

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
ESCOLA POLITÉCNICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**GABRIEL NOGAS
LUCAS PIERIN
WALLY JAREK**

**RELATÓRIO FINAL DE PROJETO INTEGRADOR
PROJETO AUTO VARAL CONTROLLER**

**CURITIBA
2015**

**GABRIEL NOGAS
LUCAS PIERIN
WALLY JAREK**

**RELATÓRIO FINAL DE PROJETO INTEGRADOR
PROJETO AUTO VARAL CONTROLLER**

Relatório de Projeto apresentado ao Curso de Engenharia de Computação da Pontifícia Universidade Católica do Paraná, como requisito parcial para a disciplina de Resolução de Problemas em Engenharia I.

Orientador: Prof. MSc Afonso Ferreira Miguel

CURITIBA

2015

AGRADECIMENTOS

Agradecemos ao professor Ivan Chueiri pelo apoio da ideia do projeto, ao professor Marcelo Gaiotto pelas dicas para criação da placa ponte-H e ao professor Guilherme Nunes Nogueira Neto que inúmeras vezes esteve disponível para orientações.

Cabe o Agradecimento, também, ao professor Afonso Ferreira Miguel pela sua dedicação e apoio, além de sua orientação durante todo o projeto, dando várias dicas muito importantes durante toda a construção e programação do Auto Varal Controller.

Um agradecimento especial aos amigos e familiares que se interessaram com a ideia desde o início do escopo e contribuíram para um bom andamento do projeto.

RESUMO

Este trabalho apresenta uma solução prática para quem pendura as roupas no varal em dias nublados e/ou precisa sair de casa. Seu sistema com sensor de chuva identifica quando começa a chover e recolhe o varal. O varal é controlado a partir da internet, podendo ser diretamente por ela, ou por um aplicativo online. A ideia do projeto surgiu do problema que algumas pessoas tinham para pendurar roupa em dias “ruins” ou precisavam sair de casa, isso ajuda a causa uma redução no gasto de energia, que seria usada em secadoras. O projeto conta com 3 Arduinos para realizar toda a comunicação por rádio e Ethernet para que os comandos sejam realizados.

Palavras-chave: Varal. Economia. Sensor.

ABSTRACT

This project presents a practical solution to the people who hang the clothes to dry at the clothes line when they need to go out in a rainy or cloudy days. Our system have rain sensor that identify when starts to rain the AVC returns to his start point. He is controlled through the internet or by an online app. The idea of the project came from the problem that some people have when they hang clothes and have to do something else, this helps reducing the amount of electric energy used in dry machines. The project uses three Arduino Boards to use both Radio and Ethernet communication.

Key-words: Clothes line. Economy. Sensor.

LISTA DE ILUSTRAÇÕES

FIGURA 1 MODELO DE VARAL COM SENSOR DE CHUVA.....	13
FIGURA 2 MODELO DE VARAL COM SENSOR DE CHUVA.....	13
FIGURA 3 MODELO DE VARAL COM SENSOR DE CHUVA.....	14
FIGURA 4 VISÃO SUPERIOR DA BASE DO VARAL.....	18
FIGURA 5 ARDUINO COM MOTORSHIELD DA BASE DO VARAL	18
FIGURA 6 IMAGEM SOFTWARE ARDUINO.....	19
FIGURA 7 FOTO DA BASE.....	35
FIGURA 8 FOTO DA BASE.....	35
FIGURA 9 FOTO DA BASE.....	36
FIGURA 10 FOTO DO AVC	36

LISTA DE TABELAS

Tabela 1 - Tabela componentes	23
-------------------------------------	----

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AVC	Auto Varal Controller
PUCPR	Pontifícia Universidade Católica do Paraná
SASR	Sistema de Ajuda para Secagem de Roupas

SUMÁRIO

1. INTRODUÇÃO	10
1.1. HISTÓRICO	10
1.2. OBJETIVOS	11
1.2.1. Objetivo Geral.....	11
1.2.2. Objetivos Específicos	12
2. ESTADO DA ARTE	13
3. REFERENCIAL TEÓRICO.....	15
4. METODOLOGIA.....	17
5. O PROJETO.....	17
6. RESULTADOS	22
7. IMPACTO AMBIENTAL.....	23
8. CONSIDERAÇÕES FINAIS	24
ANEXO A – PROGRAMAÇÃO DO ARDUINO UNO DA BASE	26
ANEXO B – PROGRAMAÇÃO DO ARDUINO MEGA DA BASE.....	30
ANEXO C – PROGRAMAÇÃO DO ARDUINO MEGA DO CARRINHO	32
ANEXO D - FOTOS DO PROJETO	35

1.INTRODUÇÃO

A ideia da criação de um varal automatizado veio pelo receio de se pendurar roupas em um dia nublado ou quando é preciso sair de casa, as pessoas ficam com medo de que comece a chover. Atualmente existem alguns varais automatizados, porém alguns tem sensor de humidade e outros tem sensor de chuva, o projeto AVC tem como seus diferenciais o deslocamento para uma área em que vai secar as roupas e possui o sensor de chuva.

O objetivo principal deste projeto é facilitar a vida das donas de casa em dias chuvosos ou quando precisam sair de casa, sem se preocupar com quem vai recolher as roupas penduradas, assim economizando tempo e trazendo sossego.

Para tanto, o projeto consiste no uso de sensor de chuva e seus motores para deslocamento para o local pré-determinado de secagem.

Um programa foi desenvolvido para interagir com o varal, apresentando as opções de estender a roupa, recolher e por fim um modo automático, onde é possível deixar que o varal recolha as roupas no momento certo. O varal lê os sensores a cada período de tempo e assim que as leituras assumem valores fora do comum, é enviado o comando de recolher a base do varal.

1.1. HISTÓRICO

A sugestão foi levantada pelo integrante do grupo WGL, Lucas Pierin. A ideia surgiu por conta da preocupação de sua mãe quando precisava sair de casa em dias nublados e a roupa estava pendurada no varal.

Percebeu-se que este problema é algo comum, então com a ajuda dos integrantes Gabriel Nogas e Wally Jarek foi decidido juntar mais informações sobre o problema para assim criar uma solução.

Buscando soluções para o problema, foi decidido utilizar 2 arduinos MEGA 2560 e 1 arduino UNO.

O integrante Lucas Pierin, provido de conhecimentos de programação, foi designado de criar as partes principais do programa do varal. Já para o integrante Wally Jarek foi encarregada a tarefa de comunicação via ethernet com o projeto. O

integrante Gabriel ficou responsável por comprar os materiais e montar o protótipo do projeto.

Durante o decorrer do projeto houveram novas ideias e alguns problemas, todos sendo adaptados e corrigidos, para que o objetivo principal fosse mantido.

Com várias opções de comunicação, foi decidido utilizar a comunicação wireless e ethernet, com o wireless fazendo a comunicação entre os arduinos e o ethernet se comunicando com a rede.

Os comandos são enviados a partir de um celular ou da própria internet.

