

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO
CURSO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Alcyr da Costa Almeida Junior

**APRIMORAMENTO DE UMA PLATAFORMA PARA ENSINO DE
CONCEITOS BÁSICOS DE LÓGICA DE PROGRAMAÇÃO
UTILIZANDO ARDUINO E ANDROID**

Belém/PA
2017

Alcyr da Costa Almeida Junior

APRIMORAMENTO DE UMA PLATAFORMA PARA ENSINO DE
CONCEITOS BÁSICOS DE LÓGICA DE PROGRAMAÇÃO
UTILIZANDO ARDUINO E ANDROID

Trabalho de Conclusão de Curso
apresentado ao Instituto de Ciências
Exatas e Naturais da Universidade
Federal do Pará como requisito parcial
para a obtenção do título de Bacharel em
Ciência da Computação.

Orientador Prof. Dr. Dionne Cavalcante
Monteiro

Belém/PA
2017

Alcyr da Costa Almeida Junior

APRIMORAMENTO DE UMA PLATAFORMA PARA ENSINO DE
CONCEITOS BÁSICOS DE LÓGICA DE PROGRAMAÇÃO
UTILIZANDO ARDUINO E ANDROID

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Aprovado em: 27 de outubro de 2017

Banca Examinadora

MSc. Fiterlinge Martins de Sousa – UFPA

Dr. Raimundo Viégas Júnior – UFPA

Dr. Dionne Cavalcante Monteiro – UFPA (orientador)

*Dedico este trabalho a todos os
que contribuíram direta ou
indiretamente na minha educação.
Especialmente à minha família.*

AGRADECIMENTOS

Agradeço à minha família pela confiança e por estar comigo, especialmente à minha mãe, Selma, pelo carinho, pela ajuda e suporte.

Agradeço à Denise Tanaka, pela companhia, pela inspiração, pela ajuda em decisões importantes e pelos momentos bons que tivemos juntos.

Agradeço ao meu orientador, Dionne Monteiro, por me guiar na elaboração desse trabalho e pelos conhecimentos no decorrer do curso.

Agradeço ao Victor Negrão, pelo incentivo, por ser um grande amigo e por se oferecer gentilmente para revisar este Trabalho.

Agradeço aos amigos do CBCC 2012, especialmente ao Alexandre Freitas, Lana Priscila, Luciana Brito, Matheus Rodrigues, Moisés Felipe e Walisson Cardoso, entre todos os que fizeram parte dessa jornada, com os quais vivi e aprendi.

Agradeço aos colegas de trabalho no Núcleo de Informática do IPAMB: André, Arnoldo Dhony, Gleise, Heloisa, Kalife, Marivaldo, Victor e os outros que não estão mais no Instituto, pelo apoio e pelas brincadeiras.

Agradeço aos demais professores que participaram da minha formação profissional.

“Tente aprender alguma coisa sobre tudo e tudo sobre alguma coisa.”

Thomas Huxley

RESUMO

A utilização da informática está se tornando cada vez mais presente no cotidiano. Porém, tem-se percebido que a dificuldade na utilização dessa tecnologia devido à alta carga de conceitos abstratos envolvida. Para tentar contornar esse problema, Negrão (2015) desenvolveu uma plataforma para ensino de lógica de programação baseada no primo (Primo Toys, 2016) com foco em crianças até 6 anos de idade, composta por um *tablet* Android e um carro com *hardware* baseado em Arduino. Nela, a criança cria um programa a partir do *tablet* e depois envia para o carro, que se move de acordo com o que foi programado. O objetivo deste trabalho é apresentar uma versão aprimorada da plataforma desenvolvida por Negrão (2015) como a criação de um programa supervisor onde o professor pode acompanhar o desenvolvimento dos alunos e a modificação da infraestrutura da rede, permitindo que vários equipamentos possam ser utilizados em uma mesma rede *wireless*.

Palavras Chave: Educação, Lógica de Programação, Android, Arduino

ABSTRACT

The use of information technology is becoming more and more present in everyday life. However, it has been realized that the difficulty in using this technology due to the high load of abstract concepts involved. To try to get around this problem, Negrão (2015) developed a platform for teaching programming based on cousin (Primo Toys, 2016) focused on children up to 6 years old, consisting of an Android tablet and a car with hardware based on Arduino. In it, the child creates a program from the tablet and then sends it to the car, which moves according to what has been programmed. The objective of this work is to present an improved version of the platform developed by Negrão (2015) as the creation of a supervisory program where the teacher can follow the development of the students and the modification of the network infrastructure, allowing several equipment to be used in a same wireless network.

Keywords: Education, Programming Logic, Android, Arduino

ÍNDICE DE FIGURAS

Figura 1: Lego Mindstorms EV3	16
Figura 2: Robô feito a partir do Lego Mindstorms EV3.....	17
Figura 3: Blocos de comandos do Lego Mindstorms EV3.....	17
Figura 4: Interface do Scratch	18
Figura 5: Componentes da plataforma Primo.....	19
Figura 6: Plataforma desenvolvida por Negrão (2015).....	19
Figura 7: Comunicação entre o Atmega 32u4 e o AR9331	21
Figura 8: Arduino Yún.....	21
Figura 9: CI ULN2003A	22
Figura 10: Motor de passo 28BYJ-48.....	22
Figura 11: Montagem na protoboard	24
Figura 12: Diagrama esquemático	24
Figura 13: Montagem do carro	25
Figura 14: Percurso s=ADFFFA&f=AE	27
Figura 15: Diagrama de Atividades do carro	28
Figura 16: Tela de login da aplicação Android	30
Figura 17: Design novo	31
Figura 18: Diagrama de atividades - Drag and Drop.....	31
Figura 19: Diagrama de casos de uso.....	33
Figura 20: Diagrama de entidade-relacionamento	33
Figura 21: Tela de login	34
Figura 22: Lista de alunos	35
Figura 23: Tela de atividades por aluno.....	35
Figura 24: Cadastro de professor	36
Figura 25: Tela de comentário	36

ÍNDICE DE TABELAS

Tabela 1: Especificações do microprocessador Linux	20
Tabela 2: Especificações do Arduino Yún.....	21
Tabela 3: Conexões.....	23

LISTA DE SÍMBOLOS

CSS	<i>Cascading Style Sheets</i>
CI	Circuito Integrado
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
IP	<i>Internet Protocol</i>
USB	<i>Universal Serial Bus</i>
XML	<i>eXtensible Markup Language</i>
V	Volt
A	Ampère
URL	<i>Uniform Resource Locator</i>
mAh	miliampère-hora

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVO GERAL	13
1.2 OBJETIVOS ESPECÍFICOS	14
1.3 METODOLOGIA.....	14
1.4 MATERIAIS UTILIZADOS	15
1.5 CONTRIBUIÇÃO DO TRABALHO	15
1.6 ORGANIZAÇÃO DO TRABALHO	15
2 TRABALHOS RELACIONADOS	16
2.1 LEGO MINDSTORMS	16
2.2 SCRATCH	17
2.3 PRIMO.....	18
2.4 PLATAFORMA DESENVOLVIDA POR NEGRÃO (2015).....	19
3 PROPOSTA DE HARDWARE E SOFTWARE DO CARRO	20
3.1 ARDUINO YÚN	20
3.2 CIRCUITO INTEGRADO ULN2003A	22
3.3 MOTOR DE PASSO 28BYJ-48	22
3.4 ALIMENTAÇÃO	23
3.5 CONEXÕES ENTRE O ARDUINO E OS MOTORES DE PASSO	23
3.6 MONTAGEM	24
3.7 SOFTWARE DO ARDUINO.....	25
3.7.1 Declarações	26
3.7.2 Metodo Setup().....	26
3.7.3 Metodo Loop()	26
4 PROPOSTA DE SOFTWARE	29
4.1 ANDROID	29
4.1.1 Classe ConsultaTxt	30

4.1.2 <i>Activity</i> Tela_Login.....	30
4.1.1 <i>Activity</i> DragDrop.....	30
4.2 SUPERVISÓRIO	32
5 AVALIAÇÃO DOS RESULTADOS	38
6 CONCLUSÃO	39
6.1 TRABALHOS FUTUROS	39
REFERÊNCIAS.....	40
ANEXOS	42

1 INTRODUÇÃO

A utilização da informática em favor do ensino tem se tornado uma realidade cada vez mais presente em nossos dias. Mattei (2016) afirma que devido ao poder de processamento de informações, hoje o computador ocupa um lugar de destaque entre as novas tecnologias e, por isso, deve ser visto como uma ferramenta que pode contribuir para o processo de aprendizagem.

Apesar disso, tem-se percebido que grande parte dos iniciantes de cursos de tecnologia apresenta dificuldades ao iniciar suas vidas como programadores (Aureliano & Tedesco, 2012). Esse problema ocorre devido à alta carga de conceitos abstratos envolvida, além de requerer o entendimento de uma linguagem específica e da lógica de programação (Ferreira et . al., 2009).

Além dos cursos de tecnologia, o conhecimento da lógica de programação auxilia no desenvolvimento de certas habilidades cognitivas que a programação exige, como raciocínio lógico, raciocínio matemático e causa e consequência. Segundo Furia (2014), os adultos pensam que o ensino da lógica de programação serve apenas para desenvolver novos profissionais de tecnologia e que as aulas e oficinas de robótica e criação de jogos e aplicativos são apenas um complemento ao ensino formal enquanto que especialistas em educação e tecnologia defendem que a programação deve fazer parte do currículo escolar desde a infância.

Essas vantagens já foram notadas em alguns lugares do mundo. Na Inglaterra, as leis obrigam que a lógica de programação seja ensinada para crianças, porém com foco apenas para o desenvolvimento de *softwares* do dia a dia. Por isso foi criado o *Raspberry Pi*, um computador de dimensões reduzidas e vendido a um custo baixo. O objetivo do *Raspberry Pi* é incentivar as crianças a criar seus próprios jogos. Segundo Mullins, conforme citado por Globo News (2012) “As crianças viam os computadores pessoais como uma caixa de Lego. Estamos tentando recriar aquela situação na qual o natural é pensar no que se quer criar com o *Raspberry Pi*, em vez de apenas pensar em que jogo se quer jogar”

1.1 OBJETIVO GERAL

O objetivo deste trabalho é aprimorar uma plataforma de ensino de lógica de programação para crianças desenvolvida por Negrão (2015), que consiste em um

smartphone Android onde a criança pode inserir uma fila de comandos como “Andar”, “Virar à direita” e “Virar à esquerda” e, em seguida, esses comandos são enviados para um carro controlado por um Arduino, que executará esses movimentos na ordem inserida. Entenda-se por aprimorar, a resolução de problemas conhecidos como a falta de precisão do carro, a modificação da infraestrutura para que seja possível a utilização de mais de um conjunto de *smartphone*/carro em uma mesma rede *wireless* e o desenvolvimento de um sistema supervisorio onde o professor possa acompanhar a atividade do aluno e registrar seus comentários sobre a mesma.

1.2 OBJETIVOS ESPECÍFICOS

- Estudar a plataforma desenvolvida por Negrão (2015) (*software* e *firmware*);
- Desenvolver uma nova versão do *hardware* do carro;
- Ampliar a rede de comunicação entre Arduino e Android;
- Desenvolver um banco de dados e uma interface de visualização dos dados enviados pelo aplicativo Android.
- Alterar o código-fonte do carro e do aplicativo Android para que se comuniquem com o servidor;

1.3 METODOLOGIA

Para o desenvolvimento deste trabalho, inicialmente foi feito um estudo da plataforma desenvolvida por Negrão (2015) a fim de fazer o levantamento das alterações necessárias para que a nova versão da plataforma alcançasse o resultado esperado.

Após este estudo, partiu-se para a escolha do *hardware* necessário para a construção do carro assim como foi realizada a sua montagem e, em seguida, foram desenvolvidos, nesta ordem, os *softwares* do carro, do *smartphone* e do sistema supervisorio. E durante cada etapa eram feitos testes a fim de verificar a funcionalidade de cada sistema e a integração entre eles.

1.4 MATERIAIS UTILIZADOS

- Um *tablet* Android modelo Samsung Tab 2;
- Um Arduino Yún;
- Dois CIs ULN2003A;
- Dois motores de passo 28BYJ-48;
- Uma *Protoboard* de 400 furos;
- Uma bateria de 5V/1A.

1.5 CONTRIBUIÇÃO DO TRABALHO

As contribuições do atual trabalho sobre o trabalho desenvolvido por Negrão (2015) são:

- Resolução do problema na alimentação do carro, que eventualmente sofria quedas de tensão resultando na reinicialização do Arduino;
- Resolução do problema de falta de precisão do carro, que variava a distancia percorrida dependendo da quantidade de carga da bateria e tipo de piso onde está andando;
- A infra-estrutura da rede foi modificada. Ao invés de utilizar a rede *ad-hoc* criada pelo Arduino, ambos (Arduino e Android) se conectam a um ponto de acesso;
- Foi desenvolvido um sistema supervisor, onde o professor pode acompanhar o desenvolvimento dos alunos.

1.6 ORGANIZAÇÃO DO TRABALHO

O restante do trabalho está organizada da seguinte forma:

Capítulo 2: Apresenta os trabalhos relacionados.

