# Bduíno - Um dispositivo para a captura de dados do contexto urbano para o uso da bicicleta.

Antonio Fernando Santos Ladeia, Orientador(a): Pablo Vieira Florentino[a]

[a]*Instituto Federal da Bahia*

*Keywords:* Microcontroladores, Dados ambientais, Bicicletas, Hardware aberto, Código aberto, Sistemas embarcados

## 1. Introdução

Enquanto os espaços para automóveis apenas aumentam no contexto atual, outros modais, não recebem a devida atenção das autoridades competentes, somando-se a isso uma possível falta de atratividade para o uso desses modais. De fato, boa parte dos investimentos públicos estão direcionados ao paradigma da mobilidade focada no veículo motorizado individual. Diversos estudos têm evidenciado este descompasso das ações públicas com outros modais, como o transporte público e os modais ativos, ou não motorizados.

Na Figura 1 é mostrado um gráfico que exemplifica a diferença de valores investidos mobilidade separados em transportes públicos e privados [1].

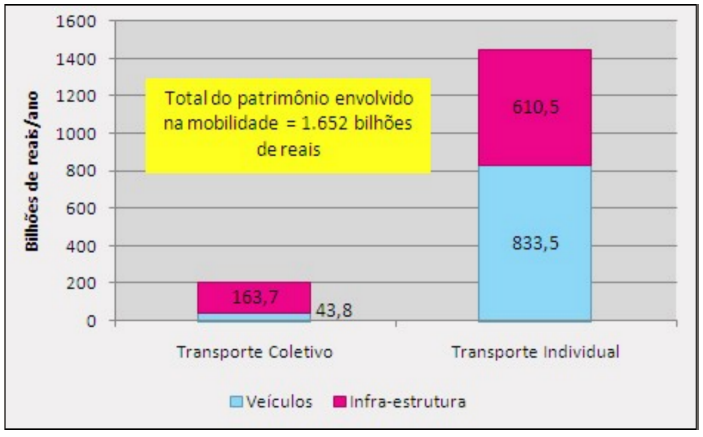


Figura 1: Investimento em mobilidade público vs privado

Questões ambientais e urbanas podem ser fatores determinantes para a popularização do uso de outros modais, como a bicicleta, por isso este trabalho posicionar-se de forma interdisciplinar, unindo áreas do Urbanismo e da Computação, no sentido de investigar os possíveis fatores que podem afastar ou trazer novos usuários para este modal.

No contexto do curso de Análise e Desenvolvimento de Sistemas do IFBA Salvador, os trabalhos SMRE - Sistema de Economia de Recursos Elétricos e Um Sistema Ubíquo para o Monitoramento Automático do Consumo de Água no Instituto Federal da Bahia, Campus Salvador são exemplos de pesquisas interdisciplinares, assim como este trabalho, de computação aplicada a outros contextos como economia de recursos e eletricidade.

Por conta da transversalidade do trabalho, faz-se necessário abordar temas de áreas correlatas ao mesmo, mas que não são necessariamente pertencentes à área da computação. Assim, alguns temas são contextualizados no sentido de detalhar o cenário e explicar como os mesmos são transversais se complementam.

Nesse intento, a aplicação Bduíno, produto desenvolvido ao longo deste trabalho, foi pensada desde a sua concepção para ser uma solução capaz de responder alguma dessas questões.

### *1.1. Justificativa*

A mobilidade urbana é um ponto de ampla discussão e impacto na sociedade atual. Os espaços reservados aos automóveis multiplicam-se, tomando o espaço de transportes coletivos e não motorizados, como calçadas ou bicicletas. Com a facilidade de acesso ao financiamento e a isenção de impostos, mais e mais pessoas ocupam as ruas com seus carros

particulares, dificultando o trânsito e prejudicando a vida da população em geral [4] [5].

Na prática, as políticas governamentais que possuam como objetivo legitimar outras formas de mobilidade, como a bicicleta, como um modal de transporte reconhecido, promovendo o seu uso e conscientizando condutores de veículos motorizados sobre as leis de trânsito para um compartilhamento pacífico e amigável das ruas são ainda muito tímidas, considerando o montante de investimentos e o espaço da cidade destinados ao modal de transporte motorizado e privado.

Na Figura 2 nos é apresentada a diferença de valores investidos em diversas ações de mobilidade [2] por parte do governo do estado da Bahia. Neste gráfico fica clara a prioridade reservada ao modelo viário. Cabe aqui lembrar que a única ação planejada pelo governo do estado para mobilidade não motorizada (uma diretriz da lei de mobilidade, sancionada pelo governo federal em 2012 [34]) não foi efetivada. O projeto Cidade Bicicleta não conseguiu ser implantado pelo governo do estado, mesmo estando em planejamento desde 2010.