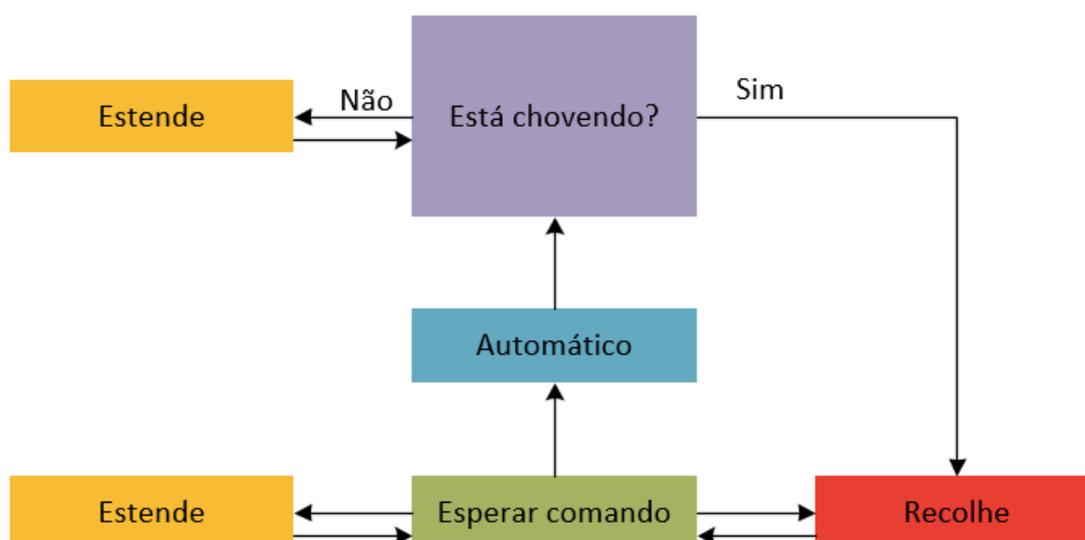
Quando houve a junção dos códigos dos sensores com o código principal do arduino, houveram problemas, que foram corrigidos o mais rápido possível.

Após a correção dos erros, foram testadas as utilidades do varal para comprovar que não havia nada fora do comum.

1.2. OBJETIVOS

1.2.1. Objetivo Geral

O objetivo deste projeto é trazer tranquilidade para as donas de casa na hora de cuidar da roupa que está secando no varal. Pois o Auto Varal Controler recolhe as roupas em caso de chuva. Isso ajuda, também, quando as roupas estão penduradas e não há ninguém em casa.



1.2.2. Objetivos Específicos

Os objetivos específicos do trabalho são:

- a) Instalação da casa do varal e encontrar um ponto estratégico para isso;
- b) Instalação do varal e dos pontos de comunicação;
- c) Teste de funcionamento.

2. ESTADO DA ARTE

Através da pesquisa de estado da arte não se identificou projetos com o mesmo funcionamento que o AVC. Entretanto, existem projetos que tem o mesmo objetivo, porém, são varais externos, sendo todos movidos por trilho. Por exemplo, o VARAL COM SENSOR DE CHUVA, segundo O site “Ele detecta umidade via sensor externo e recolhe o varal” (<http://robolivre.org/conteudo/varal-com-sensor-de-chuva> 2012). Entretanto não foi encontrado o aparelho para venda no mercado.



Figura 1 Modelo de Varal com sensor de chuva

Também foi encontrado no site: (<http://automares.blogspot.com.br> 2012) um projeto de varal inteligente que recolhe a roupa, porem no formato de trilhos.



Figura 2 Modelo de Varal com sensor de chuva



Figura 3 Modelo de Varal com sensor de chuva

Em nosso projeto o deslocamento será feito via tração por rodas. Os comandos serão emitidos via página web caso o usuário deseje fazê-lo remotamente ou por controle remoto localmente.

3.REFERENCIAL TEÓRICO

Para a realização deste projeto nós analisamos problemas vividos pelas nossas mães, fazendo algumas pesquisas sobre o tema escolhido também, e pesquisamos projetos semelhantes. Para uma referência acerca da programação utilizamos da referência disponível no próprio site do arduino e outra parte pesquisamos códigos semelhantes ou códigos disponíveis pelos fabricantes dos Shields ou adaptadores utilizados no projeto.

3.1 NRF24L01

Para a utilização do modulo de rádio NRF24L01 tivemos alguns problemas inicialmente assim tendo que fazer uma pesquisa mais detalhada sobre o mesmo, por ser um modulo bastante utilizado não foi difícil encontrar um código e explicação que se adequasse ao nosso projeto. No nosso projeto ele foi utilizado para comunicação da base fixa da casa com o Arduino que fica na base do varal.

3.2 ETHERNETS

No caso do Shield Ethernet utilizamos ele para a comunicação com a rede interna de sua casa e também ele cria um servidor web que hospeda a página de controle do varal.

3.3 Sensores de chuva

Nosso sensor de chuva é um um sensor analógico, ou seja, quando a agua cai no sensor fecha o circuito e "diz" para o Arduino que está chovendo. Assim se o varal está no modo automático vai recolher.

3.4 IMPACTOS AMBIENTAL E SUSTENTABILIDADE

Impacto ambiental é a alteração, por parte do ser humano ao meio aonde vive. O impacto pode ser negativo, ou até positivo, representando a quebra do equilíbrio ecológico. Sustentabilidade é o meio benéfico para tentar equilibrar as ações maléficas causadas anteriormente. Para ser considerado como tal, é necessário ser ecologicamente correto, economicamente viável, socialmente justo e culturalmente diverso.

O Auto Varal Controller tem esta visão de sustentabilidade, visando reduzir impacto ambiental do homem na natureza diminuindo o uso de secadoras de roupa que usam muita energia elétrica, assim reduzindo o custo para as famílias e utilizando menos dos recursos naturais que são utilizados para geração de energia.

4. METODOLOGIA

O Arduino recebe as informações do sensor de chuva que verifica se está chovendo ou não, caso esteja no modo automático o Arduino interpreta a situação, se estiver chovendo, o aplicativo envia o comando para os motores recolherem o varal de volta a sua posição inicial. O controle das funcionalidades do projeto é acessado via rede local.

Caso você escolha o modo automático e esteja chovendo, o varal não será colocado para fora, ficando parado no lugar e esperando o próximo comando do usuário.

4.1 MATERIAIS MECÂNICOS

- Base de madeira para carrinho;
- Palitos de churrasco;
- Linha de pesca;
- Parafusos.

4.2 MATERIAIS ELETRÔNICOS

- Arduino UNO;
- Arduino MEGA;
- Ethernet Shield;
- Motor Shield;
- Comunicador wirelles NRF24L01;

4.2 SOFTWARE

- Arduino IDE;
- Aptana.

5.O PROJETO

Através de uma análise da localização que será instalado o varal definimos o local onde ele ficara "recolhido" e a posição onde ele estará "estendido", tendo este dado em mãos programamos o arduino e posicionamos os sensores.

O sensor utilizado é um sensor de chuva normal que quando tem agua fecha O circuito assim o arduino detectando que está chovendo. Há três funções no varal, uma só para recolher ele, outra só para estender ele e uma terceira para deixar ele no automático, neste modo se não estiver chovendo ele vai para do lugar que ficara estendido e fica esperando por sinal de chuva ou o usuário pedir para recolher. Sendo esse modo o principal do projeto.

