

---

MINISTÉRIO DA EDUCAÇÃO – MEC

SECRETARIA DE EDUCAÇÃO SUPERIOR – SESU

PROGRAMA DE EDUCAÇÃO TUTORIAL – PET

UNIVERSIDADE FEDERAL FLUMINENSE – UFF

ESCOLA DE ENGENHARIA – TCE

GRUPO PET DO CURSO DE ENG. DE TELECOMUNICAÇÕES – PET-TELE

## Projetos PET-Tele

Sistema didático de reservatórios de detenção  
aplicados à drenagem urbana do  
LaDISan/TCE/UFF:  
novas implementações para automatização e  
acesso remoto utilizando a plataforma  
Raspberry Pi  
(Versão: A2021M10D08)

Autores: Raphael Miranda

Tutor: Alexandre Santos de la Vega

Niterói – RJ

Outubro / 2021

---

# Resumo

Este trabalho propõe uma nova infraestrutura ao projeto de integração de saberes que visa à facilitação do processo ensino-aprendizagem de Engenharia, realizado pelo grupo PET-Tele, em conjunto com o Laboratório de Drenagem, Irrigação e Saneamento Ambiental (LaDISan), da Universidade Federal Fluminense. Ele possibilitou o desenvolvimento de aulas práticas remotas, onde ensaios realizados no laboratório foram assistidos, em tempo real, pelos alunos em sala de aula teórica. A prática aborda o efeito de reservatórios de retenção em série, aplicados à minimização de cheias urbanas. O laboratório está equipado com um modelo físico de estudos hidráulicos, que simula o funcionamento de dois reservatórios de retenção conectados em cascata, apoiados por um sistema de reserva e bombeamento, que permite o reuso contínuo de água. Anteriormente, o PET-Tele implantou um sistema para automatização do processo de medição, com o registro simultâneo de vazão em três pontos: na entrada do primeiro reservatório, na conexão entre o primeiro e o segundo, e na saída do segundo. Os três hidrogramas foram registrados simultaneamente e transmitidos para a sala de aula teórica. Os resultados dessas aulas práticas remotas, realizadas em alguns períodos letivos, ampliaram-se para além de sua motivação inicial, apenas ligada à questão do pequeno espaço físico do laboratório em relação ao número de alunos por turma. Houve vantagens adicionais, tais como: a economia de tempo, maior eficácia na apreensão do conteúdo da prática, além de maior interesse e aceitação por parte dos alunos. Uma proposta de evolução da infraestrutura original é aqui apresentada com integração de uma plataforma de desenvolvimento Raspberry Pi, um servomotor e peças em impressão 3D para fins de automatização do sistema.

# Lista de Figuras

2.1	Sensor de vazão YF-S201. . . . .	9
2.2	Arduino Uno R3. . . . .	10
2.3	Raspberry Pi Revision 2.0. . . . .	11
2.4	ViewDuino: tela inicial. Versão: set/2015. . . . .	12
2.5	Desenho ilustrativo do servomotor. . . . .	13
2.6	Desenho elaborado no <i>software</i> TinkerCad para ilustrar a conexão entre um Arduino e um servomotor. . . . .	13
2.7	Exemplo de modelo em 3D, desenhado no <i>software</i> TinkerCad. . . . .	14
2.8	Troca do acabamento destinado ao acionamento manual do registro. Fotografia da condição inicial da frente (a) e em (b) fotografia do suporte acoplado. . . . .	14
2.9	Janela de <i>login</i> da aplicação CMS para Windows. . . . .	16
2.10	Janela principal da aplicação CMS para Windows. . . . .	16
3.1	Diagrama de forças no registro. . . . .	18
3.2	Servomotor MG995. . . . .	19
3.3	Diferentes ciclos de trabalho de um sinal PWM. . . . .	19
3.4	Relação entre a largura do pulso e a posição do servomotor [Electronics, 2019]. . . . .	20
3.5	Conexões entre servomotor e Raspberry Pi. . . . .	21
3.6	Esboços das estruturas de suporte. Estrutura de suporte para o servomotor (a) e em (b) um esboço do acoplamento das engrenagens. . . . .	22
3.7	Medições para modelagem em 3D do suporte. Fotografia da medição lateral da peça (a) e em (b) fotografia da medição do eixo. . . . .	23
3.8	Tabela de preços do LAB2P. . . . .	24
3.9	Diversas vistas do processo de modelagem. Em (a) é representado uma modelagem simplificada, em (b) têm-se a adição do suporte de fixação na tubulação, em (c) é representado as guias de rigidez e o envólucro do servomotor e por fim em (d) é adicionado as engrenagens no servomotor. . . . .	25
3.10	Vista da modelagem completa. . . . .	25
3.11	Fotografia da estrutura acoplada na planta didática. . . . .	26
3.12	Capa para o Raspberry Pi e Arduino. . . . .	27
3.13	Diagrama de blocos da infraestrutura. . . . .	27
3.14	Circuito esquemático do sistema de coleta. . . . .	28
3.15	Circuito esquemático entre Raspberry Pi e Arduino. . . . .	29
3.16	Visão geral das conexões. . . . .	29
3.17	Fotografia da planta didática com o sistema acoplado. . . . .	30
3.18	Diagrama de blocos do sistema. . . . .	31
3.19	Captura de tela do <i>Script</i> do servidor em execução. . . . .	32
3.20	Janela de autenticação de usuário. . . . .	32
3.21	Janela de <i>status</i> e botões de navegação. . . . .	33
3.22	Interface de usuário com experimento em operação. . . . .	33

# Lista de Tabelas

2.1	Especificações técnicas do modelo Raspberry Pi Revision 2 - Element14. . . . .	10
2.2	Descrição dos pinos do Raspberry Pi Rev 2.0 - Element 14. . . . .	11
3.1	Principais funções da biblioteca PIGPIO. . . . .	20
3.2	Tabela de descrição dos <i>scripts</i> do servidor. . . . .	31
3.3	Tabela de descrição dos <i>scripts</i> do cliente. . . . .	31
A.1	Descrição das peças impressas. . . . .	40

# Lista de Códigos

3.1	Código para efetuar giro de $n$ graus. . . . .	21
-----	--	----

# Sumário

<b>Resumo</b>	<b>1</b>
<b>Lista de Figuras</b>	<b>3</b>
<b>Lista de Tabelas</b>	<b>3</b>
<b>Lista de Códigos</b>	<b>4</b>
<b>1 Introdução</b>	<b>7</b>
1.1 Motivações . . . . .	7
1.2 Objetivo . . . . .	8
1.3 Trabalhos correlacionados . . . . .	8
1.4 Organização do documento . . . . .	8
<b>2 Materiais utilizados no projeto</b>	<b>9</b>
2.1 Sensor de vazão YF-S201 . . . . .	9
2.2 <i>Kit</i> de desenvolvimento Arduino . . . . .	9
2.3 Série de computadores Raspberry Pi . . . . .	10
2.4 Interfaces gráficas para Arduino . . . . .	12
2.5 Servomotor . . . . .	13
2.6 Modelagem e impressão em 3D . . . . .	14
2.7 Ferramentas para acesso remoto dos computadores . . . . .	15
2.8 Câmera de vídeo IP e aplicativo de controle . . . . .	15
<b>3 Metodologia e abordagens</b>	<b>17</b>
3.1 Escolha do dispositivo para controle de vazão . . . . .	17
3.2 Controle eletrônico de servomotor . . . . .	19
3.3 Elaboração do conceito da conexão entre servomotor e registro . . . . .	22
3.4 Modelagem em 3D das peças . . . . .	23
3.5 Modelo de capa para Raspberry Pi e Arduino . . . . .	26
3.6 Novo sistema proposto . . . . .	27
3.6.1 Estrutura do sistema proposto . . . . .	27
3.6.2 Programas de computador para gerenciamento do sistema . . . . .	30
3.6.3 Demonstrações no uso da interface de usuário . . . . .	32
3.7 Discussão de erros e falhas de implementação . . . . .	34
<b>4 Conclusão e trabalhos futuros</b>	<b>35</b>
4.1 Conclusão . . . . .	35
4.2 Trabalhos futuros . . . . .	35

**Referências Bibliográficas**

**35**

**A Tabela de peças**

**39**

# Capítulo 1

## Introdução

A planta tem como objetivo simular, em laboratório, regimes de chuva intermitentes e/ou contínuos para estudos de contenção de alagamentos e enchentes. Utilizando um sensor de medição de vazão d'água aliado à plataforma Arduino e uma câmera HD com transmissão sobre IP, o grupo foi capaz de trazer a experiência do laboratório para a sala de aula. Dentre as funções e componentes do sistema, o Arduino [Arduino, 2020] é o responsável pela coleta de dados. Esses dados primeiramente são reunidos em um computador, para posteriormente serem dispostos em gráficos e tabelas em um outro computador com acesso à *Internet*. Tais dados são exibidos através do *software* TeamViewer. O sistema otimiza e padroniza a coleta de dados de vazão de água em dois reservatórios, assim através de uma experiência visual e didática de qualidade, os alunos podem assistir à exibição em tempo real em um outro ambiente que acomoda por inteiro toda a classe.

O Estudo do presente relatório é a automatização da planta didática do Laboratório LaDiSan [LaDISan, 2019] já em funcionamento. O objeto inovador está na complementação da experiência remota com a utilização do Raspberry Pi [Foundation, 2020] e uma nova interface. As novas funções propostas são controlar remota e automaticamente a admissão de água na planta de modo que seja possível desenhar um padrão de vazão de acordo com a especificação do usuário. O objetivo destas novas implementações é possibilitar que o usuário esteja em posicionamento remoto seja na condição de operador ou observador. Além disso, buscamos apresentar alternativas e mostrar um roteiro para utilização remota de um laboratório de um aspecto prático e didático.

### 1.1 Motivações

O grupo PET-Tele [Grupo PET-Tele, 2021] formado por alunos de graduação da Universidade Federal Fluminense já há algum tempo desenvolve projetos com a plataforma de prototipagem Arduino [Arduino, 2020]. Esses projetos resultaram em materiais didáticos, cursos práticos, diversos experimentos educativos e criação de disciplina optativa do currículo de Engenharia de Telecomunicações.

Nosso intuito com essas atividades é servir de ferramenta educacional-pedagógica para melhorar o processo de aprendizagem tanto dos alunos da escola de Engenharia quanto de pessoas externas a universidade. A filosofia do programa PET é trabalhar a inquietação do integrante para desafiar o estado atual e realizar mudanças.

O Laboratório de Drenagem, Irrigação e Saneamento Ambiental (LaDISan) [LaDISan, 2019] fornece aos alunos dos cursos de Engenharia de Recursos Hídricos e Engenharia Ambiental experiência prática e ponto de apoio no aprendizado de sistemas de drenagem, irrigação e

saneamento ambiental. Esse mesmo laboratório é um espaço pequeno e não acomoda ao mesmo tempo todos os discentes da disciplina. Neste trabalho a motivação é fundamentada nos alunos, somos impulsionados a ajudar e melhorar a estrutura que os cercam, alterando, modernizando e propondo novas ferramentas e sistemas. Com a experiência relatada no uso do Arduino, e um processo de inovação orgânico sempre incentivado dentro do grupo, surge a introdução de uma nova plataforma de desenvolvimento, o computador Raspberry Pi [Foundation, 2020].

## 1.2 Objetivo

As novas implementações têm por objetivo a automatização da planta didática do Laboratório LaDiSan [LaDISan, 2019] por meio da implementação de novas funções para controlar de maneira remota e automática a admissão de água de modo que seja possível desenhar um padrão de vazão de acordo com a especificação do usuário, permitindo a exibição e controle em tempo real em um ambiente diferente e fisicamente distante ao laboratório.

## 1.3 Trabalhos correlacionados

Esta planta anteriormente já foi objeto de estudo do grupo, tendo como resultados: o artigo **Implantação de um Sistema Automático para Medição de Fluxo em uma Planta Didática Usando um *kit* Arduino e o Sensor Water Flow G1/2** [Coelho et al., 2017] no COBENGE 2017 [ABENGE, 2020]; o artigo **Realização de Aula Prática Remota a Partir de Laboratório Equipado com Modelo Físico Sobre Detenção de Água de Chuva** [Paiva et al., 2018] no COBENGE 2018; e a monografia **Uso de Sensores e Acesso Remoto para a Realização de Aula Prática Sobre Reservatórios de Detenção Aplicados à Drenagem urbana** [Paiva and Milhomem, 2018]. O atual relatório é uma continuação desses estudos e proporciona uma atualização e melhoria na estrutura.

## 1.4 Organização do documento

No presente documento buscamos apresentar um relatório que é fruto de uma nova implementação de atuação e acesso remoto a uma planta didática no Laboratório de Drenagem, Irrigação e Saneamento Ambiental LaDISan [LaDISan, 2019]. Os seguintes capítulos possuem objetivos de apresentação e definição de cada um dos dispositivos, ferramentas, tecnologias e métodos utilizados, que serão descritos em termos de *hardware* ou *software*. Busca-se oferecer ao leitor uma compreensão das funcionalidades e funcionamento de cada parte individual e/ou conjuntamente.

# Capítulo 2

## Materiais utilizados no projeto

### 2.1 Sensor de vazão YF-S201

Um sensor é um dispositivo que responde à um estímulo físico/químico de uma maneira específica. Tal estímulo pode ser associado, através de relações matemáticas, a uma grandeza física que, por sua vez, serve como informação para medição e/ou monitoramento.

Para medir o fluxo de água foi utilizado o sensor de vazão YF-S201b [YIFA, 2020], mostrado na Figura 2.1. Ele foi instalado em linha com o cano, para medir a quantidade de água que o transpassa. Em sua estrutura, há um válvula em formato de catavento (uma hélice), com um ímã acoplado. Ela trabalha em conjunto com um sensor de Efeito Hall, que envia um pulso de tensão a cada revolução completa da hélice. Um sensor de Efeito Hall é um transdutor [Wikipedia Members, 2021] que, sob a exposição de um campo magnético, responde com uma variação de tensão na sua saída. Através da quantidade de pulsos gerados, é possível mensurar a vazão de água na tubulação.



Figura 2.1: Sensor de vazão YF-S201.

