

CENTRO UNIVERSITÁRIO UNIFACVEST
CIÊNCIA DA COMPUTAÇÃO
GABRIEL LEMOS ALVES

**AUTOFEEDER: ALIMENTADOR AUTOMÁTICO PARA ANIMAIS
DOMÉSTICOS DE PEQUENO PORTE**

LAGES
2020

GABRIEL LEMOS ALVES

**AUTOFEEDER: ALIMENTADOR AUTOMÁTICO PARA ANIMAIS
DOMÉSTICOS DE PEQUENO PORTE**

Projeto apresentado à Banca Examinadora do
Trabalho de Conclusão de Curso II de Ciência da
Computação para análise e aprovação.

LAGES
2020

GABRIEL LEMOS ALVES

**AUTOFEEDER: ALIMENTADOR AUTOMÁTICO PARA ANIMAIS
DOMÉSTICOS DE PEQUENO PORTE**

Trabalho de conclusão de curso apresentado à
Banca Examinadora da Unifacvest como parte dos
requisitos para obtenção do título de bacharel em
Ciência da Computação.

Orientador: Me. Igor Muzeka
Coorientadores:
Me. Márcio José Sembay

Lages, SC __/__/2020.

Nota _____

Coordenador do curso de graduação

LAGES
2020

Dedico este trabalho aos meus pais e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

AGRADECIMENTOS

Agradeço aos meus pais e familiares pelo apoio e incentivo durante todos os anos em que estive na faculdade.

Aos professores e orientadores que, através do conhecimento transmitido e da orientação prestada, me permitiram a realização deste trabalho.

E a todos que direta ou indiretamente fizeram parte da minha formação.

*“O insucesso é apenas uma oportunidade para
recomeçar de novo com mais inteligência”*
(Henry Ford)

RESUMO

Em virtude da falta de tempo, os proprietários de animais domésticos têm sido cada vez menos cuidadosos no que se diz respeito à alimentação de seus pets. Devido a isso, a saúde nutricional dos mesmos pode ser afetada pela falta ou excesso de nutrientes em sua dieta; tal fator pode, ainda, trazer consequências graves, como patologias (obesidade e desnutrição) e suas complicações. Com a utilização de um alimentador automático, é possível oferecer uma alimentação equilibrada e sem a necessidade de supervisão dos donos. Esse estudo terá como objetivo desenvolver um alimentador automático controlado por aplicativo, visando *smartphones* que utilizam o sistema operacional Android. A interface do *software* será desenvolvida de forma intuitiva para que possibilite a utilização por todos os tipos de usuários, e, por meio deste, o proprietário poderá configurar a quantidade e os horários das refeições de acordo com a recomendação veterinária.

Palavras-chave: Pets. Alimentador automático. Aplicativo. *Smartphones*.

ABSTRACT

Due to the lack of time, pet owners have been less and less careful when it comes to feeding their pets. Because of this, their nutritional health can be affected by the lack or excess of nutrients in their diet; this factor can also have serious consequences, such as pathologies (obesity and malnutrition) and their complications. Through the use of an automatic feeder, it is possible to offer a balanced diet without the need of supervision. This study will aim to develop an application-driven automatic feeder, targeting smartphones that use the Android operating system. The software interface will be intuitively designed to allow use by all types of users, and through it, the owner can configure the quantity and meal frequency according to the veterinary recommendation.

Keywords: *Pets. Automatic feeder. App. Smartphones.*

RESUMEN

Debido a la falta de tiempo, los propietarios de animales domésticos han sido cada vez menos cuidadosos en lo que se refiere a la alimentación de sus mascotas. Debido a esto, la salud nutricional de los mismos puede ser afectada por la falta o exceso de nutrientes en su dieta; tal factor puede, además, traer consecuencias graves, como patologías (obesidad y desnutrición) y sus complicaciones. Con el uso de un alimentador automático, es posible ofrecer una alimentación equilibrada y sin la necesidad de supervisión de los dueños. Este estudio tendrá como objetivo desarrollar un alimentador automático controlado por aplicación, buscando smartphones que utilizan el sistema operativo Android. La interfaz del software será desarrollada de forma intuitiva para que posibilite la utilización por todo tipo de usuarios, y, por medio de éste, el propietario podrá configurar la cantidad y los horarios de las comidas de acuerdo con la recomendación veterinaria.

Palabras claves: *Mascotas. Alimentador automático. Aplicación. Smartphones.*

LISTA DE FIGURAS

Figura 1: População de animais de estimação no Brasil	17
Figura 2: Recomendação diária de ração para cães adultos de pequeno porte	19
Figura 3: Recomendação diária de ração para gatos adultos	20
Figura 4: Os sistemas operacionais mais usados no mundo em abril de 2020.....	21
Figura 5: Os pilares da programação orientada a objetos	22
Figura 6: Métodos HTTP	24
Figura 7: Exemplo de JSON.....	24
Figura 8: Placa Arduino	25
Figura 9: Módulo <i>Bluetooth</i> HC-05	26
Figura 10: Módulo RTC DS3231	26
Figura 11: Motor reversível 12v.....	27
Figura 12: Célula de carga	28
Figura 13: Diagrama de caso de uso	33
Figura 14: Diagrama de atividade	34
Figura 15: Alimentador	35
Figura 16: Sistema de transporte do alimento.....	36
Figura 17: Tela de acesso	37
Figura 18: Tela de cadastro de usuário.....	38
Figura 19: Tela principal.....	38
Figura 20: Menu do aplicativo	39
Figura 21: Tela para conectar-se ao alimentador	40
Figura 22: Tela refeições programadas	40
Figura 23: Tela cadastro de refeição	41
Figura 24: Tela para ajustar hora (RTC DS3231).....	42
Figura 25: Diagrama do banco de dados.....	43

LISTA DE QUADROS

Quadro 1: Cronograma	31
-----------------------------------	----

LISTA DE SIGLAS

ABINPET	Associação Brasileira da Indústria de Produtos para Animais de Estimação
API	Application Programming Interface
APK	Android Package
AT	Attention
CRUD	Create Read Update Delete
HTTP	HyperText Transfer Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	Integrated Development Environment
iOS	Iphone Operating System
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
OHA	Open Handset Alliance
POO	Programação Orientada a Objetos
REST	Representational State Transfer
RTC	Real Time Clock
SDK	Software Development Kit
SO	Sistema Operacional
SOAP	Simple Object Access Protocol
UML	Unified Modeling Language
USB	Universal Serial Bus
XML	eXtensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa.....	15
1.2	Importância.....	15
1.2.1	<i>Acadêmica</i>	15
1.2.2	<i>Social</i>	15
2	OBJETIVO.....	16
2.1	Objetivo Geral	16
2.2	Objetivos Específicos	16
3	REVISÃO DE LITERATURA	17
3.1	Nutrição para animais domésticos.....	17
3.1.1	Nutrição para cães.....	18
3.1.2	Nutrição para gatos	19
3.2	Android.....	20
3.3	Java	21
3.4	Programação Orientada a Objetos	22
3.5	Sistemas Distribuídos	23
3.5.1	<i>Web API</i>	23
3.5.2	HTTP	23
3.5.3	JSON.....	24
3.6	Arduino.....	24
3.6.1	Módulo <i>Bluetooth</i> HC-05	25
3.6.2	Módulo RTC DS3231	26
3.6.3	Motor reversível.....	27
3.6.4	Célula de carga.....	27
3.7	Ferramentas do projeto	28
3.7.1	Android Studio.....	28
3.7.2	Visual Studio Community	29
3.7.3	Arduino IDE	29
4	METODOLOGIA	30
4.1	Documentação	30
4.2	Natureza da Pesquisa	30
4.3	Método de Pesquisa.....	30
5	CRONOGRAMA.....	31
6	PROJETO.....	32
6.1	Diagramas UML	32
6.2	Alimentador	34
6.3	Aplicativo	37
6.4	<i>Web API</i>	42
6.5	Banco de Dados	42
7	TRABALHOS CORRELATOS.....	44
7.1	Comedouro PlayPet	44
8	LIMITAÇÕES DA PESQUISA.....	45
9	RESULTADOS	46
	REFERÊNCIAS	47

1 INTRODUÇÃO

Já passou o tempo em que os cães eram alimentados regularmente com restos de refeições, e os gatos, com leite de vaca (que além de não corresponder com suas necessidades nutricionais, pode ainda, ocasionar problemas digestivos devido à baixa produção de enzimas responsáveis pela digestão da lactose por parte dos felinos). Mas, além disso, existe outro problema relacionado à nutrição dos pets. É o antropomorfismo, que, em consequência da proximidade entre seres humanos e animais, nos passa a falsa impressão de conhecimento sobre o funcionamento dos mesmos, impondo a eles o estilo de vida humana e deixando de lado suas diferenças (GRANDJEAN, 2006).

Visando otimizar o pouco tempo que as pessoas possuem atualmente e, ainda, oferecer uma dieta equilibra aos animais, a ideia de automatizar o processo de alimentação possibilita resolver tais problemas de forma prática e eficiente.

A ascensão da era dos *smartphones* propicia o desenvolvimento de um alimentador automático controlado por aplicativo, no qual, o proprietário poderá programar o horário das refeições diárias de seu pet, bem como a quantidade de ração presente em cada uma delas de acordo com a recomendação veterinária.