Capítulo 3: Trata do *hardware* e *software* utilizados para a construção do carro.

Capítulo 4: Trata dos *software* utilizados no aplicativo Android e Supervisor.

Capítulo 5: Apresenta os resultados desta plataforma.

Capítulo 6: Apresenta a conclusão e lista possíveis trabalhos futuros.

2 TRABALHOS RELACIONADOS

Entre as soluções encontradas para a solução do problema apresentado, foram encontradas algumas plataformas já existentes para o ensino de lógica de programação, como o Lego *Mindstorms* (Elouafiq, 2012), o *Scratch* (Pereira et. al. 2012), Primo (Primo Toys, 2016) e a plataforma desenvolvida por Negrão (2015).

2.1 LEGO MINDSTORMS

O Lego *Mindstorms* é uma plataforma desenvolvida a partir de uma parceria entre a empresa Lego e o *Massachusetts Institute of Technology* (Elouafiq, 2012). O *Mindstorms* consiste em um conjunto de blocos com sensores de diferentes tipos e blocos com motores. Estes blocos interagem com um terceiro tipo de bloco, equipado com um micro controlador dentro (Elouafiq, 2012). Este bloco permite sejam executados comandos programados a partir de um computador através de *softwares* desenvolvidos com essa finalidade. Na versão atual (chamada de EV3) do Lego *Mindstorms* a programação é feita a partir de blocos de comandos, o que não exige grandes conhecimentos de programação. Na figura 1 pode-se ver o bloco do micro controlador com alguns sensores conectados, na figura 2, um robô feito com Lego *Mindstorms* e na **Erro! Fonte de referência não encontrada.**, alguns blocos utilizados para a programação da plataforma.

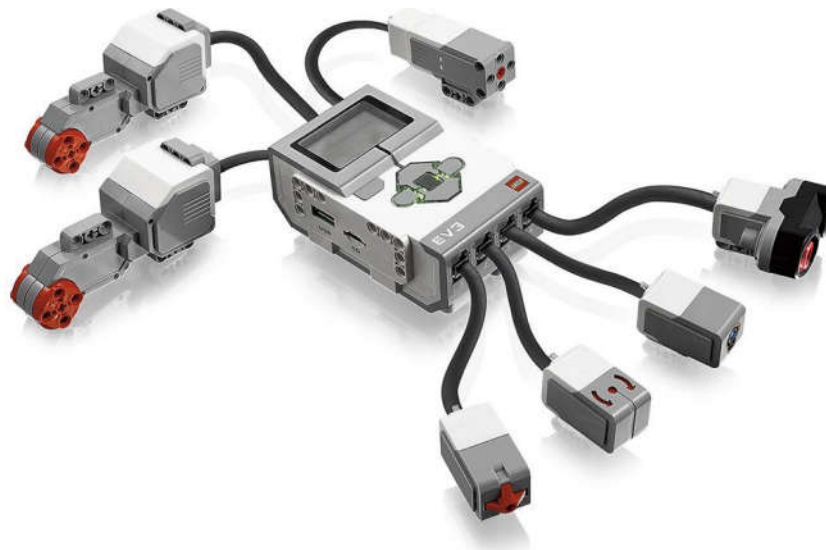


Figura 1: Lego Mindstorms EV3
Fonte: (RESOORT et. al. 2014)



Figura 2: Robô feito a partir do Lego Mindstorms EV3
Fonte: (RESOORT et. al. 2014)



Figura 3: Blocos de comandos do Lego Mindstorms EV3
Fonte: (RESOORT et. al. 2014)

2.2 SCRATCH

A linguagem *Scratch* é uma linguagem desenvolvida a partir de uma parceria entre a Portugal Telecom e o *Massachusetts Institute of Technology* inspirada nas linguagens Logo e *Squeak*, com o objetivo de ser simples, fácil de utilizar e mais intuitiva (Pereira et. al., 2012).

Sobre o *Scratch*, Pereira et. al. (2012) escreveu:

“O termo Scratch provém da técnica de scratching utilizada pelos *Disco-Jockeys* do *Hip-Hop* que giram os discos de vinil com as suas mãos para frente e para trás de modo a fazer misturas musicais de forma criativa e inesperada. Com o *Scratch* é possível fazer algo de semelhante, misturando diferentes tipos de trechos de mídia (gráficos, fotos, músicas, sons) de formas criativas.”

Assim como no Lego *Mindstorms*, a programação no *Scratch* é feita através de blocos de comandos concebidos para se encaixar de uma única forma, fazendo com que não ocorram erros de sintaxe. A figura 4 ilustra a interface do *Scratch*.

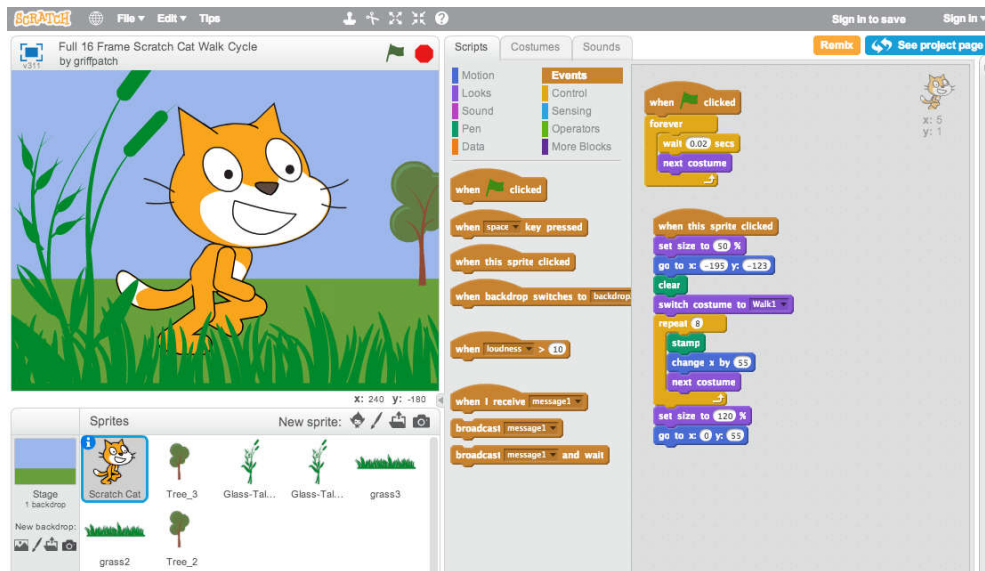


Figura 4: Interface do *Scratch*
 Fonte: MIT Media Lab (2013)

2.3 PRIMO

A plataforma Primo foi desenvolvida com base no Logo *Turtle* com o objetivo de “ensinar programação longe da tela”, ou seja, não é necessário o uso de um computador para utilizar a plataforma (Romano, 2013). Ela é composta por uma “placa de interface”, um conjunto de blocos de instrução, onde cada tipo de peça equivale a um comando e um robô chamado *Cubetto*. A programação do *Cubetto* é feita colocando-se os blocos de instrução nos espaços próprios na placa de interface, que envia a sequência de comandos ao robô. Devido à sua simplicidade, o Primo é recomendado para crianças a partir dos 3 anos de idade. A figura 5 **Erro!** **Fonte de referência não encontrada.** ilustra os componentes da plataforma.



Figura 5: Componentes da plataforma Primo
Fonte: (Primo Toys, 2016)

2.4 PLATAFORMA DESENVOLVIDA POR NEGRÃO (2015)

Baseado na plataforma *Primo*, que é *open source*, Negrão (2015) desenvolveu uma plataforma semelhante, mas com uma nova proposta: em vez de utilizar a “placa de interface” do *Primo*, o controle do carro é feito através de um aplicativo *mobile*. A figura 6 contém a interface Android e o carro.

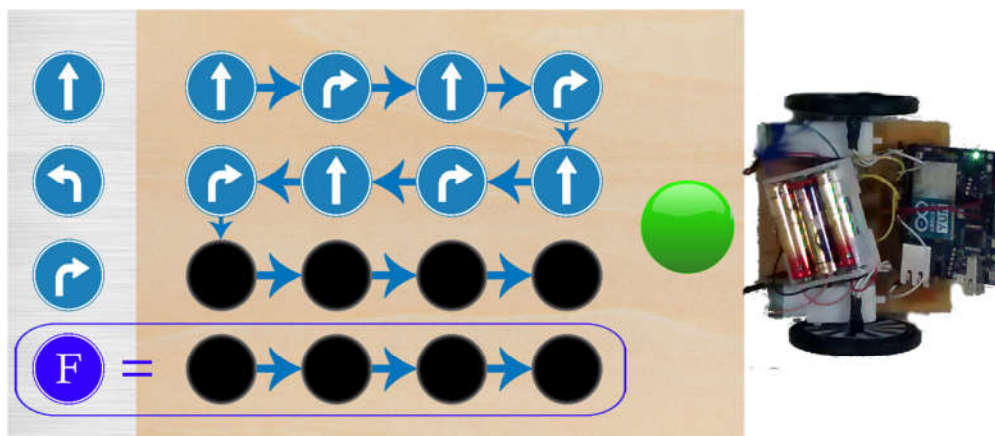


Figura 6: Plataforma desenvolvida por Negrão (2015)
Fonte: O próprio autor

Esta é uma plataforma que ainda não é completamente funcional. Para isso é necessário que sejam corrigidos problemas apresentados pelo próprio autor: a falta de precisão no movimento do carro e a ineficiência da alimentação elétrica.

3 PROPOSTA DE HARDWARE E SOFTWARE DO CARRO

3.1 ARDUINO YÚN

O Arduino é uma plataforma eletrônica de prototipagem que utiliza os conceitos de *Hardware* e de *Software* livres. Uma placa Arduino é composta por:

- Micro controlador;
- Componentes de entrada e saída digitais;
- Componentes de saída analógicos;
- Componentes de entrada e saída seriais;
- Uma interface USB.

Segundo Warren (2011), o Arduino é como um pequeno centro de comandos. Nele é possível ler o valor de um sensor e mostrar ele em um *display* ou na tela do computador, acender ou apagar luzes e tudo isso com apenas algumas linhas de código. Devido à sua versatilidade, e ao grande suporte disponível na comunidade *online* dos usuários de Arduino uma nova geração de hobistas em eletrônica está tendo o seu primeiro contato com um microcontrolador.

Durante o desenvolvimento deste trabalho foi utilizado o Arduino Yún. Um dos diferenciais desta placa em relação a outros modelos, é a presença de um adaptador *WiFi* integrado a ela, o que é uma necessidade para o projeto devido à necessidade de comunicação com dispositivos Android.

O Arduino Yún é uma placa baseada nos micro controladores ATmega32u4 e Atheros AR9331. O micro controlador Atheros contém uma distribuição Linux baseada no OpenWrt chamada Linino que oferece recursos de comunicação em rede, além do suporte a *Scripts Shell* e Phyton. A biblioteca *Bridge* facilita a comunicação entre os dois processadores, fornecendo ao Arduino a capacidade de executar scripts, se comunicar com as interfaces de rede e receber informações do processador AR9331 (Arduino, 2015.). A tabela 1 mostra as especificações do microprocessador Linux e a tabela 2 mostra as especificações do Arduino Yún.

Processador	Atheros AR9331
<i>Ethernet</i>	IEEE 802.3 10/100Mbit/s
<i>WiFi</i>	IEEE 802.11b/g/n
USB tipo A	2.0
RAM	64MB DDR2

Tabela 1: Especificações do microprocessador Linux

Microcontrolador	ATmega32U4
Voltagem de operação	5V
Pinos de entrada/saída digital	20
Pinos de entrada analógica	12
Memória	32KB (4KB reservados ao <i>Bootloader</i>)
<i>Clock</i>	16MHz

Tabela 2: Especificações do Arduino Yún

A figura 7 mostra a comunicação entre o Atmega 32u4 e o AR9331 com a biblioteca Bridge. Já a figura 8 mostra a placa Arduino Yún.