Figura 2: Investimento em mobilidade em Salvador

Além disso, deve-se considerar as práticas de legitimação da alta velocidade dos carros nas cidades brasileiras, especialmente em Salvador. Exemplificamos isso com uma listagem das avenidas e ruas de Salvador agrupadas por velocidades, consideradas inadequadas para compartilhamento com pedestres ou usuários de bicicleta pela Organização Mundial de Saúde [3]:

80 Km/h: Av. Caribé, Av. Luís Eduardo Magalhães (trecho 1), Av. Luís Viana;

70 Km/h: Av. Anita Garibaldi, Av. Antônio Carlos Magalhães, Av. Dorival Caymmi, Av. Juracy Magalhães, Av. Magalhães Neto, Av. Mário Leal Ferreira (Bonocô), Av. Orlando Gomes, Av. Otávio Mangabeira, Av. Professor Pinto de Aguiar, Av. Reitor Miguel Calmon (trecho 1), Av. Luiz Eduardo Magalhães (trecho 2);

60 Km/h: Av. Afrânio Peixoto, Av. Amaralina, Av. Conselheiro Pedro Luiz, Av. Centenário, Av. Dendezeiros, Av. Eng. Oscar Pontes, Av. General Graça Lessa, Av. Jequitaia, Av. Jorge Amado, Av. Manoel Dias da Silva, Av. Oceânica, Av. Presidente Costa e Silva, Av. Reitor Miguel Calmon(trecho 2), Av. São Cristóvão, Av. São Rafael, Av. Tancredo Neves, Av. Vale dos Barris, Av. Vasco da Gama, Praça Dr. João Mangabeira, Rua Conselheiro Pedro Luiz.

Este contexto gera um ambiente hostil para os pedestres e usuários de bicicleta, tornando-se desencorajador para aqueles que gostariam de adotar este modal de locomoção nas cidades. Especialmente na cidade de Salvador, faltam iniciativas e dados que melhor detalhem e quantifiquem este contexto adverso. Podemos citar como exemplos de aspectos observados em outras cidades ou por especialistas no assunto: altas velocidades nas ruas e avenidas; falta de sinalização; ruas arborizadas; distância entre veículos motorizados e bicicletas [6] [7] [3] [8].

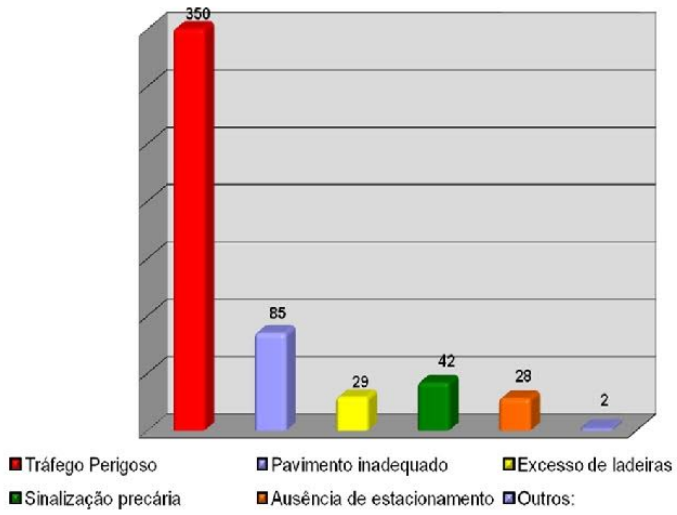


Figura 3: Características do percurso

Ainda assim, existem pesquisas como a retratada na figura 3 que mostra algumas das características de

percurso dos usuários de bicicleta, apontando como fator predominante a periculosidade do tráfego[38].

### *1.2. Objetivos*

Este trabalho possui como objetivo principal a criação de uma arquitetura composta de sistemas embarcados e sensores para a captura de informações reais e detalhadas relativas ao ambiente das ruas a partir do uso de bicicletas, para melhor entender os fatores que fazem as ruas dos grandes centros urbanos tão hostis e desmotivadores deste meio de transporte.

Busca-se, assim, tratar tanto das questões computacionais aplicadas ao monitoramento de fatores ambientais, bem como das questões de tráfego urbano e das condições, favoráveis, ou não, para o uso da bicicleta. Nesta mesma linha de raciocínio, temos como objetivos secundários:

- Referenciar geograficamente os dados coletados, no sentido de indicar os trechos com condições mais favoráveis ou menos atrativos;
- Oferecer subsídios para políticas de adequação do espaço;
- A observação da legitimação das políticas públicas de trânsito por parte de motoristas e ciclistas, podendo avaliar esta relação nas práticas de uso das pistas em séries periódicas.
- Ser uma aplicação livre e replicável;
- Ser flexível para adaptações de contexto e necessidades diferentes.

O trabalho está organizado em seções e será apresentado na seguinte ordem: na seção 2 será mostrado apresentado o referencial teórico, na seção 3 serão mostrados os trabalhos correlatos, na seção 4 será detalhada a metodologia utilizada no desenvolvimento deste trabalho, na seção 5 será discutida as tecnologias adotadas nesta solução, na seção 6 será apresentada a solução proposta por este trabalho, na seção 7 será apresentado os testes e resultados obtidos com a solução, na seção 8 será apresentada a conclusão do trabalho.

## 2. Referencial teórico

Para entender o contexto onde o trabalho está situado é importante conhecer alguns conceitos em que o mesmo se fundamenta. Por isso, esta sessão tem por objetivo pautar os trabalhos e tecnologias que serviram de referencial para o mesmo.

### *2.1. Sistemas embarcados*

Os sistemas embarcados estão presentes no cotidiano das pessoas mesmo que eles não se deem conta disso, muitos dos artefatos eletrônicos atuais, contam com um ou mais sistemas embarcados. Eles, ao contrário dos computadores, são desenvolvidos para fazer apenas uma tarefa, ou um pequeno conjunto delas, mas ainda possuindo os mesmos recursos de um computador, processador, memória, dispositivo de armazenamento, interfaces de comunicação entre outros [31].