5.1 PROJETOS: ELETRO-MECÂNICO

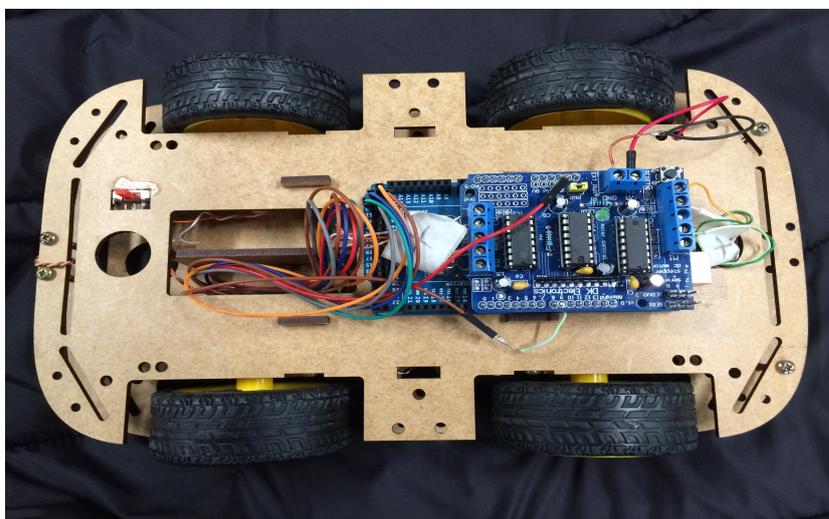


Figura 4 Visão superior da Base do varal

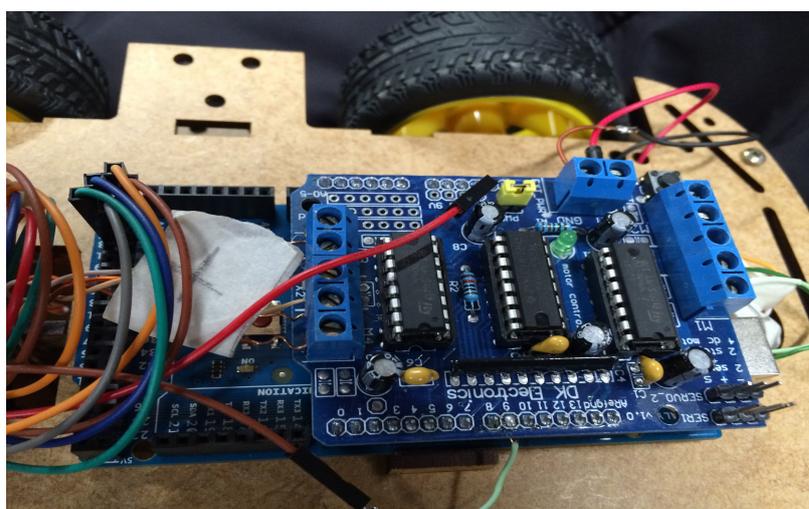
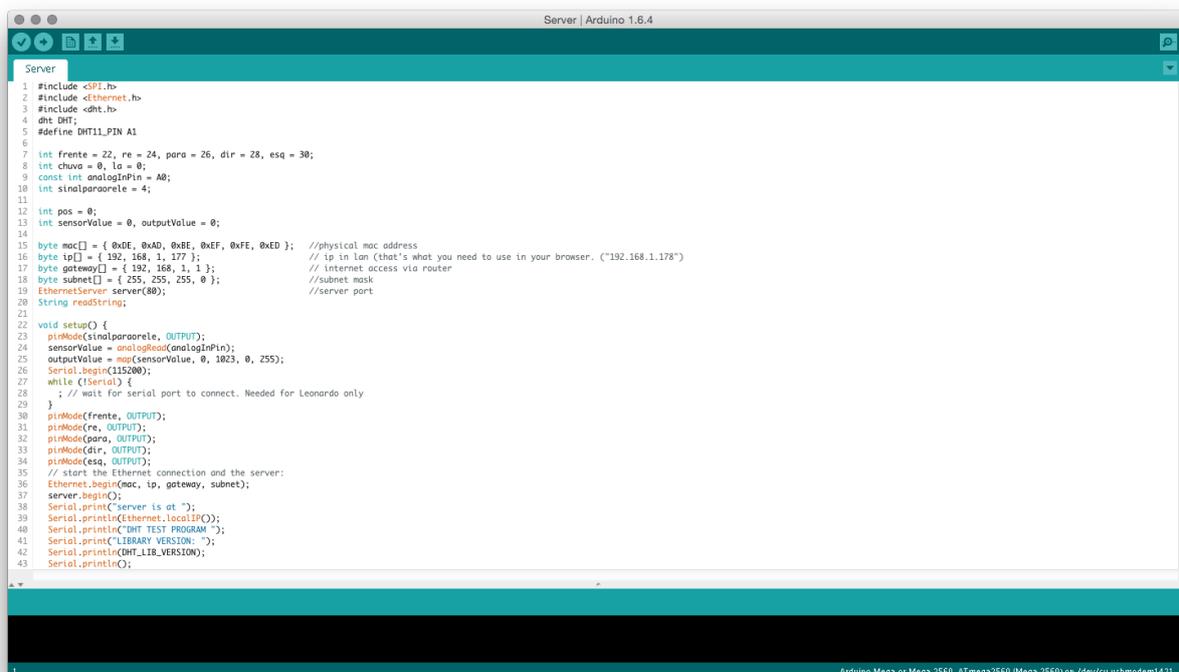


Figura 5 Arduíno com Motorshield da Base do varal

5.3 PROJETOS: SOFTWARE

Os softwares selecionados para a realização da do projeto foram:

- Arduino IDE; versão 1.5.4 (linguagem: C);