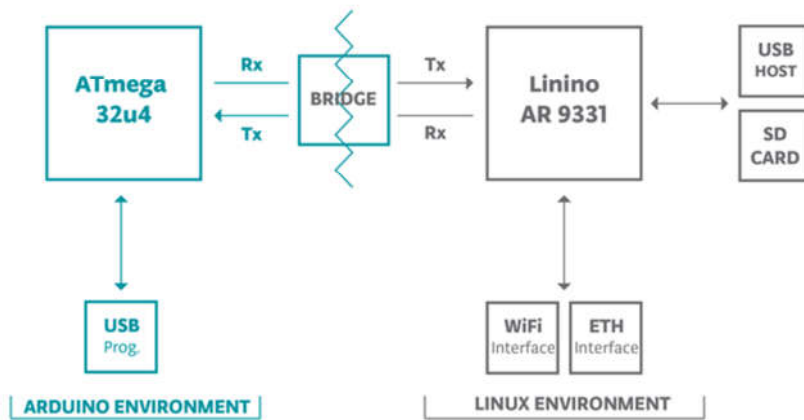


Figura 7: Comunicação entre o Atmega 32u4 e o AR9331
 Fonte: (Arduino, 2015)

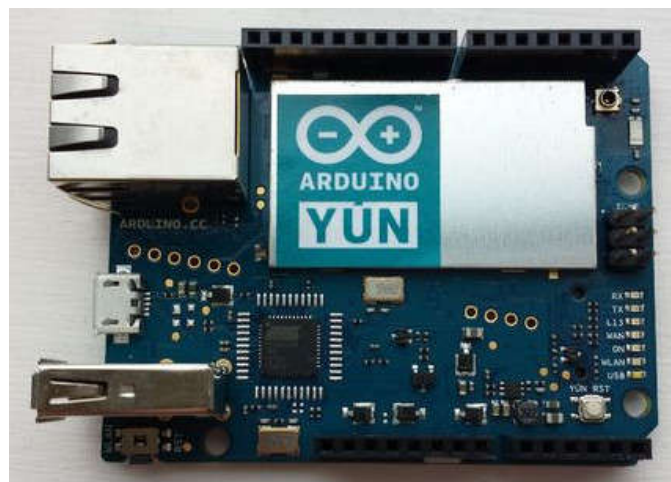


Figura 8: Arduino Yún
 Fonte: Arduino (2015)

3.2 CIRCUITO INTEGRADO ULN2003A

O CI ULN2003A (figura 9) é uma matriz com sete transistores *Darlington*. Estes transistores fornecem ao Arduino, com o auxílio de uma fonte de alimentação externa, um ganho de corrente suficiente para alimentar os motores de passo. Como o Arduino utiliza quatro saídas para controlar cada motor, é necessário utilizar dois CIs ULN2003A neste projeto.



Figura 9: CI ULN2003A
Fonte: Stan (2014)

3.3 MOTOR DE PASSO 28BYJ-48

Um motor de passo é um dispositivo que converte os impulsos elétricos em movimentos discretos (passos). Ele é composto por quatro bobinas acionadas de forma independente pelo Arduino e a combinação delas define a posição onde o motor deve ficar.

O motor de passo 28BYJ-48 (figura 10) possui 64 passos por revolução sendo que ainda conta com uma caixa de redução de 1/64, resultando em 4096 passos por revolução, ou seja, este motor pode fazer rotações de $0,088^\circ$ por passo.



Figura 10: Motor de passo 28BYJ-48
Fonte: Stan (2014)

A escolha dos motores de passo foi devido à sua alta precisão e o baixo consumo de energia comparado aos motores de corrente contínua que foram utilizados no projeto anterior.

3.4 ALIMENTAÇÃO

A fonte de alimentação utilizada é uma bateria externa para celular que possui uma saída USB que fornece uma tensão de 5V e corrente máxima de 1A. Além desta bateria, também foram utilizadas uma bateria de 9V e um conjunto de quatro pilhas AA ligadas em série. Embora todos sejam capazes de fornecer a energia necessária para alimentar o *hardware*, a bateria externa foi escolhida por ser recarregável.

3.5 CONEXÕES ENTRE O ARDUINO E OS MOTORES DE PASSO

A tabela 3 mostra como foram feitas as conexões entre as portas de saída do Arduino e as portas de entrada dos motores de passo. Nesta tabela são omitidos os CIs ULN2003A, que são ligados entre as saídas do Arduino e as entradas dos motores, com a porta de saída do Arduino ligada à entrada do CI e a saída do CI ligada à entrada do motor, como pode ser visto na **Erro! Fonte de referência não encontrada..**

Arduino	Motor de Passo	
	Lado	Conector
4	Direita	Laranja
5	Direita	Amarelo
6	Direita	Rosa
7	Direita	Azul
11	Esquerda	Laranja
10	Esquerda	Amarelo
9	Esquerda	Rosa
8	Esquerda	Azul

Tabela 3: Conexões

3.6 MONTAGEM

A montagem eletrônica do projeto foi feita utilizando uma *Protoboard*, que é uma placa com furos e conexões condutoras entre eles utilizada para a prototipagem de circuitos eletrônicos. A figura 11 mostra como foram feitas as ligações na *Protoboard* e a figura 12 contém o diagrama esquemático do projeto.

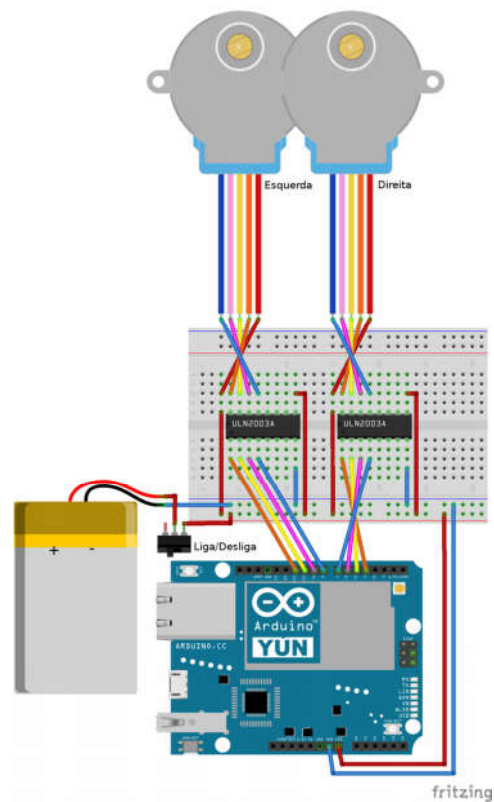


Figura 11: Montagem na protoboard
Fonte: O próprio autor

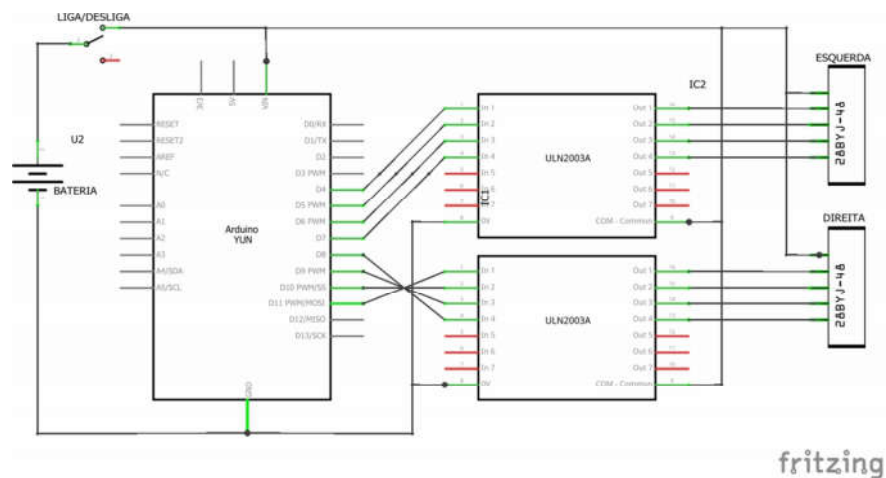


Figura 12: Diagrama esquemático
Fonte: O próprio autor

Para a estrutura física do carro foi utilizada uma vasilha plástica de 13,5x9,5x5cm (figura 13), duas rodas com 7cm de diâmetro ligadas aos motores de passo, uma “roda boba” na parte traseira e um botão de Liga/desliga instalado na parte inferior do carro.

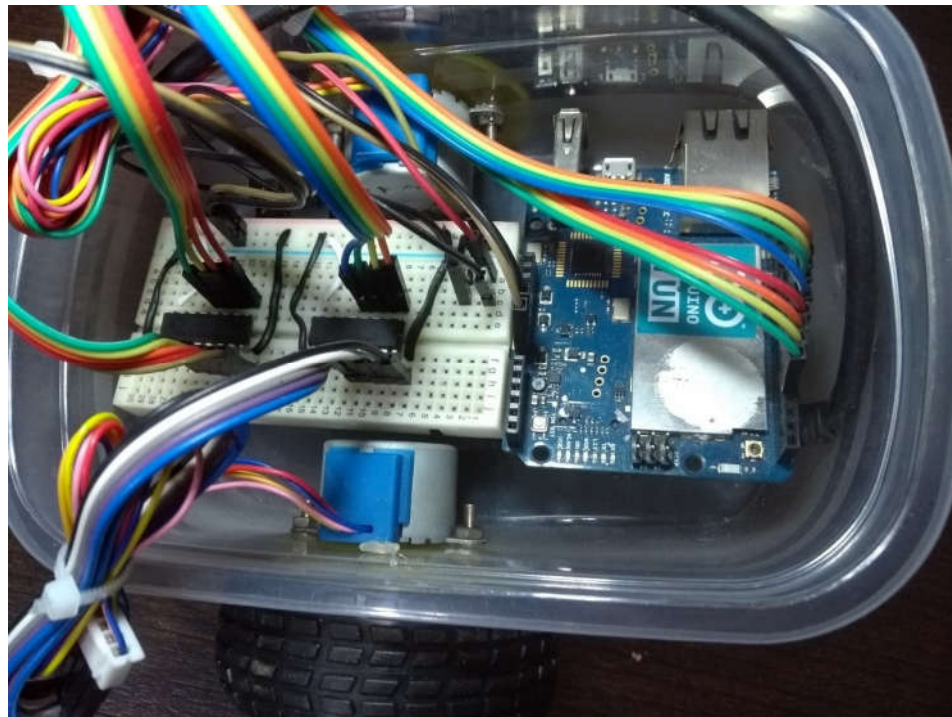


Figura 13: Montagem do carro
Fonte: O próprio autor

3.7 SOFTWARE DO ARDUINO

O *software* embarcado na placa Arduino foi desenvolvido na IDE Arduino, utilizando a linguagem *Wiring*, baseada em C/C++. Este *software* é responsável por controlar todo o funcionamento do carro. Por se tratar de um programa pequeno, esta parte foi desenvolvida em apenas um arquivo (chamado “Carro.ino”), o que também torna mais fácil a sua compreensão.

Este *software* recebe uma *string* via *socket* do aplicativo Android e aciona os motores do carro fazendo que ele se movimente de acordo com os comandos inseridos.

3.7.1 Declarações

Esta parte do código é responsável por definir as bibliotecas e as variáveis globais que serão utilizadas em todo o código. No trecho de código abaixo, temos as declarações presentes no arquivo Carro.ino.

```

1.  #include <Bridge.h>
1.  #include <YunServer.h>
2.  #include <YunClient.h>
3.  #include <Stepper.h>
4.
5.  #define passosANDA 2048;
6.  #define passosDOBRA 990;
7.
8.  #define PORTA 6666;
```

A biblioteca Bridge faz a comunicação entre os dois processadores do Arduino Yún, enquanto que YunServer e YunClient permitem a conexão do Arduino Yún com outros dispositivos.

As linhas 5 e 6 definem quanto os motores devem se deslocar (quantidade de passos) quando anda reto e quando faz curvas. Estes valores dependem de alguns fatores como tamanho das rodas, distancia entre elas e quanto o carro deve se deslocar.

A linha 8 define a porta utilizada como padrão para a recepção dos dados vindos do aplicativo Android.

3.7.2 Metodo Setup()

Este método é responsável pelas configurações dos componentes da placa Arduino que serão utilizados pelo programa. Nele, além de ser inicializada a interface *WiFi* do controlador AR9331, são definidos quais pinos serão utilizados como entrada ou saída.

3.7.3 Metodo Loop()

Este método contém a parte principal do código e é responsável pelo comportamento do carro. O conjunto de comandos presente nele é repetido de forma contínua sempre que o Arduino estiver ligado.

O método “loop()” deste projeto consiste em aguardar uma conexão *wireless* que, uma vez estabelecida, recebe um pacote vindo do programa Android contendo uma *string* no padrão “s=ADFFFA&f=AE” (sequência representada na figura 14) onde “s” representa a sequencia principal e “f” uma função secundária, que será

executada caso exista um caractere “F” na sequência principal. Os caracteres aceitos em ambas as sequências são “A”, “D” e “E” representando, respectivamente, os seguintes movimentos: andar, virar à direita e virar à esquerda. A função secundária não pode receber o valor “F”. Após receber a *string*, o programa separa em duas novas *strings*: “sequencia” e “funcao”.

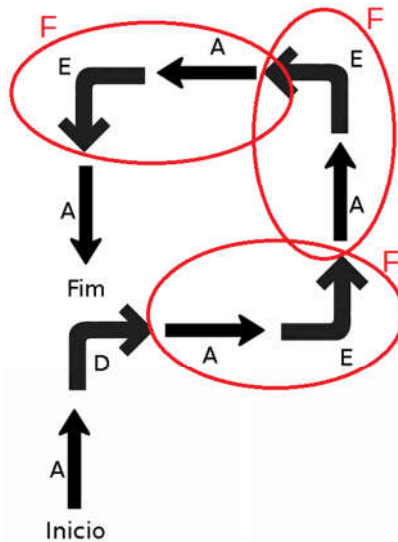


Figura 14: Percurso s=ADFFFA&f=AE
Fonte: O próprio autor

De posse das duas *strings*, o programa lê cada caractere da sequência principal e executa o método “anda(char a)”, que é responsável por enviar o sinal elétrico para o CI L293D. Porém, se o caractere lido representar a Função, o programa passa os caracteres da string “funcao” e, assim como na sequência principal, executa o método “anda(char c)”. Ao final da execução da Função a sequência principal volta a ser lida do caractere onde parou.