De acordo com Peter Marwedel [32] apud Santos [33], existem outras características inerentes aos softwares embarcados:

Os sistemas embarcados não são projetados para serem programados pelo usuário final. Eles normalmente atuam no ambiente, recebendo dados através de sensores, e modificando o ambiente através de atuadores. Este tipo de sistema deve ser confiável, passível de funcionar em ambientes críticos, uma vez que suas falhas podem causar grandes prejuízos. Além disso, sistemas desta natureza possuem formas diferentes de interação com o usuário, ao invés de teclados, mouses, e monitores, sistemas embarcados contam com botões, chaves e *displays*. Muitos dos sistemas embarcados são híbridos, compostos por partes analógicas e digitais. Normalmente estes sistemas são reativos: eles esperam uma entrada do ambiente para realizar um processamento e gerar uma saída.

### *2.2. Hardware de código aberto*

O *Hardware* de código aberto refere-se fundamentalmente as especificações de *design* de dispositivos físicos que possam ser estudados, modificados, criados e distribuídos por qualquer pessoa capacitada para tal. Na prática, ao desenvolver um *Hardware* de código aberto é necessário que ele esteja disponível ao público, seja livre de registros comerciais, e não

se pode restringir o seu uso de acordo com o adotado por seu desenvolvedor inicial [35].

O conceito de *Hardware* de código aberto surgiu apoiado no conceito de *Software* de código aberto, mas assim como este, não se restringiu apenas a parte técnica dos dispositivos físicos. O *Hardware* de código aberto extrapolou as fronteiras da computação e engenharia e hoje temos vários projetos de *Hardware* de código aberto dentro da arquitetura e do urbanismo, como os projetos *Wikihouse* [11] e o Arc e Tec [15], na criação de máquinas,como o *Open Source Ecology* [12], entre outros, que seguem a mesma ideologia de compartilhamento

Estes projetos mostram como o crescimento das ideias do *hardware* livre estão propagando-se para além das máquinas e adentrando outras áreas, como é o caso da arquitetura.

### *2.3. Microcontroladores*

Figura 4: Exemplo de microcontroladores

Os microcontroladores são pequenas placas compostas por um processador, pinos de entrada e saída e memória. Usando algoritmos implementados em linguagens específicas, geralmente de baixo nível, os microcontroladores podem ser programados para desempenharem tarefas acionando ou não suas saídas [30].

A imagem 4 mostra exemplos de microcontroladores, com seus pinos de interação. Para trabalhar com eles é necessário saber a especificação do que faz cada um destes pinos, essa informação é normalmente disponibilizada pelo desenvolvedor do *hardware*. O que diferencia os microcontroladores são os seus recursos de *hardware*, a quantidade de pinos de manipulação e a forma disponibilizada para trabalhar com os mesmos.

Possuem em comum a sua arquitetura física, que tende a encapsular o processador, a memória, os pinos de entrada e saída, além de vários microcircuitos de suporte, como temporizadores, interruptores entre outros, deixando assim o dispositivo muito pequeno e com uma alta complexidade arquitetural.

Em geral os protocolos de comunicação usados nos microcontroladores são proprietários, mas a popularização de protocolos abertos como o *Serial Peripheral Interface* - SPI e o *Inter-Integrated Circuit* - I2C contribuem para a popularização dos mesmos.

#### *2.3.1. Arduíno*

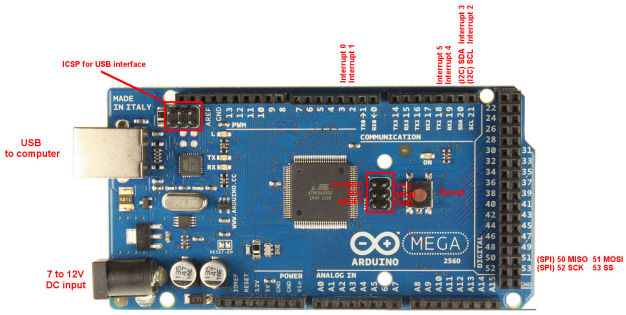


Figura 5: Exemplo do Arduíno Mega e suas conexões

O Arduíno [9] é uma família de ferramentas e placas de circuito integrado utilizadas para prototipagem de circuitos. Ele é utilizado para permitir que computadores possam controlar dispositivos eletrônicos existentes no mundo físico além dos próprios computadores pessoais. É uma plataforma física de arquitetura aberta voltada para computação baseada num microcontrolador, que funciona atrelado a uma interface de comunicação com outros dispositivos através de pinos digitais e analógicos. A imagem 5 exibe um Arduíno modelo mega 2560, indicando suas principais formas de conexão.

O Arduíno conta ainda com um ambiente de desenvolvimento integrado para escrever, testar, com-

pilar e embarcar os programas para o conjunto de componentes e dispositivos [27].

### 2.4. *Raspberry Pi*

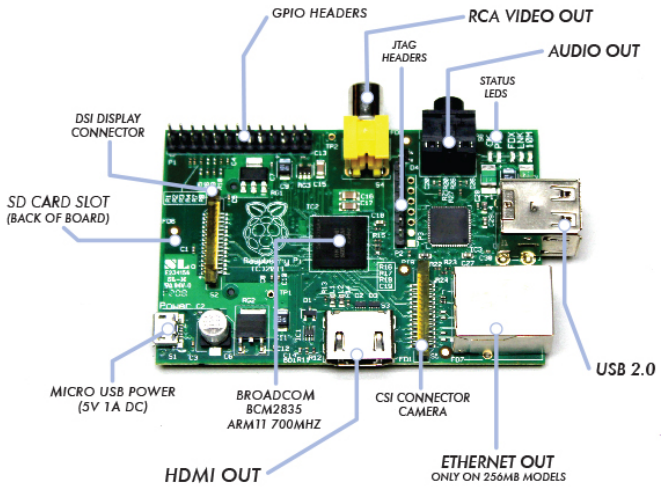


Figura 6: Raspberry Pi

O Raspberry Pi é um micro-computador embarcado numa pequena placa. Ele possui um processador single core com barramento de 700 MHZ, uma memória RAM de 512MB [26] duas portas USB e uma saída de vídeo HDMI. O Raspberry Pi conta ainda com um slot para um cartão SD (Secure Digital). Neste projeto foi usado um cartão com capacidade de 8GB e classe 10 (proporcina maior velocidade de leitura e escrita de dados), onde estarão armazenados o sistema operacional que controla o Raspberry Pi, os *scripts* da aplicação responsáveis pela comunicação, recebimento, tratamento e armazenamento dos dados vindos do Arduíno.