### 2.2 *Kit* de desenvolvimento Arduino

O Arduino é uma plataforma para prototipagem eletrônica com *hardware* livre. É constituído de uma placa de circuito impresso (PCI), um microcontrolador e diversos componentes eletrônicos. Possui uma IDE (Ambiente de Desenvolvimento Integrado) para programação, que se utiliza de uma linguagem similar às linguagens C e C++. Tais características tornam mais fácil e barata a montagem de diversos tipos de circuitos e projetos de eletrônica.

As placas Arduino existem em diversos modelos e versões e são conectadas a um computador para a programação de seu microcontrolador. Todas possuem algum tipo de conexão USB (*Universal Serial Bus*) [USB Implementers Forum, 2020] seja do tipo B, C ou micro USB. Outra função para a porta USB é a transmissão de dados coletados pelos sensores do Arduino para um computador.

Neste projeto foi utilizado um Arduino Uno R3 como mostrado na Figura 2.2.

A premissa básica para a utilização do Arduino em um projeto de aquisição de dados é usá-lo conectado a sensores, que transformarão as variáveis do ambiente, normalmente analógicas, em sinais elétricos, que, por sua vez, serão recebidos pelos pinos da placa.

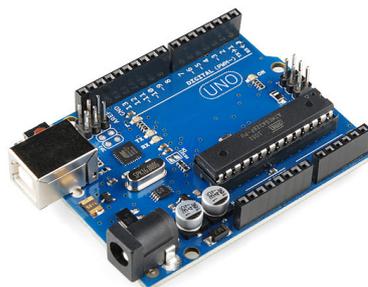


Figura 2.2: Arduino Uno R3.

## 2.3 Série de computadores Raspberry Pi

O Raspberry Pi é uma série de computadores de placa única e tamanho reduzido, que recebem a denominação SoC (*System on Chip*) [Members, 2019b] à qual são conectados os seguintes dispositivos: monitor, teclado, e mouse.

Desenvolvido no Reino Unido pela Raspberry Pi Foundation tendo, como principais objetivos, contribuir para inclusão digital, promoção de ensino básico em ciência da computação e empoderamento social. Uma alternativa de ensino com baixo custo para escolas e estudantes [Members, 2019a]. A Tabela 2.1 apresenta as especificações técnicas do modelo Raspberry Pi Revision 2 - Element 14, utilizado pelo grupo PET-Tele.

Chip	Broadcom BCM2835 SoC Full HD Processador de Aplicações Multimídia
CPU	700 Mhz ARM1176JZ-F Processador de Baixa Potência de aplicações
GPU	Dois Núcleos, VideoCore IV, Co-Processador de Multimídia
Memória	512MB SDRAM
Ethernet	Onboard 10/100 Ethernet com conector RJ-45
USB 2.0	Dois Conectores USB 2.0
Saída de Vídeo	HDMI e Composição RCA (PAL e NTSC)
Saída de Áudio	Conector 3.5 mm ou HDMI
Armazenamento	SD, MMC, SDIO Card Slot
Dimensões	8.6 cm X 5.4 cm X 1.7 cm

Tabela 2.1: Especificações técnicas do modelo Raspberry Pi Revision 2 - Element14.

Na Figura 2.3 é mostrado uma fotografia do modelo. Dentre os 25 Pinos presentes neste Raspberry, 17 podem ser usados como entradas ou saídas de uso geral, 5 como terminais comuns (GND), 2 como fontes de tensão de valor +5V e 2 como fontes de tensão de valor +3.3V.

A Tabela 2.2 reúne uma descrição de todos os pinos.

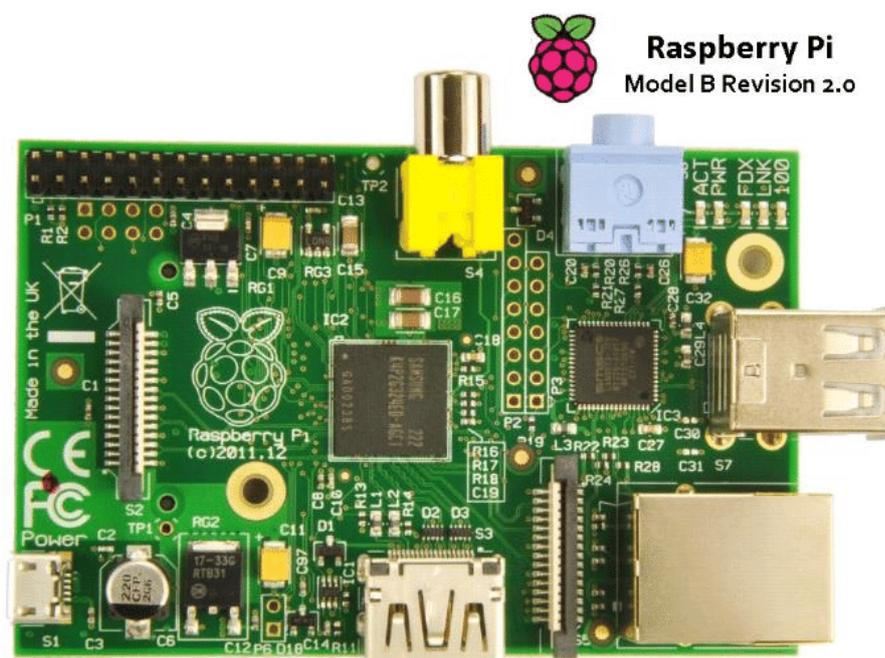


Figura 2.3: Raspberry Pi Revision 2.0.

N	Descrição	N	Descrição
1	Saída de +3.3V	14	Terminal Comum GND
2	Saída de +5V	15	GPIO22
3	GPIO2   Pull-Up   I2C-SDA	16	GPIO23
4	Saída de +5V	17	3V3
5	GPIO3   Pull-Up   I2C-SCE	18	GPIO24
6	Terminal Comum GND	19	GPIO10   SPI   MOSI
7	GPIO4	20	Terminal Comum GND
8	GPIO14   UART   TXD	21	GPIO9   SPI   MISO
9	Terminal Comum GND	22	GPIO25
10	GPIO15   UART   RXD	23	GPIO11   SPI   CLK
11	GPIO17   UART-RTS	24	GPIO8   SPI   CE0
12	GPIO18   PWM	25	Terminal Comum GND
13	GPIO27	26	GPIO7   SPI   CE1

Tabela 2.2: Descrição dos pinos do Raspberry Pi Rev 2.0 - Element 14.

## 2.4 Interfaces gráficas para Arduino

A placa de desenvolvimento Arduino é programável através de seu ambiente de desenvolvimento integrado (IDE) [Codecademy, Inc., 2020], que possui código aberto e encontra-se disponível em arduino.cc. Nesta ferramenta podemos escrever, compilar, carregar e inspecionar códigos. Além disso, possui a característica de se comunicar em tempo real com o Arduino para receber ou enviar informações via Barramento Serial Universal (USB) [USB Implementers Forum, 2020]. Esse tipo de comunicação é interfaceado ao usuário por meio do monitor de porta serial *Serial Monitor* e o plotador serial *Serial Plotter*. O plotador serial mostra na tela uma interpolação de dados da variável escolhida e o monitor serial mostra na tela cada valor dessa variável. Não é possível utilizar os dois recursos ao mesmo tempo.

Buscando soluções alternativas para utilização em conjunto desses recursos o grupo se depara com uma iniciativa brasileira da Universidade Federal de Santa Catarina (UFSC) o programa ViewDuino [Girardi, 2017b] do projeto de ensino TeachDuino [Girardi, 2017a]. O programa fornece ferramentas de coleta e análise de dados oriundos de um Arduino. Possui diversas funcionalidades: Ajustar retas, montar histogramas, mostrar gráficos durante *loops* de coleta, entre outras. Em sua versão de 06 de Junho de 2017, os dados podem ser recebidos por conexão *Bluetooth*.

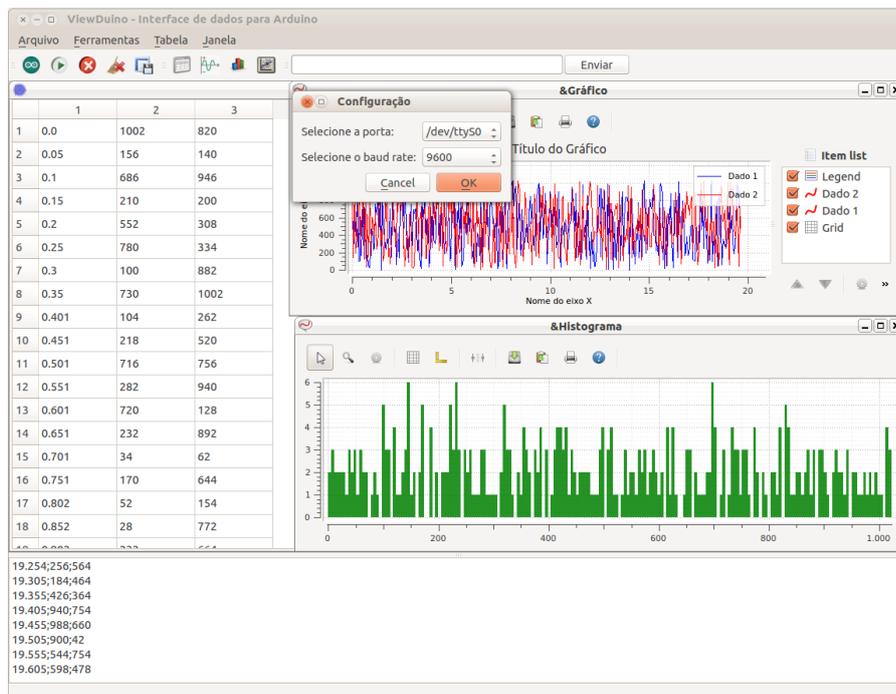


Figura 2.4: ViewDuino: tela inicial. Versão: set/2015.

Na Figura 2.4 é mostrado a janela de inicialização do ViewDuino. Sua interface é completamente desenvolvida em Python. Logo, é facilmente transportada entre sistemas operacionais com Linux, MacOS ou Windows. Os requisitos de funcionamento da interface são instalações de bibliotecas de Python em seu sistema operacional conforme especificado na *webpage* do ViewDuino.

Uma segunda alternativa ao ViewDuino, com funções similares, é a aplicação *SerialPlot - Realtime Plotting Software* [Özderya, 2017], do engenheiro eletrônico Hasan Yavuz Özderya. Essa aplicação encontra-se disponível para sistemas operacionais Windows e Linux.

## 2.5 Servomotor

O servomotor é um dispositivo eletromecânico utilizado para gerar movimentos precisos a partir de sinais elétricos. No desenvolvimento do projeto, escolhemos utilizar esse tipo de motor pelo sistema de controle integrado de fácil utilização com Arduino. A ilustração da Figura 2.5 provê uma vista interna da peça para melhor compreensão.

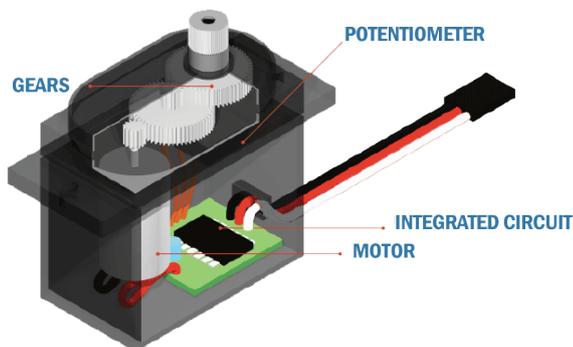


Figura 2.5: Desenho ilustrativo do servomotor.

Para controlar um servomotor com um Arduino, por exemplo, basta alimentar os terminais de energia com uma fonte de tensão contínua de +5V (se for separada, lembrar de utilizar o mesmo aterramento para a placa e a fonte). Observe a carga será aplicada ao servomotor, pode ocorrer da saída de +5V do Arduino não suprir a demanda de potência exigida. Em seguida, com o servomotor alimentado, conectar um pino de *Pulse Width Modulation* (PWM) [Pinckney, 2006] do Arduino ao terminal de controle do motor. A Figura 2.6 ilustra a montagem do circuito com a utilização da saída +5V do Arduino. Para acionar o motor, basta enviar o sinal PWM a partir do pino PWM conectado. As características desse sinal PWM serão explicadas na Seção 3.2.

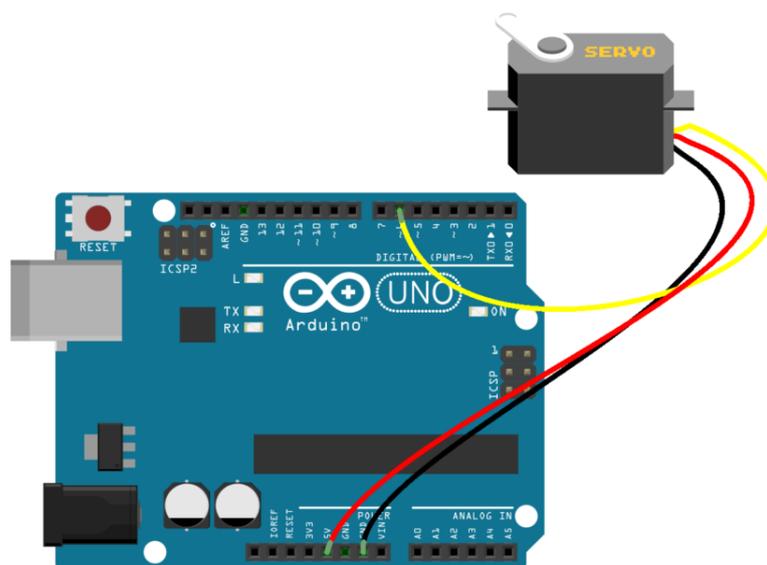


Figura 2.6: Desenho elaborado no *software* TinkerCad para ilustrar a conexão entre um Arduino e um servomotor.

## 2.6 Modelagem e impressão em 3D

A modelagem e impressão em 3D é um recurso bastante utilizado quando se precisa de uma prototipagem rápida. É uma forma de tecnologia de fabricação em que um objeto tridimensional é criado por meio da deposição de sucessivas camadas de algum tipo de material, polimérico. Isso confere aos desenvolvedores de produtos a possibilidade de, em um processo simples, imprimir diversos componentes com diferentes propriedades físico-mecânicas.