Após terminar a sequência, o programa volta ao estado inicial, aguardando novamente uma conexão *wireless*.

A figura 15 contém o Diagrama de Atividades do carro.

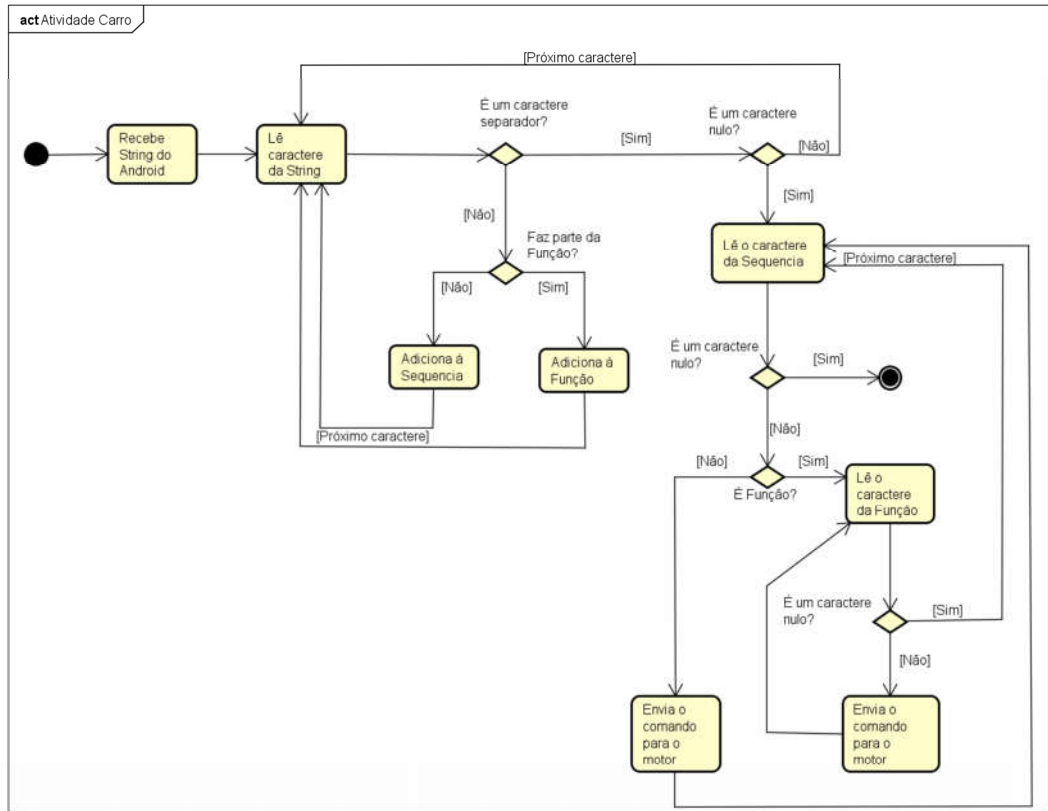


Figura 15: Diagrama de Atividades do carro
Fonte: O próprio autor

4 PROPOSTA DE SOFTWARE

O *software* deste projeto é dividido em três partes: uma embarcada na placa Arduino, já descrito na seção 3.7, outra no dispositivo Android e a terceira relativa ao sistema supervisorio descritos a seguir.

4.1 ANDROID

O aplicativo presente no *tablet* Android foi desenvolvido utilizando a IDE *Android Studio*, utilizando a linguagem Java. É a partir deste aplicativo que a criança desenvolve o trajeto que será enviado para o carro. Ele também tem a função de enviar este trajeto para a aplicação supervisoría, onde o professor pode acompanhar as atividades dos alunos.

Em um aplicativo Android toda a interação com o usuário é feita através de *Activities*. Segundo Android (2016), uma *Activity* é um componente do aplicativo que fornece uma tela que exibe a interface onde o usuário pode interagir com o programa. Geralmente essa tela preenche toda a tela, mas também pode ser menor que a tela ou flutuar sobre outras janelas.

Toda *Activity* no Android deve conter o método `onCreate()` implementado. Ele é executado assim que a *Activity* entra em ação e inicializa todos os elementos essenciais da atividade, como o método `setContentView()` que define o *layout* da interface do usuário.

A organização dos arquivos em um aplicativo Android é composta basicamente das seguintes pastas:

- `drawable/`: Contém as imagens utilizadas no projeto.
- `layout/`: Contém os arquivos XML do *layout* da *Activity*.
- `mipmap/ic_launcher.png/`: Contém o ícone do aplicativo.
- `Values/`: Contém arquivos XML com a definição de valores como cores, dimensões e *Strings*.

As *Activities* e as classes utilizadas neste aplicativo serão descritas nos próximos subitens.

4.1.1 Classe ConsultaTxt

Esta classe implementa o método `consultaTxt()` que realiza, através de um URL fornecido, consultas a servidores HTTP e armazena a resposta, caso exista, em uma *String*.

4.1.2 *Activity* Tela_Login

Esta *Activity* é exibida assim que o aplicativo é aberto. Nela o aluno deve informar seu nome e o número do carro que utilizará durante as atividades. O método `consultaTxt()` consulta o servidor PHP que, caso os dados existam no banco de dados, retorna a identificação do aluno e o *IP* do carro e carrega a *Activity DragDrop* caso contrário exibe uma mensagem de erro. É exibida a janela de *login* mostrada na figura 16.



Figura 16: Tela de login da aplicação Android
Fonte: O próprio autor

4.1.1 *Activity DragDrop*

É nesta *Activity* onde os comandos serão inseridos. Inicialmente foi utilizado um *design* semelhante ao desenvolvido por NEGRÃO (2015), onde as figuras dos comandos ficam disponíveis à esquerda dos soquetes vazios e o preenchimento desses soquetes é feito arrastando as figuras para eles. Porém foi uma modificação na segunda linha: Ao invés de ser preenchida da direita para a esquerda, também

passou a ser preenchida da esquerda para a direita. A versão final pode ser vista na figura 17.

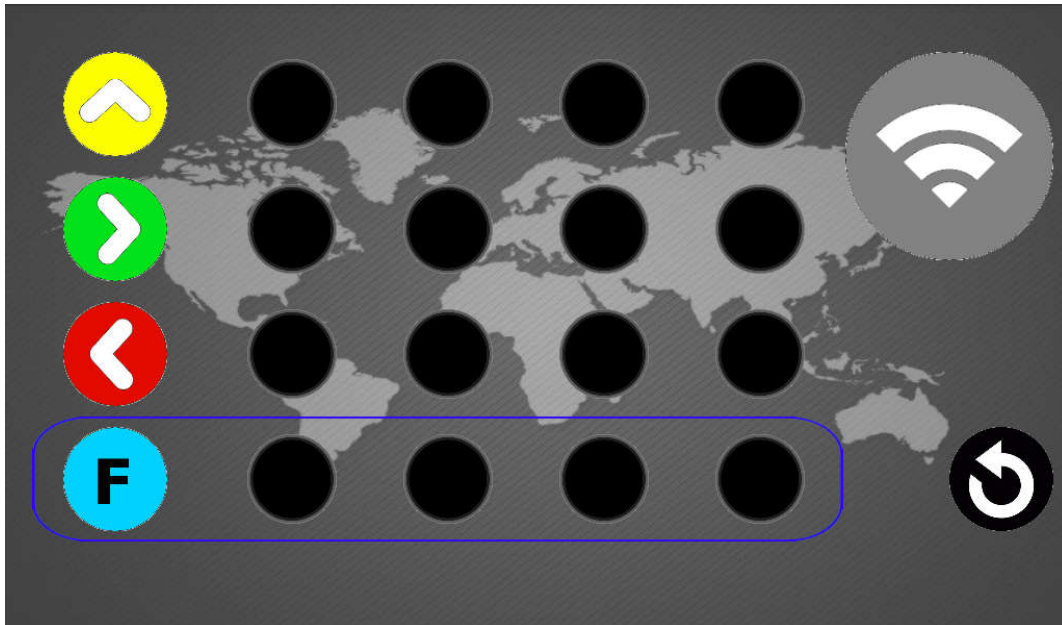


Figura 17: Design novo
Fonte: O próprio autor

Para analisar a execução deste tipo de tarefa, foi desenvolvido o diagrama de atividades mostrado na figura 18 **Erro! Fonte de referência não encontrada..**

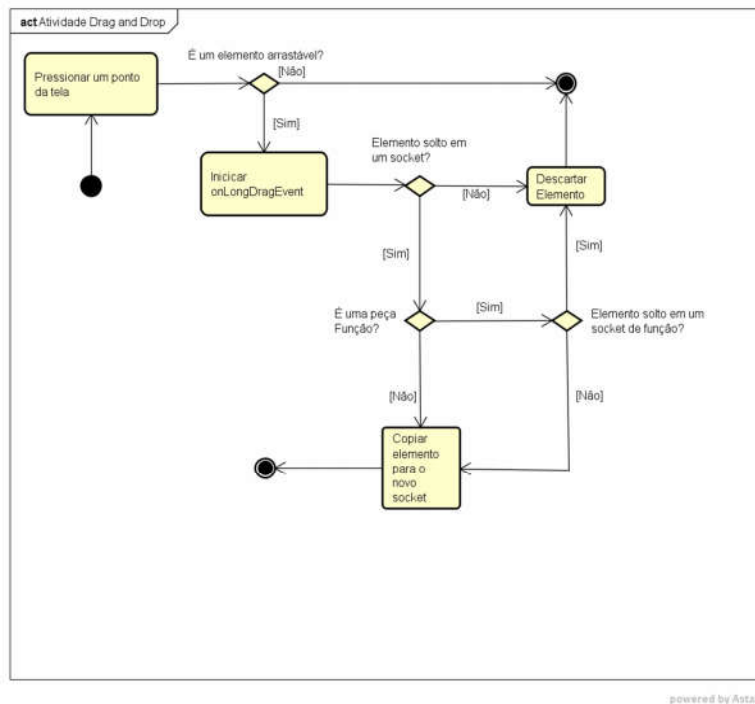


Figura 18: Diagrama de atividades - Drag and Drop
Fonte: O próprio autor

Os principais métodos desta *Activity* são:

- enviar()
- limpar()
- onLongClick()
- onDrag()

4.1.1.1 Método enviar()

Quando o botão enviar é pressionado, este método lê o valor de cada soquete preenchido e agrupa em uma *String* que será enviada por *socket* para o carro e por HTTP para o aplicativo supervisorio.

4.1.1.2 Método limpar()

Este método tem o único objetivo esvaziar todos os *sockets* para que sejam preenchidos novamente.

4.1.1.3 Método onLongClick()

Método existente na classe *OnLongClickListener*. Este método é executado sempre que a tela é pressionada. Caso o toque seja feito em uma figura de comando, é criada uma sombra da figura, que acompanhará o dedo do usuário até que ele solte a tela.

4.1.1.4 Método onDrag()

Método existente na classe *OnDragListener*. Este método permanece em execução enquanto houver alguma imagem sendo arrastada na tela. Ele também define as ações executadas quando o objeto começa a ser arrastado e quando é solto.

No caso deste aplicativo, quando a imagem é solta ele verifica se a posição final é um *socket* válido.

4.2 SUPERVISÓRIO

O aplicativo supervisorio foi escrito na linguagem PHP utilizando a IDE Adobe Brackets. Este aplicativo destina-se a fornecer ao professor uma plataforma em que possa acompanhar o desenvolvimento, inserir comentários e gerar relatórios sobre

as atividades realizadas pelos alunos, conforme o diagrama de casos de uso da figura 19.

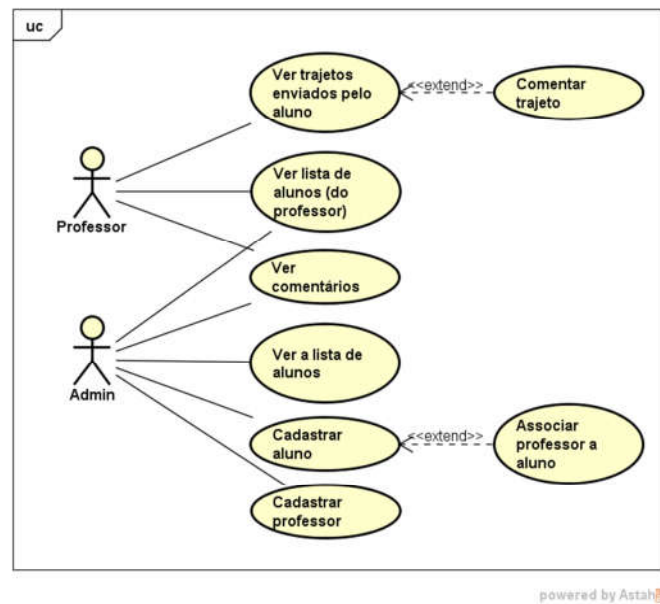


Figura 19: Diagrama de casos de uso
Fonte: O próprio autor

Neste aplicativo foi utilizado o gestor de banco de dados MySQL com o diagrama de entidade-relacionamento exibido na figura 20. Neste diagrama pode-se perceber que além das tabelas necessárias ao acompanhamento das atividades dos alunos pelo professor existe, também, a tabela 'Carro'. Como o aplicativo Android necessita que seja informado o endereço IP do carro que receberá os comandos, esta tabela associa o número de identificação do carro com seu endereço IP.