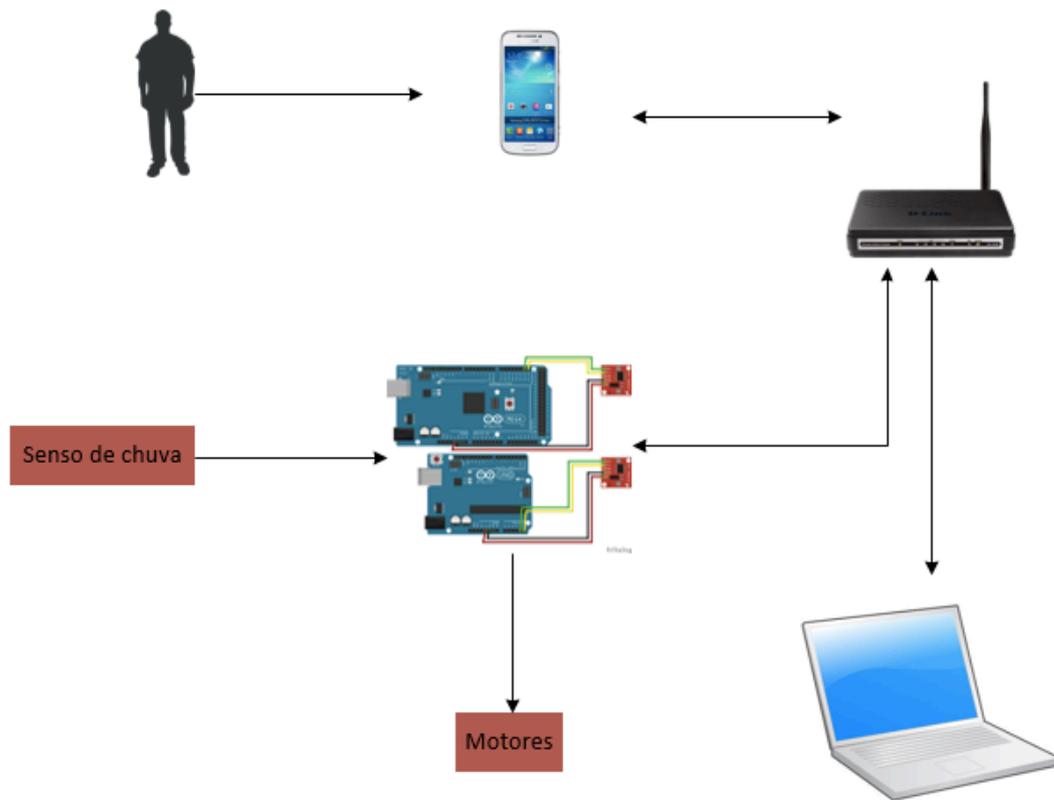
```
Server | Arduino 1.6.4
Server
1 #include <SPI.h>
2 #include <Ethernet.h>
3 #include <dht.h>
4 #define DHT;
5 #define DHT11_PIN A1
6
7 int frente = 22, re = 24, para = 26, dir = 28, esq = 30;
8 int chuva = 0, lo = 0;
9 const int analogInPin = A0;
10 int sinalparaorele = 4;
11
12 int pos = 0;
13 int sensorValue = 0, outputValue = 0;
14
15 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //physical mac address
16 byte ip[] = { 192, 168, 1, 177 }; // ip in lan (that's what you need to use in your browser. ("192.168.1.178")
17 byte gateway[] = { 192, 168, 1, 1 }; // internet access via router
18 byte subnet[] = { 255, 255, 255, 0 }; //subnet mask
19 EthernetServer server(80); //server port
20 String readString;
21
22 void setup() {
23   pinMode(sinalparaorele, OUTPUT);
24   sensorValue = analogRead(analogInPin);
25   outputValue = map(sensorValue, 0, 1023, 0, 255);
26   Serial.begin(115200);
27   while (!Serial) {
28     ; // wait for serial port to connect. Needed for Leonardo only
29   }
30   pinMode(frente, OUTPUT);
31   pinMode(re, OUTPUT);
32   pinMode(para, OUTPUT);
33   pinMode(dir, OUTPUT);
34   pinMode(esq, OUTPUT);
35   // start the Ethernet connection and the server:
36   Ethernet.begin(mac, ip, gateway, subnet);
37   server.begin();
38   Serial.print("server is at ");
39   Serial.println(Ethernet.localIP());
40   Serial.println("DHT TEST PROGRAM ");
41   Serial.print("LIBRARY VERSION: ");
42   Serial.println(DHT_LIB_VERSION);
43   Serial.println();
44 }
```

1 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) en /dev/cu.usbmodem1421

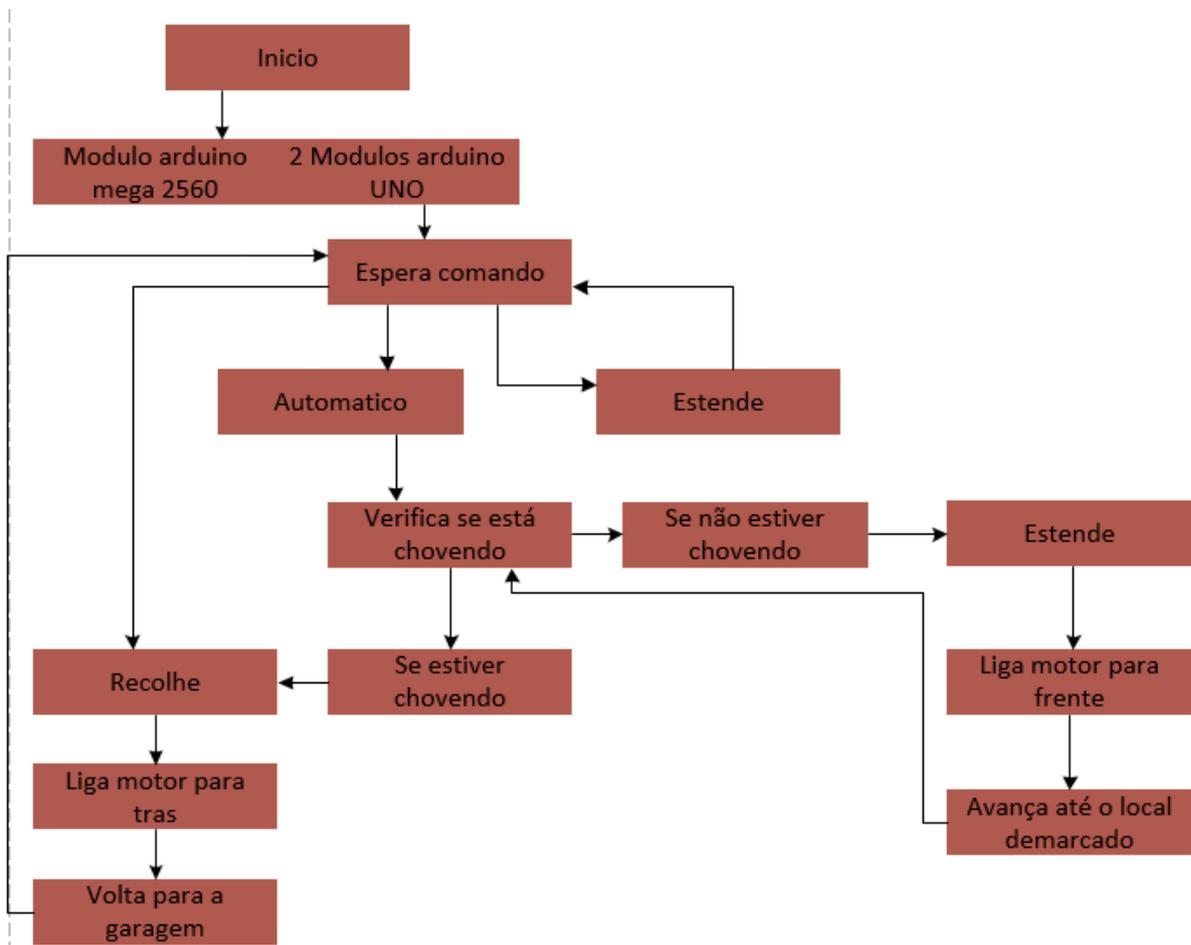
Figura 6 Imagem Software Arduino

- Aptana Studio 3; versão 3.6.1 (Linguagem PHP+HTML+CSS);
Arduino IDE

5.4 FLUXOGRAMA ESTRUTURAL



5.5 FLUXOGRAMA COMPORTAMENTAL



6.RESULTADOS

Com a divisão de tarefas, cada qual ficou encarregado de fazer uma parte do Trabalho, portanto foram feitos trabalhos simultâneos, com a obtenção do hardware Arduino Mega teve o início das atividades, a princípio criamos uma ponte H para controle do motor que faria o deslocamento, porém decidimos que seria necessária uma ponte H para 4 motores independentes.

Adquirindo o modulo de comunicação wi fi se desencadearam vários problemas de comunicação, assim foi definido que a comunicação seria feita separadamente. Uma local entre os arduinos e uma para a ethernet para o controle remoto.