A imagem 6 exibe um Raspberry Pi, indicando suas principais formas de conexão e componentes.

O Raspberry Pi não é um *Hardware* de código aberto, ele é criado com muitos componentes proprietários, como o seu processador ARM, e o próprio *design* da placa, apesar disso, seu seu baixo custo, facilidade de operação e portabilidade pelo seu tamanho diminuto, o faz uma boa escolha para o projeto.

## 3. Trabalhos correlatos

Abaixo são apontados alguns trabalhos que correlacionam-se, em maior ou menor grau, com algumas das funcionalidades deste projeto, ou serviram de inspiração na concepção e no andamento deste trabalho.

### 3.1. *Sensorium*

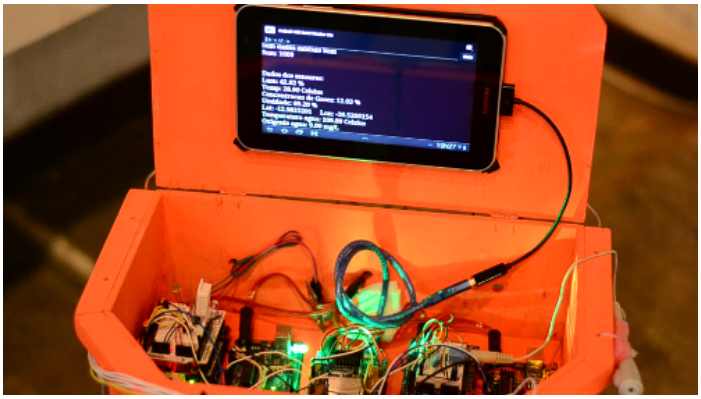

Figura 7: Protótipo do sensorium

Sensorium é um projeto de arte, tecnologia e inovação criado pelo grupo de pesquisa Ecoarte – IHAC/Universidade Federal da Bahia - UFBA que consiste na criação de um dispositivo móvel com o uso de sensores para fazer a interação com o ambiente. O projeto Sensorium foi dividido em 4 fases, sendo elas a criação do dispositivo, uma *performance* com ação da comunidade, a visualização dos dados coletados e a exposição do projeto artístico e seus processos [29].

A imagem 7 exibe o protótipo do Sensorium e parte dos seus componentes.

A criação do dispositivo envolveu prioritariamente soluções livres, tanto de *hardware* quanto de *software*, a facilidade operacional do dispositivo, a geração e a leitura dos dados obtidos por pessoas não ligadas a área técnica/tecnológica. Para tanto, o protótipo foi construído com o microcontrolador Arduíno, utilizando os sensores de temperatura do ar e da água, umidade relativa do ar, salinidade, níveis de intensidades sonoras, níveis de CO2, GPS e Ultra violeta.

### 3.2. *The Copenhagen Wheel*

Este projeto foi criado por um grupo no Instituto de tecnologia de Massachucets (*Massachucets Institute of Thecnologic*) ou MIT no ano de 2005 e tem como objetivo transformar biciletas comuns em bicicletas eletrônicas, ou e-bikes [18]. A imagem 8 exibe


Figura 8: Protótipo do The Copenhagen Wheel

uma bicicleta já com o The Copenhagen Wheel instalada, assim podemos ter noção do que é necessário alterar na bicicleta para a inserção deste produto.

O projeto usa o *smartphone* para controlar o dispositivo que está acoplado à bicicleta e mapeia os níveis de poluição, congestionamento do tráfego e as condicões da estrada em tempo real [28]. Enquanto a bicicleta é usada, os sensores da bicicleta capturaram informações sobre suas preferências pessoais de ciclismo, como quanto esforço é posto nisso, quantas calorias foram queimadas e etc além de captar as informações sobre o ambiente, incluindo quantidade de monóxido de carbono, óxido de nitrogênio (NOx), níveis de ruído, temperatura ambiental e umidade relativa.

### *3.3. Comparativo*

O Sensorium diferencia-se basicamente do Bduíno por sua concepção, apesar de coletar dados e localizá-los espacialmente, premissa do Bduíno, o Sensorium parte da coleta através do modal pedestre, além disso, ele conta com um tablet para fazer parte do seu processamento, o que traz mais recursos ao conjunto.

O The Copenhagen Wheel, apesar de parecido com o Bduíno, diferencia-se deste em diversos fatores, o primeiro deles é que para instalar o The Copenhagen Wheel, é necessário remover a uma das rodas e substituir pelo protótipo, além disso ele conta com o uso de um *smartphone* do usuário para seu funcionamento e por fim ele é um dispositivo proprietário, o que o Bduíno tenta adaptar-se a qualquer bicicleta sem necessidade de troca de peças, além de ser um sistema auto-contido e de código aberto, além disso, o Bduíno observa a distância dos carros, aspecto que o C. Wheel não registra.