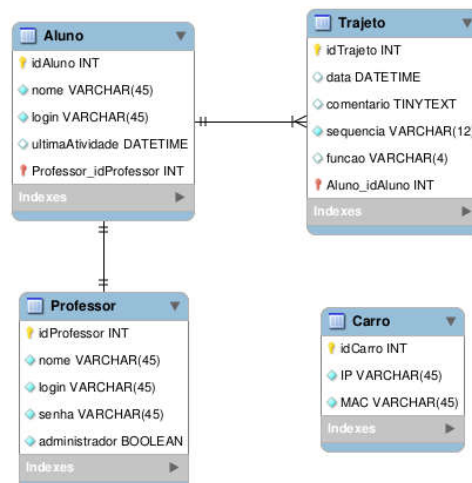


Figura 20: Diagrama de entidade-relacionamento
Fonte: O próprio autor

Nos próximos itens são descritos os arquivos que compõem este aplicativo assim como os respectivos *layouts*.

- index.php: este arquivo contém a tela de login (figura 21)

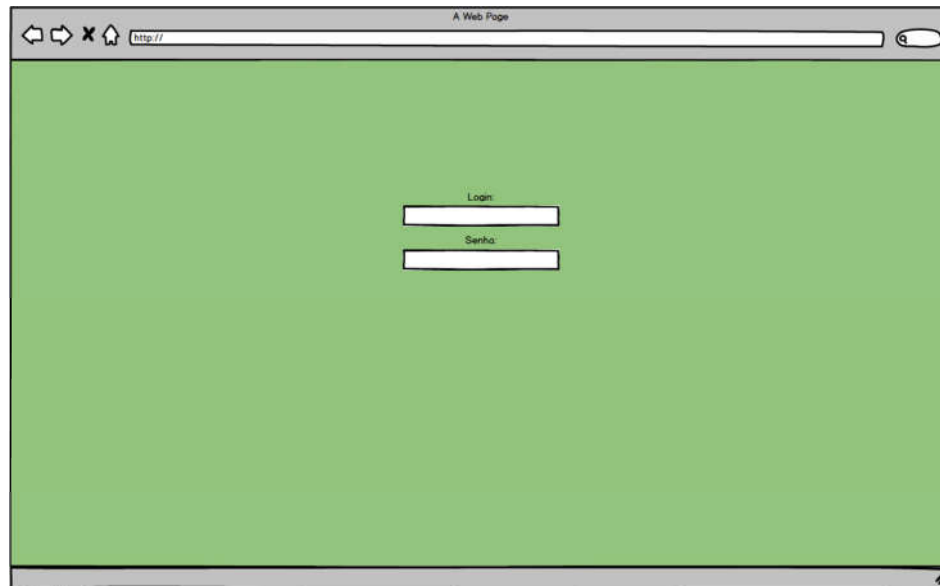


Figura 21: Tela de login
Fonte: O próprio autor

- home.php: exibe a lista de alunos do professor, junto ao dia e horário em que realizou a última atividade. É possível organizar esta lista tanto pelo nome do aluno, em ordem alfabética, quanto pela data da última atividade. (figura 22 **Erro! Fonte de referência não encontrada.**)

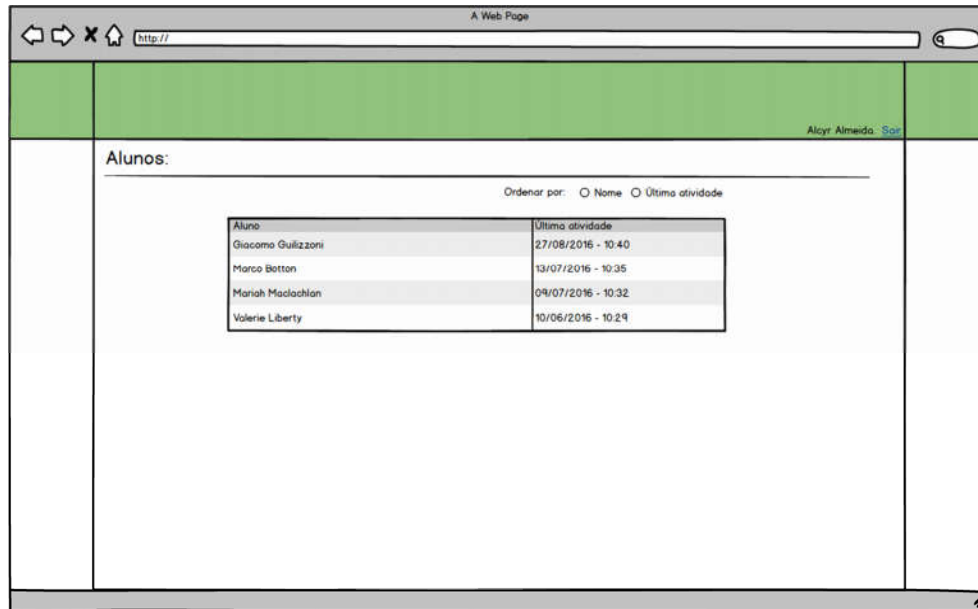


Figura 22: Lista de alunos
Fonte: O próprio autor

- alunoAtividades.php: exibe as atividades realizadas pelo aluno, ordenadas pela data, e o comentário feito pelo professor. (figura 23 **Erro! Fonte de referência não encontrada.**)

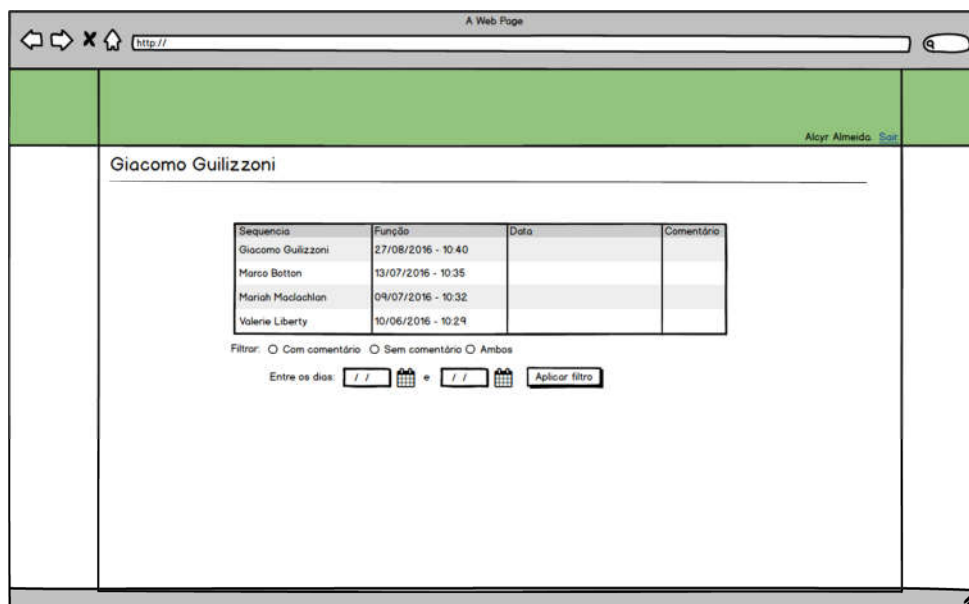


Figura 23: Tela de atividades por aluno
Fonte: O próprio autor

- cadastroAluno_form.php e cadastroProfessor_form.php: permitem cadastrar ou alterar os dados do aluno e do professor, respectivamente. (figura 24)

Figura 24: Cadastro de professor
Fonte: O próprio autor

- comentário.php: permite ao professor inserir ou alterar o comentário em uma atividade. (figura 25)

cadastroAluno_form.php e cadastroProfessor_form.php: permitem cadastrar ou alterar os dados do aluno e do professor, respectivamente. (**Erro! Fonte de referência não encontrada.**).

Figura 25: Tela de comentário
Fonte: O próprio autor

- conectaDB.php: realiza a conexão do aplicativo ao banco de dados.
- trajeto.php: este arquivo recebe o trajeto do aplicativo Android e armazena no banco de dados.

- loginCarro.php: verifica se o nome de usuário e o número do carro informados no aplicativo Android estão corretos. Caso estes dados estejam corretos ele retorna o número de identificação do aluno e o endereço IP do carro.
- salvaComentário.php: insere um novo comentário ou atualiza um comentário já existente com o texto inserido na tela de comentário da figura 23.
- estilo.css: contém os estilos de formatação dos elementos presentes nos arquivos HTML.

5 AVALIAÇÃO DOS RESULTADOS

Devido ao fato que o autor não tem conhecimento sobre como avaliar o desempenho das crianças utilizando esta plataforma, a avaliação dos resultados será focada unicamente nos resultados técnicos.

Considerando que o carro utilizado nos testes possui, aproximadamente, 13,5cm entre as duas rodas, as rodas possuem 3,5cm de raio, e que ele anda 2048 passos quando é solicitado que ande reto e 990 passos ao dobrar para qualquer lado, a distancia percorrida ao andar reto é de 22cm e as curvas são feitas em ângulos de 90°.

Com relação à alimentação, foi utilizada uma bateria de 5V / 1A / 3.600mAh, que foi suficiente para manter o carro ligado por aproximadamente duas horas.

6 CONCLUSÃO

Durante os testes realizados pelo autor, a plataforma se mostrou funcional embora ainda seja necessária a elaboração de uma carcaça que isole os componentes eletrônicos do carro e o torne esteticamente mais atraente. Porém permanece o problema apresentado por Negrão (2015) sobre a falta de conhecimento do autor acerca da quantização do aprendizado limitando-se os testes à facilidade de uso da aplicação Android e à comunicação entre os componentes envolvidos (aplicação Android, Arduino e aplicação supervisória).

Outra limitação deste Trabalho é que a aplicação supervisória não permite a criação de turmas, limitando-se a uma turma por professor.

6.1 TRABALHOS FUTUROS

Algumas propostas para trabalhos futuros são:

Instalar um sensor no carro que indique se ao final da execução do percurso inserido, ele parou em um local pré-definido;

- O aprimoramento do *design* do carro para que se torne visualmente atraente para as crianças e proteja os componentes eletrônicos do mesmo;
- A adaptação da aplicação supervisória para que seja possível a criação de turmas de alunos;
- O teste em campo a fim de verificar a curva de aprendizado e as vantagens da utilização da plataforma.

REFERÊNCIAS

- ANDROID, **Atividades**. Disponível em <<https://developer.android.com/guide/components/activities.html?hl=pt-br>>. Acesso em 15 de agosto de 2016.
- ARDUINO, **Arduino – Reference**. 2015. Disponível em <<https://www.arduino.cc/en/Guide/ArduinoYunLin>>. Acesso em 11 de julho de 2016.
- AURELIANO, Viviane Cristina Oliveira; TEDESCO, PC de AR. **Avaliando o uso do Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação**. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. 2012. p. 1-10.
- ELOUAFIQ, Ali. **The Lego Mindstorms Robotics Invention Systems 2.0 Toolkit: A Study Case**. arXiv preprint arXiv:1204.1650, 2012.
- FERREIRA, Cláudia; GONZAGA, Flávio; SANTOS, Rodrigo. **Um Estudo sobre a Aprendizagem de Lógica de Programação Utilizando Programação por Demonstração**. Faculdade Governador Ozanam Coelho/Universidade Federal de Alfenas/Universidade Federal do Rio de Janeiro. v. 23, 2009.
- FURIA, Fernanda. **Crianças que programam: uma habilidade para toda a vida**. Playground da Inovação. 07 Dez. 2014. Disponível em <<http://www.playground-inovacao.com.br/criancas-que-programam-uma-habilidade-para-toda-a-vida>>. Acesso em 12 de Junho de 2016.
- GLOBO NEWS. **Cientistas criam computador para ensinar programação às crianças**. G1, 09 Out. 2012. Disponível em <<http://g1.globo.com/globo-news/noticia/2012/10/cientistas-criam-computador-para-ensinar-programacao-criancas.html>>. Acesso em 12 Jun. 2016.
- MATTEI, Claudinéia. **O prazer de aprender com a informática na Educação Infantil**. Curso de especialização em Psicopedagogia, ICPG. Disponível em: <<http://www.icpg.com.br/artigos/rev02-11.pdf>>. Acesso em 12 de Junho de 2016.
- MIT Media Lab. **Kids coding in the cloud**. 2013. Disponível em <<http://news.mit.edu/2013/scratch-two-released-0514>>. Acesso em 20 de Julho de 2016.
- NEGRÃO, Victor Luiz Santos. **Desenvolvimento de Uma Plataforma para Ensino Básico para Ensino de Conceitos Básicos de Lógica de Programação utilizando arduino e Android**. 2015. in Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Instituto de Ciências Exatas e Naturais, Universidade Federal do Pará.
- PEREIRA, P. de S.; MEDEIROS, M.; MENEZES, J. W. M. Análise do Scratch como ferramenta de auxílio ao ensino de programação de computadores. In: **XL Congresso Brasileiro de Educação em Engenharia, Belém, Brasil**. 2012.