Além disso, o uso do aplicativo torna a programação do alimentador muito mais fácil para o usuário, pois atualmente grande parte da população possui familiaridade com o celular. Segundo Pereira (2016), os *smartphones* estão amplamente presentes no cotidiano da sociedade contemporânea, uma vez que sua aquisição se tornou acessível a todas as camadas da sociedade. Um dos fatores responsáveis por isso é a grande variedade de aparelhos encontrados no mercado e suas diversas faixas de preço. Outro fator importante é a facilidade de pagamento que as lojas oferecem ao consumidor.

O protótipo do alimentador será impresso em uma impressora 3D e a plataforma Arduino ficará responsável por seu funcionamento interno. Para o desenvolvimento do aplicativo foi escolhido o sistema operacional Android em virtude de ser o principal SO dos dispositivos móveis da atualidade.

Pensando em segurança e no armazenamento de informações de acesso, será necessária a criação de uma *Web API*, um *framework* utilizado para construir serviços HTTP sobre o *.Net Framework* (DEV MEDIA, 2019), que neste caso, ficará responsável pela autenticação no aplicativo e pelo armazenamento de informações complexas.

1.1 Justificativa

Assim como a alimentação é essencial para a saúde dos seres humanos, com os animais não é diferente. Portanto, oferecer uma dieta equilibrada é uma tarefa que exige muita atenção por parte dos proprietários. Com a utilização de um alimentador automático é possível alcançar tal objetivo com muito menos esforço, pois o único trabalho será manter o nível de ração recomendado no reservatório e realizar a limpeza quando necessário.

1.2 Importância

A seguir serão apresentadas as importâncias acadêmicas e sociais deste trabalho.

1.2.1 Acadêmica

Utilizar o conhecimento acadêmico adquirido na instituição e ampliá-lo com o uso de tecnologias como Arduino, *Web API* e sistemas *mobile* para automatizar a alimentação dos pets e expandir a experiência na área.

1.2.2 Social

O projeto em questão proporciona mais comodidade aos proprietários de animais domésticos. Além de contribuir para a saúde e o bem-estar do animal, conseqüentemente melhora a relação entre as duas partes.

2 OBJETIVO

Neste capítulo serão apresentados os objetivos gerais e específicos deste trabalho.

2.1 Objetivo Geral

Automatizar o processo de alimentação de animais domésticos e gerenciar a disponibilidade de alimento utilizando módulos da plataforma Arduino que serão programados por meio de um sistema *mobile*.

2.2 Objetivos Específicos

Os objetivos específicos são:

- a) Controlar a disponibilidade de alimento utilizando módulos do Arduino.
- b) Desenvolver um aplicativo para facilitar a programação das refeições via *bluetooth*.
- c) Criar uma *Web API* para autenticação de usuário e para possibilitar a consulta de algumas informações a qualquer momento.

3 REVISÃO DE LITERATURA

Neste capítulo serão abordados os principais tópicos relacionados ao desenvolvimento do projeto AutoFeeder, tendo por objetivo realizar uma breve explicação sobre seus principais componentes, bem como evidenciar os benefícios que uma alimentação equilibrada oferece à saúde dos pets. Na seção Nutrição para animais domésticos, será dada ênfase à importância de uma alimentação regular para animais de estimação. Em Android, Java, Programação Orientada a Objetos e Sistemas Distribuídos será abordado um pouco da história e das características de cada escopo em questão. Em seguida, na seção Arduino, serão apresentados alguns dos componentes eletrônicos usados no projeto. Por fim, serão apresentadas as principais ferramentas utilizadas.

3.1 Nutrição para animais domésticos

O número de animais de estimação no Brasil cresce cada vez mais. Segundo o IBGE, em 2013 esse número alcançava a casa dos 132,4 milhões, dos quais 52,2 milhões representavam a população de cães e 22,1 milhões, a de gatos.



Fonte: IBGE – Pesquisa quinzenal. Elaboração: Abinpet. Dados 2013. *Estimativa Abinpet para outros animais de estimação (répteis e pequenos mamíferos).

“A nutrição veterinária é a ciência que tem por objetivo descobrir nutrientes essenciais e as suas respectivas funções, em doses benéficas para o animal, em função das quantidades ingeridas” (GRANDJEAN, 2006, p. 5).

Assim como os seres humanos, os animais precisam de uma alimentação balanceada, que varia de acordo com seu porte e espécie. Ela é de suma importância para que o animal tenha uma vida saudável, porém, muitas vezes essa questão é negligenciada pelos seus donos. Muitas pessoas, por exemplo, oferecem comida destinada a seres humanos para seus animais de estimação, e essa pode causar doenças nutricionais devido à carência ou excesso de nutrientes.

Animais fundamentalmente ativos retiram a energia de que necessitam das gorduras e não apresentam os problemas de “colesterol” como os humanos, mas podem sofrer de obesidade (e das suas consequências graves: cardíacas, diabéticas, articulares, etc.), caso não se respeite a alimentação e a dosagem adaptadas (GRANDJEAN, 2006, p. 6).

A ideia de um alimentador automático pode diminuir significativamente esses e outros problemas ocasionados por uma alimentação desapropriada e desequilibrada.

Segundo Grandjean (2006), outro erro comum é não estabelecer um limite na alimentação de seu pet, ou oferecer uma quantidade diária muito maior ou menor que a recomendada. Visto que alguns não possuem autocontrole, caso o limite não seja estabelecido, os animais continuarão ingerindo alimento enquanto este estiver disponível e isso ocasionará sérios problemas para sua saúde.

3.1.1 Nutrição para cães

A quantidade recomendada de ração que um cão deve ingerir diariamente varia de acordo com a marca, em virtude das concentrações dos nutrientes variarem entre uma e outra. Seguir essas recomendações é de suma importância para a saúde do animal, pois previne doenças graves, como a obesidade canina.

Outros fatores determinantes são o porte e o nível de atividade física diária. Quanto maior o cão, maior será seu peso e conseqüentemente sua necessidade energética, a mesma regra se aplica para cães sedentários e ativos.

Para manter-se saudável, seu cão precisa de uma dieta balanceada em termos de proteínas, gorduras, carboidratos, vitaminas e minerais. Cada nutriente deve estar em equilíbrio com o conteúdo energético da dieta e com cada um dos demais nutrientes. A quantidade correta de calorias irá depender do tamanho, raça e quantidade de exercício que o animal pratica. As fases da vida do animal também fazem uma grande diferença no tipo de alimentação: filhotes, cães adultos, cães idosos, fêmeas prenhas ou lactantes precisam de diferentes quantidades de nutrientes (PEDIGREE, 2019).

Figura 2: Recomendação diária de ração para cães adultos de pequeno porte

Peso do Cão Peso del Perro Weight of Dog	Baixa Atividade Baja Actividad Low Activity		Atividade Moderada Actividad Moderada Moderate Activity		Alta Atividade Alta Actividad High Activity	
	(g)	copos tazas cups	(g)	copos tazas cups	(g)	copos tazas cups
2 kg	40	1/2	46	1/2	53	5/8
3 kg	54	5/8	63	5/8	71	3/4
4 kg	67	3/4	78	7/8	88	1
5 kg	79	7/8	92	1	104	1+1/8
6 kg	91	1	105	1+1/8	120	1+1/4
7 kg	102	1+1/8	118	1+1/4	134	1+1/2
8 kg	113	1+1/4	131	1+3/8	148	1+5/8
9 kg	123	1+3/8	143	1+5/8	162	1+3/4
10 kg	133	1+1/2	154	1+3/4	175	1+7/8

1 copo = 240 ml = 91 gramas
 1 taza = 240 ml = 91 gramos
 1 cup = 240 ml = 91 grams

Fonte: <https://www.geracaopet.com.br/racao-royal-canin-mini-adulto>

Na Figura 2, podem-se observar as diferentes necessidades energéticas diárias para cada cão, de acordo com seu porte e nível de atividade. Recomenda-se ainda que a quantidade seja dividida entre duas refeições.

3.1.2 Nutrição para gatos

Assim como em todos os animais, a alimentação está diretamente ligada ao crescimento e a saúde dos gatos. Eles também possuem sua ingestão de ração diária recomendada variável de acordo com o porte, idade e nível de atividade. Portanto, ao comprar a ração é importante sempre verificar o rótulo, a fim de saber se a mesma é adequada para o animal.

A alimentação dos gatos deve ser rica em proteínas de qualidade, vitaminas e sais minerais em quantidades adequadas. Por isso, é imprescindível que o gato coma apenas alimentos desenvolvidos especificamente para ele. A ração para gatos é a melhor opção para a alimentação do felino. Por terem o organismo mais sensível que o nosso, os animais reagem de forma negativa quando ingerem alimentos que não são próprios para eles, o que pode gerar diversos problemas de saúde, como a diarreia felina. (MAX TOTAL ALIMENTOS, 2019).

Segundo Grandjean (2006), diferentemente dos cães, os gatos podem ter acesso livre à quantidade diária total do alimento, pois fazem inúmeras refeições em pequenas quantidades durante todo o dia.