7.IMPACTO AMBIENTAL

Sustentabilidade e impacto ambiental, palavras que hoje em dia tem um significado muito grande. É considerada em praticamente todo tipo de projeto criado, para que não haja o desperdício dos materiais, tudo que é criado deve ter uma forma de ser reutilizado ou reciclado. O projeto está envolvido na sustentabilidade, pois seus componentes podem ser reutilizados e ajudam a economizar energia elétrica que seria gasta em uma secadora de roupas. A criação em larga escala, pode gerar uma grande economia de energia, quando um dia está nublado e as pessoas saem de casa, além de seus materiais que podem ser reutilizados e reciclados.

	REUTILIZAÇÃO	RECICLAGEM	DESCARTE	LEI DE CHUMBO
ARDUINO MEGA	SIM	SIM	CENTRO DE TRIAGEM DE RECICLAGEM	RESPEITA
ARDUINO UNO	SIM	SIM	CENTRO DE TRIAGEM DE RECICLAGEM	RESPEITA
ETHERNET SHIELD	SIM	SIM	CENTRO DE TRIAGEM DE RECICLAGEM	RESPEITA
NRF24L01	SIM	SIM	CENTRO DE TRIAGEM DE RECICLAGEM	RESPEITA
MOTOR	SIM	SIM	CENTRO DE TRIAGEM DE RECICLAGEM	RESPEITA
PALITOS DE MADEIRA	SIM	SIM	LIXO RECICLÁVEL	RESPEITA
COMPENSADO	SIM	SIM	LIXO RECICLÁVEL	RESPEITA
ROTEADOR	SIM	SIM	CENTRO DE TRIAGEM DE RECICLAGEM	RESPEITA
LINHA DE PESCA	NÃO	SIM	LIXO RECICLÁVEL	RESPEITA
RODAS	SIM	SIM	LIXO RECICLÁVEL	RESPEITA

8. CONSIDERAÇÕES FINAIS

Com o objetivo de otimizar o processo de secagem de roupas, o projeto mostrou-se bastante eficiente ao evitar a exposição da roupa à chuva. Desta forma as roupas permanecem expostas durante bom tempo, fazendo o melhor aproveitamento das áreas descobertas - sacadas e terraços – quando as condições climáticas permitem e proporcionando o recolhimento automático do varal na detecção de alta humidade e grande probabilidade de chuva.

Além da recolha automática do varal para uma área coberta e pré-determinada, este projeto oferece a possibilidade de secagem por ar, quanto à exposição natural não possibilitou a secagem total das roupas. Este processo evita a necessidade da remoção das roupas do varal para secagem através de secadora, eletrodoméstico de alto consumo energético.

Esta opção poderá ser desativada pelo usuário, quando entender que a secagem pode ser finalizada sob a área coberta – especialmente em dias quentes.

O projeto busca proporcionar comodidade nas atividades domésticas, principalmente a indivíduos que passam o dia todo fora de casa e não conseguem aproveitar as áreas descobertas de suas residências no período de incidência de sol e vento, partindo como uma proposta sustentável, que permite o aproveitamento dos recursos naturais e reduz significativamente a utilização de secadoras de roupa, que são equipamentos de alto consumo energético.

REFERÊNCIAS

Robô Livre. **Varal com Sensor de Chuva.** Disponível em: <http://robolivre.org/conteudo/varal-com-sensor-de-chuva>. Acesso em: 04 jun. 2015.

Patentes Online. **Varal eletrônico com sensor de umidade.** Disponível em: <http://www.patentesonline.com.br/varal-eletr-nico-com-sensor-de-umidade-102032.html> Acesso em: 04 jun. 2015.

Arduino Playground, Nrf24L01. **Arduino Sensor nrf24l01.** Disponível em: <http://playground.arduino.cc/InterfacingWithHardware/Nrf24L01>

Arduino e Cia. **Comunicação wireless com Arduino e módulo NRF24L01 2.4GHz.** Disponível em: <http://www.arduinoecia.com.br/2015/02/comunicacao-wireless-arduino-nrf24l01.html>

ANEXO A – PROGRAMAÇÃO DO ARDUINO UNO DA BASE

Neste Arduino temos conectado o ethernet shield e o sensor de chuva, sendo ele (o arduino) a central de comunicação externa e controle do sistema.

```
#include <SPI.h>
#include <Ethernet.h>

int frente = 22, re = 24, para = 26, dir = 28, esq = 30;
int chuva = 0, la = 0;
const int analogInPin = A0;

int pos = 0;
int sensorValue = 0, outputValue = 0;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 1, 177 };
byte gateway[] = { 192, 168, 1, 1 };
byte subnet[] = { 255, 255, 255, 0 };
EthernetServer server(80);
String readString;

void setup() {
  sensorValue = analogRead(analogInPin);
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
  pinMode(frente, OUTPUT);
  pinMode(re, OUTPUT);
  pinMode(para, OUTPUT);
  pinMode(dir, OUTPUT);
  pinMode(esq, OUTPUT);
  // start the Ethernet connection and the server:
  Ethernet.begin(mac, ip, gateway, subnet);
  server.begin();
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  // READ DATA
  Serial.print("Agua: ");
  Serial.println(outputValue);
  delay(1000);
  // Create a client connection
  EthernetClient client = server.available();
  if (client) {
    while (client.connected()) {
```

```

if (client.available()) {
    char c = client.read();

    //read char by char HTTP request
    if (readString.length() < 100) {
        //store characters to string
        readString += c;
        //Serial.print(c);
    }
}
//INICIO DA PAGINA WEB COM INCLUSAO DE CSS E JAVASCRIPT EXTERNO
//if HTTP request has ended
if (c == '\n') {
    client.println("HTTP/1.1 200 OK"); //send new page
    client.println("Content-Type: text/html");
    client.println();
    client.println("<HTML>");
    client.println("<HEAD>");
    client.println("<link rel='stylesheet' type='text/css'
href='http://192.168.1.2:8888/css/bootstrap.css' />");
    client.println("<link rel='stylesheet' type='text/css'
href='http://192.168.1.2:8888/WGL/wgl.css' />");
    client.println("<TITLE>WGL Corporation</TITLE>");
    client.println("</HEAD>");
    client.println("<BODY>");
    client.println("<nav class='navbar navbar-inverse navbar-fixed-top'>");
    client.println("<div class='container'>");
    client.println("<div class='navbar-header'>");
    client.println("<button type='button' class='navbar-toggle collapsed' data-toggle='collapse'
data-target='#navbar' aria-expanded='false' aria-controls='navbar'>");
    client.println("<span class='sr-only'>Toggle navigation</span>");
    client.println("<span class='icon-bar'></span><span class='icon-bar'></span><span
class='icon-bar'></span>");
    client.println("</button>");
    client.println("<a class='navbar-brand' href='http://www.wglcorp.com'>WGL</a>");
    client.println("</div>");
    client.println("<div id='navbar' class='collapse navbar-collapse'>");
    client.println("<ul class='nav navbar-nav'>");
    client.println("<li><a href='http://www.wglcorp.com'>Home</a></li>");
    client.println("<li><a href='http://www.wglcorp.com/sobre.html'>Sobre</a></li>");
    client.println("<li><a href='http://www.wglcorp.com/contato.html'>Contato</a></li>");
    client.println("</ul></div></div></nav>");
    client.println("<div class='container'><div class='jumbotron'>");
    client.println("<p align='center'><H2>Auto Varal Controller Prototype</H2>");
    client.println("<br />");
    client.println("<a class='btn btn-success' href='\"/?fren\"'>Frente</a>");
    client.println("<a class='btn btn-success' href='\"/?re\"'>Re</a>");
    client.println("<a class='btn btn-danger' href='\"/?par\"'>PARA</a>");
    client.println("<a class='btn btn-warning' href='\"/?dir\"'>Esquerda</a>");
    client.println("<a class='btn btn-warning' href='\"/?esq\"'>Direita</a>");
    client.println("<a class='btn btn-primary' href='\"/?au\"'>AUTO</a><br /></p>");
    client.println("</div></div>");
    client.println("<script src='http://192.168.1.2:8888/js/bootstrap.min.js'></script>");
    client.println("</BODY>");
    client.println("</HTML>");
    delay(1);
}

