

Plataforma Embarcada: um rastreador de veículos de transporte terrestre

Clodoaldo José de Souza

Fatec – Faculdade de Tecnologia de Garça - clodoaldo.souza01@fatec.sp.gov.br

Luiz Carlos Querino Filho

Fatec – Faculdade de Tecnologia de Garça - luiz.querino@fatec.sp.gov.br

Resumo

A telemetria possibilita o envio e recebimento de dados para comando, medição e rastreamento de objetos distantes. Uma forma de telemetria moderna é a recepção de dados de localização de um objeto equipado com um receptor do sistema de posicionamento global (GPS – Global Positioning System). Este projeto apresenta uma plataforma de software e hardware que poderá ser utilizada em um equipamento de rastreamento previamente instalado em veículos de transporte terrestre. A plataforma envolve o uso de um microcontrolador Arduino acoplado a um receptor de sinal GPS e um dispositivo de GPRS (General Packet Radio Service) para envio dos dados a um website, onde os mesmos poderão ser analisados. Planeja-se com essa plataforma oferecer a empresas e indivíduos meios de otimizar seus processos de transporte e logística.

Abstract

Telemetry enables the sending and receiving of data for command, measurement and tracking of distant objects. One form of modern telemetry is the reception of location data from an object with a Global Positioning System (GPS) receiver. This project presents a software and hardware platform that can be used in ground transportation vehicle tracking equipment. The platform involves the use of an Arduino microcontroller coupled to a GPS signal receiver and a General Packet Radio Service (GPRS) device for transmitting data from a site where it can be further analyzed. It is planned with this platform to offer companies and individuals with ways to improve its transport and logistics processes.

1 INTRODUÇÃO, JUSTIFICATIVA E OBJETIVO

No século XX, a ciência e as tecnologias alcançaram avanços profundos e significativos. Nesse contexto, nasceu a Informática que se desenvolveu em nível acelerado e adentrou o século XXI com grandes perspectivas no cenário mundial.

No universo de abrangência da Informática, encontra-se o tema escolhido para a pesquisa, Programação Embarcada para Rastreamento de Veículos de Transporte Terrestre, para pessoas ou cargas, cuja função é informar a posição do veículo por meio de instrumentos de telemetria, ou seja, rastreamento via satélite.

A telemetria é um sistema tecnológico de monitoramento, que não utiliza cabos, que envia e recebe dados por meio de comunicação via rádio ou satélite, para comandar, medir ou rastrear alguma coisa à distância. A palavra telemetria se originou do grego *tele* que significa “remoto” e *metron* que significa “medida”. Esta tecnologia é muito utilizada em objetos que estão sempre em movimento, como os carros (BRANCO, 2006).

Por meio da telemetria, é informada a posição do veículo por meio de um sistema com programação embarcada que contará com um circuito integrado

microcontrolador baseado na plataforma Arduino e estará conectado a um dispositivo de GPS que por meio de satélite recebe informações de coordenadas, e um dispositivo de GPRS para envio de dados para um site web que fará o monitoramento.

Oportunizar as empresas e as pessoas que fazem uso das tecnologias em questão, eficácia na dinâmica do processo de embarque e desembarque de pessoas e coleta e entrega de produtos, para maior qualidade nos serviços prestados pelas empresas.

O objetivo desse trabalho é apresentar uma plataforma de *hardware* e *software* que possa ser usada para identificar as falhas no processo logístico em transporte de cargas e passageiros, as quais implicam na perda de tempo, em custos e benefícios, e na qualidade dos serviços prestados.

Desenvolver um software e um hardware utilizados em um equipamento de rastreamento, previamente instalados em veículos de transporte que farão a transmissão de dados por meio de telemetria, a partir de um serviço de rede de internet móvel.

2 METODOLOGIA UTILIZADA

A ferramenta utilizada para o desenvolvimento deste artigo é um hardware utilizando a programação embarcada na plataforma Arduino que utiliza a linguagem de programação C++.

O hardware contará com duas placas denominadas *Shields*, que serão conectadas à placa principal da plataforma Arduino, sendo uma placa de GPS NEO6M e uma placa GPRS SIM800L.

O software apresentará dados do veículo rastreado como, por exemplo, a localização, a velocidade em que o veículo estiver naquele momento específico, entre outras informações pertinentes.

A metodologia de desenvolvimento utilizada é a Prototipação. A metodologia de prototipação é importante neste projeto para verificar se está sendo executado de maneira correta, caso contrário, pode-se conhecer as falhas e suas possíveis soluções (COSTA, 2011 e Nepomuceno, 2012).