A ferramenta escolhida para fazer o desenho da peça mostrada na Figura 2.7 foi o aplicativo TinkerCad [Autodesk, Inc., 2019], com acesso gratuito, disponibilizado via navegador pela AutoDesk Inc. Ao total foram impressas duas braçadeiras, cada uma com partes superior e inferior, um invólucro para o servomotor também com partes superior e inferior e duas engrenagens. As oito peças permitem ao servomotor fixação na planta didática do LaDISan, posicionamento correto e transmissão de movimento do eixo do motor para o registro. Além dessas peças duas hastes metálicas, para conferir maior rigidez, foram utilizadas. A Tabela A.1, presente no Apêndice A, detalha a função de cada peça.

No processo de confecção das peças, foram utilizadas as impressoras 3D do laboratório LAB2P [LAB2P, 2019] da UFF. Todas as ferramentas utilizadas na modelagem e confecção foram de extrema importância para a criação de partes únicas e sob medida. Tais partes possuem funções únicas de acoplamento e transmissão de movimento fundamentais para o funcionamento proposto neste relatório. Suas especificações são de difícil obtenção em um mercado varejo convencional. A planta didática com seu acoplamento original é mostrada na Figura 2.8a. Ao lado a Figura 2.8b mostra a incorporação das peças na planta.

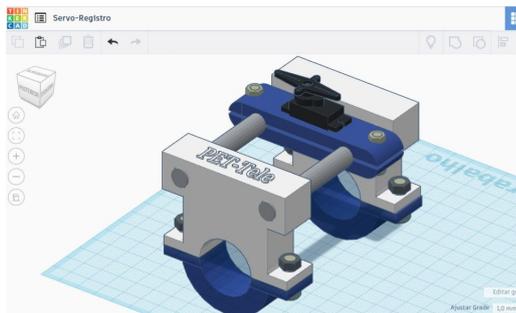
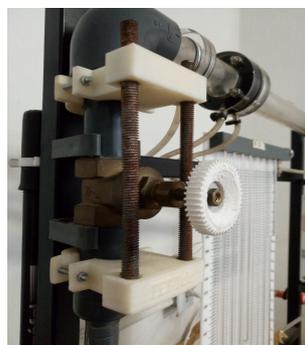


Figura 2.7: Exemplo de modelo em 3D, desenhado no *software* TinkerCad.



(a) Condição inicial frente.



(b) Suporte Acoplado.

Figura 2.8: Troca do acabamento destinado ao acionamento manual do registro. Fotografia da condição inicial da frente (a) e em (b) fotografia do suporte acoplado.

## 2.7 Ferramentas para acesso remoto dos computadores

O termo acesso remoto pode ser definido como o acesso de um computador ou sistema de computador a partir de uma localização à distância por meio de uma rede de conexão. As conexões de acesso remoto não se limitam a conexões via *Internet*, redes locais ou redes locais sem fio também podem ser utilizadas. O termo acesso remoto atualmente é comumente utilizado para um serviço mais específico, o acesso remoto de recursos de *software*. A popularidade do acesso remoto de recursos é devida ao grande número de empresas que fornecem serviços de reparos virtualmente. Não há necessidade de profissionais de TI se locomoverem até o computador do cliente. Toda a atuação desses profissionais pode ser feita através dos *softwares* de Suporte remoto. Neste projeto estamos utilizando dois desses softwares, o TeamViewer e o VNC Connect. Ambos estão sob licença proprietária das empresas TeamViewer Group e RealVNC Limited, respectivamente, que oferecem os aplicativos de forma gratuita para uso pessoal.

Em um primeiro momento deste projeto, foram utilizados os aplicativos TeamViewer e VNC Connect para compartilhamento de tela e transferência de arquivos. Eles permitem a conexão com um computador remoto, a partir de qualquer lugar do mundo, empregando a infraestrutura da *Internet*. Na função de compartilhamento de tela é possível prover ao usuário controle do computador remoto, como se estivesse sentado à sua frente. Ambos os *softwares* podem ser utilizados comutativamente, pois apresentam funções similares. Neste projeto a escolha de um em relação ao outro é entendida como uma escolha substituta e não apresenta prejuízos. O PET-Tele já desenvolveu um material didático para instalação e utilização do TeamViewer [Miranda and de la Vega, 2018].

Para fins de clareza o termo computador *host* será empregado neste relatório descritivo para referência à máquina acessada remotamente que se encontra presente no laboratório LaDISan. O termo computador cliente ou só “cliente” será utilizado para referência à máquina que está presente em sala de aula distante ao laboratório.

Em uma outra parte do projeto foi utilizado o protocolo *Secure Shell* (SSH) este não possui interface gráfica como os outros, e portanto, está restrito ao terminal de usuário do sistema operacional utilizado. O SSH é um protocolo de rede criptográfico destinado à operação de serviços de rede de forma segura, sobre uma rede insegura. É normalmente usado na operação de *login* em uma máquina remota e na execução remota de comandos, mas também suporta tunelamento de rede [Teleco, ] e redirecionamento de portas TCP [Duke et al., 2015]. Ele pode transferir arquivos usando os protocolos *SSH File Transfer* (SFTP) [Ylonen and Lehtinen, 2001] e *Secure Copy* (SCP) [Suehring and Salowey, 2005].

Uma vez que o sistema proposto necessita de acesso remoto, essas ferramentas desempenham, juntas, e em sentido amplo, o papel de canais de comunicação por onde vão fluir as informações.

## 2.8 Câmera de vídeo IP e aplicativo de controle

Como fundamentado anteriormente, um dos objetivos da atividade de aula remota é oferecer, tanto ao professor quanto aos alunos, novas ferramentas para ensino e uma experiência completa que simule sua presença no laboratório. Portanto, o dispositivo câmera de vídeo é essencial para o acompanhamento visual do experimento remoto. A função da câmera de vídeo é captar e acompanhar cada passo/processo do experimento. O corrente projeto utiliza, em sua maioria, a *Internet* como meio de transmissão. Logo, é interessante que a transmissão de vídeo seja feita por ela também. Uma vez que, atualmente as câmeras de vídeo com transmissão via IP

(*Internet Protocol*) [RFC Editors, 2020] apresentam baixo custo, uma câmera deste tipo foi utilizada no sistema implantado.

Para configuração do dispositivo, foram seguidos os seguintes passos:

1. Foi criada uma conta no aplicativo da Yoosee [Yoosee, 2017], para IOS ou Android, e ela foi vinculada a um endereço de *E-mail*.
2. Foi dada preferência a uma configuração via Wi-Fi, pois, mesmo que, posteriormente, venha-se a utilizar a câmera em conexão cabeada, esse é um processo mais simples de configuração.
3. Após a vinculação do dispositivo com a conta de *E-mail*, tornou-se possível utilizar qualquer tipo de conexão.

Após a instalação e configuração do aplicativo Yoosee, é possível acessar a aplicação CMS [Yoosee, 2017] da Yoosee, para sistemas operacionais Windows, conforme mostrado na Figura 2.9. O CMS é uma aplicação que permite gerenciar, selecionar, gravar e movimentar, a câmera IP vinculada. A Figura 2.10 mostra a principal interface da aplicação. Todos os procedimentos são análogos para as versões de dispositivos Android e IOS.

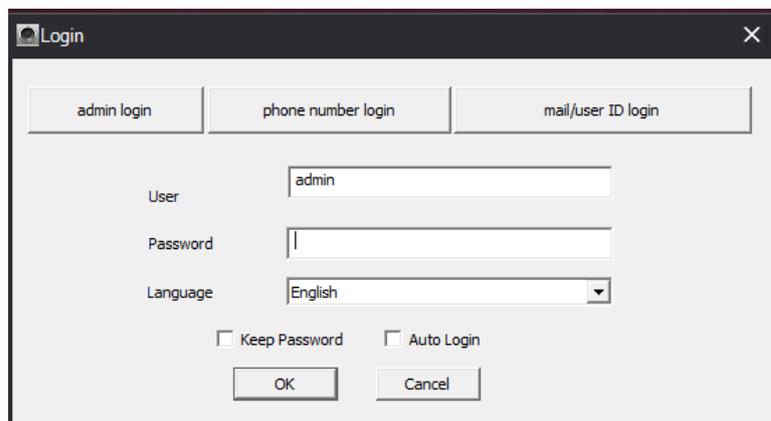


Figura 2.9: Janela de *login* da aplicação CMS para Windows.

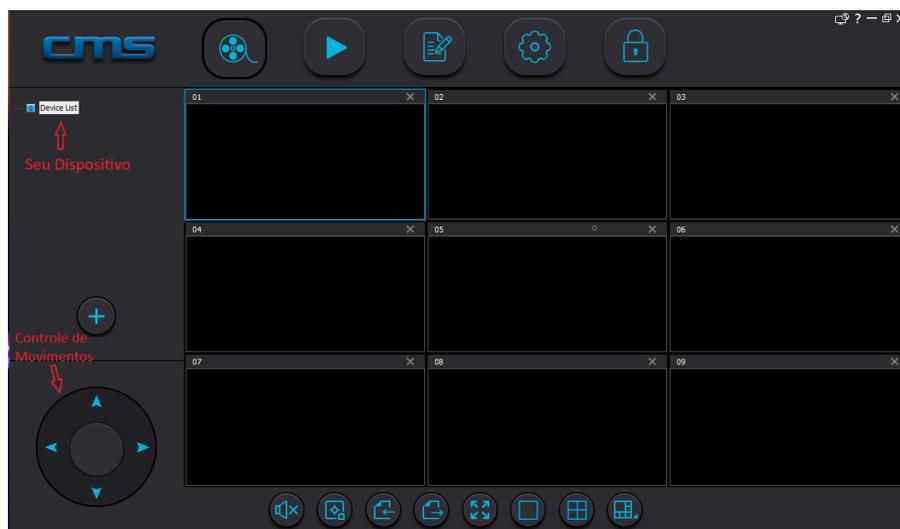


Figura 2.10: Janela principal da aplicação CMS para Windows.

# Capítulo 3

## Metodologia e abordagens

### 3.1 Escolha do dispositivo para controle de vazão

Um dos pontos cruciais do projeto é o controle de vazão da água que passa pelo cano. Existem diversas soluções possíveis, sejam elas hidráulicas, mecânicas ou elétricas. O objetivo é encontrar um mecanismo que possa ser controlado remotamente, via *software*. A princípio, podem-se cogitar duas alternativas. A primeira alternativa é a atuação na vazão da água a partir da inserção de uma bomba DC, em linha ao encanamento. Teoricamente uma abordagem como essa seria possível pela utilização de um controle PWM sobre o motor. A segunda abordagem seria aproveitar o registro de admissão de água já existente na tubulação e atuar diretamente sobre ele, empregando um motor com controle eletro/eletrônico. Para o projeto, foi verificado que a primeira abordagem seria complexa e onerosa, pois a inserção da bomba à planta resultaria em maior número de desvios de tubulação sendo necessária a ocupação de mais espaço do laboratório. Outro ponto observado é o fato de que não seria aproveitado parte do equipamento que contém uma bomba d'água comum. Dessa forma, escolheu-se utilizar a segunda abordagem, na qual seria usado um motor com controle eletro/eletrônico para girar o registro existente na planta hidráulica, simulando o movimento de abrir/fechar desempenhado por uma pessoa.

Com a abordagem definida, surgiram questões sobre como deveria ser a atuação do motor, tais como: qual tipo de motor elétrico a ser utilizado; como realizar o controle do motor eletronicamente; e qual torque constante que ele deveria desempenhar, a ponto de girar o registro com boa adequação de movimento. Todas essas questões foram problematizadas em uma pesquisa para verificar se um servomotor seria o tipo adequado de abordagem. Das soluções e alternativas propostas, pode-se destacar a característica do motor apresentar um sistema de controle por PWM, que o torna integrável com qualquer placa de prototipagem já utilizada pelo grupo. Porém, retornando a atenção para os outros problemas, ainda resta uma questão crucial para o acionamento: o torque desempenhado pelo servomotor deve se sobrepor ao torque de resistência mecânica do registro, o que é matematicamente dado por

$$\tau_{servo} \geq \tau_{arrasto} . \quad (3.1)$$

Para descobrir o torque de arrasto do registro, foram feitas medições com a utilização de um paquímetro e um dinamômetro, preso à borda do registro por um barbante. Para isso, foi utilizada a configuração representada na Figura 3.1. Com simplificações de modelagem, é possível medir o raio externo do registro e a força tangencial necessária para colocar o registro em movimento. Com esses dois valores, encontra-se o torque mínimo necessário para o servomotor atuar, dado por

$$\tau_{arrasto} = F_{arrasto} \cdot R , \quad (3.2)$$

onde  $\tau_{arrasto}$  é o torque de arrasto,  $F_{arrasto}$  é força tangencial de arrasto no registro e  $R$  é o raio externo do registro.

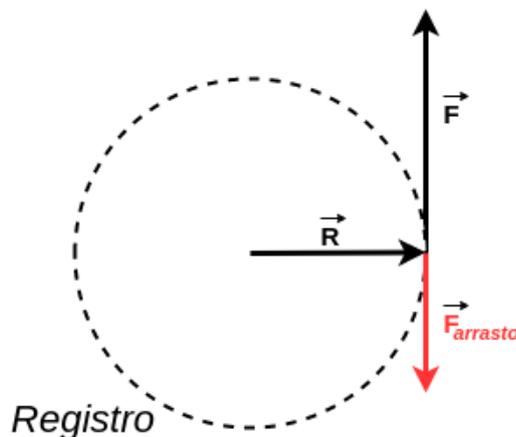


Figura 3.1: Diagrama de forças no registro.