Figura 3: Recomendação diária de ração para gatos adultos

Quantidade diária recomendada						
Peso do gato (kg)	Baixa atividade		Atividade normal		Ativo	
	g	copo*	g	copo*	g	copo*
2	25	1/2	30	1/2	35	1/2
3	35	1/2	40	1/2	50	1/2
4	40	1/2	50	1/2	60	1/2
5	45	1/2	60	1/2	70	1
6	55	1/2	65	1	80	1
7	60	1/2	75	1	90	1
8	65	1	85	1	200	1
9	70	1	90	1	110	1
10	75	1	100	1	115	1 1/2

*1 copo 240 ml = 94 gramas

Fonte: <https://www.geracaopet.com.br/racao-royal-canin-premium-cat-gatos-adultos>

Na Figura 3, fica evidente a importância de se respeitar as quantidades diárias para cada perfil de animal, pois tanto o excesso quanto a falta de alimento são extremamente prejudiciais à sua saúde.

3.2 Android

O Android é um sistema operacional para dispositivos móveis liderado pela Google e é líder absoluto desse mercado. Porém, essa liderança não se deve apenas ao fato de ser desenvolvido por uma empresa de grande expressão no cenário mundial. Um grupo chamado OHA (*Open Handset Alliance*) ajuda no desenvolvimento da plataforma, esse grupo é composto por grandes fabricantes de celulares como Samsung, LG, Motorola, Nextel, Asus, etc. e parte do sucesso que a plataforma tem atualmente deve-se à parceria feita por essas empresas que possuem a visão de melhorar a experiência móvel de todos os consumidores (LECHETA, 2016).

A plataforma é de código-fonte aberto e foi desenvolvida baseada no *kernel* do Linux, que é responsável pelo gerenciamento de memória, processos, *threads*, rede, *drivers* e por toda a segurança dos arquivos pessoais do usuário (GRIFFITHS; GRIFFITHS, 2016; LECHETA, 2016).

Os aplicativos desenvolvidos para Android são programados na linguagem de programação Java, com uma mistura de XML para a parte visual (GRIFFITHS; GRIFFITHS, 2016). As ferramentas do Android SDK são responsáveis pela compilação do código juntamente de todos os arquivos e recursos utilizados. A compilação gera um APK, que é um

arquivo de extensão .apk contendo todo o conteúdo, esse arquivo será utilizado posteriormente por dispositivos como *smartphones* e *tablets* para a instalação do aplicativo (ANDROID, 2019b).

Figura 4: Os sistemas operacionais mais usados no mundo em abril de 2020.



Fonte: StatCounter, 2020.

Na Figura 4, é possível observar que, atualmente, o sistema operacional Android é o mais utilizado no mundo, superando o Windows da poderosa Microsoft e com grande vantagem sobre o iOS da Apple, seu principal concorrente no mercado de dispositivos móveis.

3.3 Java

Em 1991, a empresa Sun Microsystems financiou uma pesquisa corporativa chamada *Green*. Essa pesquisa tinha por objetivo possibilitar a comunicação entre aparelhos eletrônicos. Desta forma, o chefe do projeto James Gosling criou a linguagem de programação chamada Oak, porém, mais tarde verificou-se que já existia uma linguagem de programação com esse nome. A equipe da Sun costumava visitar uma cafeteria local cujo nome era Java (cidade origem de um tipo de café importado) e, devido a isso, a linguagem passou a ser assim chamada, adotando também uma xícara de café como logotipo (DEITEL; DEITEL, 2003).

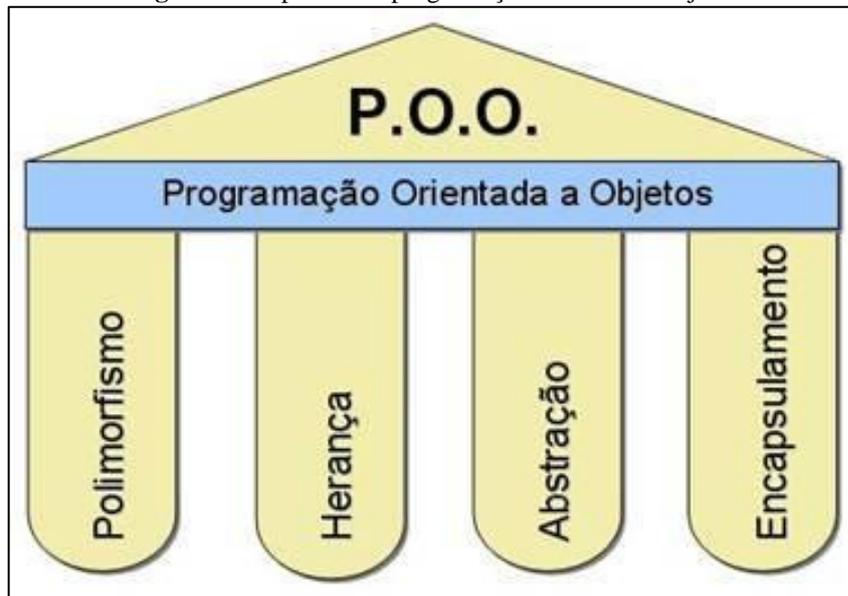
Trata-se de uma linguagem de alto nível, simples e de fácil uso, em virtude de ser orientada a objetos. Uma das principais vantagens dessa linguagem é a JVM (*Java Virtual Machine*), um interpretador capaz de ler os *bytecodes* gerados pela compilação e repassá-los para o processador do aparelho em questão, permitindo assim que o *software* seja executado em qualquer sistema operacional, sem a necessidade de recompilar o mesmo, tornando-o portátil (CLARO; SOBRAL, 2008).

3.4 Programação Orientada a Objetos

A programação orientada a objetos (POO) pode ser conceituada como um padrão de desenvolvimento muito utilizado atualmente em linguagens de alto nível, como Java e C#.

Dentro da POO existem quatro pilares (Figura 5), que podem ser entendidos como os requisitos que uma linguagem deve atender para que ela possa ser considerada dentro desse paradigma.

Figura 5: Os pilares da programação orientada a objetos



Fonte: <https://infogram.com/principios-da-programacao-orientada-a-objetos-1gv4m75ze1wkp18>

Abstração: o processo de abstração consiste na elaboração de um modelo abstrato de algo real através de seus atributos e métodos a fim de posteriormente transformá-lo em uma classe e instanciá-la para que ela possa ser acessada por meio de seu objeto.

Classe: uma classe é um conjunto de atributos (características) e métodos (ações) que descrevem um objeto (MICROSOFT, 2019c).

Objeto: um objeto é uma instância utilizável de uma classe (MICROSOFT, 2019c).

Encapsulamento: o encapsulamento permite que apenas as informações essenciais das operações se tornem acessáveis (públicas) e tem por objetivo proteger a integridade da classe, para que a mesma não possa ser manipulada de forma direta (RICARTE, 2001).

Herança: segundo Ricarte (2001), herança é a derivação de características comuns entre diversas classes a partir de uma classe base, ou superclasse, bastando implantar nas classes derivadas, ou subclasses, apenas suas particularidades.

Polimorfismo: o polimorfismo possibilita que duas ou mais classes derivadas de uma superclasse possam invocar métodos de mesma assinatura, mas com comportamentos diferentes (RICARTE, 2001).

3.5 Sistemas Distribuídos

Devido ao avanço da tecnologia, atualmente é possível construir sistemas compostos por um grande número de computadores conectados entre si em uma rede de alta velocidade. Esses sistemas são conhecidos como sistemas distribuídos (TANENBAUM; STEEN, 2007).

Tal prática possibilita, por exemplo, diminuir o custo de operação, onde um *mainframe* pode ser substituído por vários microcomputadores, a fim de distribuir o processamento através do compartilhamento de recursos, de forma transparente e tolerante a falhas.

Segundo Tanenbaum e Steen (2007, p.1), “um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”.

Outra vantagem dos sistemas distribuídos é a escalabilidade, onde facilmente podem ser adicionados computadores à rede.

3.5.1 Web API

A tecnologia *Web API* foi criada pela Microsoft e tem como função disponibilizar serviços HTTP para que estes sejam consumidos por *browsers* e dispositivos móveis (DEV MEDIA, 2019).

Essa ferramenta utiliza HTTP (*HyperText Transfer Protocol*) com REST (*Representational State Transfer*), que diferente do SOAP (*Simple Object Access Protocol*), não trafega dados exclusivamente em XML, nesse caso é usado por padrão o JSON (*JavaScript Object Notation*) que é um formato mais leve, legível e livre de plataformas (PIRES, 2019).

3.5.2 HTTP

O HTTP (*HyperText Transfer Protocol*) é um protocolo de transferência de hipertexto utilizado em toda a *World Wide Web*. Ele determina as mensagens que os clientes podem enviar aos servidores e que respostas serão recebidas. O protocolo foi criado visando futuras

aplicações orientadas a objetos e devido a isso ele aceita operações, os chamados métodos HTTP (TANENBAUM, 2003), que estão listados na Figura 6.