```

```

//stopping client
client.stop();
//controls the Arduino if you press the buttons
if (readString.indexOf("?fren") > 0) {
  digitalWrite(frente, HIGH);
  digitalWrite(re, LOW);
  digitalWrite(para, LOW);
  digitalWrite(dir, LOW);
  digitalWrite(esq, LOW);
}
if (readString.indexOf("?re") > 0) {
  digitalWrite(frente, LOW);
  digitalWrite(re, HIGH);
  digitalWrite(para, LOW);
  digitalWrite(dir, LOW);
  digitalWrite(esq, LOW);
}
if (readString.indexOf("?par") > 0) {
  digitalWrite(frente, LOW);
  digitalWrite(re, LOW);
  digitalWrite(para, HIGH);
  digitalWrite(dir, LOW);
  digitalWrite(esq, LOW);
}
if (readString.indexOf("?dir") > 0) {
  digitalWrite(frente, LOW);
  digitalWrite(re, LOW);
  digitalWrite(para, LOW);
  digitalWrite(dir, HIGH);
  digitalWrite(esq, LOW);
}
if (readString.indexOf("?esq") > 0) {
  digitalWrite(frente, LOW);
  digitalWrite(re, LOW);
  digitalWrite(para, LOW);
  digitalWrite(dir, LOW);
  digitalWrite(esq, HIGH);
}
if (readString.indexOf("?au") > 0) {
  Serial.println("Enviado : A - AUTOMATICO");
  if (outputValue > 250 ) {
    Serial.println("Sem Chuva");
    Serial.print("Agua: ");
    Serial.println(outputValue);
    if (la == 0) {
      digitalWrite(frente, HIGH);
      digitalWrite(re, LOW);
      digitalWrite(para, LOW);
      digitalWrite(dir, LOW);
      digitalWrite(esq, LOW);
      Serial.println("Saiu");
      delay(1000);
      la == 1;
    }
  }
  while (chuva == 0) {

```

```

sensorValue = analogRead(analogInPin);
outputValue = map(sensorValue, 0, 1023, 0, 255);
Serial.println("While");
Serial.print("Umidade: ");
Serial.println(DHT.humidity);
Serial.print("Agua: ");
Serial.println(outputValue);
delay(1000);
if (outputValue < 250) {
  //RE
  digitalWrite(frente, LOW);
  digitalWrite(re, HIGH);
  digitalWrite(para, LOW);
  digitalWrite(dir, LOW);
  digitalWrite(esq, LOW);
  delay(1000);
  //FIMRE
  Serial.println("Chuva! Agua");
  chuva = 1;
  la = 0;
}
digitalWrite(frente, LOW);
digitalWrite(re, LOW);
digitalWrite(para, HIGH);
digitalWrite(dir, LOW);
digitalWrite(esq, LOW);
}
readString = "";
} else {
  Serial.println("Chuva!");
  digitalWrite(frente, LOW);
  digitalWrite(re, LOW);
  digitalWrite(para, HIGH);
  digitalWrite(dir, LOW);
  digitalWrite(esq, LOW);
  la = 0;
}
}
//Zerar string para a próxima leitura
readString = "";
}}}}

```

ANEXO B – PROGRAMAÇÃO DO ARDUINO MEGA DA BASE

Este Arduino será o responsável por transmitir os comandos dados via Ethernet para o arduino que está instalado no carrinho.

```
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"

int frente = 3, re = 4, para = 5, dir = 6, esq = 7, agua = 80;
int f = 0, r = 0, p = 0, d = 0, e = 0, a = 0, chuva = 0;

// Armazena caractere digitado na serial
char valor[1];

// Armazena os dados enviados
int dados[1];

// Inicializa o NRF24L01 nos pinos 9 (CE) e 10 (CS) do Arduino Uno
RF24 radio(9, 10);

// Define o endereco para comunicacao entre os modulos
const uint64_t pipe = 0xE13CBAF433LL;

void setup()
{
  pinMode(frente, INPUT);
  pinMode(re, INPUT);
  pinMode(para, INPUT);
  pinMode(dir, INPUT);
  pinMode(esq, INPUT);
  pinMode(agua, INPUT);

  // Inicializa a serial
  Serial.begin(57600);
  //Serial.println("Digite F para Frente, T para Re e P para Parar");

  // Inicializa a comunicacao do NRF24L01
  radio.begin();
  // Entra em modo de transmissao
  radio.openWritingPipe(pipe);
}

void loop()
{
  //INICIO PINOS ETHERNET
  f = digitalRead(frente);
  r = digitalRead(re);
  p = digitalRead(para);
  d = digitalRead(dir);
  e = digitalRead(esq);
  a = digitalRead(agua);
  //FIM PINOS ETHERNET
  // Le o caractere digitado na serial
```

```
if (Serial.available() > 0)
{
  // valor[0] = Serial.read();
}
// Envia 1 via radio caso seja digitado o valor F
if (f == HIGH || valor[0] == 'f')
{
  Serial.println("Enviado : F - FRENTE");
  dados[0] = 1;
  radio.write(dados, 1);
  delay(100);
  valor[0] = 0;
}
// Envia 2 via radio caso seja digitado o valor R
if (r == HIGH || valor[0] == 'r')
{
  Serial.println("Enviado : T - RE");
  dados[0] = 2;
  radio.write(dados, 1);
  delay(100);
  valor[0] = 0;
}
if (p == HIGH || valor[0] == 'p')
{
  Serial.println("Enviado : P - PAROU");
  dados[0] = 3;
  radio.write(dados, 1);
  delay(100);
  valor[0] = 0;
}
if (d == HIGH || valor[0] == 'd')
{
  Serial.println("Enviado : D - DIREITA");
  dados[0] = 4;
  radio.write(dados, 1);
  delay(100);
  valor[0] = 0;
}
if (e == HIGH || valor[0] == 'e')
{
  Serial.println("Enviado : E - ESQUERDA");
  dados[0] = 5;
  radio.write(dados, 1);
  delay(100);
  valor[0] = 0;
}
}
```