Após fazer medidas da distância externa até o eixo e suas respectivas forças de arrasto correspondentes, consegue-se chegar a um torque de arrasto de  $\tau_{arrasto} \approx 4,5 \text{ kgf} \cdot \text{cm}$ . Seguindo o critério da Equação (3.1), obtém-se

$$\tau_{servo} > 4,5 \text{ kgf} \cdot \text{cm} . \quad (3.3)$$

A partir desse ponto, deu-se início a etapa de busca por servomotores capazes de trabalhar acima dessa restrição. Como resultado, foi encontrado o servomotor MG995 [Tower-Pro, 2019], mostrado na Figura 3.2, que, além de satisfazer os critérios, possui engrenagens metálicas, o que representa maior durabilidade e resistência. Dentre as informações úteis para o projeto, encontradas no *datasheet* do servomotor, destacam-se:

- Peso: 55 g.
- Dimensões aproximadas: 40.7 x 19.7 x 42.9 mm.
- *Stall torque*: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V).
- Velocidade de operação: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V).
- Tensão de operação: 4.8 V a 7.2 V.
- *Dead bandwidth*: 5  $\mu\text{s}$ .
- Estável e a prova de choque mecânico.
- Faixa de temperatura: 0° C – 55° C.

A partir desses dados, pode-se dizer que o servomotor era a opção disponível que melhor atendia às especificações para o projeto. O fato de ter um torque de 8.5 kgf·cm é o principal para destacá-lo como melhor escolha. Esse torque é suficientemente alto para realizar o movimento proposto, dispensando assim o uso de qualquer tipo de caixa redutora e/ou outros esforços para funcionamento dentro das especificações de carga propostas.

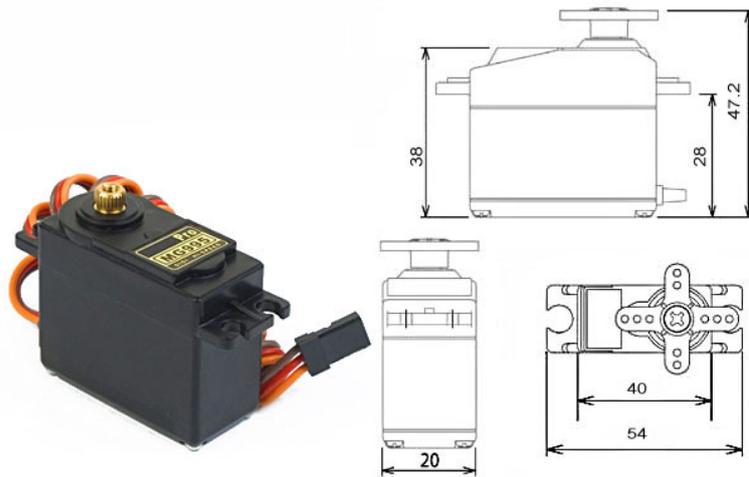


Figura 3.2: Servomotor MG995.

### 3.2 Controle eletrônico de servomotor

Servomotores são controlados por meio de pulsos elétricos de tamanhos variáveis do tipo PWM (*Pulse Width Modulation*) [Pinckney, 2006]. Nesta técnica, existem um pulso de largura variável (entre um valor mínimo e um valor máximo) e uma cadência de repetição deste pulso, definida por um período ou uma frequência. Um sinal PWM pode ser parametrizado em função da frequência da onda retangular e do ciclo de trabalho dessa onda. O ciclo de trabalho (*duty cycle*) é um valor percentual da duração do pulso em nível alto em relação à duração de um período. A Figura 3.3 apresenta, de forma gráfica, diferentes ciclos de trabalho de um sinal PWM.

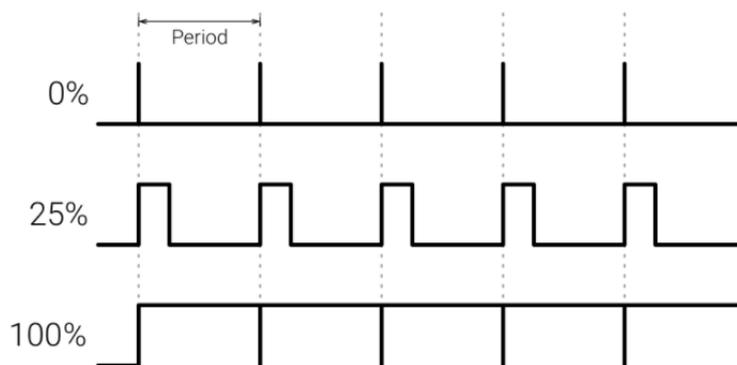


Figura 3.3: Diferentes ciclos de trabalho de um sinal PWM.

Um servomotor normalmente pode rotacionar até  $90^\circ$  em um determinado sentido de rotação. Se forem considerados os sentidos horário e anti-horário, pode-se dizer que ele é capaz de rotacionar de  $180^\circ$ . A posição neutra do servomotor é definida como a posição em que o servomotor tem o mesmo potencial de rotação para o sentido horário e anti-horário, conforme mostrado na Figura 3.4. A maioria dos servomotores, incluindo o utilizado nesse projeto, espera um pulso a cada 20 ms e a largura do pulso irá determinar até onde o motor deve rotacionar. Por exemplo, um pulso de 1.5 ms irá ocasionar uma rotação até a posição de  $90^\circ$  denominada posição neutra, um pulso menor que 1.5 ms ocasiona um giro no sentido anti-horário em encontro a posição  $0^\circ$  e um pulso maior que 1.5 ms ocasiona um giro no sentido horário de encontro a posição  $180^\circ$  [Electronics, 2019].

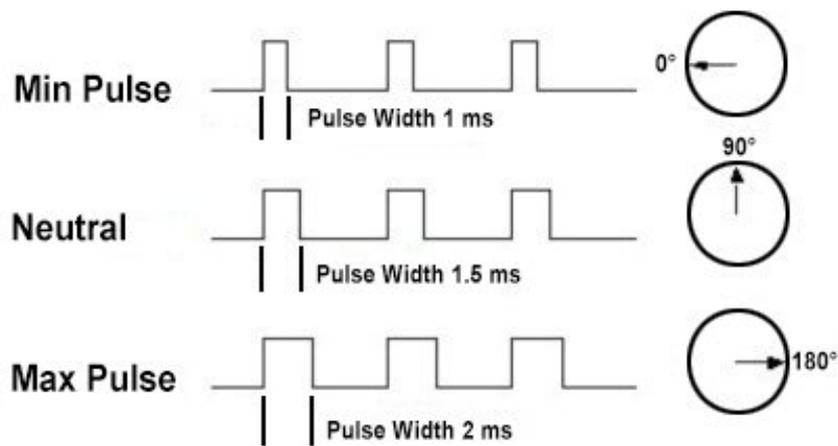


Figura 3.4: Relação entre a largura do pulso e a posição do servomotor [Electronics, 2019].

Para utilizar a técnica PWM no Raspberry Pi, os processos adotados são semelhantes aos aplicados ao Arduino. Existem pequenas diferenças em algumas especificações da plataforma. Além disso, enquanto o Arduino trabalha com sua linguagem própria, no Raspberry Pi podemos utilizar uma variedade de outras linguagens. Nesse projeto foi adotado Python por sua fácil integração modular. No Arduino modelo UNO a frequência PWM é 976 Hz. A frequência de PWM presentes em placas Raspberry Pi de modelos mais recentes pode ser um valor escolhido de 10 Hz a 8 kHz. Além disso, duas bibliotecas da linguagem Python foram pesquisadas para utilização no controle do servomotor: a PIGPIO [joan2937, 2019] e a RPI.GPIO [Croston, 2019]. Ao final de diversos testes, notou-se um melhor desempenho e uma quantidade menor de erros ao se utilizar a biblioteca PIGPIO. A Tabela 3.1 mostra as principais funções da biblioteca PIGPIO.

<code>pigpio.pi</code>	Inicia a conexão com a PigPio
<code>stop</code>	Encerra a conexão com a PigPio
<code>set_mode</code>	Define o tipo do pino GPIO
<code>get_mode</code>	Retorna o tipo do pino GPIO
<code>set_pull_up_down</code>	Define/reinicia um resistor Pull up/down no pino GPIO
<code>read</code>	Realiza a leitura do pino GPIO
<code>write</code>	Realiza a escrita no Pino GPIO
<code>set_servo_pulsewidth</code>	Começa/Termina um pulso tipo servo no pino GPIO
<code>get_servo_pulsewidth</code>	Retorna a largura do Pulso definida no pino GPIO
<code>set_PWM_dutycycle</code>	Começa/Termina um pulso PWM no pino GPIO
<code>set_PWM_frequency</code>	Determina uma frequência PWM no pino GPIO
<code>set_PWM_range</code>	Configura o alcance PWM do pino GPIO

Tabela 3.1: Principais funções da biblioteca PIGPIO.

Conforme dito acima e representado na Figura 3.5 a montagem do circuito eletrônico é bastante semelhante à do Arduino. A conexão a ser feita entre Raspberry Pi e o servomotor, utiliza um pino de alimentação, um pino terminal comum e um pino de controle. O pino de controle fornece o sinal de acionamento do servomotor e pode ser qualquer um dentre todos os pinos de propósito geral **GPIO** presentes no Raspberry Pi. No caso da Figura 3.5, foi escolhido o GPIO-4, que é o 4º pino na ordem de contagem cima para baixo e direita para esquerda.

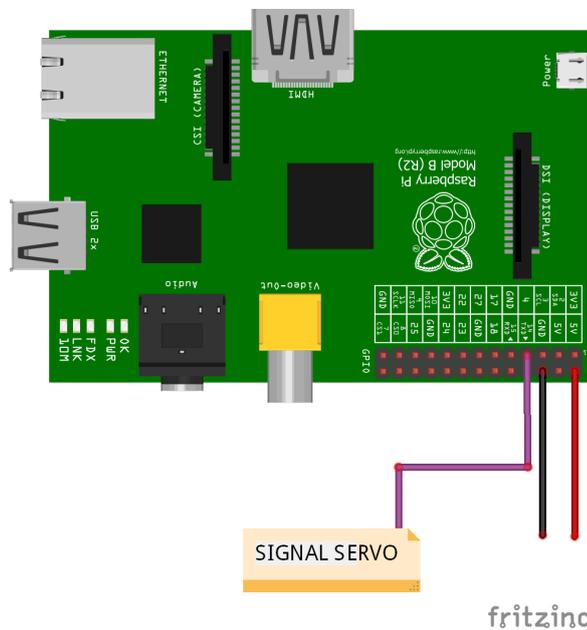


Figura 3.5: Conexões entre servomotor e Raspberry Pi.

Para o primeiro teste, foi desenvolvido o programa de computador mostrado na Listagem 3.1. O objetivo era servir de ponto de partida e para isso necessitava de simplicidade. Assim como no Arduino, o Raspberry possui uma interação entre *software* e *hardware* programável pelo desenvolvedor. Os primeiros momentos são utilizados para atribuir variáveis para guardar números de pinos e definir os pinos como Entrada ou Saída. Em seguida, existe um verificador para ver se a conexão PIGPIO foi estabelecida. Se sim, o programa entra em laço para requisitar ângulos do usuário e realizar o movimento do motor.

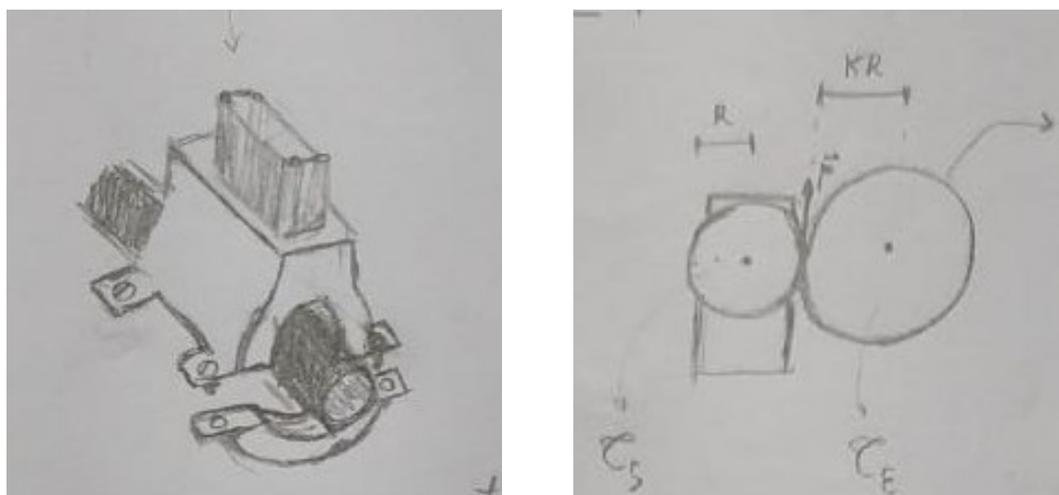
```
#!/usr/bin/env python
import sys, tty, termios, time, pigpio, RPi.GPIO
servo = 4
pi = pigpio.pi()
if pi.connected:
    print('pigpio is conected!!')
else:
    exit()
pi.set_mode(4, pigpio.OUTPUT)
while True:
    angulo = float(input('Digite o angulo:'))
    width = (100/9)*angulo + 500
    pi.set_servo_pulsewidth(4, width)
    time.sleep(1)
pi.stop()
```

Código 3.1: Código para efetuar giro de  $n$  graus.

Nos primeiros testes realizados, o conjunto Motor-Raspberry não apresentou qualquer anomalia. Contudo, a inserção do conjunto no restante do sistema não gerou o efeito esperado na operação de abertura e/ou fechamento do registro d'água. A solução foi simples, porém a detecção do problema foi árdua. O servomotor apresentava respostas diferentes para um mesmo programa de controle, em testes sequenciais ou intermitentes. A princípio, o dispositivo funcionava corretamente, mas, no teste seguinte, sem qualquer alteração, apresentava erro. A causa do problema foi a energização do motor pela própria placa Raspberry Pi, que não supria a alta demanda de corrente por parte do servomotor, quando este tentava produzir alto torque. O problema foi solucionado com uma fonte DC chaveada de maior potência.