Figura 6: Métodos HTTP

Método	Descrição
GET	Solicita a leitura de uma página da Web
HEAD	Solicita a leitura de um cabeçalho de página da Web
PUT	Solicita o armazenamento de uma página da Web
POST	Acrescenta a um recurso (por exemplo, uma página da Web)
DELETE	Remove a página da Web
TRACE	Ecoa a solicitação recebida
CONNECT	Reservado para uso futuro
OPTIONS	Consulta certas opções

Fonte: TANENBAUM, 2003 pg. 494

3.5.3 JSON

O JSON (*JavaScript Object Notation*) é um formato de codificação de dados que permite a troca de informações entre navegadores clientes e serviços *Web* habilitados para AJAX (MICROSOFT, 2019b). Ele produz um texto formatado e remove a maioria dos espaços do documento, o que conseqüentemente torna-o mais leve, além de ser legível independente da plataforma.

Figura 7: Exemplo de JSON

```
[
  {
    "id": 1,
    "nome": "Usuario 1"
  },
  {
    "id": 2,
    "nome": "Usuario 2"
  },
  {
    "id": 3,
    "nome": "Usuario 3"
  },
  {
    "id": 4,
    "nome": "Usuario 4"
  }
]
```

Fonte: Autoria própria

3.6 Arduino

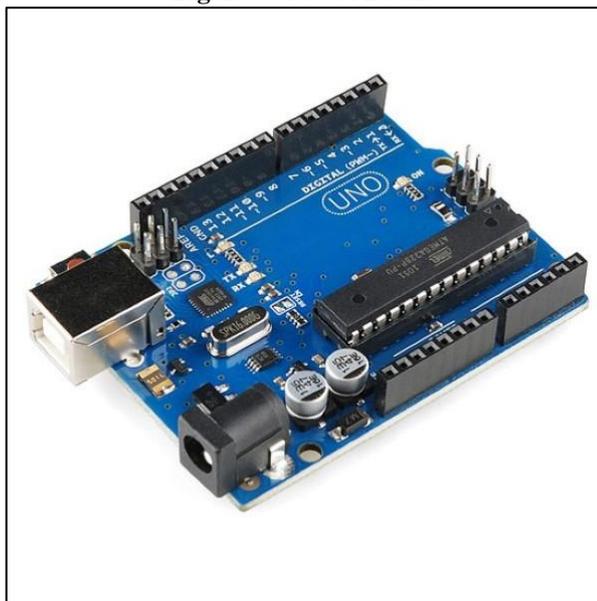
O Arduino foi criado em 2005 pelos pesquisadores Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis (EMBARCADOS, 2019). É uma plataforma de

computação física ou embarcada, programada geralmente na linguagem C/C++, que torna possível a interação de um sistema com seu ambiente através de *hardware* e *software* (MCROBERTS, 2011).

Uma placa Arduino é alimentada através de conexão USB (utilizada também para comunicação com o computador) ou de uma fonte externa. Seu componente principal é o microcontrolador, um chip programável responsável por processar as entradas e saídas entre a placa e os componentes externos conectados a ela (MCROBERTS, 2011; EMBARCADOS, 2019).

É uma plataforma de *hardware* e *software open-source*, ou seja, tanto o *hardware* quanto o *software* podem ser utilizados de forma gratuita, permitindo também a modificação e distribuição do mesmo.

Figura 8: Placa Arduino

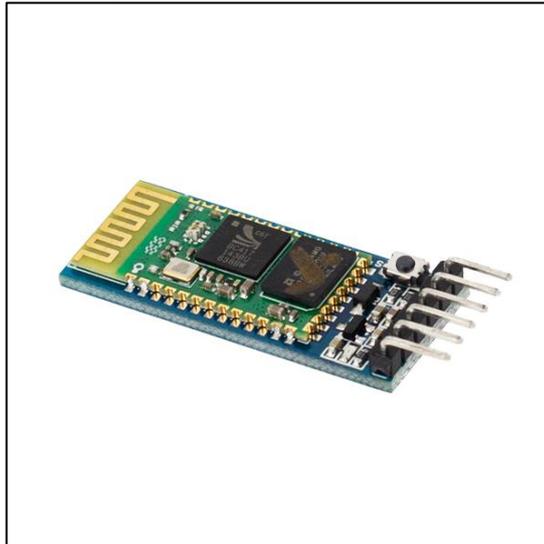


Fonte: <https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino>

3.6.1 Módulo *Bluetooth* HC-05

O módulo *bluetooth* HC-05 é indicado para projetos que necessitam de uma conexão sem fio simples e confiável, com um alcance de até 10 metros. Ele pode ser configurado por meio de comandos AT e funciona tanto em modo escravo (aceita conexões de outros dispositivos) como em modo mestre (realiza conexão com outros dispositivos). Com isso, é possível conectá-lo a um celular, ou criar uma rede de sensores intercomunicados (MULTILÓGICA-SHOP, 2020).

Figura 9: Módulo *Bluetooth* HC-05



Fonte: <https://www.filipeflop.com/produto/modulo-bluetooth-rs232-hc-05>

3.6.2 Módulo RTC DS3231

O *Real Time Clock* DS3231 é um relógio em tempo real com alta precisão, com ele é possível obter informações como: horas, minutos, segundos, dia, mês e ano. Além disso, o módulo conta com um circuito de detecção de falha de energia, onde a bateria é acionada automaticamente caso ocorra algum problema de alimentação no sistema, evitando a perda de informações (FILIFEFLOP, 2020b).

Por meio desse módulo o Arduino poderá saber quando a refeição deve ser despejada pelo alimentador.

Figura 10: Módulo RTC DS3231



Fonte: <https://www.filipeflop.com/produto/real-time-clock-rtc-ds3231>

3.6.3 Motor reversível

O motor reversível é um motor de turbo compressor que possui travamento automático, ou seja, o eixo de saída é travado automaticamente caso não haja corrente elétrica atuando sobre o mesmo, evitando que gire de um lado para o outro. Ele ainda permite inverter o sentido de rotação do eixo quando a tensão aplicada nos terminais do motor é alterada (FILIFEFLOP, 2020a).

Figura 11: Motor reversível 12v



Fonte: <https://www.filipeflop.com/produto/motor-reversivel-de-alto-torque-gw370-12v-30rpm>

3.6.4 Célula de carga

Uma célula de carga é um dispositivo capaz de medir a deformação ou flexão de um corpo e transformar essa ação em uma saída de tensão ou de corrente. A deformação sofrida pela célula será diretamente proporcional ao peso aplicado sobre ela, ou seja, quanto menor o peso aplicado menor será a deformação (TOLEDO DO BRASIL, 2019).

Células de carga com sensor de deformação possuem uma precisão de 0,25% a 0,03% do fundo de escala e são amplamente utilizadas em aplicações industriais (OMEGA, 2019a).

Deformação Mecânica é a mudança na forma de um material resultante da aplicação de força, e é medida pela mudança em seu comprimento. Ela pode ser de extensão ou de compressão. Um sensor de deformação mecânica é um equipamento que varia sua resistência elétrica proporcionalmente à deformação mecânica. A deformação mecânica pode resultar de uma variedade de influências internas ou externas, incluindo pressão, temperatura ou mudança estrutural (OMEGA, 2019b).

O sensor de peso para Arduino utiliza uma célula de carga com sensor de deformação mecânica do tipo *Strain Gauges* e deve ser usado em conjunto com um módulo HX711 (conversor analógico/digital projetado para transformar os valores de resistência dos sensores em dados digitais).

Figura 12: Célula de carga



Fonte: <https://www.vidadesilicio.com.br/sensor-de-peso-celula-de-carga-5kg>

3.7 Ferramentas do projeto

Neste capítulo, serão apresentadas as principais ferramentas utilizadas no desenvolvimento deste projeto.

3.7.1 Android Studio

O Android Studio é o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicativos para o sistema operacional Android. O *software* conta com um editor de código e ferramentas de desenvolvedor do IntelliJ (uma ferramenta integrada a IDE que visa dar assistência ao programador durante o desenvolvimento do código). Outro recurso importante do Android Studio é o emulador nativo, que permite testar rapidamente os aplicativos em um dispositivo virtual (ANDROID, 2019a).

3.7.2 Visual Studio Community

A IDE do Visual Studio é um painel de inicialização onde é possível criar, editar, compilar e publicar códigos escritos em linguagens como C#, C++, Visual Basic, Python, entre outras. Ele também conta com compiladores, ferramentas de preenchimento de código, *designers* gráficos e muitos outros recursos para facilitar o processo de desenvolvimento de *software* (MICROSOFT, 2019a). A versão *Community* é gratuita e destinada para desenvolvedores individuais.

3.7.3 Arduino IDE

O *software* Arduino IDE é uma ferramenta de código aberto que facilita o processo de gravação de código em qualquer modelo de placa Arduino. O ambiente é escrito em Java e compatível com os sistemas operacionais Windows, Mac OS X e Linux (ARDUINO, 2019).

4 METODOLOGIA

A seguir, serão descritos os instrumentos de pesquisa utilizados neste estudo.

4.1 Documentação

“É a fase da pesquisa realizada com intuito de recolher informações prévias sobre o campo de interesse” (MARCONI; LAKATOS, 2003). Para a realização deste trabalho foi utilizada a documentação indireta, com baseamento em publicações já existentes.