ANEXO C – PROGRAMAÇÃO DO ARDUINO MEGA DO CARRINHO

Este Arduino sera o responsável por receber os comandos e controlar os motores da base do nosso varal.

```
#include <AFMotor.h>
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"

AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

// Armazena os dados recebidos
int recebidos[1];
// Inicializa o NRF24L01 nos pinos 9 (CE) e 53 (CS) do Arduino Mega
RF24 radio(9, 53);
// Define o endereco para comunicacao entre os modulos
const uint64_t pipe = 0xE13CBAF433LL;
// Define o pino do led
int LED1 = 46, velo = 190;

void setup()
{
  motor1.setSpeed(velo);
  motor2.setSpeed(velo);
  motor3.setSpeed(velo);
  motor4.setSpeed(velo);
  // Define o pino do led como saida
  pinMode(LED1, OUTPUT);
  // Inicializa a serial
  Serial.begin(57600);
  // Inicializa a comunicacao do NRF24L01
  radio.begin();
  // Entra em modo de recepcao
  radio.openReadingPipe(1, pipe);
  radio.startListening();
  // Mensagem inicial
  Serial.println("Aguardando dados...");
}

void loop()
{
  // Verifica sinal de radio
  if (radio.available())
  {
    digitalWrite(LED1, HIGH);
    bool done = false;
    while (!done)
    {
      done = radio.read(recebidos, 1);
      Serial.print("Recebido : ");
      Serial.print(recebidos[0]);
    }
  }
}
```

```
// Se recebeu 1, movimenta o servo para a esquerda
if (recebidos[0] == 1)
{
  motor1.setSpeed(velo);
  motor2.setSpeed(velo);
  motor3.setSpeed(velo);
  motor4.setSpeed(velo);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  delay(10);
  Serial.println(" -> FRENTE");
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  delay(2000);
}

// Se recebeu 2, movimenta o servo para a direita
if (recebidos[0] == 2)
{
  motor1.setSpeed(velo);
  motor2.setSpeed(velo);
  motor3.setSpeed(velo);
  motor4.setSpeed(velo);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  delay(10);
  Serial.println(" -> RE");
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
  delay(2000);
}
if (recebidos[0] == 3)
{
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  delay(10);
  Serial.println(" -> PAROU");
}
if (recebidos[0] == 4)
{
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
```

```
delay(10);
motor2.setSpeed(velo / 2);
motor3.setSpeed(velo / 2);
Serial.println(" -> Direita");
motor1.run(FORWARD);
motor2.run(FORWARD);
motor3.run(FORWARD);
motor4.run(FORWARD);
}
if (recebidos[0] == 5)
{
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  delay(10);
  motor1.setSpeed(velo / 2);
  motor4.setSpeed(velo / 2);
  Serial.println(" -> Esquerda");
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
}
}
}
```