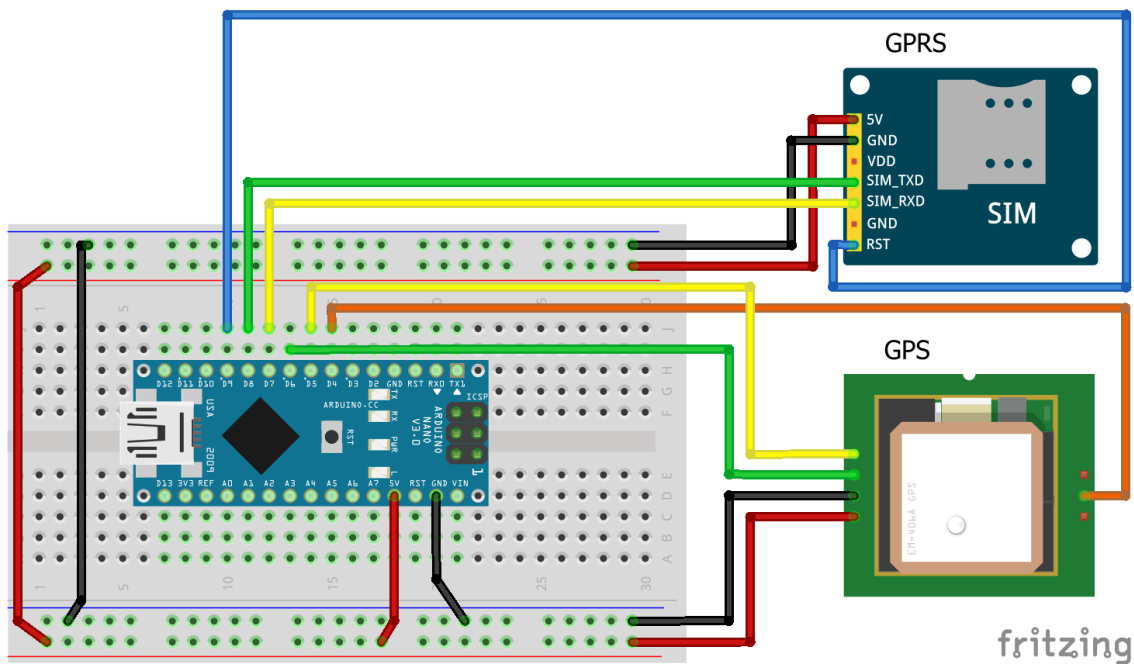
A experiência que é adquirida no desenvolvimento do protótipo será utilizada nas próximas etapas do desenvolvimento do sistema, permitindo melhores resultados e redução de custos que ajudará no processo de definição de requisitos

3 DESENVOLVIMENTO

Uma abstração das conexões do Arduino com os *Shields*¹ de GPS e GPRS é apresentado na Figura 1.

Figura 1 – Abstração das conexões do Arduino Nano e Shields.

¹ Shield (escudo, em inglês) é o nome usado para as placas e módulos de expansão que podem ser conectadas em um microcontrolador Arduino (ARDUINO, 2018)



Fonte: Próprio autor

Como pode ser observado na Figura 1, os *Shields* de GPRS e GPS estão conectados a uma placa de prototipagem (*protoboard*), na qual também se encontra o microcontrolador Arduino. Esta abstração reflete também o ambiente de teste utilizado para o desenvolvimento.

Após realizar as conexões, no ambiente IDE do Arduino foram adicionadas as bibliotecas e atribuídos os nomes para os pinos utilizados nas conexões. Essa atribuição de nomes para os pinos ajudará e facilitará no desenvolvimento do código fonte. A parte do código pode ser visto na Figura 2.

Figura 2 – Adicionando os nomes das bibliotecas e nomenclatura dos pinos utilizados.

```

1  /* PROJETO RASTRO FATEC
2
3  Autor: Clodoaldo José de Souza
4
5
6  */
7
8  // Inclusão de bibliotecas
9  #include <TinyGPS++.h>
10 #include <SoftwareSerial.h>
11
12 // Definição dos pinos
13 // GPS PINOS
14 #define GPS_LED 4
15 #define GPS_RX 5
16 #define GPS_TX 6
17 // GSM PINOS
18 #define GSM_RX 7
19 #define GSM_TX 8
20 #define GSM_RST 9
21

```

Fonte: Próprio autor

Como forma de identificar o dispositivo é utilizado o código IMEI. O IMEI (*International Mobile Equipment Identity*, ou identidade internacional de equipamento

móvel) é um conjunto de 15 números que formam um código único e global (ANATEL, 2017).