**COMPARATIVO**

| | Sensorium | T. C. W. | Bduíno |
|---|---|---|---|
| Modal | A pé | Bicicleta | Bicicleta |
| Armazenamento | Tablet | Celular | Bduíno |
| Alimentação | Indefinido | Baterias | Bateria c/ carregador solar |
| Instalação | Não existe | Trocar rodas | Amarração |
| Distribuição | Open source? | Venda | Open source |

Figura 9: Comparativo entre os dispositivos.

A figura 9 exibe um quadro comparativo simplificados entre os dispositivos dos trabalhos correlatos e o Bduíno. As especificações técnicas e de desempenho desses dispositivos não foram encontradas para uma melhor comparação.

## 4. Metodologia

Primeiramente foi feita uma pesquisa na literatura sobre as melhores práticas de ambiente urbano para o uso da bicicleta na cidade. Em seguida, foram analisados os resultados de uma pesquisa sobre o uso da bicicleta e fatores relativos, realizada no ano de 2012 entre pessoas de diferentes perfis e majoritariamente participantes de grupos de usuários de bicicletas em Salvador [25].

A partir disso, foi formulado um questionário baseando-se nos aspectos observados na literatura e na pesquisa, o qual foi compartilhado com usuários ou potenciais usuários da bicicleta em Salvador. Analisadas essas informações, foram definidos diversos aspectos que poderiam ser aferidos através de um dispositivo eletrônico a ser anexado a uma bicicleta, a fim de medir alguns destes fatores. Com os resultados obtidos, os mesmos deverão ser ponderados pelas respostas obtidas indicando os aspectos que mais obtiveram respostas negativas quanto à influência sobre o ato de pedalar.

Com a revisão de literatura e a partir dos resultados das pesquisas, optou-se por analisar os seguintes fatores ambientais:

- Temperatura do ar.
- Umidade do ar.
- Luminosidade.
- Nível de ruído.

assim como fatores urbanos:

- Nível de trepidação da pista.
- Distância em que outros veículos motorizados passam pelo usuário de bicicleta.
- Velocidade em que outros veículos motorizados passam pelo usuário.

### *4.1. Pesquisa com usuários e possíveis usuários de bicicleta*

A pesquisa foi feita para colher informações que serviram como fonte de informação sobre quade através da internet usando um formulário eletrônico com as perguntas a seguir:

- Você é usuário de bicicleta?
- Caso tenha respondido Não na pergunta anterior, você gostaria de utilizá-la?
- Você acredita que os investimentos públicos para o uso da bicicleta em salvador são suficientes para popularizar o seu uso?
- A temperatura das vias é um fator que influencia, ou pode influenciar no seu uso da bicicleta?
- A umidade do ar nas vias é um fator que influencia, ou pode influenciar no seu uso da bicicleta?
- O ruído nas vias é um fator que influencia, ou pode influenciar no seu uso da bicicleta?
- A iluminação das vias é um fator que influencia, ou pode influenciar no seu uso da bicicleta?
- O estado físico das vias é um fator que influencia, ou pode influenciar no seu uso da bicicleta?
- A distância que os automóveis ultrapassam as bicicletas nas vias é um fator que influencia, ou pode influenciar no seu uso da bicicleta?

Foram recebidos 183 respostas válidas. A partir dos resultados desta pesquisa, foi possível criar gráficos que ajudassem a entender quais fatores poderiam inferir a popularização da bicicleta. Os gráficos gerados são apresentados abaixo.

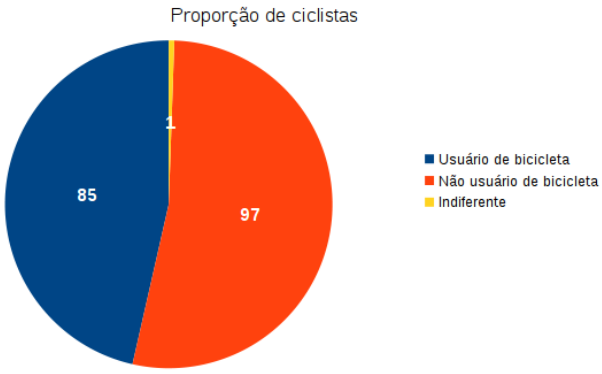


Figura 10:

A Figura 10 exibe a proporção de entrevistados que fazem, ou não, uso da bicicleta com meio de transporte.

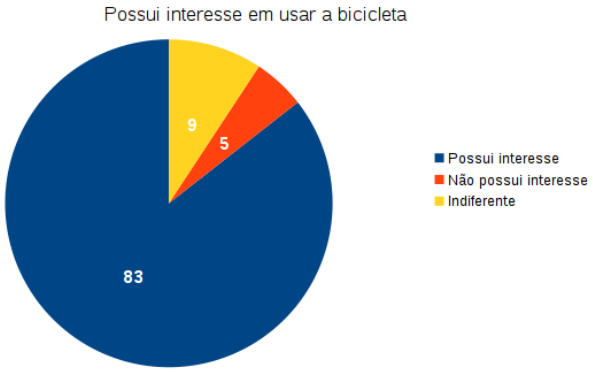


Figura 11:

A Figura 11 exibe a proporção de entrevistados que não usam a bicicleta, mas gostariam ou não de usá-la.

O gráfico 12 exibe a proporção de entrevistados que acham que os investimentos públicos para o uso da bicicleta são suficientes, ou não.

O gráfico 13 exibe a proporção de entrevistados que acreditam que a temperatura é um fator que pode influenciar, ou não, o uso da bicicleta.