ANEXO D - FOTOS DO PROJETO

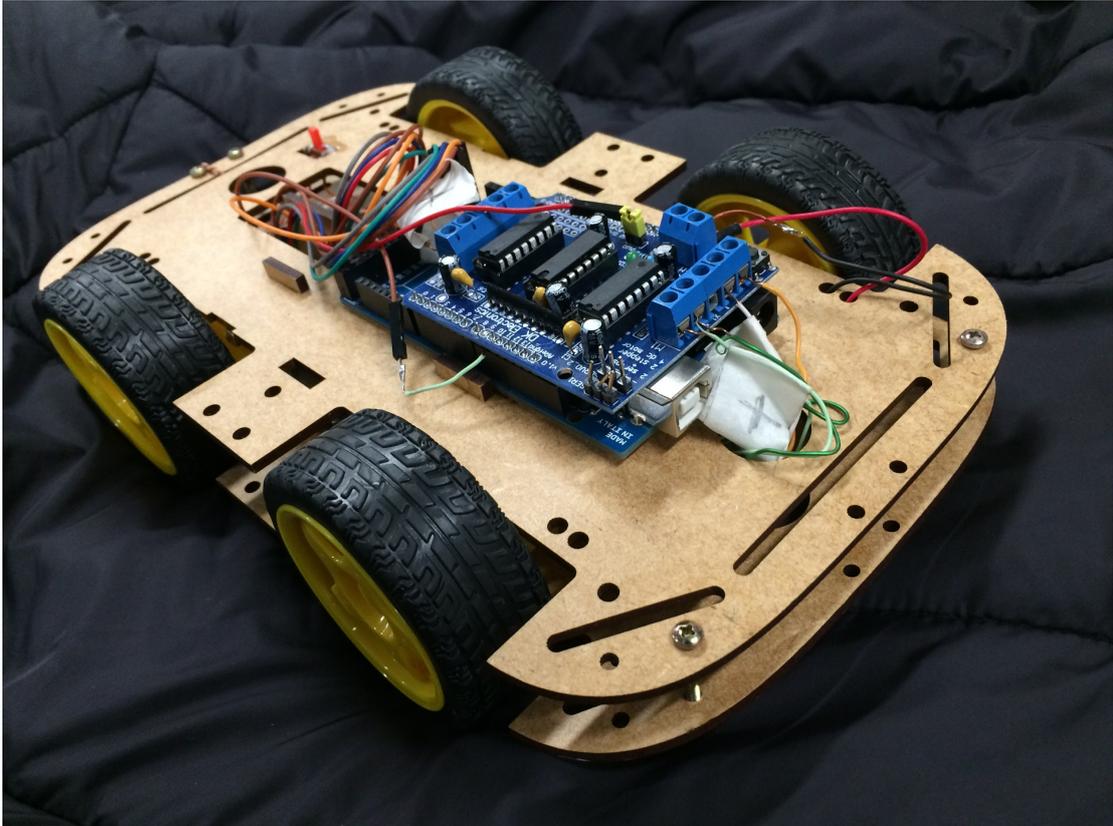


Figura 7 Foto da base

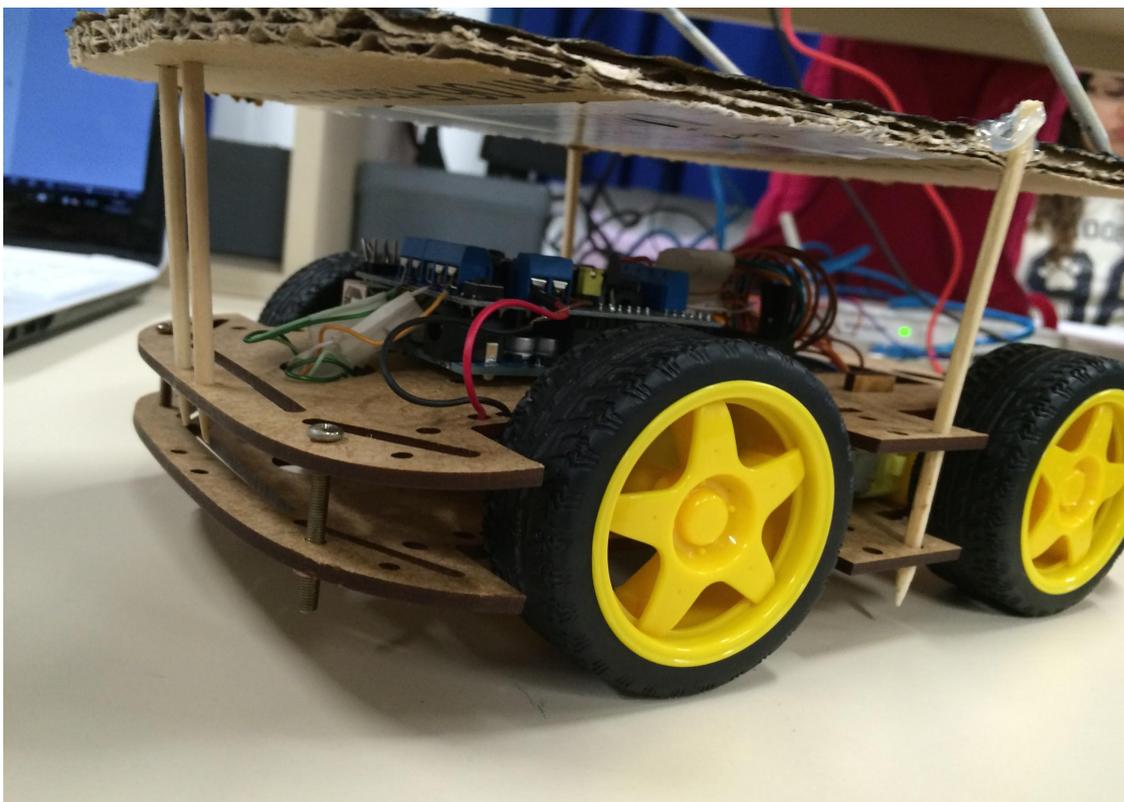


Figura 8 Foto da base

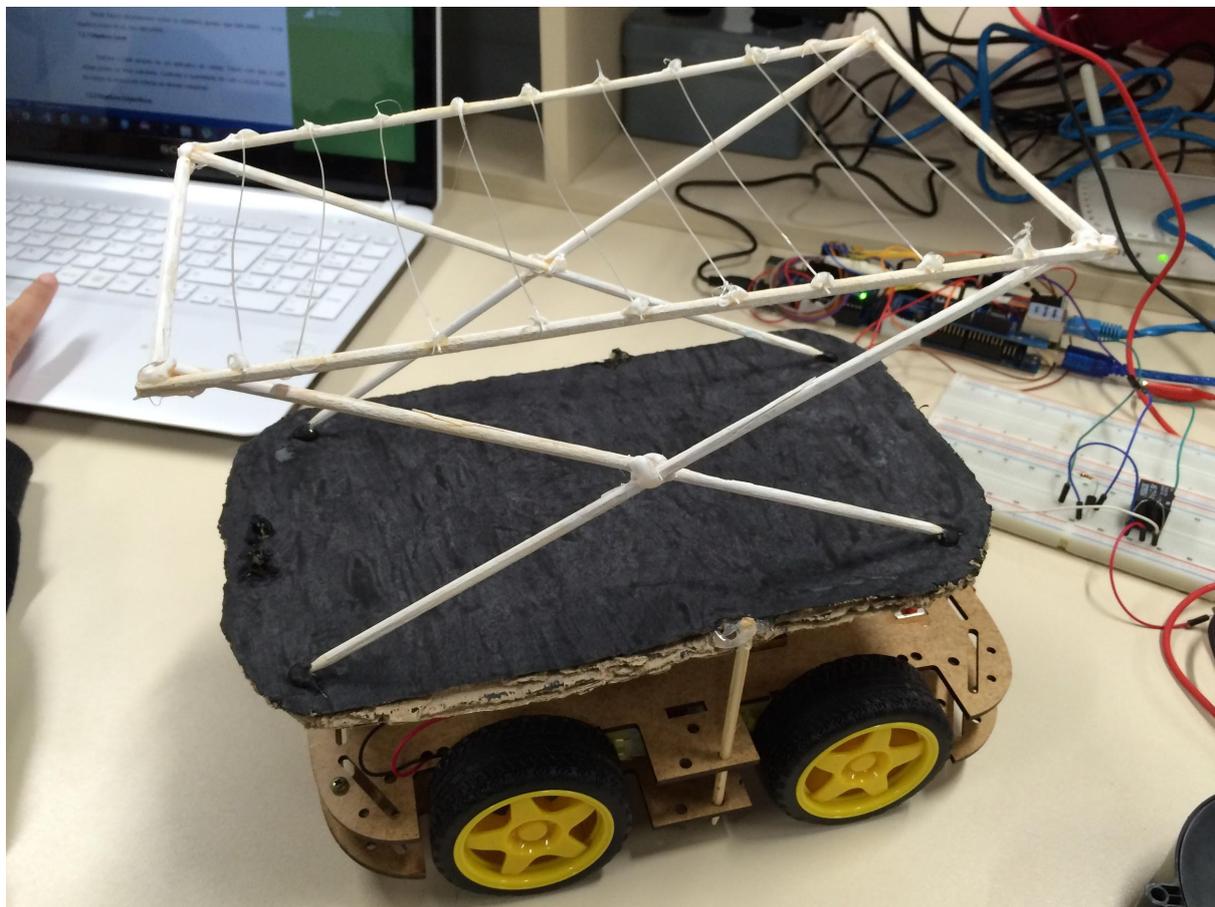


Figura 10 Foto do AVC

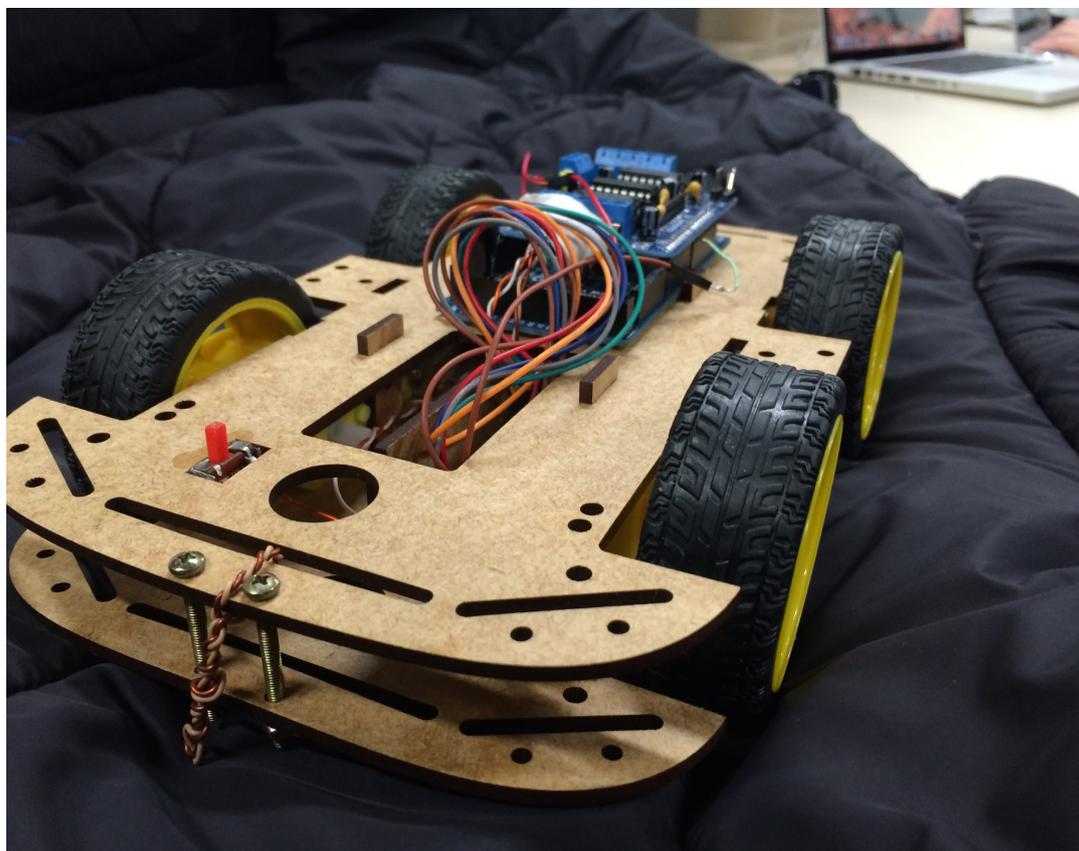


Figura 9 Foto da base