualmente pelo sistema ou então através do agendamento pelo sistema, onde vai ser feita a alimentação automaticamente;
- UC. 007 – Controlar Nível de Ração: o usuário determinará quantidade de ração a ser despejada para o animal e o dispositivo é responsável por manter sempre o nível determinado pelo usuário;
- UC. 008 – Realizar Comunicação com o Dispositivo: através dos casos de uso Manter Dispositivo de Alimentação e Realizar Alimentação, onde através do sistema irá ser feita toda a comunicação com o dispositivo, onde terá agendamento de horário para funcionamento, alimentação manual pelo sistema;

Apêndice B: Visão lógica

A visão lógica da arquitetura utilizada para o desenvolvimento é o padrão *MVC*.



Model: Pacote responsável pela leitura e escrita de dados, e também de suas validações.

Controller: Pacote responsável por receber todas as requisições do usuário. Seus métodos chamados *actions* são responsáveis por uma página, controlando qual *model* usar e qual *view* será mostrado ao usuário.

View: Pacote que contém camada a interface e responsável pela interação com o usuário. Faz a exibição dos dados, sendo ele por meio de um *HTML* (*HyperText Markup Language*) ou *XML* (*Extensible Markup Language*).

Apêndice C: Repositório com códigos de fonte

Repositório Github:

<https://github.com/wellington29/Sistema-de-alimenta-o-autom-tica-para-animais.git>