### 3.3 Elaboração do conceito da conexão entre servomotor e registro

Após a decisão de acionar o registro por meio de um servomotor, o planejamento precisou evoluir para a forma como seria transferido o movimento do motor para o registro. Os esboços da Figura 3.6 mostram essa etapa do planejamento. Definido o problema, realizou-se diversas reuniões e uma reunião no dia 13 de novembro de 2018 teve possíveis resultados. Na reunião, duas possibilidades foram trazidas à discussão. A possibilidade do servomotor ser acoplado diretamente ao registro, fazendo ambos compartilharem o mesmo eixo de rotação e a possibilidade de transferir o movimento do servomotor a partir de um par de engrenagens, acopladas ao eixo do servomotor e ao eixo do registro respectivamente. Nessa segunda possibilidade, não há necessidade de estabelecer relações de alteração do torque. As duas engrenagens podem ser do mesmo tamanho e acopladas lado a lado. Em reunião, optou-se pelas engrenagens e a colocação lado a lado. Dessa forma, para posicionar o servomotor lado a lado com o registro, foi necessário desenvolver uma estrutura de suporte a ser fixada na tubulação. Essa nova estrutura de suporte é esboçada e apresentada na Figura 3.6.



(a) Esboço do suporte para o servo.

(b) Esboço acoplamento das engrenagens.

Figura 3.6: Esboços das estruturas de suporte. Estrutura de suporte para o servomotor (a) e em (b) um esboço do acoplamento das engrenagens.

Com o conceito inicial pré-estabelecido, foram feitas medidas na estrutura para começar a desenhar um modelo em 3D para o suporte. Dentre as diversas medidas realizadas algumas são mostradas nas Figuras 3.7a e 3.7b.



(a) Primeira medição.



(b) Segunda medição.

Figura 3.7: Medições para modelagem em 3D do suporte. Fotografia da medição lateral da peça (a) e em (b) fotografia da medição do eixo.

### 3.4 Modelagem em 3D das peças

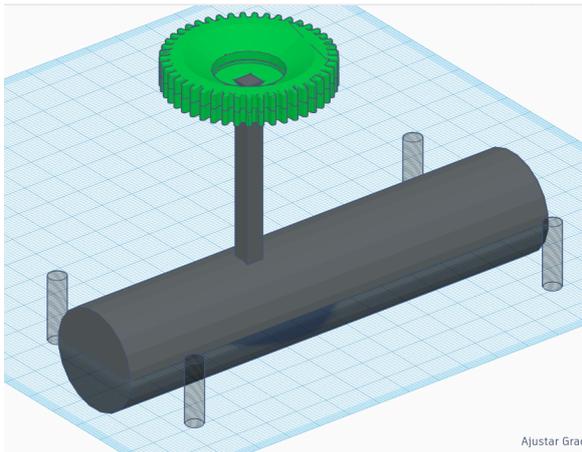
Por diversos fatores, como, por exemplo, custo, facilidade de acesso, versatilidade e o caráter único das peças, optou-se por utilizar a impressão 3D. As peças, em sua maioria, foram impressas no Laboratório Produto e Processo (LAB2P) [LAB2P, 2019], localizado na Escola de Engenharia da Universidade Federal Fluminense. O custo para produção de cada peça foi calculado de acordo com a Tabela da Figura 3.8.

A partir das medições realizadas, foi utilizado o TinkerCad [Autodesk, Inc., 2019], que é uma ferramenta *online* para construção de modelos em ambiente 3D. A Figura 3.9 mostra as diversas etapas do processo de criação no Tinkercad. Na etapa inicial é rascunhado um formato para representação da tubulação e registro. Como resultado, tem-se o primeiro desenho em 3D simplificado conforme mostrado na Figura 3.9a. Essa primeira representação simplificada funciona como uma base modular para o acréscimo seguinte das outras características do modelo. Após o cano estar devidamente representado com suas devidas medidas, parte-se para a modelagem de uma estrutura de suporte para fixação longitudinal ao cano, com a utilização de braçadeiras. A estrutura de suporte é composta por duas braçadeiras idênticas e impressas com o polímero do tipo ABS. Outras duas hastes metálicas cilíndricas foram utilizadas para conferir resistência. Além disso, servem de base para a devida fixação do servomotor conforme mostrado na Figura 3.9c. Uma observação interessante é que a engrenagem verde mostrada na Figura 3.9a foi modelada com maior espessura em relação a sua correspondente em metal para contornar qualquer possível erro de alinhamento no acoplamento com a engrenagem laranja do servomotor, conforme a Figura 3.9d. Na Figura 3.9d, as peças em cor de rosa são divididas em duas partes, superior e inferior, que, juntas, são fixadas entre si por meio de parafusos formando um encaixe central para colocação do servomotor. Com essa montagem também são formados dois orifícios paralelos em direção longitudinal com a tubulação, que servem para o acoplamento do suporte em cor de rosa nas hastes metálicas. Na cor preta, temos a representação do servomotor modelo MG995. Por último, a Figura 3.9d mostra a engrenagem do servomotor, em laranja, que foi acoplada ao servomotor por meio de um parafuso no eixo do motor.

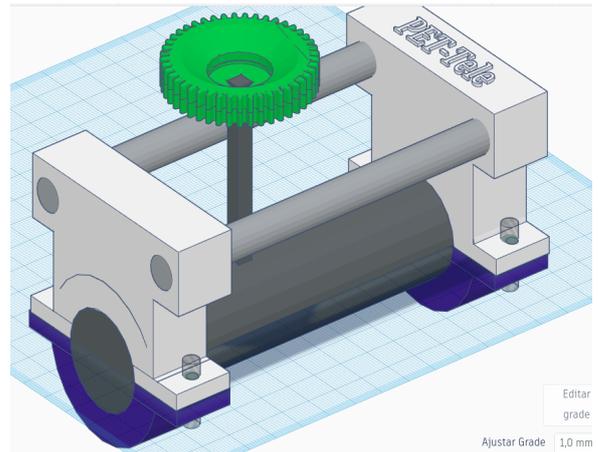
A Figura 3.10 mostra o resultado final da modelagem com a representação de todas as peças visíveis que compõem a estrutura.

<b>Impressões LAB2P (2019)</b>			
<b>Dados iniciais</b>			
Data do pedido	18/02	18/02	18/02
Cliente/ Responsável	Bruno Fischer	Bruno Fischer	Bruno Fischer
Contato do cliente	brunoarguelhes@gmail.com	brunoarguelhes@gmail.com	brunoarguelhes@gmail.com
Monitor(a)	Bianca Barbosa	Bianca Barbosa	Bianca Barbosa
Finalidade do produto	Protótipo	Protótipo	Protótipo
Nome da peça	Braçadeira superior	Braçadeira inferior	Braçadeira inferior
Quantidade	02 peças	02 peças	01 peça
Prazo de entrega pretendido	Até 21/02	Até 21/02	Até 21/02
<b>Dados de Impressão</b>			
Filamento utilizado	ABS	ABS	ABS
Temperatura de Extrusão	235°C	235°C	235°C
Correção de altura	6,7	6,7	6,7
Quantid. preenchimento interno	25%	25%	25%
Velocidade de impressão	95%	95%	95%
Qualidade da impressão	Regular	Regular	Regular
<b>Dados da peça</b>			
Comprimento total da peça (mm)	70	70	23,49
Largura total da peça (mm)	53	103	23,48
Altura total da peça (mm)	20	20	6,05
Quant. Filam. Gasto (grs)	27	77	1
Tempo estimado de impressão (h)	03h 34min.	09h 49min.	0h 15min.
<b>Valores</b>			
Custo do kg do filamento	R\$ 130,00	R\$ 130,00	R\$ 130,00
Custo da impressora	R\$ 4.990,00	R\$ 4.990,00	R\$ 4.990,00
Custo hora/trabalho da máquina	R\$ 1,42	R\$ 1,42	R\$ 1,42
Custo do filamento utilizado	R\$ 3,90	R\$ 11,23	R\$ 0,21
Custo de uso máquina	R\$ 4,28	R\$ 11,78	R\$ 0,30
Custo Unitário de Produção	R\$ 8,18	R\$ 23,01	R\$ 0,51
Desconto	-	-	-
<b>Custo total da impressão</b>	<b>R\$ 8,18</b>	<b>R\$ 23,01</b>	<b>R\$ 0,51</b>

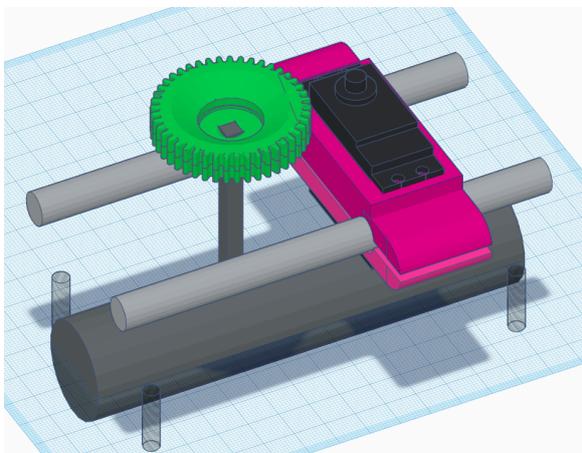
Figura 3.8: Tabela de preços do LAB2P.



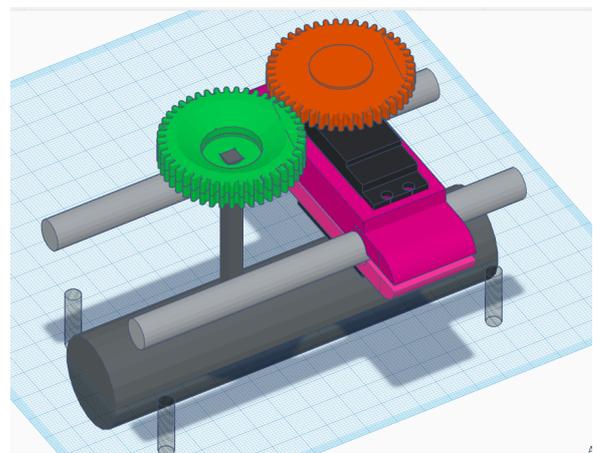
(a) Representação inicial simplificada.



(b) Suporte acoplado na representação inicial.



(c) Guias de fixação e envólucro do servomotor.



(d) Servomotor com engrenagens acopladas.

Figura 3.9: Diversas vistas do processo de modelagem. Em (a) é representado uma modelagem simplificada, em (b) têm-se a adição do suporte de fixação na tubulação, em (c) é representado as guias de rigidez e o envólucro do servomotor e por fim em (d) é adicionado as engrenagens no servomotor.

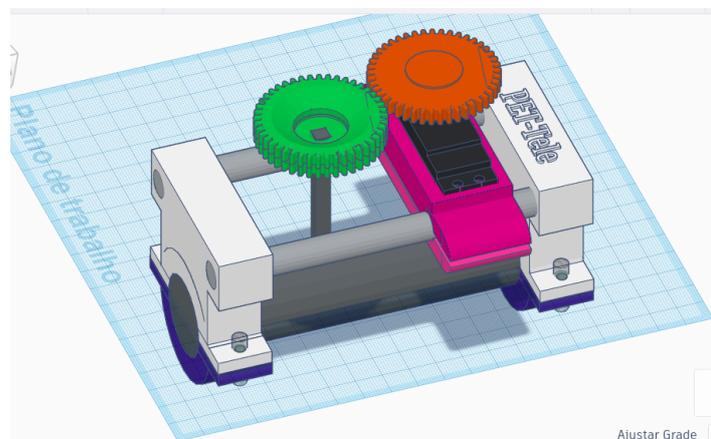


Figura 3.10: Vista da modelagem completa.

Uma vez terminada a modelagem, iniciou-se a impressão por partes do modelo em 3D. Considerando fatores de distorção entre modelagem e impressão foram requeridos alguns ajustes manuais nas peças. Por exemplo, os diâmetros dos furos para a colocação da maioria dos parafusos de fixação na tubulação ficaram menores que o desejado. Foi preciso expandí-los com uma broca e furadeira. Apesar dos ajustes necessários, pode-se dizer que foi obtido um excelente resultado com a impressão final das peças. A Figura 3.11 mostra a estrutura completa acoplada à planta didática após os devidos ajustes.

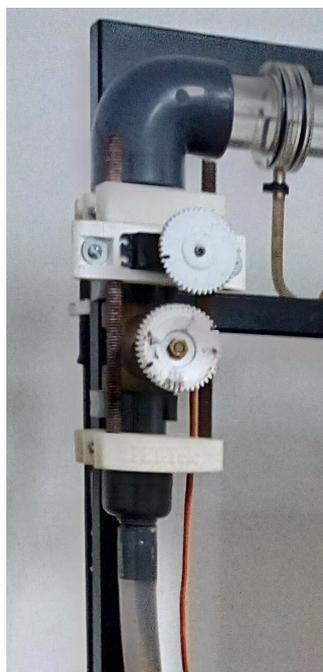


Figura 3.11: Fotografia da estrutura acoplada na planta didática.

### 3.5 Modelo de capa para Raspberry Pi e Arduino

Ao verificar a facilidade de interação com o *software* TinkerCad, considerou-se modelar uma capa para proteção dos componentes eletrônicos e das placas de prototipagem utilizadas no projeto. Essa parte da atividade não foi implementada. Ela foi apenas projetada. A ideia da capa veio da necessidade de atender a uma questão estrutural importante, que é a organização e proteção dos componentes eletrônicos que forma o sistema de controle. Para isso, decidiu-se criar uma capa conforme aquela mostrada na Figura 3.12, que une e envolve tanto o Arduino quanto o Raspberry Pi em uma peça única.

A capa protetora mostrada na Figura 3.12, que ainda está em etapa de estudo, pode ser dividida em dois setores, um superior e um inferior, separados por uma peça intermediária. O superior é o local de fixação do Raspberry Pi, enquanto o inferior é destinado ao Arduino UNO. A capa possui também diversos espaços vazados para dar acesso aos pinos de entrada e saída de cada placa. Em vista das atividades que já ocorreram no LaDISan a capa é um subproduto e têm a pretensão de facilitar o transporte, utilização, proteção e diminuição de folgas entre os fios, que por vezes ocasionam mal contato.