Figura 12:

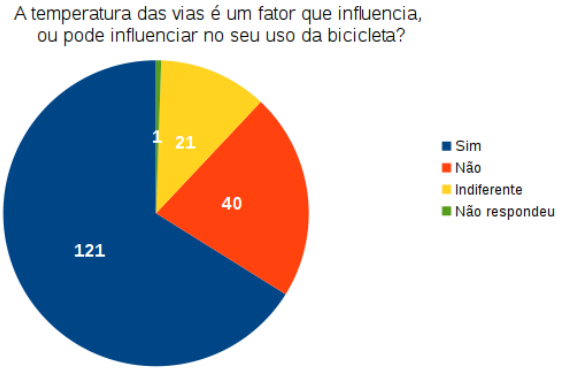


Figura 13:

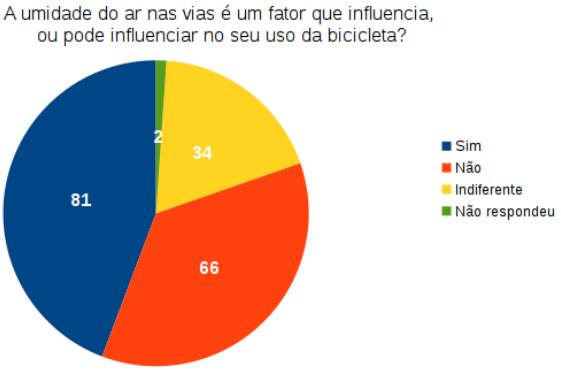


Figura 14:

O gráfico 14 exibe a proporção de entrevistados que acreditam que a umidade do ar é um fator que pode influenciar, ou não, o uso da bicicleta.

O gráfico 15 exibe a proporção de entrevistados que acreditam que o ruído nas vias é um fator que pode influenciar, ou não, o uso da bicicleta.

O gráfico 16 exibe a proporção de entrevistados que acreditam que a luminosidade é um fator que

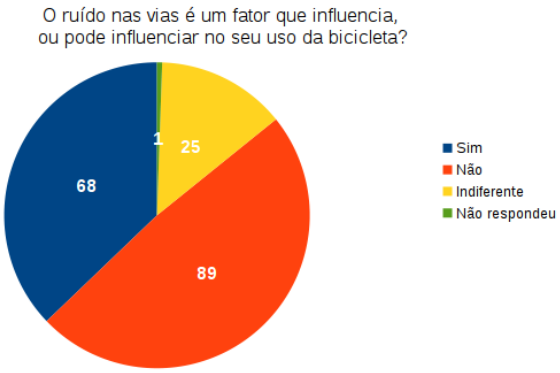


Figura 15:


Figura 16:

pode influenciar, ou não, o uso da bicicleta.

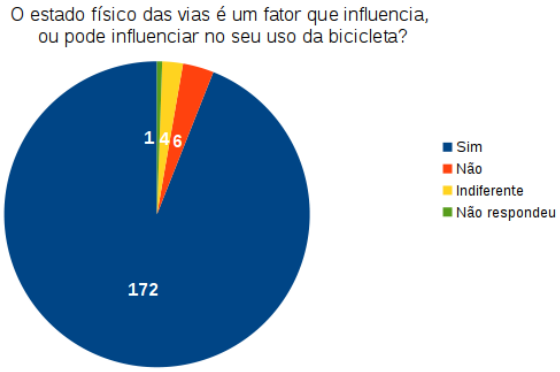


Figura 17:

O gráfico 17 exibe a proporção de entrevistados que acreditam que o estado das pistas é um fator que pode influenciar, ou não, o uso da bicicleta.

O gráfico 18 exibe a proporção de entrevistados que acreditam que a distância de ultrapassagem de um veículo motorizado é um fator que pode influenciar, ou não, o uso da bicicleta.


Figura 18:

## 4.2. *Soluções adotadas*

A criação do dispositivo, utilizando-se de incrementos funcionais aplicados ao dispositivo que visam a criação de aplicações de testes, a integração dessas aplicações e as correções dos erros encontrados durante o processo de desenvolvimento do sistema.

### 4.2.1. *Arquitetura da aplicação*

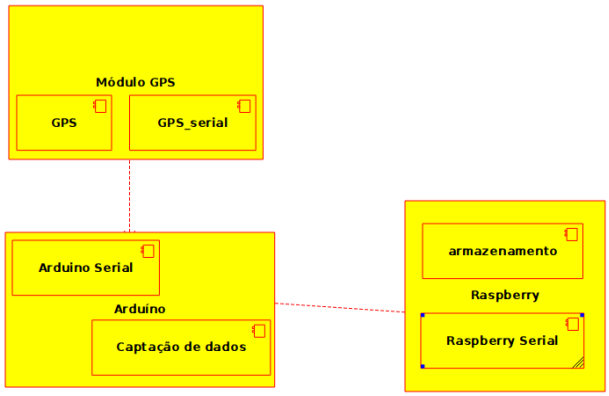


Figura 19: Diagrama de componentes

O diagrama na Figura 19 representa os componentes mais importantes da aplicação, todos aqueles que possuem algum processamento mais elaborado ou necessitam de uma atenção especial na montagem. No módulo GPS, ele possui embarcado um código responsável por receber os dados do satélite, e um módulo responsável por passar essas informações através de uma conexão serial, o Arduíno possui um módulo responsável por captar os dados dos diversos sensores, e um módulo de conexão serial para receber dados do módulo GPS e enviar dados para o Raspberry Pi. Por fim, o Raspberry Pi possui um módulo para receber os dados do Arduíno, e um módulo de armazenamento de dados.