Figura 3 – Código para verificar IMEI do aparelho.

```
87 serial_GSM.write("AT+CGSN\r"); // envia o comando de aquisição do imei
88 if (serial_GSM.available()) // Se tem algo impresso na serial_GSM, armazena em buffer
89 {
90     while (serial_GSM.available())
91     {
92         buffer[count++] = serial_GSM.read();
93         if (count == tamanho)break; // Se tamanho for igual count, para
94     }
95     aux = 0;
96     for (int i = 0; i < tamanho; i++)
97     {
98         if ( ( buffer[i] == '0' ) || ( buffer[i] == '1' ) || ( buffer[i] == '2' ) || ( buffer[i] == '3' )
99             || ( buffer[i] == '4' ) || ( buffer[i] == '5' ) || ( buffer[i] == '6' )
100             || ( buffer[i] == '7' ) || ( buffer[i] == '8' ) || ( buffer[i] == '9' ) )
101         {
102             if (aux < 15)
103             {
104                 imei[aux] = buffer[i];
105                 aux++;
106             }
107         }
108     }
109 }
```

Fonte: Próprio autor

No trecho de código apresentado na Figura 3, é exemplificado o processo de comunicação do Arduino com um de seus *Shields*: enquanto forem obtidas leituras do módulo GSM, seus dados de IMEI são guardados em um vetor.

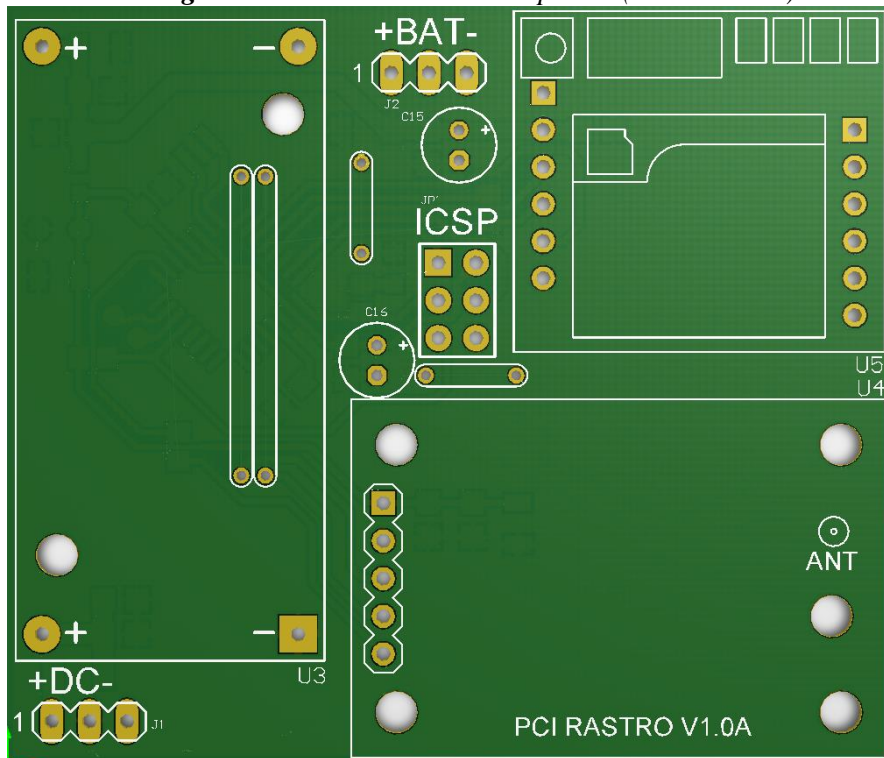
A seguir, são explicados os principais procedimentos realizados no código listado na Figura 3:

- Linha 87: é enviado via serial o comando AT responsável por responder o IMEI.
- Linha 88: verifica se há resposta do *Shield* de GPRS.
- Linha 90: verifica enquanto houver dados na comunicação serial.
- Linha 92: vetor recebe a resposta da serial.
- Linha 93: verifica se a contagem é igual ao tamanho do vetor de resposta da serial.
- Linha 95: a variável aux é zerada. Será utilizada para ser utilizada no índice do vetor.
- Linha 96: realiza a contagem do tamanho do vetor de resposta da serial, onde contém o IMEI e outros dados.
- Linhas 97, 98 e 99: realiza a checagem índice a índice até encontrar os valores numéricos.
- Linha 102: verifica se aux tem o tamanho menor que quinze.
- Linha 104: enquanto a variável aux for menor que quinze, vai guardando no vetor IMEI os números recebidos pela serial.
- Linha 105: a variável aux é incrementada.