4.2 Natureza da Pesquisa

O estudo em questão possui natureza qualitativa, que de acordo com Wainer (2020), tem como base a observação minuciosa dos ambientes em que o sistema será instalado, e também procura analisar vários aspectos referentes aos usuários ou potenciais usuários desse sistema.

4.3 Método de Pesquisa

O método de pesquisa adotado para este trabalho é a pesquisa bibliográfica. Segundo Marconi e Lakatos (2003), esse tipo de pesquisa engloba toda a bibliografia pública sobre o assunto em questão, desde publicações avulsas, boletins, jornais, revistas, livros, pesquisas, monografias, teses, material cartográfico até meios de comunicação orais como rádio, filme e televisão. Ela tem por objetivo manter o pesquisador conectado com tudo aquilo que foi escrito sobre o tema.

5 CRONOGRAMA

Foi utilizado o seguinte cronograma para o desenvolvimento deste trabalho.

Quadro 1: Cronograma

Atividade	Fev	Mar	Abr	Mai	Jun	Jul
Revisão de literatura	X	X	X			
Estudo de técnicas		X	X			
Especificação do protótipo (UML)		X	X			
Desenvolvimento da <i>Web API</i>		X	X			
Desenvolvimento do aplicativo		X	X	X		
Desenvolvimento do alimentador				X	X	
Ajustes metodologia					X	
Entrega do TCC à coordenação					X	
Defesa TCC						X

Fonte: Autoria própria

6 PROJETO

Devido à rotina corrida, muitas pessoas passam o dia inteiro fora de casa e as que possuem animais de estimação devem ter uma atenção especial no que se diz respeito à alimentação dos mesmos. A utilização de um alimentador automático possibilita resolver esse problema de forma prática e eficiente, trazendo mais comodidade ao proprietário e consequentemente influencia de forma positiva na saúde e no bem estar do animal.

Com a intenção de resolver esse problema foi desenvolvido um aplicativo para o sistema operacional Android, proporcionado ao usuário realizar a programação das refeições de forma fácil. A *Web API* realiza todas as ações que possuem um alto custo de processamento e foi hospedada em um servidor *on-line*, onde retorna os dados no formato de texto (JSON) para o aplicativo. Para um melhor acabamento foi optado por realizar a impressão do alimentador em uma impressora 3D.

6.1 Diagramas UML

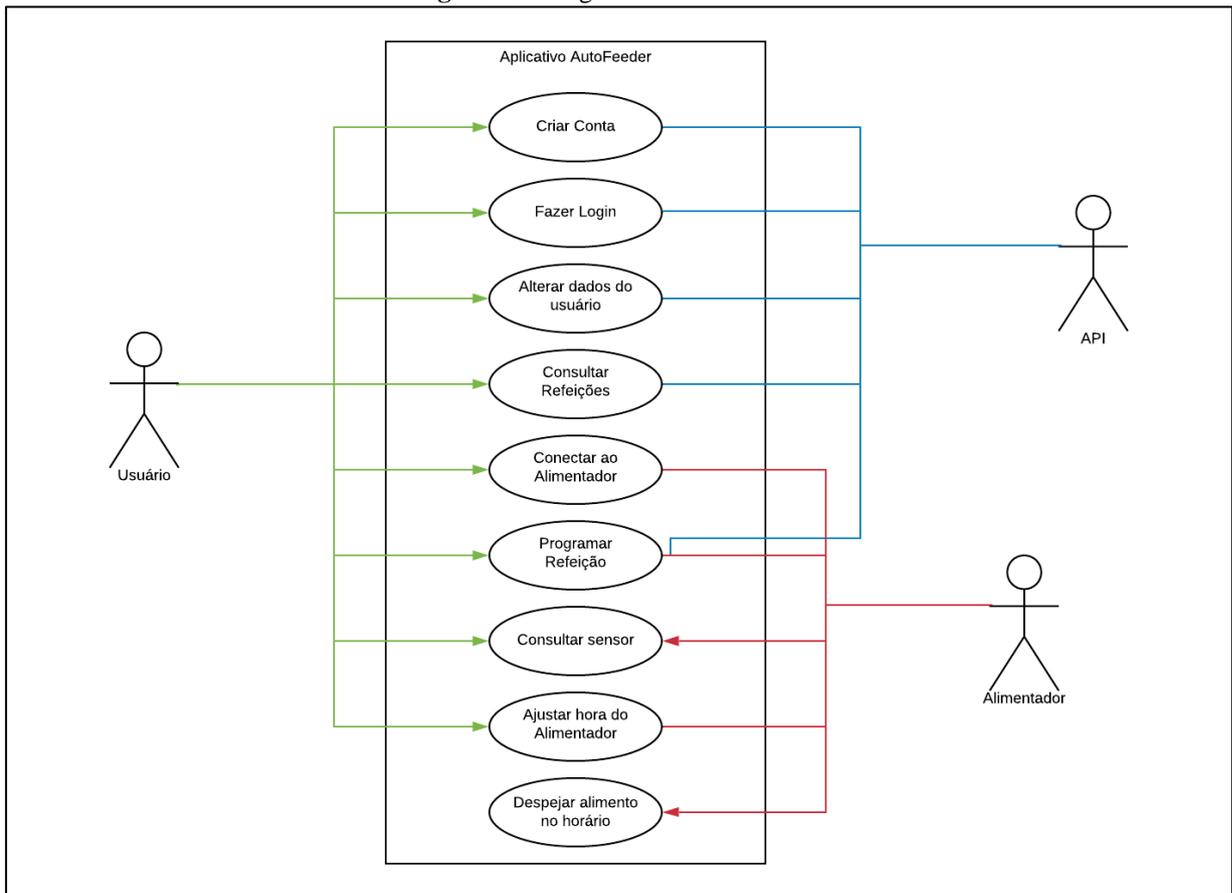
A UML (*Unified Modeling Language*) é uma linguagem utilizada para construir modelos de *software* precisos e completos de forma extensível. Ela permite modelar todos os tipos de sistema, o entendimento e utilização pelos usuários, a leitura por máquinas e também atende os analistas e projetistas em todo o ciclo de vida do sistema (PEREIRA, 2011).

Caso de uso

Os diagramas de caso de uso representam a relação dos atores com um conjunto de ações executadas durante alguma funcionalidade do sistema. Os casos de uso são denotados por ovais ou elipses que representam alguma funcionalidade dentro da aplicação e seus nomes devem ser escritos com um verbo no infinitivo concatenado a um substantivo, por exemplo, criar conta. O termo ator se refere ao papel que alguma coisa interpreta no contexto, na maioria das vezes os atores são seres humanos, mas também podem representar dispositivos eletrônicos ou outros sistemas que se relacionam com o sistema estudado (PEREIRA, 2011).

O diagrama a seguir mostra de forma simplificada as principais ações que podem ser realizadas pelo usuário dentro do aplicativo e suas respectivas relações com os outros sistemas desenvolvidos nesse projeto.