- PRIMO TOYS. **Bring coding to life in your classroom with Cubetto**. Primo Toys. 2016. Disponível em <<https://www.primotoys.com/education>>. Acesso em 20 de Julho. 2016.
- RESOORT, Jeroen; SCHRIJVER, Bert Jan; WOLTRING, Ivo, **LEGO Mindstorms EV3 - Java en LEGO hand in hand?**. 2014. Disponível em <<http://www.nljug.org/databasejava/lego-mindstorms-ev3-java-en-lego-hand-hand/>>. Acesso em 11 de julho de 2016.
- ROMANO, Zoe. **Teach programming logic to kids with Primo – Now on Kickstarter!**. Arduino. 22 Nov. 2013. Disponível em <<https://blog.arduino.cc/2013/11/22/teach-programming-logic-to-kids-with-primo>>. Acesso em 23 Jun. 2016.
- STAN. **28BYJ-48 Stepper Motor with ULN2003 driver and Arduino Uno**. 2014. Disponível em <<http://42bots.com/tutorials/28byj-48-stepper-motor-with-uln2003-driver-and-arduino-uno/>>. Acesso em 11 de julho de 2016.
- WARREN, John-David; ADAMS, Josh; MOLLE, Harald. **Arduino for Robotics**. Apress, 2011.

ANEXOS

Anexo A – Código embarcado no Arduino

```

#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
#include <Stepper.h>

#define passosANDA 2048;
#define passosDOBRA 990;

#define PORTA 6666
YunServer server(PORTA);
static boolean clienteAtivo = false;

const int stepsPerRevolution = 256;
const int velocidade = 100;
Stepper StepperD(stepsPerRevolution, 8,10,9,11);
Stepper StepperE(stepsPerRevolution, 4,6,5,7);

int d, e, passos;
String sequencia="", funcao="";

void setup(){
  Bridge.begin();
  server.begin();

  Serial.begin(9600);

  StepperD.setSpeed(velocidade);
  StepperE.setSpeed(velocidade);
}

void loop(){
  YunClient cliente = server.accept();
  int cont=0;
  while (cliente.connected()){
    char entrada;
    if (cliente.available()){
      entrada = cliente.read();
      Serial.print(entrada);
      if (cont==1 && entrada != '&') sequencia+=entrada;
      if (cont==3 && entrada != '&') funcao+=entrada;
      if (entrada == '=' || entrada == '&') cont++;
      if(entrada=='&' && cont==4){
        cliente.stop();
        Serial.println();
      }
    }
  }

  if (sequencia.length() > 0){
    for (int i=0;i<sequencia.length();i++){
      if (sequencia[i]=='F'){
        for (int j=0;j<funcao.length();j++){
          anda(funcao[j]);
        }
      }
    }

    if (sequencia[i]!='F'){
      anda(sequencia[i]);
    }
  }
}

```

```
    }  
  }  
  sequencia="", funcao="";  
}  
}  
  
void anda(char a){  
  switch (a){  
    case 'A':  
      d=-1;  
      e=-1;  
  
      passos=passosANDA;  
      break;  
  
    case 'D':  
      d=-1;  
      e=1;  
  
      passos=passosDOBRA;  
      break;  
  
    case 'E':  
      d=1;  
      e=-1;  
      passos=passosDOBRA;  
      break;  
  }  
  
  for (int i = 0; i<=passos; i++){  
    StepperD.step(d);  
    StepperE.step(-e);  
  }  
  
  delay(500);  
}
```

Anexo B – Código embarcado no aplicativo Android

B.1: Tela_login.java

```

package com.almeida.tcc;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import static java.lang.Integer.parseInt;

public class Tela_login extends Activity implements ConsultaInterface {
    private TextView msgErro;

    private String ipServidor = "192.168.43.240";
    private int resultado=1;
    private String idAluno=null;
    private String ipCarro=null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tela_login);

        this.msgErro = (TextView) findViewById(R.id.msgErro);
    }

    public void login(View v) throws InterruptedException {
        EditText nome= (EditText) findViewById(R.id.usuario_et);
        EditText carro=(EditText) findViewById(R.id.carro_et);

        String
url="http://" + ipServidor + "/Supervisorio/loginCarro.php?aluno=" + nome.getText()
().toString() + "&carro=" + carro.getText().toString();

        ConsultaTxt consultaTxt = new ConsultaTxt(this, this);
        consultaTxt.execute(url);
    }

    // @SuppressWarnings("LongLogTag")
    public void downloadTxt(String string) {
        String campos[] = string.split(",");

        resultado = parseInt(campos[0]);

        if (resultado==0) {
            idAluno=campos[1];
            ipCarro=campos[2];

            Bundle params = new Bundle();
            params.putString("idAluno", idAluno);
            params.putString("ipCarro", ipCarro);

            Intent intent = new Intent(this, DragDrop.class);
            intent.putExtras(params);

            startActivity(intent);

```

```
    }  
    else if (resultado == -1){  
msgErro.setText(getString(R.string.erroLogin)); }  
    else if (resultado == -2){  
msgErro.setText(getString(R.string.erroCarro)); }  
    else if (resultado == -3){  
msgErro.setText(getString(R.string.erroLoginCarro)); }  
    }  
}
```

B.2: DragDrop.java

```

package com.almeida.tcc;

import android.content.Intent;
import android.os.Bundle;
import android.app.Activity;
import android.content.ClipData;
import android.graphics.Canvas;
import android.graphics.Point;
import android.graphics.drawable.Drawable;
import android.view.DragEvent;
import android.view.Menu;
import android.view.View;
import android.view.View.OnDragListener;
import android.view.View.OnLongClickListener;
import android.widget.ImageView;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;

public class DragDrop extends Activity implements ConsultaInterface {

    static int LIMPAR=0;
    static int LER=1;

    ImageView andar, direita, esquerda, funcao, apagar, socketVazio, f01,
    f02, f03, f04;

    String idAluno, ipCarro, ipServidor="192.168.43.240", saidaArduino,
    saidaHTTP;

    private Socket socket = null;

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_drag_drop);

        findViewById(R.id.andar).setOnLongClickListener(longListener);
        findViewById(R.id.direita).setOnLongClickListener(longListener);
        findViewById(R.id.esquerda).setOnLongClickListener(longListener);
        findViewById(R.id.funcao).setOnLongClickListener(longListener);

        findViewById(R.id.s01).setOnDragListener(dragListener);
        findViewById(R.id.s02).setOnDragListener(dragListener);
        findViewById(R.id.s03).setOnDragListener(dragListener);
        findViewById(R.id.s04).setOnDragListener(dragListener);
        findViewById(R.id.s05).setOnDragListener(dragListener);
        findViewById(R.id.s06).setOnDragListener(dragListener);
        findViewById(R.id.s07).setOnDragListener(dragListener);
        findViewById(R.id.s08).setOnDragListener(dragListener);
    }
}

```

```

findViewById(R.id.s09).setOnDragListener(dragListener);
findViewById(R.id.s10).setOnDragListener(dragListener);
findViewById(R.id.s11).setOnDragListener(dragListener);
findViewById(R.id.s12).setOnDragListener(dragListener);
findViewById(R.id.f01).setOnDragListener(dragListener);
findViewById(R.id.f02).setOnDragListener(dragListener);
findViewById(R.id.f03).setOnDragListener(dragListener);
findViewById(R.id.f04).setOnDragListener(dragListener);

andar = (ImageView) findViewById(R.id.andar);
direita = (ImageView) findViewById(R.id.direita);
esquerda = (ImageView) findViewById(R.id.esquerda);
funcao = (ImageView) findViewById(R.id.funcao);
socketVazio = (ImageView) findViewById(R.id.socketVazio);
f01 = (ImageView) findViewById(R.id.f01);
f02 = (ImageView) findViewById(R.id.f02);
f03 = (ImageView) findViewById(R.id.f03);
f04 = (ImageView) findViewById(R.id.f04);

Intent intent = getIntent();
if (intent != null){
    Bundle params = intent.getExtras();

    if(params != null){
        idAluno = params.getString("idAluno");
        ipCarro = params.getString("ipCarro");
        //ipServidor = params.getString("ipServidor");
    }
}

public void enviar(View v){
    saidaArduino="s=";
    saidaArduino+=valor((ImageView) findViewById(R.id.s01), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s02), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s03), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s04), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s05), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s06), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s07), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s08), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s09), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s10), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s11), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.s12), LER);
    saidaArduino+="&f=";
    saidaArduino+=valor((ImageView) findViewById(R.id.f01), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.f02), LER);
    saidaArduino+=valor((ImageView) findViewById(R.id.f03), LER);
    saidaArduino+= valor((ImageView) findViewById(R.id.f04), LER);
    saidaArduino+="&";

    saidaHTTP="http://" + ipServidor + "/Supervisorio/trajeto.php?" + saidaArduino + "a
luno=" + idAluno;

    ConsultaTxt consulta = new ConsultaTxt(this, this);
    consulta.execute(saidaHTTP);

    new Thread(new ClienteThread()).start();
    Thread.interrupted();
}

```

```

}

public void limpar(View v){
    valor((ImageView) findViewById(R.id.s01), LIMPAR);
    valor((ImageView) findViewById(R.id.s02), LIMPAR);
    valor((ImageView) findViewById(R.id.s03), LIMPAR);
    valor((ImageView) findViewById(R.id.s04), LIMPAR);
    valor((ImageView) findViewById(R.id.s05), LIMPAR);
    valor((ImageView) findViewById(R.id.s06), LIMPAR);
    valor((ImageView) findViewById(R.id.s07), LIMPAR);
    valor((ImageView) findViewById(R.id.s08), LIMPAR);
    valor((ImageView) findViewById(R.id.s09), LIMPAR);
    valor((ImageView) findViewById(R.id.s10), LIMPAR);
    valor((ImageView) findViewById(R.id.s11), LIMPAR);
    valor((ImageView) findViewById(R.id.s12), LIMPAR);
    valor((ImageView) findViewById(R.id.f01), LIMPAR);
    valor((ImageView) findViewById(R.id.f02), LIMPAR);
    valor((ImageView) findViewById(R.id.f03), LIMPAR);
    valor((ImageView) findViewById(R.id.f04), LIMPAR);
}

public String valor(ImageView iv, int valor){
    if(valor ==LER)
        if (iv.getDrawable() == andar.getDrawable()) return "A";
        else if (iv.getDrawable() == direita.getDrawable()) return "D";
        else if (iv.getDrawable() == esquerda.getDrawable()) return
"E";
        else if (iv.getDrawable() == funcao.getDrawable()) return "F";
        else return "";

    else if (valor==LIMPAR){
        iv.setImageDrawable(socketVazio.getDrawable());
        return "";
    }

    return "";
}

OnLongClickListener longListener = new OnLongClickListener() {
    @Override
    public boolean onLongClick(View v) {
        ImageView iv = (ImageView) v;

        View.DragShadowBuilder myShadowBuilder = new
MyShadowBuilder(iv);

        ClipData data = ClipData.newPlainText("", "");
        v.startDrag(data, myShadowBuilder, iv, 0);

        return true;
    }
};

OnDragListener dragListener = new OnDragListener(){
    @Override
    public boolean onDrag(View v, DragEvent event){
        ImageView iv = (ImageView) v;
        ImageView e = (ImageView) event.getLocalState();
        int dragEvent = event.getAction();

        if (dragEvent == DragEvent.ACTION_DROP)
            if (e != funcao || iv != f01 && iv != f02 && iv != f03 &&

```

```

iv != f04)
    if (e == apagar)
iv.setImageDrawable(socketVazio.getDrawable());
    else iv.setImageDrawable(e.getDrawable());

    return true;
}
};

private class MyShadowBuilder extends View.DragShadowBuilder{
    private Drawable shadow;

    public MyShadowBuilder(ImageView iv){
        super(iv);
        shadow = iv.getDrawable();
    }

    @Override
    public void onDrawShadow(Canvas canvas){
        shadow.draw(canvas);
    }

    @Override
    public void onProvideShadowMetrics(Point shadowSize, Point
shadowTouchPoint){
        int height = getView().getHeight();
        int width = getView().getHeight();

        shadow.setBounds(0, 0, width, height);

        shadowSize.set(width, height);
        shadowTouchPoint.set(width/2, height/2);
    }
}

class ClienteThread implements Runnable {
    @Override
    public void run() {
        try {
            InetAddress servidor = InetAddress.getByName(ipCarro);
            socket = new Socket(servidor, 6666);
            PrintWriter out = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream()), true);
            out.println(saidaArduino);
        } catch (UnknownHostException e1) {
            e1.printStackTrace();
        } catch (IOException e1) {
            e1.printStackTrace();
        }
    }
}

public void downloadTxt(String string){}
}