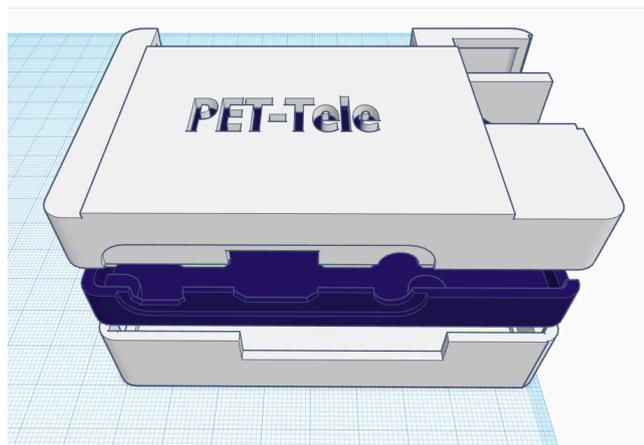


Figura 3.12: Capa para o Raspberry Pi e Arduino.

## 3.6 Novo sistema proposto

### 3.6.1 Estrutura do sistema proposto

Essa seção dedica-se à apresentação dos mecanismos de funcionamento do sistema de atuação remota proposto. A ideia geral representada na Figura 3.13 é o Raspberry Pi funcionando como uma central de automação. Foram escritos *scripts* em Python para a coleta de dados do Arduino e também para a atuação do servomotor no controle de admissão de água. Em termos de conexão, o Raspberry Pi está ligado à rede local da UFF (Eduroam [RNP, 2021]) com permissão para acesso por senha de um usuário da rede local. Neste projeto, o usuário é o computador presente na sala de aula. A partir de um *software* ou protocolo de acesso remoto (VNCViewer, TeamViewer, SSH), o computador-usuário consegue visualizar os dados de vazão em tempo real e mandar comandos de início e fim do experimento para o servomotor que os executa.

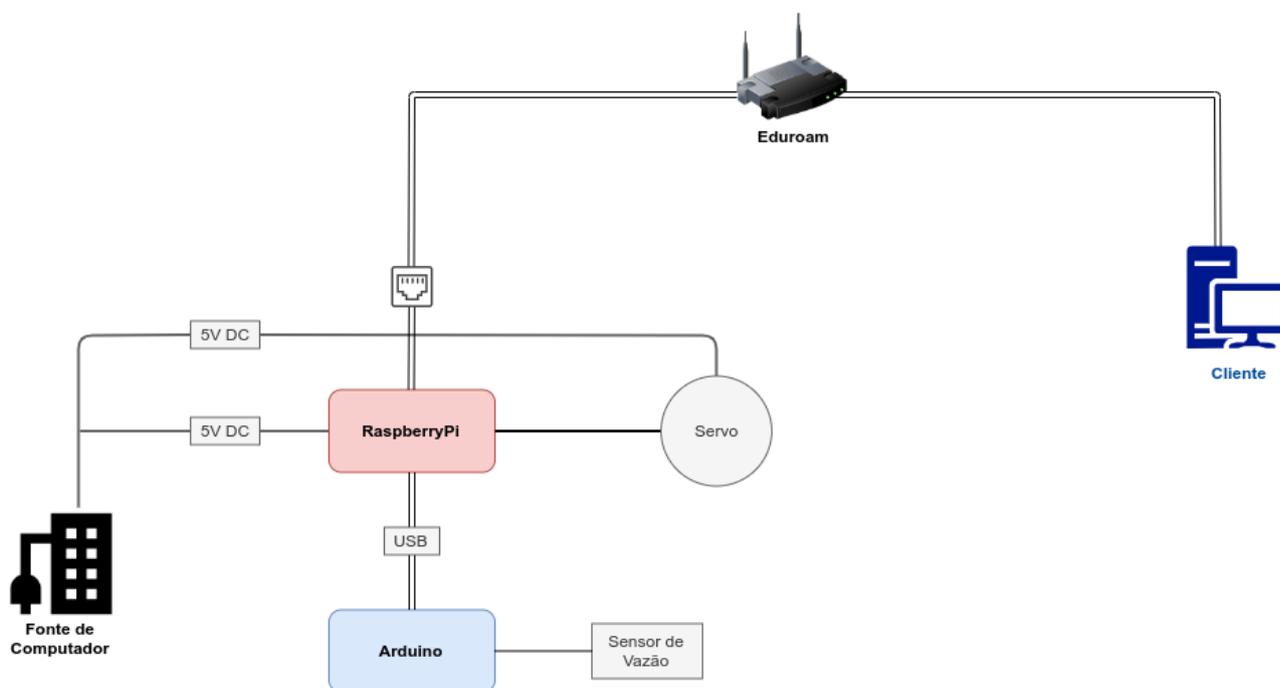


Figura 3.13: Diagrama de blocos da infraestrutura.

A seguir serão apresentados detalhes específicos sobre cada parte constituinte.

A Figura 3.14 mostra o diagrama esquemático elétrico da conexão entre Arduino e os sensores de vazão. Os sensores YF-S201 foram apresentados na Seção 2.1 e são conectados em portas digitais do Arduino, que estão programadas para reconhecer esses sensores. O Arduino entra em seu estado de *loop* e realiza as medições de acordo com certos intervalos pré-definidos.

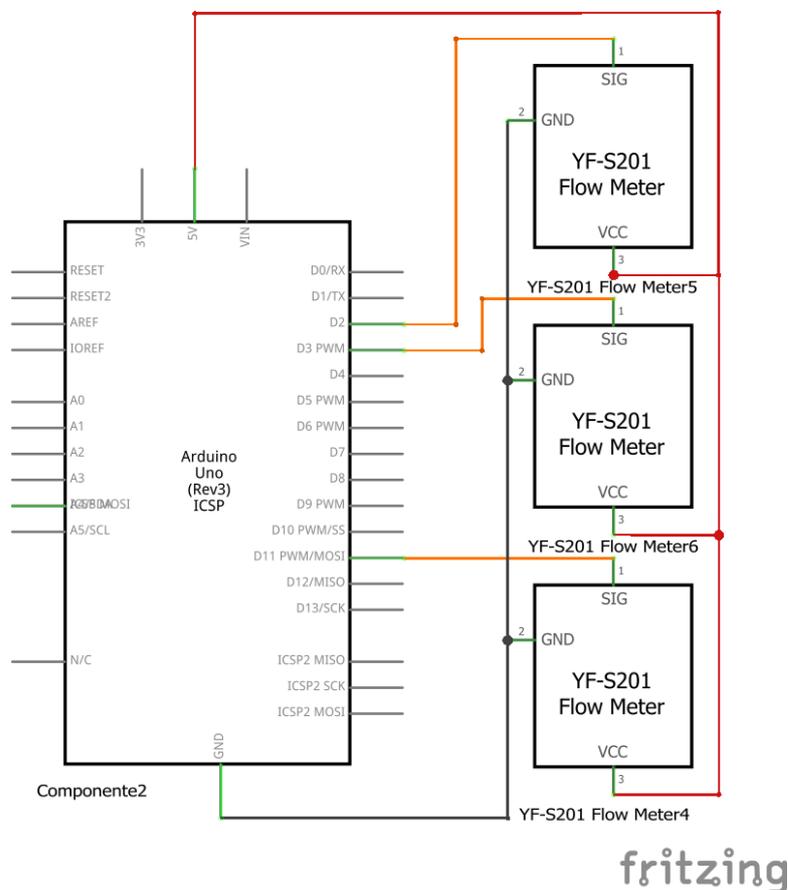


Figura 3.14: Circuito esquemático do sistema de coleta.

Ao invés de reprojeter a coleta e a organização dos dados de vazão, escolheu-se adaptar a infraestrutura original (*hardware* e *software*), baseada no Arduino. No novo sistema proposto, o Arduino é ligado, via USB, ao Raspberry Pi. Os dados captados dos sensores pelo Arduino são transferidos para o Raspberry Pi, ao invés de serem enviados ao *notebook* presente no laboratório, como no projeto original. Essa configuração além de receber os dados através da porta serial permite fornecer energia ao Arduino pelo Raspberry Pi.

A Figura 3.15 exemplifica a conexão Arduino-Raspberry Pi via USB. Além disso, a conexão do servomotor na placa Raspberry Pi se dá através de um pino PWM que é preparado para receber o motor. Para o fornecimento de energia do componente foi utilizada uma fonte DC do tipo ATX para computadores. Como mencionado na Seção 3.7, houve a necessidade do uso dessa fonte pela necessidade elevada de potência requerida pelo servomotor para realizar o trabalho com a carga do sistema. Com o objetivo de melhorar a compreensão dos métodos, dispositivos e elementos aqui desenvolvidos, a Figura 3.16 apresenta uma ilustração referente ao modelo esquemático.

Na foto da Figura 3.17 é exposta a estrutura que foi desenvolvida em ferramenta de modelagem em 3D já instalada na planta hidráulica.

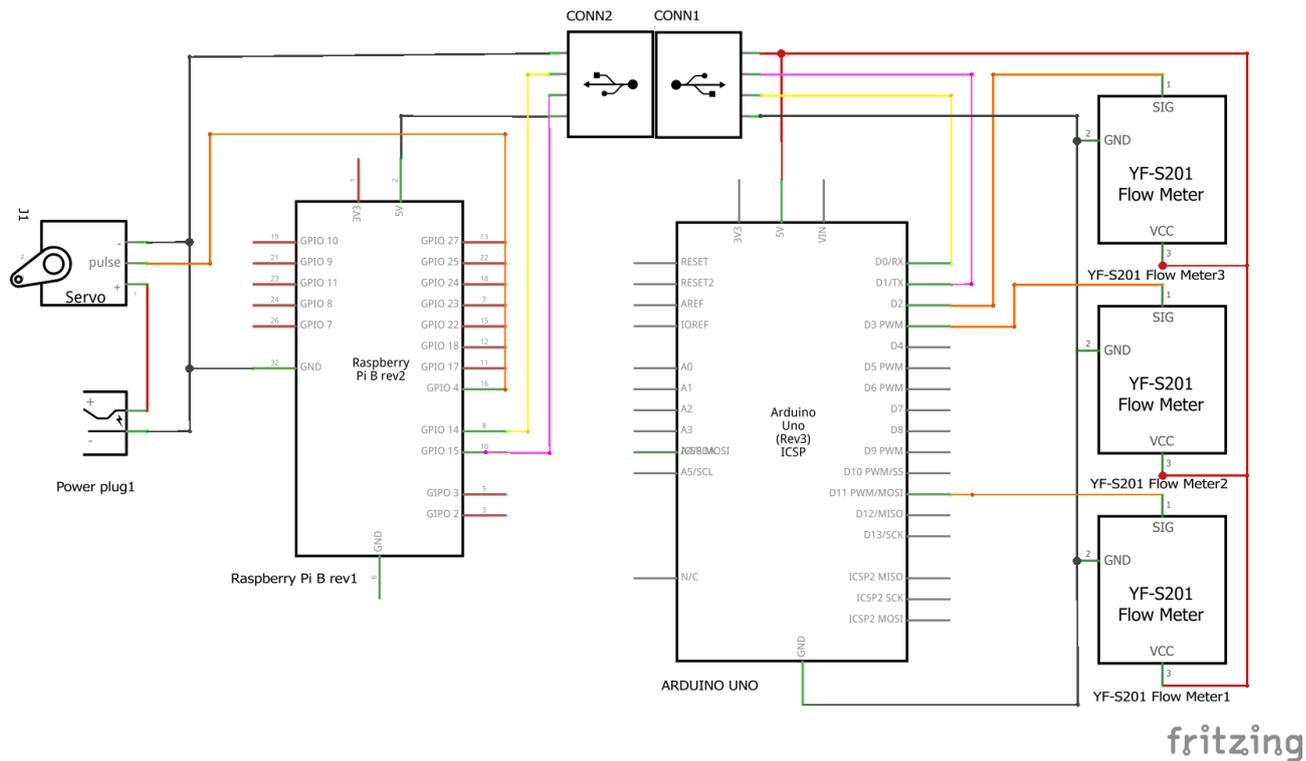


Figura 3.15: Circuito esquemático entre Raspberry Pi e Arduino.

## Sistema de Admissão e controle de Vazão Automatizados

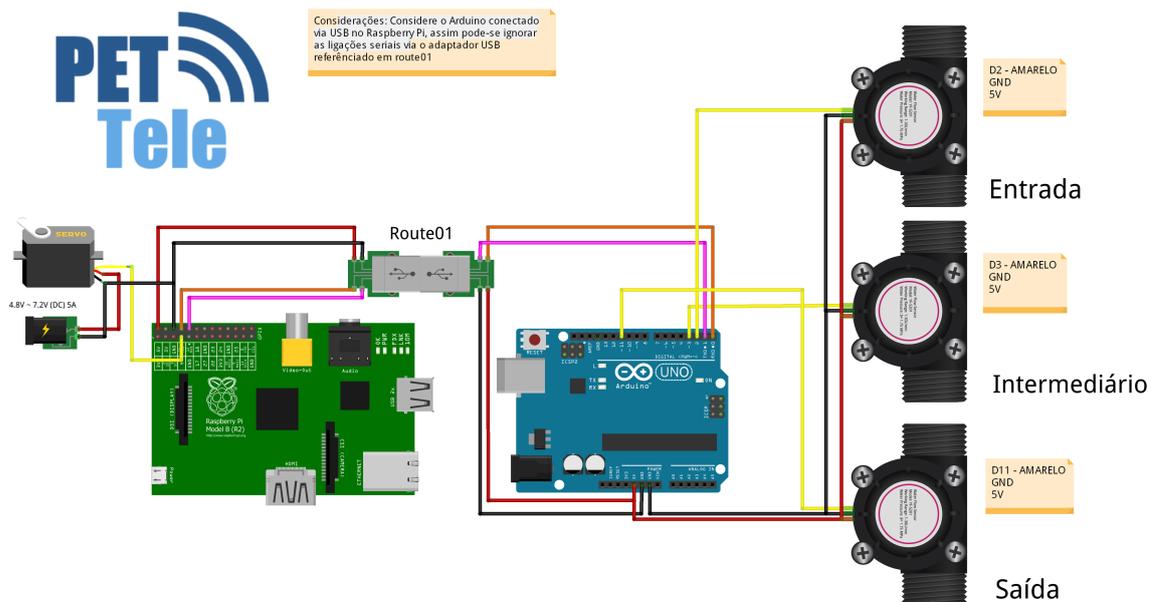


Figura 3.16: Visão geral das conexões.