A arquitetura da aplicação é naturalmente modular, pela natureza da solução construída, por isso foi utilizado o estilo arquitetural em camadas. Cada camada só envia informações para outra camada através de conexão serial, e não retorna nenhum tipo de dado para a camada anterior.O conector usado entre as camadas é o fluxo ou *stream*.

A aplicação também se utilizou de outro padrão arquitetural, o *Sense-Compute-Control*, que é o padrão quando trabalha-se com sistemas embarcados de controle. Neste projeto, o Arduíno funciona como módulo central, que recebe os dados dos sensores, depois atua sobre o Raspberry Pi para guardar os dados obtidos.

Como os dispositivos possuem em geral pouca memória e processamento, por questões de desempenho e melhor aproveitamento de recursos, a implementação dos códigos da aplicação não está dividida em módulos. Por se tratar de dispositivos embarcados com limitações de *hardware*, principalmente em memória e processamento, quanto menos recursos computacionais forem usados na implementação, obteremos uma aplicação final mais enxuta.

### 4.2.2. *Funcionamento do conjunto*

O funcionamento do Bduíno é composta de três dispositivos que executam em três ambientes heterogêneos, a saber, o GPS, o Arduíno, e o Rapberry Pi. Cada aplicação desse sistema, funciona de forma independente da outra devendo apenas executar uma pequena fração da aplicação total.

A primeira aplicação é nativa, e embarcada no módulo GPS, ela é fechada e não é possível manipular a sua programação, para o seu funcionamento, é necessário apenas estar conectado ao Arduíno, ter suas conFigurações físicas feitas, e com isso ele irá funcionar e enviar os dados através das portas serias conFiguradas.

O segundo módulo, foi desenvolvido e executará na plataforma Raspberry Pi. Este módulo é encarregado de manter uma comunicação serial e receber os dados que por ela chegarem, armazenando-os em um arquivo texto simples. A evolução dessa aplicação

se deu com a mesma inicializando juntamente com o Sistema operacional do Raspberry Pi, a fim de começar a coletar os dados automaticamente, além de gravar os dados recebidos de uma forma estruturada em um banco de dados relacional.

O terceiro módulo foi desenvolvido, e é uma aplicação embarcada que executará na plataforma Arduíno. Este módulo é responsável pela execução de algumas tarefas: iniciar uma comunicação serial com o GPS, iniciar uma comunicação serial com o Raspberry Pi, coletar os dados dos sensores e atuadores nele conectados e enviar os dados coletados através da conexão serial criada.

### *4.3. Sensores*

Sensores são dispositivos físicos que captam uma determinada variação de uma grandeza que ele se propõe a mensurar, de forma analógica ou digital.

#### *4.3.1. Sensor de temperatura e umidade*

Para a medição da temperatura e da umidade do ar, foi utilizado um módulo chamado DHT11, que faz a medição de temperatura e umidade do ar de forma digital.

O DHT11 é um sensor de temperatura e umidade que permite fazer leituras de temperaturas com variações entre 0 a 50 graus Celsius e umidade entre 20 a 90%. Este sensor é muito usado para projetos com Arduíno.

O módulo sensor conta internamente com um termistor do tipo NTC e o sensor de umidade é do tipo HR202. O módulo internamente faz a leitura dos sensores e envia os dados através de um pino serial de via única.

#### *4.3.2. Sensor de iluminação*

Para fazer a medição da iluminação foi utilizado um resistor dependente de luz (*Light Dependent Resistor*) ou LDR. O LDR é um resistor analógico cuja resistência aplicada ao circuito varia de acordo com o nível de iluminação captado, capturando assim a variação da resistência, é possível mensurar a intensidade da luz recebida.

#### *4.3.3. Sensor de som*

O nível de ruído será capturado através de um módulo de som, que consiste em um pequeno microfone com um potênciometro para regular a sensibilidade do mesmo.

O objetivo do sensor de som KY-038 é medir a intensidade sonora do ambiente ao seu redor, variando o estado de sua saída digital caso detectado um sinal sonoro. O limite de detecção pode ser ajustado através do ajuste do potenciômetro presente no sensor, que regulará a saída digital D0. Contudo, para ter uma resolução melhor, é possível utilizar a saída analógica A0 e conectar a um conversor analógico/digital, como a presente no Arduíno por exemplo.

#### *4.3.4. Acelerômetro*

O nível de trepidação será capturado através de um módulo digital de um acelerômetro de 3 eixos modelo MMA7361, fixando o eito y como nível vertical, ele capturará as variações neste eixo enquanto o dispositivo está ligado, podendo assim perceber trechos de maior variação indicando possivelmente um terreno mais acidentado e por conseguinte menos propício ao uso da bicicleta.

O MMA7361 é um módulo que possui um micro capacitor para medição do sinal. O módulo conta também com um filtro para atenuação de ruídos externos e um circuito próprio para compensação da temperatura, estes conjuntos aliados formam um acelerômetro de alta sensibilidade.

Um facilidade encontrada neste modelo, é que ele já possui os pinos soldados ao módulo, tornado a montagem do mesmo mais fácil e rápida.

#### *4.3.5. Sonar ultra-sônico*

A distância entre um veículo motorizado e a bicicleta será medida por um sensor ultra-sônico modelo HC-SR04. Este sensor emite ondas ultra-sônicas e as recebe de volta depois que as ondas são rebatidas por algum objeto, ele então calcula a distância do objeto com base na diferença do tempo entre emissão e recebimento.

O sensor ultrassônico HC-SR04 é capaz de medir distâncias de 2cm a 4m com ótima precisão. Este módulo possui um circuito pronto com emissor e receptor acoplados e 4 pinos (VCC, Trigger, ECHO, GND) para medição.