4 RESULTADOS OBTIDOS

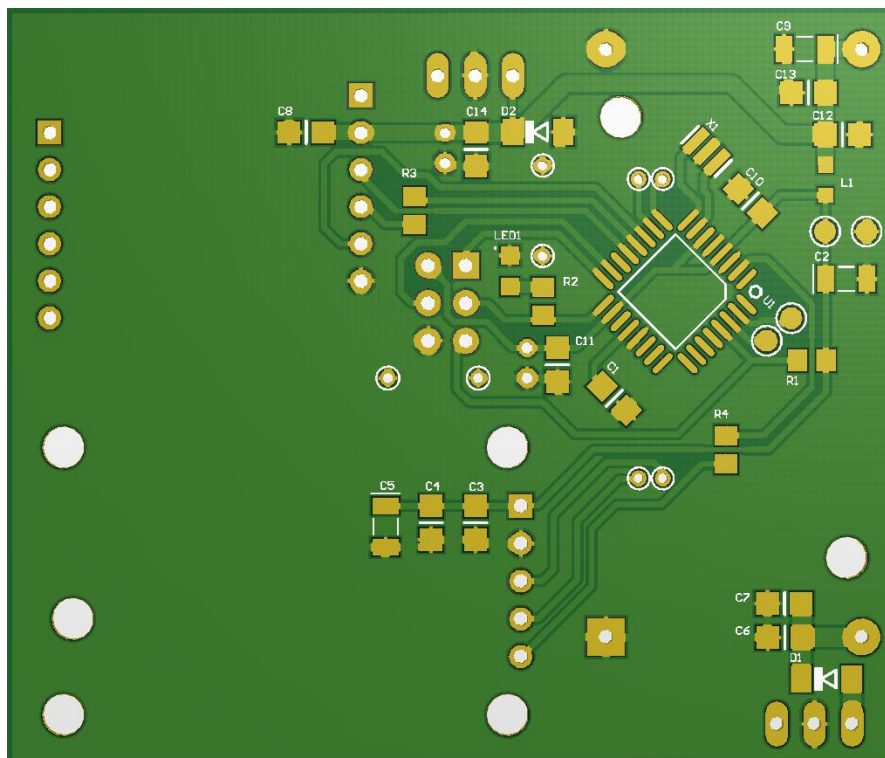
Um dos resultados obtidos é a confecção da placa de circuito impresso, em tamanho reduzido de 6x6 cm, esta montagem do protótipo sem a utilização de protoboard melhorou as conexões dos *Shields* com o microcontrolador ATMEGA328 (Atmega328 datasheet) utilizado no Arduino Nano.

Figura 4 – Placa de Circuito Impresso (vista de cima).



Fonte: Próprio autor

Figura 5 – Placa de Circuito Impresso (vista de baixo).



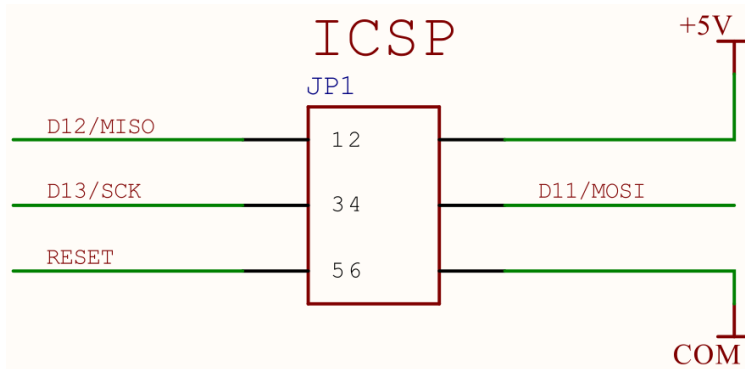
Fonte: Próprio autor

O processo de gravação na placa é utilizando o formato ICSP (*In Circuit Serial Programming*), desta maneira é possível gravar o *software/firmware* no

microcontrolador diretamente soldado na placa, sem a necessidade de retirá-lo, gravá-lo e soldá-lo novamente.