Figura 13: Diagrama de caso de uso



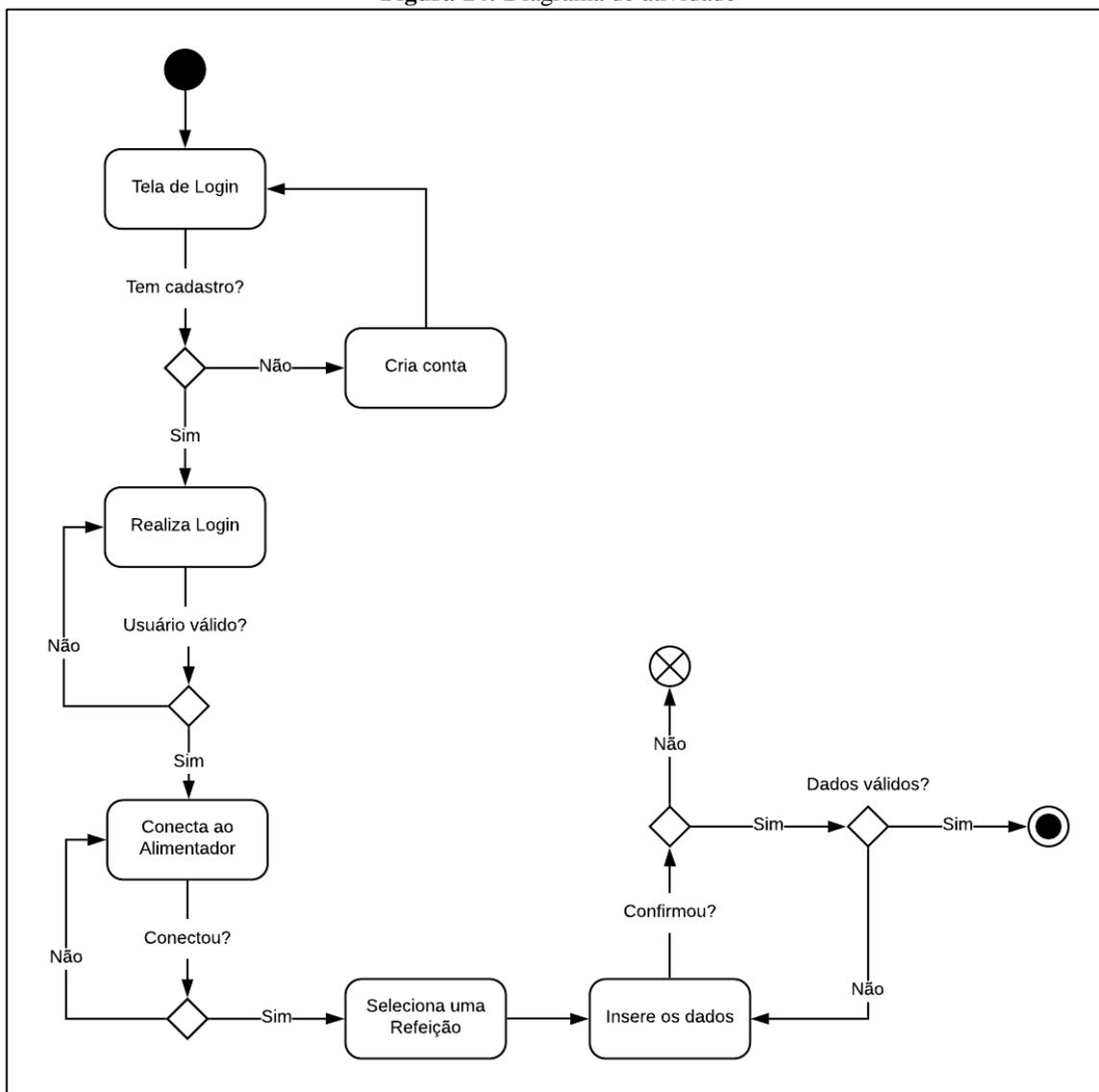
Fonte: Autoria própria

Diagrama de atividade

Os diagramas de atividade concentram-se no fluxo de controle entre as ações que representam um processo qualquer dentro do sistema. Eles apontam a ordem cronológica dessas ações, envolvendo a dimensão temporal do modelo de um *software*. Esse tipo de diagrama vem sendo usado em várias aplicações, até mesmo fora da área de desenvolvimento de sistemas, como especificação de algoritmos complexos, modelagem de processos de negócio, etc. Além disso, se utilizado na especificação dos casos de uso, a representação das ações dentro de um sistema pode ser modelada de forma mais fácil, pois é possível observar visualmente o início do processo e os possíveis fins que ele pode atingir (PEREIRA, 2011).

O diagrama de atividade da Figura 14 representa o fluxo para a programação de uma refeição pelo usuário.

Figura 14: Diagrama de atividade



Fonte: Autoria própria

6.2 Alimentador

O projeto 3D do alimentador foi desenvolvido por Krichbaumer (2020) e está disponível de forma gratuita em seu site. A impressão do protótipo foi realizada em uma impressora Sethi3D S3, com resolução de 0,3 mm por camada.

Ele conta com um compartimento para fixação do motor e instalação de componentes eletrônicos, um reservatório para ração com tampa, um sistema para transporte de comida e um pote onde será despejado o alimento.

Através do módulo RTC DS3231, o Arduino realiza a comparação do horário atual com os horários das refeições programadas pelo usuário, e, se essa condição for verdadeira, o motor é acionado, iniciando o processo de transporte de ração do reservatório até o pote.

A célula de carga foi instalada na base do alimentador, embaixo do pote, e é responsável por informar ao Arduino a quantidade de ração existente naquele momento, fazendo com que o motor seja parado assim que a porção de ração presente no pote for igual a que foi programada pelo usuário no aplicativo.

O alimentador possui uma chave liga/desliga e é alimentado por uma fonte 12 volts.

Figura 15: Alimentador



Fonte: Autoria própria

O sistema de transporte do alimento funciona por meio de um parafuso sem fim, que também foi impresso em 3D, e está acoplado ao eixo do motor. Quando o motor gira no sentido horário, o alimento presente no tubo é empurrado para frente, até chegar ao pote. O motor foi programado para realizar o movimento no sentido anti-horário de tempos em tempos, a fim de evitar o entupimento do tubo ou o travamento do parafuso.

Na Figura 16 é possível observar melhor como funciona o sistema de transporte de alimento entre o reservatório e o pote.

Figura 16: Sistema de transporte do alimento

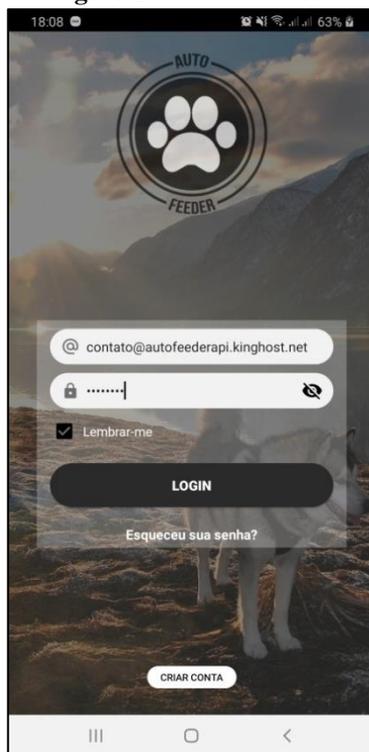


Fonte: Autoria própria

6.3 Aplicativo

O aplicativo AutoFeeder tem a função de programar as refeições diárias do pet, verificar o a quantidade de comida presente no alimentador remotamente e também realizar algumas configurações. Ele iniciará em uma tela de *login* (Figura 17), onde o proprietário deverá inserir seu usuário e senha para ter acesso às informações e funcionalidades do mesmo.

Figura 17: Tela de acesso

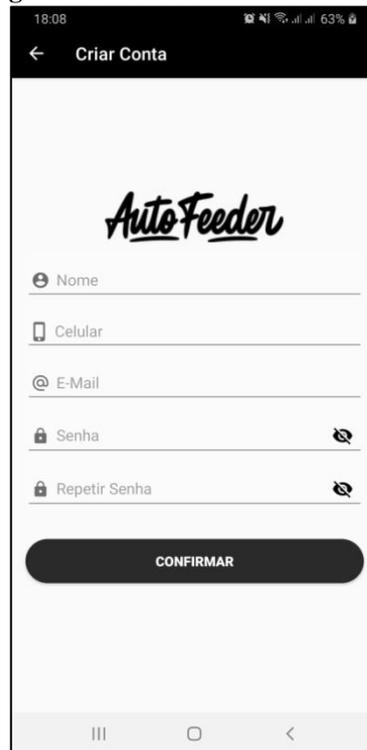


Fonte: Autoria própria

Caso ele já possua uma conta, basta informar os dados e clicar no botão *login* para ter acesso completo ao aplicativo. Do contrário, clicando sobre o botão criar conta, será aberta uma nova tela (Figura 18) para que seja feito um novo cadastro.

Se ocorrer o esquecimento da senha, será possível recuperá-la clicando sobre “Esqueceu sua senha?”, onde serão enviadas para o *e-mail* do usuário, as instruções referentes à criação de uma nova.

Figura 18: Tela de cadastro de usuário



Fonte: Autoria própria

Realizado o acesso, será aberta a tela principal do aplicativo (Figura 19).

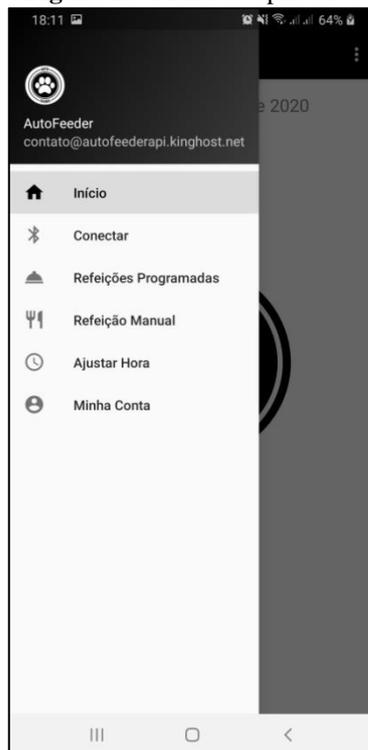
Figura 19: Tela principal



Fonte: Autoria própria

Na tela principal, será possível visualizar em sua área central a próxima refeição do animal, e caso o mesmo esteja conectado ao alimentador, na parte inferior será mostrada a quantidade de alimento disponível para o consumo naquele momento. Ao clicar no botão menu (ícone localizado no canto superior esquerdo) será aberto o menu de opções do aplicativo.

Figura 20: Menu do aplicativo

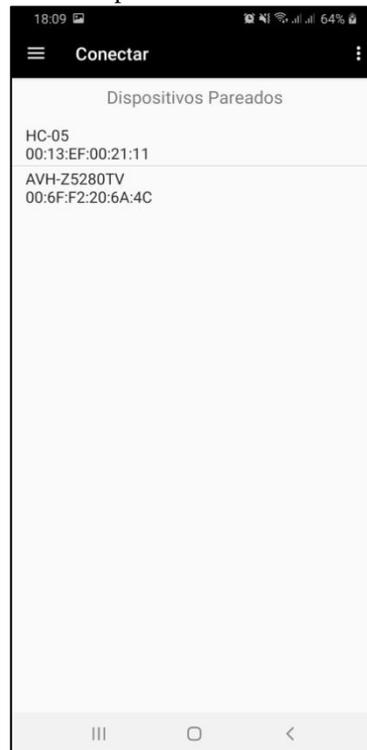


Fonte: Autoria própria

Clicando sobre a opção conectar, serão listados os dispositivos *bluetooth* pareados com o celular. O usuário deve realizar o pareamento com o módulo HC-05 fora do aplicativo, uma única vez, utilizando a senha padrão 1234. Assim o mesmo será listado para receber a conexão como é mostrado na Figura 21. Para conectar-se basta clicar sobre o nome do dispositivo e aguardar a mensagem informando que a conexão foi estabelecida com sucesso.