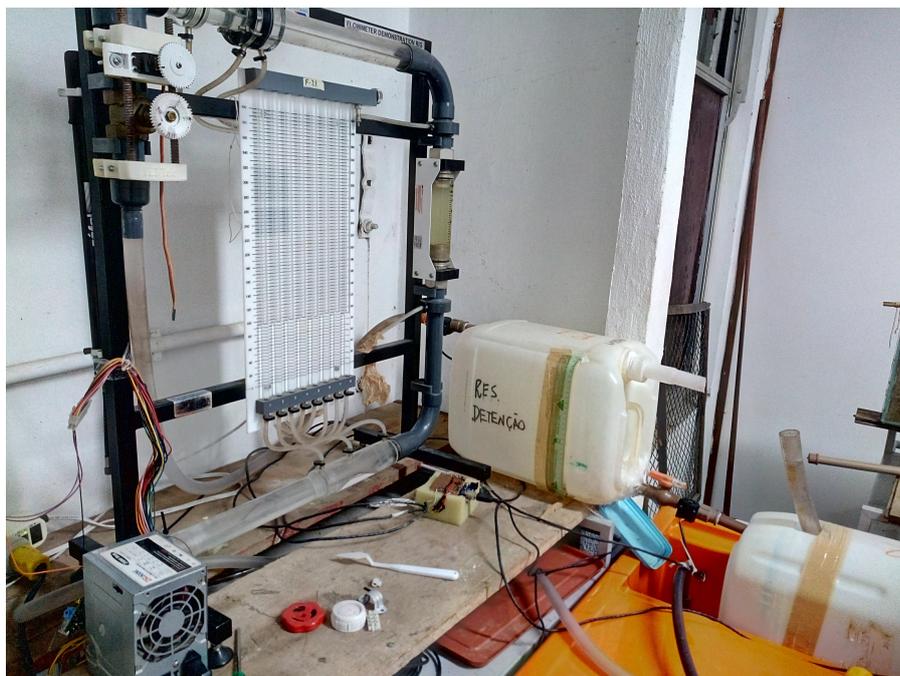


Figura 3.17: Fotografia da planta didática com o sistema acoplado.

### 3.6.2 Programas de computador para gerenciamento do sistema

Nesta seção, será apresentado o processo de desenvolvimento dos códigos utilizados na placa de prototipagem Raspberry Pi, para controle do servomotor, automatização, interface, gerenciamento e os mecanismos necessários para a utilização remota do laboratório para esse experimento.

Todo o código desenvolvido para controle do motor, automatização e interface foi implementado na linguagem de programação Python, em sua versão 3. Por ser uma linguagem de propósito geral, não houve dificuldade de encontrar os módulos necessários para elaborar o projeto. Vale citar a integração e compatibilidade nativa que o Raspberry Pi possui com Python. Essa integração acabou sendo um motivo a mais para sua utilização. No diagrama da Figura 3.18 são citados os nomes dos principais *scripts* criados para automatizar o experimento. Em versão futura desse documento, esses *scripts* serão disponibilizados em um apêndice apropriado. Até o momento, o *software* desenvolvido está disponível apenas para ambientes Linux, tanto no lado cliente quanto no do Servidor.

No diagrama da Figura 3.18 é possível perceber os programas Python com a extensão de arquivos “.py”. Cada um deles é responsável por uma etapa, seja coleta, atuação, transmissão ou tratamento dos dados. Há uma separação dessas atividades em relação ao que é executado no lado do cliente (computador em sala de aula) e ao que é executado no servidor (Raspberry Pi - laboratório). No cliente são executados programas que visam receber os dados coletados, exibí-los para o usuário de forma amigável e também permitir o envio de comandos para o servidor. No servidor, as atividades são as seguintes: coletar os dados de vazão, enviar esses dados para o cliente e controlar o servomotor de acordo com os comandos recebidos do usuário.

É possível notar também os arquivos com extensão “.txt”, que estão sendo utilizados como uma forma de comunicação entre os processos correntes (IPC ou *Inter Process Communication*). Uma observação importante sobre esse tipo de método é que ele é o menos eficiente de todos os tipos de IPC [IBM, 2021]. A justificativa para utilizá-lo reside na conveniência para prototipação rápida e salvar os dados do experimento. Mais tarde, esse tipo de IPC será substituído por um método mais eficiente.

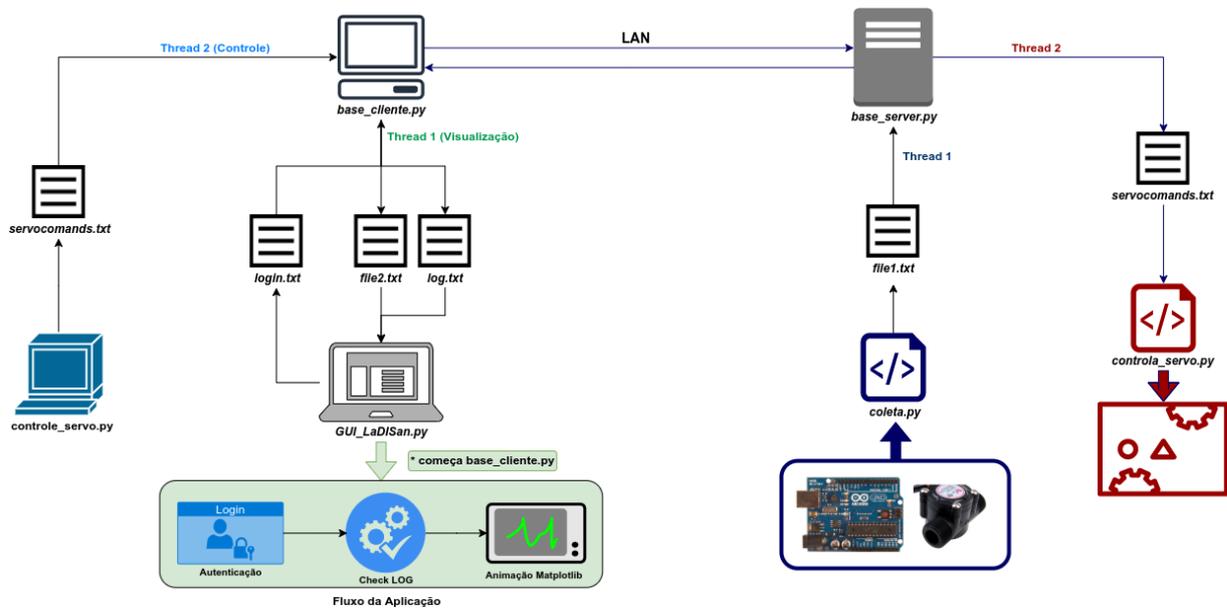


Figura 3.18: Diagrama de blocos do sistema.

As Tabelas 3.2 e 3.3 apresentam a descrição da função de cada *script* em ambos os lados da conexão.

base_server.py	O script "base_server.py" cria um servidor <i>socket</i> que possui duas funções básicas: A primeira é ler o arquivo "file1.txt" e enviar esses dados de coleta para o cliente. A segunda atividade é receber, do cliente, os comandos de início e fim da rotação do servomotor e publicá-los no arquivo "servocomands.txt", para serem lidos pelo <i>script</i> "controla_servo.py". Para isso, existem duas <i>threads</i> , cada uma agindo em uma dada porta de rede.
controla_servo.py	Apenas lê o arquivo "servocomands.txt" e reproduz os comandos que aparecem linha a linha.

Tabela 3.2: Tabela de descrição dos *scripts* do servidor.

GULLaDISan	GULLaDISan é o <i>script</i> responsável por construir a interface gráfica e executar os procedimentos inicialização do "base_cliente.py", <i>login</i> , visualização dos gráficos e escrever os comandos de começo e fim do experimento no arquivo "servocomands.txt".
base_cliente.py	O <i>script</i> "base_cliente.py" possui 2 funções. A primeira é receber os dados de vazão do servidor e salvá-los no arquivo "file2.txt", para o "GULLaDISan.py" gerar os gráficos de vazão. A segunda função é a de enviar os comandos presentes no arquivo "servocomands.txt" para o servidor. Para isso, existem duas <i>threads</i> , cada uma agindo em uma dada porta de rede.

Tabela 3.3: Tabela de descrição dos *scripts* do cliente.



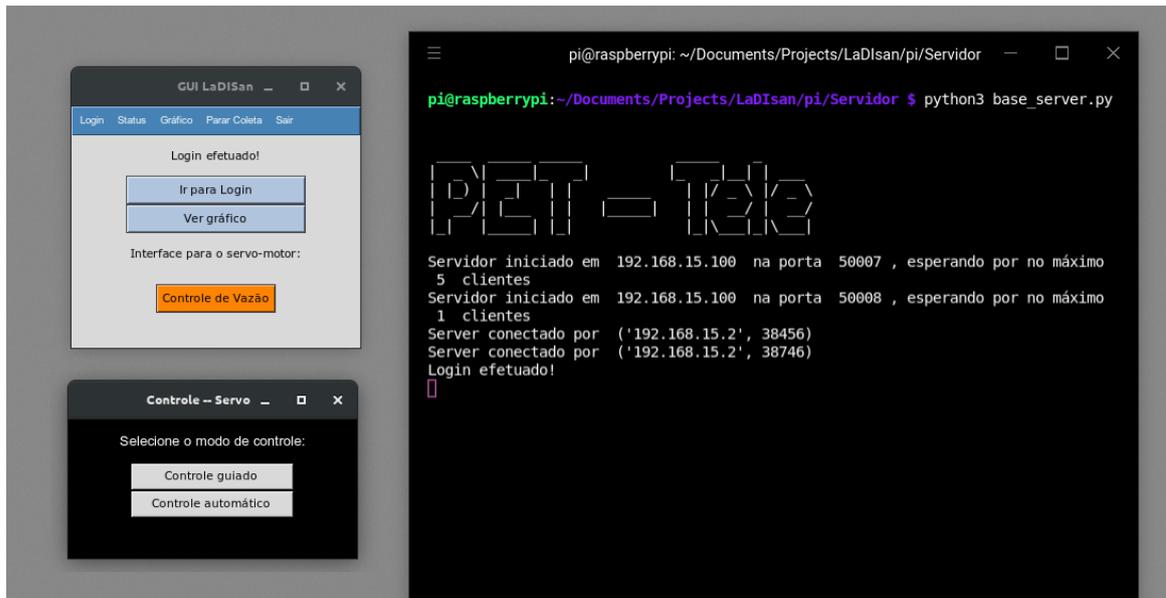


Figura 3.21: Janela de *status* e botões de navegação.

No decorrer do experimento, a interface de usuário é modificada, para indicar a etapa de operação do servomotor, a curva de vazão e o volume de água presente nos reservatórios, conforme mostrado na Figura 3.22. Quando o servomotor inicia seu movimento de abertura de válvula, a animação do arquivo “.gif” é executada na interface “Inicializador de Curva”. Essa interface fica aberta lado a lado à construção da curva em tempo real.

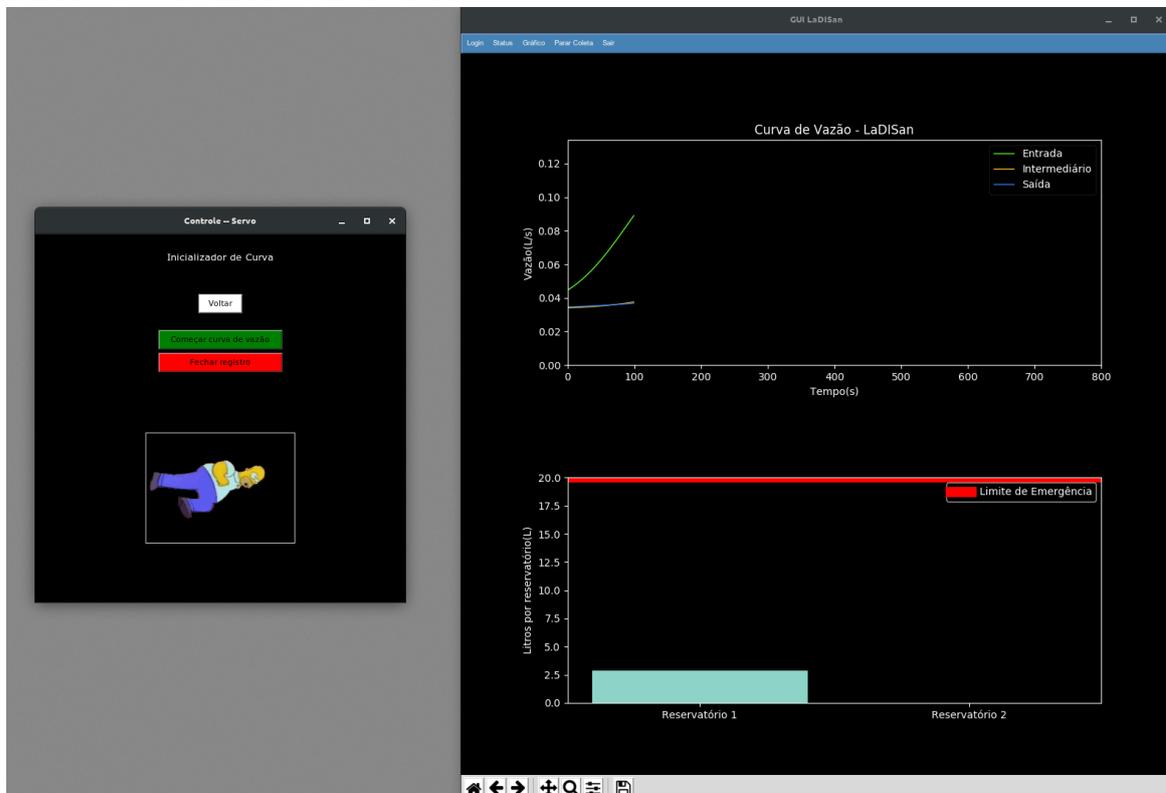


Figura 3.22: Interface de usuário com experimento em operação.

### 3.7 Discussão de erros e falhas de implementação

Essa seção apresenta os erros encontrados, solucionados e não solucionados.