O jumper de programação ICSP utiliza os pinos de alimentação VCC (representados pelos pinos 4 e 6) e GND (representados pelos pinos 3,5 e 21) além dos pinos de Reset (representado pelo pino 29), MISO (representado pelo pino 16), MOSI (representado pelo pino 15) e o SCK (representado pelo pino 17).

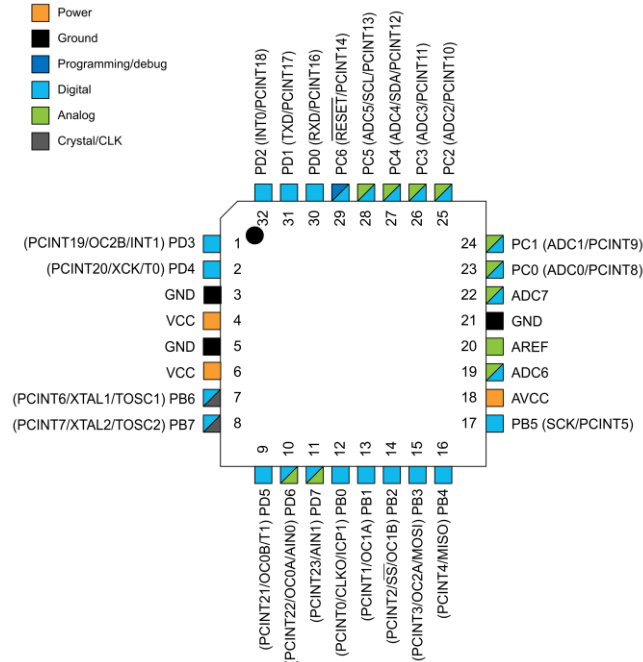
Figura 6 – Placa de Circuito Impresso (vista de baixo).



Fonte: Próprio autor

A localização dos pinos utilizados para a gravação pode ser visualizada na Figura 7.

Figura 7 – Pinout do Atmega328.



Fonte: Próprio autor

O código fonte da figura a seguir foi escrito em PHP para testar a recepção dos dados de localização gerados pelo módulo GPS e transmitido por meio do módulo de GPRS.

Figura 8 – Código fonte em PHP para checar os dados enviados pelo protótipo.

```
1 <title>Inserindo dados...</title>
2 <?php
3 $servername = "localhost";
4 $username = "id222222229_teste";
5 $password = "123456";
6 $dbname = "id222222229_sim800";
7 // Cria a conexao com o banco de dados
8 $conn = new mysqli($servername, $username, $password, $dbname);
9 // Verifica a conexao
10 if ($conn->connect_error) {
11     die("Falha de conexao: " . $conn->connect_error);
12 }
13 if (!empty($_GET["id"])) // Checa pra ver se não está vazio
14 {
15     $sql = "INSERT INTO sim800 (imei, latitude, longitude, velocidade) VALUES
16         ('".$_GET["id"]."','".$_GET["lat"]."','".$_GET["lon"]."','".$_GET["vel"]."')";
17
18     if ($conn->query($sql) == TRUE) {
19         echo "Valores Adicionados com sucesso!!!";
20     } else {
21         echo "Erro: " . $sql . "<br>" . $conn->error;
22     }
23     $conn->close();
24 }
25 else
26 {
27     echo "Parametros invalidos detectados.";
28 }
29 ?>
30
```