A conexão permanecerá ativa até que o aplicativo seja encerrado pelo usuário, ou que o aparelho esteja fora do alcance do alimentador.

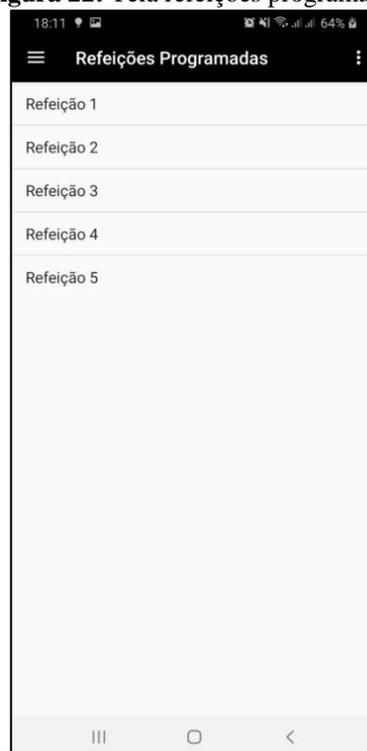
Figura 21: Tela para conectar-se ao alimentador



Fonte: Autoria própria

Ao clicar no item Refeições Programadas será mostrada uma lista com as cinco refeições programáveis pelo usuário (Figura 22).

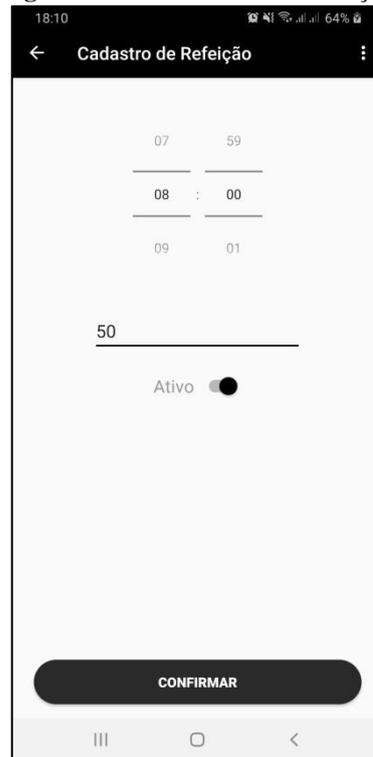
Figura 22: Tela refeições programadas



Fonte: Autoria própria

Escolhendo qualquer uma das refeições será aberta a tela de cadastro de refeição (Figura 23), onde é possível visualizar as informações de cada uma delas. Para fazer alterações o usuário deve estar conectado ao alimentador, além de respeitar algumas regras que foram impostas, como ordem cronológica das refeições e as quantidades mínima e máxima de alimento.

Figura 23: Tela cadastro de refeição

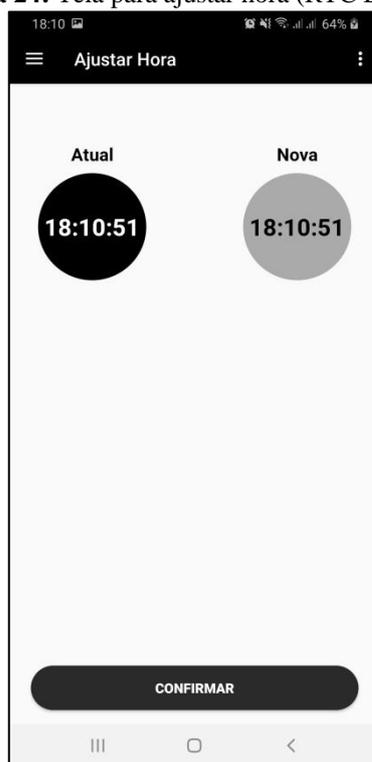


Fonte: Autoria própria

Outra função importante do aplicativo é realizar a configuração da hora do módulo RTC DS3231 que está conectado ao Arduino. Para isso basta estar conectado ao alimentador e clicar sobre o item Ajustar Hora, onde será aberta uma tela informando a hora atual do módulo e o novo horário caso o usuário confirme a operação, como é mostrado na Figura 24.

O usuário pode ainda verificar seus dados cadastrais e alterar sua senha acessando o menu Minha Conta.

Figura 24: Tela para ajustar hora (RTC DS3231)



Fonte: Autoria própria

6.4 Web API

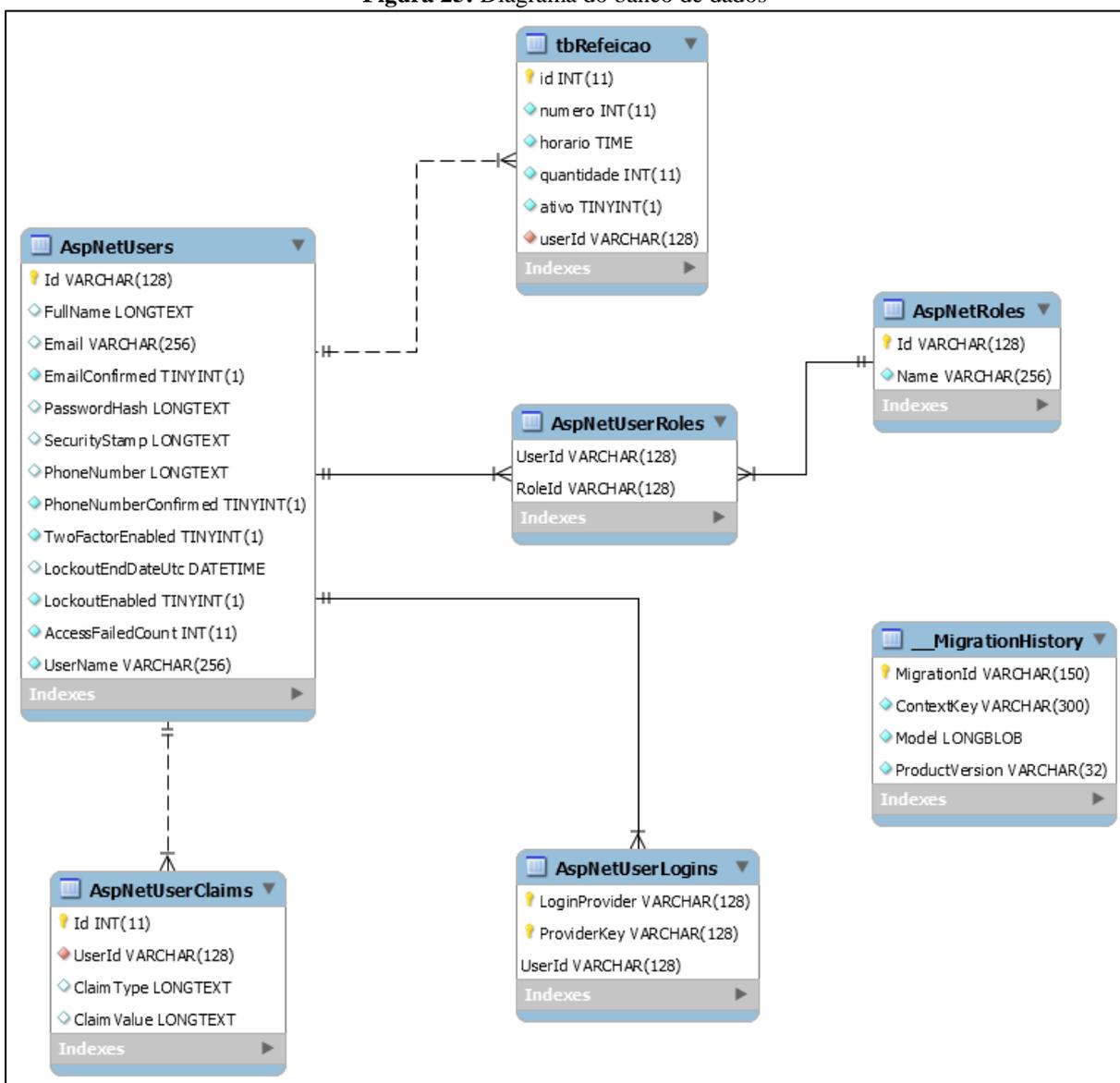
A *Web API* criada para esse projeto terá a finalidade realizar a integração entre o aplicativo e o banco de dados, autenticação de usuários e também fazer operações de CRUD, deixando todo o processamento pesado a encargo do servidor.