```

B.3: Consulta.java

```

package com.almeida.tcc;

import android.content.Context;
import android.os.AsyncTask;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

class ConsultaTxt extends AsyncTask<String, String, String> {//Parametro,
progresso, resultado
    private static final int USER_AGENT = 0;
    private Context context;
    private ConsultaInterface ci;

    public ConsultaTxt(Context context, ConsultaInterface ci){
        this.context = context;
        this.ci = ci;
    }

    protected void onPreExecute() {}

    protected String doInBackground(String... params) {
        StringBuffer response = new StringBuffer();

        URL url = null;
        try {
            url = new URL(params[0]);
            HttpURLConnection conexao = null;
            conexao = (HttpURLConnection) url.openConnection();
            conexao.setRequestMethod("GET");
            //adiciona o cabeçalho
            conexao.setRequestProperty("User-Agent",
String.valueOf(USER_AGENT));

            BufferedReader in = null;
            in = new BufferedReader(
                new InputStreamReader(conexao.getInputStream()));

            String inputLine;
            response = new StringBuffer();

            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }

            in.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

        return response.toString();
    }

    protected void onProgressUpdate(String... params) { }

    protected void onPostExecute(String params) {
        ci.downloadTxt(params);
    }
}

```

}
}

B.4: activity_drag-drop.xml

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@drawable/fundo"
    tools:context="com.almeida.tcc.DragDrop">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:id="@+id/aluno_carro"
        android:textSize="30dp"
        android:textColor="#FFFFFF"
        android:layout_below="@+id/enviar"
        android:layout_toRightOf="@+id/carro_tv"
        android:layout_alignRight="@+id/textView2"
        android:layout_alignEnd="@+id/textView2" />

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:id="@+id/andar"
            android:layout_width="@dimen/comando"
            android:layout_height="@dimen/comando"
            android:layout_marginLeft="@dimen/espacoLeft"
            android:layout_marginTop="@dimen/espacoTop"
            android:src="@drawable/andar" />

        <ImageView
            android:id="@+id/direita"
            android:layout_width="@dimen/comando"

            android:layout_height="@dimen/comando"
            android:layout_alignLeft="@+id/andar"
            android:layout_alignTop="@+id/andar"
            android:layout_marginTop="@dimen/distanciaH"
            android:src="@drawable/direita" />

        <ImageView
            android:id="@+id/esquerda"
            android:layout_width="@dimen/comando"

            android:layout_height="@dimen/comando"
            android:layout_alignLeft="@+id/andar"
            android:layout_alignTop="@+id/direita"
            android:layout_marginTop="@dimen/distanciaH"
            android:src="@drawable/esquerda" />

        <ImageView
            android:id="@+id/funcao"
            android:layout_width="@dimen/comando"

```

```

        android:layout_height="@dimen/comando"
        android:layout_alignLeft="@+id/andar"
        android:layout_alignTop="@+id/esquerda"
        android:layout_marginTop="@dimen/distanciaH"
        android:src="@drawable/funcao" />

<!-- -->

<ImageView
    android:id="@+id/s01"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/andar"
    android:layout_alignTop="@+id/andar"
    android:layout_marginLeft="@dimen/distanciaV0"
    android:src="@drawable/circulo" />

<ImageView
    android:id="@+id/s02"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s01"
    android:layout_alignTop="@+id/andar"
    android:layout_marginLeft="@dimen/distanciaV"
    android:src="@drawable/circulo" />

<ImageView
    android:id="@+id/s03"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s02"
    android:layout_alignTop="@+id/andar"
    android:layout_marginLeft="@dimen/distanciaV"
    android:src="@drawable/circulo" />

<ImageView
    android:id="@+id/s04"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s03"
    android:layout_alignTop="@+id/andar"
    android:layout_marginLeft="@dimen/distanciaV"
    android:src="@drawable/circulo" />

<!-- -->

<ImageView
    android:id="@+id/s05"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s09"
    android:layout_alignStart="@+id/s09"
    android:layout_alignTop="@+id/s07"
    android:src="@drawable/circulo" />

<ImageView
    android:id="@+id/s06"

```

```

    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s02"
    android:layout_alignTop="@+id/direita"
    android:src="@drawable/circulo" />

```

```

<ImageView
    android:id="@+id/s07"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s03"
    android:layout_alignTop="@+id/direita"
    android:src="@drawable/circulo" />

```

```

<ImageView
    android:id="@+id/s08"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s04"
    android:layout_alignTop="@+id/direita"
    android:src="@drawable/circulo" />

```

```

<!-- -->

```

```

<ImageView
    android:id="@+id/s09"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s01"
    android:layout_alignTop="@+id/esquerda"
    android:src="@drawable/circulo" />

```

```

<ImageView
    android:id="@+id/s10"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s02"
    android:layout_alignTop="@+id/esquerda"
    android:src="@drawable/circulo" />

```

```

<ImageView
    android:id="@+id/s11"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s03"
    android:layout_alignTop="@+id/esquerda"
    android:src="@drawable/circulo" />

```

```

<ImageView
    android:id="@+id/s12"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:src="@drawable/circulo"
    android:layout_alignTop="@+id/s11"
    android:layout_alignLeft="@+id/s08"
    android:layout_alignStart="@+id/s08" />

```

```

<!-- -->

<ImageView
    android:id="@+id/f01"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s01"
    android:layout_alignTop="@+id/funcao"
    android:src="@drawable/circulo" />

<ImageView
    android:id="@+id/f02"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s02"
    android:layout_alignTop="@+id/funcao"
    android:src="@drawable/circulo" />

<ImageView
    android:id="@+id/f03"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s03"
    android:layout_alignTop="@+id/funcao"
    android:src="@drawable/circulo" />

<ImageView
    android:id="@+id/f04"
    android:layout_width="@dimen/comando"

    android:layout_height="@dimen/comando"
    android:layout_alignLeft="@+id/s04"
    android:layout_alignTop="@+id/funcao"
    android:src="@drawable/circulo" />

<!-- -->

<ImageView
    android:id="@+id/enviar"
    android:layout_width="@dimen/enviar"
    android:layout_height="@dimen/enviar"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignTop="@+id/s04"
    android:onClick="enviar"
    android:src="@drawable/enviar" />

<ImageView
    android:id="@+id/limpar"
    android:layout_width="@dimen/comando"
    android:layout_height="@dimen/comando"
    android:layout_alignRight="@+id/enviar"
    android:layout_alignTop="@+id/funcao"
    android:onClick="limpar"
    android:src="@drawable/limpar" />

<ImageView
    android:id="@+id/retangulo"
    android:layout_width="780dp"

```

```
        android:layout_height="140dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="370dp"
        android:src="@drawable/retangulo"
    />

    <ImageView
        android:id="@+id/socketVazio"
        android:layout_width="@dimen/comando"
        android:layout_height="@dimen/comando"
        android:src="@drawable/circulo"
        android:visibility="invisible" />

    </RelativeLayout>
</LinearLayout>
```

B.5: activity_tela_login.xml

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fundo"
    tools:context="com.almeida.tcc.Tela_login">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:textColor="#FFFFFF"
            android:id="@+id/usuario_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:layout_marginLeft="210dp"
            android:layout_marginStart="210dp"
            android:layout_marginTop="114dp"
            android:text="@string/login"
            android:textSize="30dp" />

        <EditText
            android:textColor="#FFFFFF"
            android:id="@+id/usuario_et"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignTop="@+id/usuario_tv"
            android:layout_marginLeft="51dp"
            android:layout_marginStart="51dp"
            android:layout_toEndOf="@+id/usuario_tv"
            android:layout_toRightOf="@+id/usuario_tv"
            android:textSize="30dp" />

        <TextView
            android:textColor="#FFFFFF"
            android:id="@+id/textView4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignLeft="@+id/usuario_tv"
            android:layout_below="@+id/usuario_et"
            android:text="@string/carro"
            android:textSize="30dp" />

        <EditText
            android:textColor="#FFFFFF"
            android:id="@+id/carro_et"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:layout_alignLeft="@+id/usuario_et"
            android:layout_alignStart="@+id/usuario_et"
            android:layout_alignTop="@+id/textView4" />

        <Button
            android:textColor="#FFFFFF"
            android:id="@+id/login"

```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="login"
        android:text="@string/botaoLogin"
        android:layout_marginTop="230dp"
        android:layout_centerHorizontal="true"/>

        <TextView
            android:textColor="#FF0000"
            android:id="@+id/msgErro"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="75dp"
            android:layout_centerHorizontal="true" />

    </RelativeLayout>
</FrameLayout>
```

B.6: strings.xml

```
<resources>
  <string name="app_name">TCC</string>

  <string name="dummy_button">Dummy Button</string>
  <string name="dummy_content">DUMMY\nCONTENT</string>

  <string name="title_activity_drag_drop">Drag_drop</string>
  <string name="action_settings">Settings</string>

  <string name="NomeServidor">1.1.10.1</string>

  <!-- tela de login -->
  <string name="login">Login:</string>
  <string name="carro">Carro:</string>
  <string name="botaoLogin">Login</string>

  <string name="erroLogin">Usuário não encontrado.</string>
  <string name="erroCarro">Carro não encontrado.</string>
  <string name="erroLoginCarro">Usuário e carro não encontrados.</string>
</resources>
```

Anexo C – Código do sistema Supervisório

C.1: index.php

```
<!doctype html>
<html>
<head>
<title></title>
<link rel="stylesheet" type="text/css" href="estilo.css"/>

<?php
    session_start();
    session_destroy();
?>

</head>
<body id="login">
<div id="login">
<form id="login" method="post" action="verificaLogin.php">
    <label for="cLoginUsuario">Usuário:</label><br>
    <input type="text" name="tLoginUsuario" id="cLoginUsuario"
size="40"><br><br>
    <label for="senha">Senha:</label><br>
    <input type="password" name="tSenha" id="senha" size="40"><br>
    <input type="submit" name="bLogin" value="Login">
</form>
</div>
</body>
</html>
```

C.2: home.php

```

<!doctype html>
<html>
<head>
<meta charset="iso-8859-1">
<title>Home</title>
<link rel="stylesheet" type="text/css" href="estilo.css"/>
<?php
    include 'conectaDB.php';
    session_start();

    if (!isset($_SESSION["PROFESSOR_ID"])){
        header("Location: index.php");
    }

    $professor_id=$_SESSION["PROFESSOR_ID"];

    $ordenar = $_GET["ordenar"];

    if ($ordenar == 'nome'){
        $ordem = 'A.nome ASC';
    }
    else if ($ordenar == 'atividade'){
        $ordem = 'A.ultimaAtividade ASC';
    }

    $sqlL="SELECT A.idAluno, A.nome, A.ultimaAtividade FROM Aluno A WHERE
A.Professor_idProfessor = '$professor_id' ORDER BY $ordem";
    $resultado=$mysqli->query($sqlL);
    $buscaL = mysqli_query($mysqli,$sqlL);
    $colunas = mysqli_num_rows($buscaL);
?>
</head>

<body>
<div id="limite">
    <div id="barraSuperior">
        <?php if ($_SESSION["ADMIN"]=="1"){ ?>
        <p style="position:absolute; top:50px;">
        Cadastrar:<br>
        <a href="cadastroProfessor_form.php">Professor</a> | <a
href="cadastroAluno_form.php">Aluno</a> | <a
href="cadastroCarro_form.php">Carro</a>

        <?php }?>
        <p style="position:absolute; top:70px; right:0px;" > <?php echo
$_SESSION["PROFESSOR_NOME"]; ?> [<a href="index.php">Sair</a>]</p>
    </div>

    <div id="conteudo">
    <h1> Lista de alunos </h1>
    <table>
    <tr bgcolor="#CCCCCC">
        <td width="500"><?php if ($ordenar=='nome') echo '<strong>' ?><a
href="home.php?ordenar=nome">Aluno</a><?php if ($ordenar=='nome') echo
'</strong>' ?></td>
        <td width="200"><?php if ($ordenar=='atividade') echo '<strong>'
?><a href="home.php?ordenar=atividade">Última atividade</a><?php if
($ordenar=='atividade') echo '</strong>' ?></td>
    </tr>

```

```
<?php while ($linha = $resultado->fetch_row()){ ?>
<tr>
  <td> <a href="alunoAtividades.php?aluno=<?php echo $linha[0]
?>"><?php echo $linha[1] ?></a> </td> <!-- Aluno -->
  <td> <?php echo date("d/m/Y - H:i:s", strtotime($linha[2])) ?>
</td> <!-- Última atividade -->
</tr>

<?php } ?>
</table>
</div>
</div>
</body>
</html>
```

C.3: alunoAtividades.php

```

<!doctype html>
<html>
<head>
<meta charset="iso-8859-1">
<title>Home</title>
<link rel="stylesheet" type="text/css" href="estilo.css"/>
<?php
    include 'conectaDB.php';

    session_start();
    if (!isset($_SESSION["PROFESSOR_ID"])){
        header("Location: index.php");
    }

    $alunoID = $_GET["aluno"];

    /* Nome do Aluno */
    $sql="SELECT * FROM Aluno WHERE idAluno='$alunoID'";
    $buscaLogin = mysqli_query($mysqli,$sql);
    $coluna = mysqli_fetch_array($buscaLogin);
    $alunoNome = $coluna['nome'];

    /* Trajetos */
    $sqlL="SELECT T.idTrajeto, T.sequencia, T.funcao, T.data,
T.comentario FROM Trajeto T WHERE T.Aluno_idAluno = '$alunoID'";
    $resultado=mysqli_query($mysqli,$sqlL);
    $buscaL = mysqli_query($mysqli,$sqlL);
    $colunas = mysqli_num_rows($buscaL);
?>
</head>