Fonte: Próprio autor

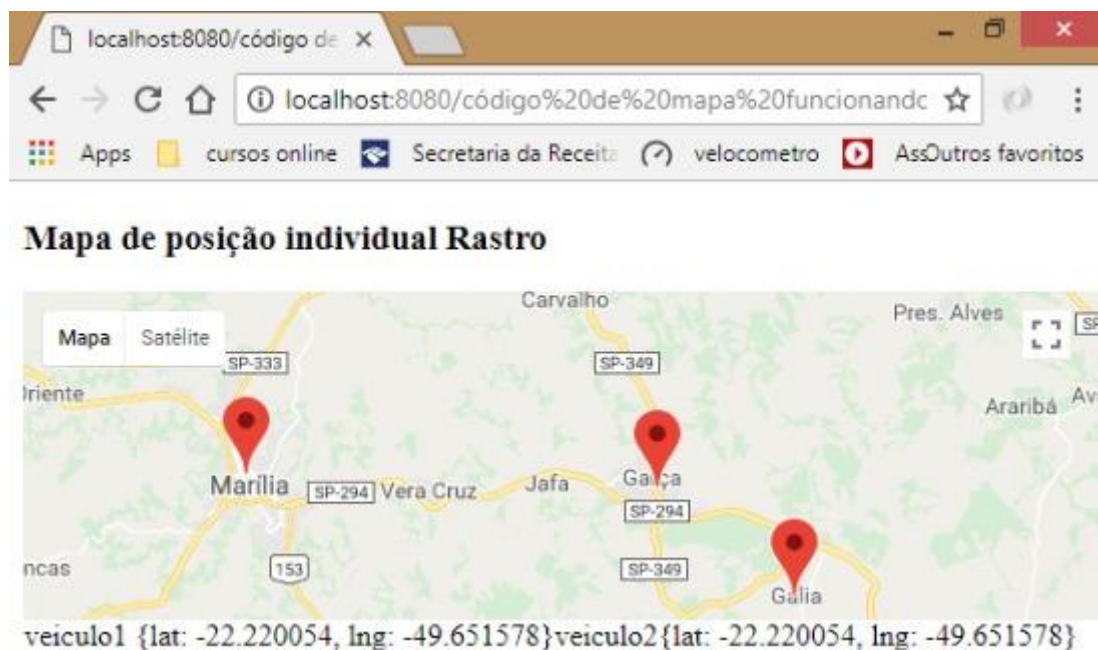
O protótipo envia por método POST os dados para a página escrita em linguagem PHP diretamente pela url do site (exemplo, <https://rastreo.com.br/rastrear.php/?imei=123456789012345&lat=-22.00&long=-49.00&vel=60>).

A seguir, são explicados os principais procedimentos realizados no código listado na figura 8:

- Linha 1: é o título da página PHP.
- Linha 2: comando que inicia a linguagem PHP.
- Linhas 3, 4, 5 e 6: são responsáveis pelo local do servidor de banco de dados SQL, usuário de acesso ao banco de dados, senha de acesso ao banco de dados e por fim, o nome do banco de dados.
- Linha 8: realiza a conexão com o bando de dados.
- Linhas 10 e 11: se houver erro com o banco de dados, ele fecha a conexão.
- Linha 13: verifica se o “id” não está vazio.
- Linhas 15 e 16: havendo dados recebidos, estes são inclusos no banco por método de uma query que os adicionará.
- Linhas 18 e 19: exibe a mensagem de dados inseridos com sucesso.
- Linhas 20 e 21: exibe a mensagem caso haja erro.
- Linha 23: fecha a conexão com o banco de dados.
- Linhas 25, 26, 27 e 28: exibe mensagem, caso “id” esteja vazio.
- Linha 29: comando que finaliza a linguagem PHP.

O resultado dos dados é plotado no mapa, conforme Figura 9.

Figura 9 – Visualização das posições dos veículos no mapa.



Fonte: Próprio autor

5 CONSIDERAÇÕES FINAIS

Este trabalho se propôs, como objetivo geral, criar uma plataforma de *hardware* e *software* que possa informar sua localização e outros dados. Os resultados apresentados foram satisfatórios com base nos testes de campo.

Como objetivos futuros a melhoria no código do microcontrolador, para que se torne mais eficiente em consumo de memória e dados de telefonia, customização do intervalo de tempo entre as transmissões, possibilidade de adicionar novas funcionalidades, como por exemplo monitoramento.

REFERÊNCIAS

ANATEL. 2017. Disponível em: < <http://www.anatel.gov.br/celularlegal/imei>>. Acesso em: 20 maio 2018.

ARDUINO, 2018. Disponível em <http://www.arduino.org>> Acesso em: 23 maio 2018

Atmega328 datasheet. Disponível em: <<http://www.microchip.com/wwwproducts/en/ATmega328>>. Acesso em: 23 maio 2018.

BRANCO, Agatha. Telemetria. 2006. Disponível em: <<http://www.administradores.com.br/artigos/marketing/telemetria-tudo-sobre-controle/29567/>>. Acesso em: 23 mar. 2017.

COSTA, L. 2011. Disponível em: <<http://www.semeru.com.br/blog/as-metodologias-tradicionais-de-desenvolvimento-de-software/>>. Acesso em: 20 maio 2018.

*FERREIRA, Fernanda. História do GPS. 2014. Disponível em: <
<http://www.gpspoint.com.br/gps/94-historia-do-gps> >. Acesso em: 23 mar. 2017.*

*Nepomuceno, Dênys. 2012. Disponível em:
<<http://enghariaadesoftwareuesb.blogspot.com.br/2012/12/blog-post.html>>. Acesso em: 20 maio 2018.*