O primeiro problema encontrado foi a falta de força dos servomotores. Em testes iniciais ao ligarmos o servomotor especificado neste relatório diretamente no Raspberry Pi, foi obtido a correspondência desejada entre controle eletrônico e movimentação física. Ao fazer o transporte dessa configuração e montagem do servomotor na estrutura do laboratório os resultados obtidos não eram condizentes. Certo ponto não apresentava correspondência alguma entre controle e movimentação. Além disso, o servomotor apresentava chiado interno, não respondia aos comandos e nenhuma movimentação aparente. Para resolução desses problemas foram feitas medições de tensões e correntes nos terminais do motor e do Raspberry Pi. Foi verificado a existência de uma queda de tensão, acima do normal, ao realizar uma movimentação enquanto o sistema estava acoplado a carga. O termo carga aqui aplicado se refere ao registro mecânico que controla a vazão de água na tubulação. A solução para esse problema foi simples em contrapartida à árdua detecção. Levou-se meses para finalmente ter-se uma conclusão. Ao fazer a comparação das medições obtidas com o *datasheet* do servomotor e do Raspberry Pi, foi colocada em evidência e dada a devida atenção para o fato de que o Raspberry Pi, com a fonte utilizada, não fornecer potência suficiente para suprir o servomotor com a carga acoplada em seu eixo.

Então, a solução foi reaproveitar uma antiga Fonte DC de computador que estava disponível na sala do grupo.

# Capítulo 4

## Conclusão e trabalhos futuros

### 4.1 Conclusão

Utilizando, como base, o documento **Uso de Sensores e Acesso Remoto para a Realização de Aula Prática Sobre Reservatórios de Detenção Aplicados à Drenagem urbana** [Paiva and Milhomem, 2018] e o artigo **Realização de Aula Prática Remota a Partir de Laboratório Equipado com Modelo Físico Sobre Detenção de Água de Chuva** [Paiva et al., 2018], publicado no COBENGE 2018 [ABENGE, 2020], foi desenvolvido um conjunto de melhorias e novos processos que permitem monitoramento e maior e melhor atuação do “cliente” (computador remoto localizado na sala de aula) na planta didática que opera no Laboratório LaDISan [LaDISan, 2019]. Esse conjunto de melhorias compreende uma câmera com controle remoto, um servomotor acoplado ao registro de admissão de água, uma interface de acesso e um dispositivo Raspberry Pi presente no laboratório operando como controlador remoto do servomotor e computador auxiliar para o Arduino.

Os resultados foram satisfatórios no que tange aos objetivos propostos. Foram realizadas duas aulas no ano de 2019. Sendo uma no primeiro semestre do ano correspondente e outra no segundo semestre. Existem ainda diversos erros de implementação conforme citados no Capítulo 3, que vão ser corrigidos conforme novas propostas e atualizações de projeto.

### 4.2 Trabalhos futuros

Neste relatório foi apresentado conceitos e resultados estudados ao longo do desenvolvimento do projeto. Observando a amplitude do estudo e dos conceitos, existe a necessidade de realização de mais testes no sistema, para que todas as funcionalidades propostas estejam com funcionamento correto. Para isso, é de fundamental importância novos trabalhos com a elaboração de novos casos e situações para teste, a repetição dos casos já estudados e que implemente novas atualizações com versões otimizadas dos programas. Além disso, para trabalhos futuros é necessário a realização de um experimento prático com a nova interface proposta na Seção 3.6.3.

# Referências Bibliográficas

- [ABENGE, 2020] ABENGE (2020). **URL do COBENGE**. Disponível em: “<http://www.abenge.org.br/cobenge.php>”. Acesso em: 25/01/2021.
- [Arduino, 2020] Arduino (2020). **URL do Arduino**. Disponível em: “<https://www.arduino.cc/>”. Acesso em: 25/01/2021.
- [Autodesk, Inc., 2019] Autodesk, Inc. (2019). **URL da plataforma TinkerCad AutoDesk**. Disponível em: “<https://www.tinkercad.com/>”. Acesso em: 25/01/2021.
- [Codecademy, Inc., 2020] Codecademy, Inc. (2020). **Integrated Development Environment (IDE)**. Disponível em: “<https://www.codecademy.com/articles/what-is-an-ide>”. Acesso em: 25/01/2021.
- [Coelho et al., 2017] Coelho, T. C., Siqueira, L. P., de Oliveira, F. B., and de la Vega, A. S. (2017). **Implantação de um Sistema Automático para Medição de Fluxo em uma Planta Didática Usando um Kit Arduino e o Sensor Water Flow G1/2**. *COBENGE*, page 8. Disponível em: “[http://www.telecom.uff.br/pet/petws/downloads/artigos/cobenge\\_2017/COBENGE17\\_00017\\_00000123\\_Planta\\_LaDISan\\_2017\\_07\\_15.pdf](http://www.telecom.uff.br/pet/petws/downloads/artigos/cobenge_2017/COBENGE17_00017_00000123_Planta_LaDISan_2017_07_15.pdf)”.
- [Croston, 2019] Croston, B. (2019). **URL da biblioteca RPI.GPIO 0.7.0 para Python**. Disponível em: “<https://pypi.org/project/RPi.GPIO/>”. Acesso em: 25/01/2021.
- [Duke et al., 2015] Duke, M., Braden, R., Eddy, W. M., Blanton, E., and Zimmermann, A. (2015). **A Roadmap for Transmission Control Protocol (TCP) Specification Documents**. Disponível em: “<https://tools.ietf.org/html/rfc7414>”. Acesso em: 25/01/2021.
- [Electronics, 2019] Electronics, J. (2019). **How do servo motors work?** Disponível em: “<https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html>”. Acesso em: 25/01/2021.
- [Foundation, 2020] Foundation, R. P. (2020). **URL da plataforma Raspberry Pi**. Disponível em: “<https://www.raspberrypi.org/>”. Acesso em: 25/01/2021.
- [Girardi, 2017a] Girardi, D. (2017a). **URL do projeto Teachduino (UFSC)**. Disponível em: “<https://teachduino.ufsc.br/>”. Acesso em: 25/01/2021.
- [Girardi, 2017b] Girardi, D. (2017b). **ViewDuino - Uma interface gráfica para dados do Arduino**. Disponível em: “<https://teachduino.ufsc.br/2017/09/06/viewduino-uma-interface-grafica-para-dados-do-arduino-bluetooth/>”. Acesso em: 25/01/2021.
- [Grupo PET-Tele, 2021] Grupo PET-Tele (2021). Disponível em: “<http://www.telecom.uff.br/pet>”. Acesso em: 25/01/2021.

- [IBM, 2021] IBM (2021). **The Interprocess Communication (IPC) Overview**. Disponível em: “<https://www.ibm.com/support/pages/interprocess-communication-ipc-overview>”. Acesso em: 25/01/2021.
- [joan2937, 2019] joan2937 (2019). **URL da biblioteca PIGPIO para Python**. Disponível em: “<https://github.com/joan2937/pigpio>”. Acesso em: 25/01/2021.
- [LAB2P, 2019] LAB2P (2019). **URL do Laboratório de Produtos e Processos (TGP/UFF)**. Disponível em: “Sem\_URL\_disponivel”. Acesso em: 25/01/2021.
- [LaDISan, 2019] LaDISan (2019). **URL do Laboratório de Drenagem, Irrigação e Saneamento Ambiental (TER/UFF)**. Disponível em: “<http://www.uff.br/?q=setor/laboratorio-de-drenagem-irrigacao-e-saneamento-ambiental-ladisan>”. Acesso em: 25/01/2021.
- [Members, 2019a] Members, W. (2019a). **URL do termo Raspberry Pi**. Disponível em: “[https://pt.wikipedia.org/wiki/Raspberry\\_Pi](https://pt.wikipedia.org/wiki/Raspberry_Pi)”. Acesso em: 25/01/2021.
- [Members, 2019b] Members, W. (2019b). **URL do termo *System on a chip* (SoC)**. Disponível em: “<https://pt.wikipedia.org/wiki/System-on-a-chip>”. Acesso em: 25/01/2021.
- [Miranda and de la Vega, 2018] Miranda, R. and de la Vega, A. S. (2018). **Tutorial sobre aplicativo computacional TeamViewer**. Disponível em: “[http://www.telecom.uff.br/pet/petws/downloads/tutoriais/TeamViewer/Tut\\_TeamViewer.pdf](http://www.telecom.uff.br/pet/petws/downloads/tutoriais/TeamViewer/Tut_TeamViewer.pdf)”. Acesso em: 25/01/2021.
- [Paiva and Milhomem, 2018] Paiva, L. M. M. and Milhomem, M. C. (2018). **Uso de sensores e acesso remoto para a realização de aula prática sobre reservatórios de detenção aplicados à drenagem urbana**. *UFF*, page 70. Disponível em: “<https://app.uff.br/riuff/handle/1/7080>”.
- [Paiva et al., 2018] Paiva, L. M. M., Milhomem, M. C., and de la Vega, A. S. (2018). **Realização de Aula Prática Remota a Partir de Laboratório Equipado com Modelo Físico sobre Detenção de Água de Chuva**. *COBENGE*, page 10. Disponível em: “[http://www.telecom.uff.br/pet/petws/downloads/artigos/cobenge\\_2018/COBENGE18\\_00063\\_00001258\\_Aula\\_Remota\\_LaDISan\\_2018\\_07\\_09.pdf](http://www.telecom.uff.br/pet/petws/downloads/artigos/cobenge_2018/COBENGE18_00063_00001258_Aula_Remota_LaDISan_2018_07_09.pdf)”.
- [Pinckney, 2006] Pinckney, N. (2006). Pulse-width modulation for microcontroller servo control. *IEEE Potentials*, 25(1):27–29.
- [RFC Editors, 2020] RFC Editors (2020). **URL do padrão do protocolo IP**. Disponível em: “<https://www.rfc-editor.org/standards>”. Acesso em: 25/01/2021.
- [RNP, 2021] RNP (2021). **EDUROAM - Rede Nacional de Ensino e Pesquisa (RNP)**. Disponível em: “<https://www.rnp.br/servicos/alunos-e-professores/colaboracao-a-distancia/eduroam>”. Acesso em: 25/01/2021.
- [Suehring and Salowey, 2005] Suehring, S. and Salowey, J. (2005). **SCP/SFTP/SSH URI Format draft-ietf-secsh-scp-sftp-ssh-uri-02**. Disponível em: “<https://tools.ietf.org/html/draft-ietf-secsh-scp-sftp-ssh-uri-02#page-7>”. Acesso em: 25/01/2021.
- [Teleco, ] Teleco (-). **Seção: Tutoriais Banda Larga**. Disponível em: “[https://www.teleco.com.br/tutoriais/tutorialvpn/pagina\\_3.asp](https://www.teleco.com.br/tutoriais/tutorialvpn/pagina_3.asp)”. Acesso em: 25/01/2021.

- [Tower-Pro, 2019] Tower-Pro (2019). **URL do Datasheet do MG995**. Disponível em: “<https://storage.googleapis.com/baudaeletronicadatasheet/MG995.pdf>”. Acesso em: 25/01/2021.
- [USB Implementers Forum, 2020] USB Implementers Forum, I. (2020). **URL da tecnologia *Universal Serial Bus (USB)***. Acesso em: 25/01/2021.
- [Wikipedia Members, 2021] Wikipedia Members (2021). **Transdutor**. Disponível em: “<https://pt.wikipedia.org/wiki/Transdutor>”. Acesso em: 25/01/2021.
- [YIFA, 2020] YIFA (2020). **Sensor Water Flow YF-S201**. Disponível em: “<http://www.hobbytronics.co.uk/datasheets/sensors/YF-S201.pdf>”. Acesso em: 25/01/2021.
- [Ylonen and Lehtinen, 2001] Ylonen, T. and Lehtinen, S. (2001). **SSH File Transfer Protocol draft-ietf-secsh-filexfer-02**. Disponível em: “<https://tools.ietf.org/html/draft-ietf-secsh-filexfer-02>”. Acesso em: 25/01/2021.
- [Yoosee, 2017] Yoosee (2017). **URL da fabricante Yoosee**. Disponível em: “<https://pt.yooseecamera.com/>”. Acesso em: 25/01/2021.
- [Yoosee, 2017] Yoosee (2017). **URL do Manual do *software CMS***. Disponível em: “[http://support.hkvstar.com/file/Yoosee\\_Instruction\\_Manual.pdf](http://support.hkvstar.com/file/Yoosee_Instruction_Manual.pdf)”. Acesso em: 25/01/2021.
- [Özderya, 2017] Özderya, H. Y. (2017). **URL da plataforma SerialPlot - *Realtime Plotting Software***. Disponível em: “<https://hackaday.io/project/5334-serialplot-realtime-plotting-software>”. Acesso em: 25/01/2021.

# **Apêndice A**

## **Tabela de peças**

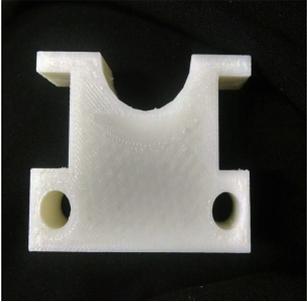
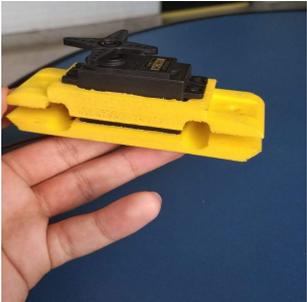
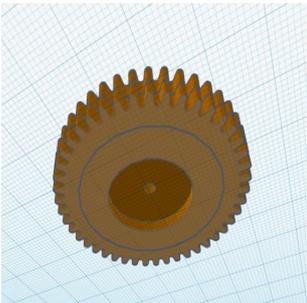
Peça	Nome	Função
	Parte superior da braçadeira.	Em conjunto com a parte inferior, serve para fixar a estrutura na tubulação. Possui dois furos para transpassar as barras metálicas.
	Parte inferior da braçadeira.	Com a parte superior da braçadeira, serve para fixar a estrutura na tubulação. Possui dois furos para inserção de parafusos rosqueados.
	Invólucro do servomotor.	Serve para fixar o servomotor na estrutura criada. Possui dois furos para transpassar as barras metálicas.
	Engrenagem dentada.	Realizar a transmissão de movimento entre o motor e o registro.

Tabela A.1: Descrição das peças impressas.