<body>
<div id="limite">
    <div id="barraSuperior">
        <p style="position:absolute; top:70px; right:0px;" > <a
href="home.php?ordenar=nome"><?php echo $_SESSION["PROFESSOR_NOME"]; ?></a>
[<a href="index.php">Sair</a>]</p>
    </div>

    <div id="conteudo">
<h1> <?php echo $alunoNome;      ?> </h1>
<table>
<tr bgcolor="#CCCCCC">
    <td>Trajeto_id</td>
    <td>Sequencia</td>
    <td>Função</td>
    <td>Hora</td>
    <td>Comentário</td>

</tr>

<?php while ($linha = $resultado->fetch_row()){ ?>
<tr>
    <td><?php echo $linha[0] ?></td>
    <td><?php echo $linha[1] ?></td>
    <td><?php echo $linha[2] ?></td>
    <td><?php echo date("d/m/Y - H:i:s", strtotime($linha[3])) ?></td>
    <td><a href="comentario.php?trajeto=<?php echo $linha[0] ?>"
target="_new"></a><?php echo $linha[4] ?></td>

```

```
</tr>
  <?php } ?>
</table>
</div>
</div>
</body>
</html>
```

C.4: cadastroAluno_form.php

```

<!doctype html>
<html>
<head>
<meta charset="iso-8859-1">
<title>Home</title>
<link rel="stylesheet" type="text/css" href="estilo.css"/>
<?php
    include 'conectaDB.php';

    session_start();
    if (!isset($_SESSION["PROFESSOR_ID"])){
        header("Location: index.php");
    }

    $sqlL="SELECT P.idProfessor, P.Nome FROM Professor P";
    $resultado=$mysqli->query($sqlL);
    $buscaL = mysqli_query($mysqli,$sqlL);
    $colunas = mysqli_num_rows($buscaL);
?>
</head>

<body>
<div id="limite">
    <div id="barraSuperior">
        <p style="position:absolute; top:70px; right:0px;" > <?php echo
        $_SESSION["PROFESSOR_NOME"]; ?> [<a href="index.php">Sair</a>]</p>
    </div>

    <div id="conteudo">
<h1> Cadastrar Aluno </h1>

    <form id="cadastroAluno" method="post" action="cadastroAluno.php">

        <label for="cAlunoNome">Nome:</label>
        <input type="text" name="nome" id="cAlunoNome" size="40"><br><br>

        <label for="cAlunoLogin">Login:</label>
        <input type="text" name="login" id="cAunoLogin"
size="40"><br><br>

        <label for="cAlunoProfessor">Professor:</label>
        <select name="professor">
            <?php while ($linha = $resultado->fetch_row()){ ?>
                <option value="<?php echo $linha[0]?>"><?php echo
$linha[1]?></option>
            <?php }?>
        </select><br><br>
        <input type="submit" name="bLogin" value="Salvar">
    </form>

    </div>
</div>
</body>
</html>

```

C.5: cadastroAluno.php

```
<?php
    include 'conectaDB.php';
    session_start();

    $nome = $_POST['nome'];
    $login = $_POST['login'];
    $professor = $_POST['professor'];

    $sql = "INSERT INTO Aluno (nome, login, Professor_idProfessor) VALUES
('$nome', '$login', '$professor')";
    $insere=mysqli_query($mysqli, $sql);

    date_default_timezone_set('America/Belem');
    $dia = date('Y-m-d H:i:s');

    $sqlQ = "UPDATE Aluno SET ultimaAtividade='$dia' WHERE
login='$login'";
    $update=$mysqli->query($sqlQ);

    header("Location: home.php?ordenar=nome");
?>
```

C.6: cadastroProfessor_form.php

```

<!doctype html>
<html>
<head>
<meta charset="iso-8859-1">
<title>Home</title>
<link rel="stylesheet" type="text/css" href="estilo.css"/>
<?php
    include 'conectaDB.php';

    session_start();
    if (!isset($_SESSION["PROFESSOR_ID"])){
        header("Location: index.php");
    }

    $alunoID = $_GET["aluno"];
?>
</head>

<body>
<div id="limite">
    <div id="barraSuperior">
        <p style="position:absolute; top:70px; right:0px;" > <?php echo
$_SESSION["PROFESSOR_NOME"]; ?> [<a href="index.php">Sair</a>]</p>
    </div>

    <div id="conteudo">
<h1> Cadastrar Professor </h1>

    <form id="cadastroProfessor" method="post"
action="cadastroProfessor.php">

        <label for="cProfessorNome">Nome:</label>
        <input type="text" name="nome" id="cProfessorNome" size="40"><br><br>

        <label for="cProfessorLogin">Login:</label>
        <input type="text" name="login" id="cProfessorLogin"
size="40"><br><br>

        <label for="sSenha">Senha:</label>
        <input type="password" name="senha" id="senha" size="40"><br><br>

        <label for="bAdministrador">Administrador</label>
        <input type="checkbox" name="administrador" value="1"><br><br>

        <input type="submit" name="bLogin" value="Salvar">
</form>

    </div>
</div>
</body>
</html>

```

C.7: cadastroProfessor.php

```
<?php
    include 'conectaDB.php';
    session_start();

    $nome = $_POST['nome'];
    $login = $_POST['login'];
    $senha = $_POST['senha'];
    $administrador = $_POST['administrador'];

    if ($administrador != '1'){
        $administrador =0;
    }

    $sql = "INSERT INTO Professor (nome, login, senha, administrador)
VALUES ('$nome', '$login', '$senha', '$administrador')";
    $insere=mysqli_query($mysqli, $sql);

    header("Location: home.php?ordenar=nome");
?>
```

C.8: cadastroCarro_form.php

```

<!doctype html>
<html>
<head>
<meta charset="iso-8859-1">
<title>Home</title>
<link rel="stylesheet" type="text/css" href="estilo.css"/>
<?php
    include 'conectaDB.php';

    session_start();
    if (!isset($_SESSION["PROFESSOR_ID"])){
        header("Location: index.php");
    }

    $alunoID = $_GET["aluno"];
?>
</head>

<body>
<div id="limite">
    <div id="barraSuperior">
        <p style="position:absolute; top:70px; right:0px;" > <?php echo
$_SESSION["PROFESSOR_NOME"]; ?> [<a href="index.php">Sair</a>]</p>
    </div>

    <div id="conteudo">
<h1> Cadastrar carro </h1>

    <form id="cadastroCarro" method="post" action="cadastroCarro.php">

        <label for="cCarroMAC">MAC:</label>
        <input type="text" name="MAC" id="cCarroMAC" size="40"><br><br>

        <label for="cCarroIP">IP:</label>
        <input type="text" name="IP" id="cCarroIP" size="40"><br><br>

        <input type="submit" name="bLogin" value="Salvar">
    </form>

    </div>
</div>
</body>
</html>

```

C.9: cadastroCarro.php

```
<?php
    include 'conectaDB.php';
    session_start();

    $MAC = $_POST['MAC'];
    $IP = $_POST['IP'];

    $sql = "INSERT INTO Carro (MAC, IP) VALUES ('$MAC', '$IP')";
    $insere=mysqli_query($mysqli, $sql);

    header("Location: home.php?ordenar=nome");
?>
```

C.10: comentário.php

```

<!doctype html>
<html>
<head>
<meta charset="iso-8859-1">
<title>Comentario</title>
<link rel="stylesheet" type="text/css" href="estilo.css"/>
<?php
    include 'conectaDB.php';
    session_start();
    if (!isset($_SESSION["PROFESSOR_ID"])){
        header("Location: index.php");
    }

    $trajeto = $_GET["trajeto"];

    $sql="SELECT A.nome, T.data, T.sequencia, T.funcao, T.comentario FROM
Aluno A, Trajeto T WHERE T.Aluno_idAluno = A.idAluno AND
T.idTrajeto='$trajeto'";
    $buscaLogin = mysqli_query($mysqli,$sql);
    $coluna = mysqli_fetch_array($buscaLogin);
?>
</head>

<body id="comentario">
<h1>Editar comentário</h1>
<p>Aluno: <?php echo $coluna['nome'] ?></p>
<p>Data da atividade: <?php echo date("d/m/Y - H:i:s",
strtotime($coluna['data']))?></p>
<p> Sequencia: <?php echo $coluna['sequencia'] ?></p>
<p> Função: <?php echo $coluna['funcao'] ?></p>

<form id="comentario" method="post"
action="salvaComentario.php?trajeto=<?php echo $trajeto?>">
    <textarea name="comentario" rows="10" cols="40"><?php echo
$coluna['comentario'] ?></textarea><br>
    <input type="image" src="IMG_ok.png" width="50px" height="50px;
"alt="enviar" />
    <a href="home.php?ordenar=nome"></a>
</form>
</body>
</html>

```

C.11: salvaComentario.php

```
<?php
    include 'conectaDB.php';
    session_start();

    $comentario = $_POST['comentario'];
    $trajeto = $_GET['trajeto'];

    $sqlQ = "UPDATE Trajeto SET comentario='$comentario' WHERE
idTrajeto='$trajeto'";
    $update=$mysqli->query($sqlQ);

    header("Location: home.php?ordenar=nome");
?>
```

C.12: loginCarro.php

```
<?php
include "conectaDB.php";

$aluno = $_GET["aluno"];
$carro = $_GET["carro"];

$sql = "SELECT idAluno FROM Aluno WHERE login='$aluno'";
$busca = mysqli_query($mysqli,$sql);
$numLinhasAluno = mysqli_num_rows($busca);
$idAluno = mysqli_fetch_array($busca);

$sql = "SELECT IP FROM Carro WHERE idCarro='$carro'";
$busca = mysqli_query($mysqli,$sql);
$numLinhasIp = mysqli_num_rows($busca);
$ip = mysqli_fetch_array($busca);

$resultado=0;

if ($numLinhasAluno == 1 && $numLinhasIp == 1){
    echo $resultado,',',,$idAluno['idAluno'],'',,$ip['IP'];
}
else {
    if ($numLinhasAluno != 1){
        $resultado-=1;
    }
    if ($numLinhasIp != 1){
        $resultado-=2;
    }
    echo $resultado;
}

?>
```

C.13: trajeto.php

```
<?php
include "conectaDB.php";

$s = $_GET["s"];
$f = $_GET["f"];
$idAluno = $_GET["aluno"];

date_default_timezone_set('America/Belem');
$dia = date('Y-m-d H:i:s');

$sql = "INSERT INTO Trajeto (sequencia, funcao, data, Aluno_idAluno) VALUES
('$s', '$f', '$dia', '$idAluno')";
$insert = mysqli_query($mysqli, $sql);

$sqlQ = "UPDATE Aluno SET ultimaAtividade='$dia' WHERE idAluno='$idAluno'";
$update=mysqli->query($sqlQ);

?>
```

C.14: verificaLogin.php

```
<?php
include 'conectaDB.php';
session_start();

$usuario=$_POST['tLoginUsuario'];
$senha=$_POST['tSenha'];

$sql="SELECT * FROM Professor WHERE login='$usuario' AND
senha='$senha'";
$buscaLogin = mysqli_query($mysqli,$sql);
$colunas = mysqli_num_rows($buscaLogin);
$coluna = mysqli_fetch_array($buscaLogin);

if ($colunas == 1)
{
    $_SESSION["PROFESSOR_ID"]=$coluna['idProfessor'];
    $_SESSION["PROFESSOR_NOME"]=$coluna['nome'];
    $_SESSION["ADMIN"]=$coluna['administrador'];
    header("Location: home.php?ordenar=nome");
}

else
{
    header("Location: index.php");
}

?>
```

C15: conectaDB.php

```
<?php
    header('Content-Type: text/html; charset=iso-8859-1');

    $servidor = 'localhost';
    $usuario = 'root';
    $senha = 'root';
    $banco = 'supervisorio';

    $mysqli = new mysqli($servidor, $usuario, $senha, $banco);
    if (mysqli_connect_errno())
        trigger_error(mysqli_connect_error());
?>
```

C.16: estilo.css

```
/* CSS Document */

body {
    background-color: #EEE;
    font-family:"Gill Sans", "Gill Sans MT", "Myriad Pro", "DejaVu Sans
Condensed", Helvetica, Arial, sans-serif;

    background-image:url(fundo.jpg);
    background-repeat:repeat-x;
}

a {
    text-decoration:none;
    color:#000;
}

h1 {
    font-family:"Lucida Grande", "Lucida Sans Unicode", "Lucida Sans",
"DejaVu Sans", Verdana, sans-serif;
    size: 12px;
    border-bottom:inset;
}

table, tr, td {
    border-collapse: collapse;
    border:1px solid #666;
}

body#login {
    background-color: #6C9;
}

div#login{
    position: absolute;
    text-align:center;
    width: 100%;
    top: 250px;
}

body#comentario {
    background:#EEE;
    width: 450px;
}

div#limite {
    position:absolute;
    width: 900px;
    margin-left:-450px;
    left:50%;
    top: 0px;
    height:100%;
}

div#barraSuperior {
    height:110px;
    top:0px;
}
```