6.5 Banco de Dados

O banco de dados ficará responsável por armazenar as informações do usuário e as refeições programadas por cada um deles, para que possam ser consultadas mesmo que o aplicativo não esteja conectado ao alimentador. No projeto em questão foi escolhido o banco de dados MySQL, que será hospedado no mesmo servidor da *Web API*.

Na Figura 25 é possível observar o diagrama contendo todas as tabelas do banco. Elas foram criadas automaticamente pelo *Entity Framework Code First*, que analisa as classes do *software* e realiza a criação da base de dados para o programador.

Figura 25: Diagrama do banco de dados



Fonte: Autoria própria

7 TRABALHOS CORRELATOS

7.1 Comedouro PlayPet

O comedouro Playpet é um alimentador inteligente para animais de estimação presente no mercado. Ele é controlado por aplicativo e pode ser acionado remotamente ou em tempo real, o comedouro também conta com o agendamento de refeições para quando o proprietário estiver fora de casa e é indicado apenas para alimentos de grãos secos e pequenos (PLAYPET, 2019).

O diferencial do projeto AutoFeeder é adicionar ao alimentador sensores que possibilitem ao usuário saber remotamente a quantidade de ração para consumo do animal. Além disso, os alimentadores que se dizem inteligentes muitas vezes não são, pois ao não contar com sensores é impossível saber se o animal tem comido a ração que foi despejada, e caso isso não aconteça, ele irá continuar despejando alimento, tornando a funcionalidade de um produto caro igual à de um simples alimentador por gravidade.

8 LIMITAÇÕES DA PESQUISA

O projeto AutoFeeder foi desenvolvido para ser utilizado apenas com animais domésticos de pequeno porte e que se alimentem à base de ração, outros tipos de alimentos ou até mesmo água não serão abordados.

A programação das refeições e a verificação de informações referentes aos módulos do Arduino por meio do aplicativo será totalmente dependente da conexão *bluetooth* entre ambas as partes.

Por se tratar de apenas uma unidade do protótipo, o funcionamento do sistema está limitado a um único alimentador que é controlado por um único usuário.

Como futuras implementações e no caso de serem produzidas mais unidades do projeto pretende-se alterar essa limitação, para que vários usuários possam utilizar um ou mais alimentadores.

9 RESULTADOS

Com a utilização do alimentador foi possível proporcionar uma alimentação adequada de forma prática e eficiente aos pets, tornando-o útil tanto para os proprietários que possuem escassez de tempo, quanto para os que procuram facilidades para o seu dia a dia.

Além disso, o aplicativo deixou a programação das refeições muito mais fácil para o usuário, tornando o grupo de possíveis utilizadores ainda maior, e conseqüentemente reduziu o custo do projeto, pois não foi preciso instalar botões e *display's* para realizar as configurações.

A hospedagem da *Web API* e da base de dados possibilitou a consulta de informações salvas sem a necessidade da conexão *bluetooth*, além de contribuir para o bom funcionamento do sistema, com recursos como: autenticação de usuários, confirmação obrigatória de e-mail e recuperação de senha.

REFERÊNCIAS

ANDROID. **Conheça o Android Studio**. Disponível em: <<https://developer.android.com/studio/intro?hl=pt-br>>. Acesso em: 30 jun. 2019a.

ANDROID. **Fundamentos de aplicativos**. Disponível em: <<https://developer.android.com/guide/components/fundamentals.html?hl=pt-br>>. Acesso em: 07 abr. 2019b.

ARDUINO. **Software**. Disponível em: <<https://www.arduino.cc/en/Main/Software>>. Acesso em: 30 jun. 2019.

CLARO, Daniela Barreiro; SOBRAL, João Bosco Manguiera. **Programação em JAVA**. Florianópolis: Copyleft Pearson Education, 2008.

DEITEL, Paul; DEITEL, Harvey. **Java: como programar**. 4. ed. Porto Alegre: Bookman, 2003.

DEVMEDIA. **Introdução ao ASP.NET Web API**. Disponível em: <<https://www.devmedia.com.br/introducao-ao-asp-net-web-api/25180>>. Acesso em: 01 jun. 2019.

EMBARCADOS. **Arduino UNO**. Disponível em: <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em: 19 abr. 2019.

FILIFELOP. **Motor Reversível de Alto Torque GW370 12V 30RPM**. Disponível em: <<https://www.filipeflop.com/produto/motor-reversivel-de-alto-torque-gw370-12v-30rpm>>. Acesso em: 12 maio 2020a.

FILIFELOP. **Real Time Clock RTC DS3231**. Disponível em: <<https://www.filipeflop.com/produto/real-time-clock-rtc-ds3231>>. Acesso em: 10 maio 2020b.

GRANDJEAN, Dominique. **Tudo o que você deve saber sobre nutrientes para saúde de cães e gatos**. Paris: Editora Aniwa, 2006.

GRIFFITHS, Dawn; GRIFFITHS, David. **Use a Cabeça! Desenvolvendo para Android**. Rio de Janeiro: Alta Books Editora, 2016.

KRICHBAUMER, Thomas. **The Cat-feeder**. Disponível em: <<https://www.machs-selbst.net/kopie-von-retropie-emulationstation>>. Acesso em: 01 maio 2020.

LECHETA, Ricardo R.. **Android Essencial**: Edição resumida do livro Google Android. São Paulo: Novatec Editora Ltda., 2016.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de Metodologia Científica**. 5. ed. São Paulo: Editora Atlas, 2003.

MAX TOTAL ALIMENTOS. **Quantidade ideal de ração para gatos: respeite a porção diária.** Disponível em: <<https://www.maxtotalalimentos.com.br/dica-gato/alimentacao-felina/quantidade-ideal-racao-para-gatos-respeite-porcao-diaria>>. Acesso em: 07 abr. 2019.

MCROBERTS, Michael. **Arduino Básico.** São Paulo: Novatec Editora Ltda., 2011.
MICROSOFT. **Bem-vindo ao IDE do Visual Studio.** Disponível em: <<https://docs.microsoft.com/pt-br/visualstudio/get-started/visual-studio-ide?view=vs-2019>>. Acesso em: 30 jun. 2019a.

MICROSOFT. **Como Serializar e desserializar dados JSON.** Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/framework/wcf/feature-details/how-to-serialize-and-deserialize-json-data>>. Acesso em: 01 jun. 2019b.

MICROSOFT. **Object-Oriented Programming (C#).** Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/concepts/object-oriented-programming>>. Acesso em: 18 abr. 2019c.

MULTILÓGICA-SHOP. **Módulo Bluetooth HC-05.** Disponível em: <<https://multilogica-shop.com/modulo-bluetooth-hc-05>>. Acesso em: 10 maio 2020.

OMEGA. **Célula de Carga.** Disponível em: <<https://br.omega.com/prodinfo/celulas-de-carga.html>>. Acesso em: 21 abr. 2019a.

OMEGA. **Otimizando Desempenho de Deformação Mecânica para Materiais Não Homogêneos.** Disponível em: <<https://br.omega.com/artigos-tecnicos/otimizando-o-desempenho-de-sensores-de-deformacao-mecanica.html>>. Acesso em: 21 abr. 2019b.

PEDIGREE. **Alimentando seu cão.** Disponível em: <<https://www.pedigree.com.br/artigos/adultos/alimentando-seu-cao>>. Acesso em: 07 abr. 2019.

PEREIRA, Luiz Antônio de Moraes. **Análise e Modelagem de Sistemas com a UML: Com Dicas e Exercícios Resolvidos.** Rio de Janeiro: Luiz Antônio M. Pereira, 2011.

PEREIRA, Jaqueline da Silva. Do consumo as apropriações: o uso de smartphones por estudantes do ensino médio em Cuiabá. **Revista Anagrama**, São Paulo, v. 10, n. 1, p.1-19, 04 jan. 2016. Disponível em: <<http://www.revistas.usp.br/anagrama/article/view/108978/107451>>. Acesso em: 26 maio 2019.

PIRES, Eduardo. **ASP.Net Web API – Meu primeiro serviço REST.** Disponível em: <<https://www.eduardopires.net.br/2013/07/asp-net-web-api-meu-primeiro-servico-rest/>>. Acesso em: 03 jun. 2019.

PLAYPET. **Comedouro Automático Com Aplicativo Playpet.** Disponível em: <<https://www.playpet.io/comprar/comedouro-automatico-com-aplicativo-playpet/>>. Acesso em: 02 jul. 2019

RICARTE, Ivan Luiz Marques. **Programação Orientada a Objetos: Uma Abordagem com Java.** Campinas, 2001.

TANENBAUM, Andrew S. **Redes de Computadores**. 4. ed. Rio de Janeiro: Editora Campus, 2003.

TANENBAUM, A. S.; STEEN, M. V. **Sistemas distribuídos: Princípios e paradigmas**. 2. ed. São Paulo: Pearson Prentice Hall, 2007. 402 p

TOLEDO DO BRASIL. **Você sabe como funciona uma célula de carga?** Disponível em: <<https://www.toledobrasil.com.br/blog/artigos/detalhe/voce-sabe-como-funciona-uma-celula-de-carga>>. Acesso em: 21 abr. 2019.

WAINER, Jacques. **Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação**. Disponível em: <<https://www.ic.unicamp.br/~wainer/papers/metod07.pdf>>. Acesso em: 01 jun. 2020.