A imagem 20 exibe um módulo ultrassônico e nele podemos ver os seus 4 pinos.

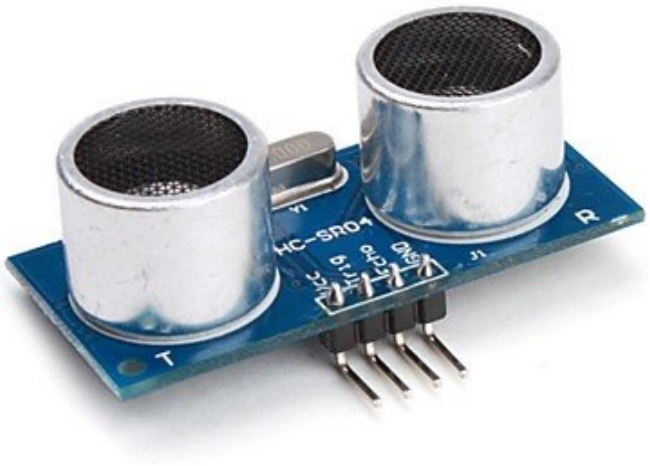


Figura 20: Sonar ultra-sônico

Para medir a distância entre o sensor e um objeto, é preciso alimentar o pino Trigger do sensor com uma tensão alta, durante no mínimo 10 microssegundos, então o sensor irá disparar uma onda ultrassônica que quando encontrar um objeto, será rebatida novamente ao sensor e será captada pelo pino ECHO do sensor.

A distância entre o sensor e o objeto, pode ser calculada de acordo com o tempo em que o pino ECHO demorou para receber o retorno da onda disparada pelo pino Trigger.

A fórmula da distância pode ser descrita como, Distância = [Tempo em que o ECHO demorou para receber o retorno * Velocidade do Som] / 2.

A velocidade do som poder ser considerada idealmente igual a 340 m/s, assim o resultado pode ser obtido em metros, bastando considerar o tempo em segundos. Na fórmula a divisão por 2 deve-se ao fato que a onda é enviada e rebatida, logo ela percorre 2 vezes a distância procurada, uma vez para ir até o objeto, e outra para voltar ao sensor.

### *4.4. Geolocalizador*

A geolocalização será feita através de um *shield* GPS, *global positioning system*, fabricado pela empresa ITEAD Studio. Um *shield* é um módulo criado especialmente para o Arduíno, normalmente com o mesmo tamanho, que possui os pinos alinhados com o mesmo, facilitando a sua montagem, precisando apenas encaixar o *shield* no Arduíno, diferente de um módulo que deve ser montado num circuito a parte.

Este *shield* para Arduíno é um módulo GPS projetado para o receptor do Sistema de Posicionamento Global com a interface SD. Apesar de poder ser usado para armazenamento dos dados, o módulo de micro SD não será utilizado nesse projeto. Este *shield* pode operar com nível de tensão de operação compatível com 5V e 3.3V a fim de torná-lo compatível com placas Arduíno e outras placas compatíveis Arduíno, neste projeto foi usado o a tensão de 5V padrão do Arduíno.

Ele é adequado para as seguintes aplicações com Arduíno:

- Navegação Automotiva
- Posicionamento pessoal
- Gerenciamento da frota
- Navegação marítima

Este *shield* GPS possui seu próprio microcontrolador e seu *software* embarcado é responsável por encontrar os satélites, fazer a conexão com os mesmos e repassar as informações recebidas através de uma conexão serial, por tanto se faz necessário criar uma conexão serial entre o *shield* GPS e o Arduíno.

#### *4.4.1. NMEA 0183*

O *National Marine Electronics Association* (NMEA) desenvolveu uma especificação de um conjunto de protocolos que definem a interface de comunicação entre diversos equipamentos eletrônicos marinhos. Este conjunto, permite a troca de informações entre computadores e equipamentos eletrônicos [14]. As informações recebidas e repassadas pelo módulo GPS estão neste formato, foi utilizado a biblioteca TinyGPS para extrair as informações deste módulo. Mais informações encontram-se no anexo deste trabalho.

### *4.5. Uso de cabo de dados*

A comunicação entre o Arduíno e o Raspberry Pi é um ponto chave do funcionamento do sistema sendo assim, um ponto de atenção e possivelmente um ponto limítrofe do mesmo.

A comunicação entre o Raspberry Pi e o Arduíno feita através de meio físico pode ser feita através de cabos por onde passam as informações entre as duas placas, nesse quesito encontram-se a comunicação serial entre o GPIO do Raspberry Pi e os pinos seriais do Arduíno, a comunicação serial entre os pinos I2C do Raspberry Pi e os pinos seriais do Arduíno, e por fim o uso do cabo USB entre as saídas USB dos dois dispositivos.

Para o projeto Bduíno, foi escolhido o método mais simples de comunicação, a comunicação através da USB usando uma biblioteca de abstração de comunicação serial da própria linguagem, a PySerial do Python2. Além do trafego de dados, a conexão com cabo USB, encarrega-se também da alimentação elétrica do Arduíno através da porta USB do Raspberry Pi.

## 5. Criação de dispositivo

O desenvolvimento do dispositivo se deu através da plataforma, Arduíno, juntando se a isso os sensores necessários para a checagem das informações ambientais e urbanas. A partir desse conjunto, os dados serão capturados, sem nenhum tipo de tratamento, e serão enviados, através de uma conexão serial[19], para outro dispositivo, com mais capacidade de processamento, mas ainda assim portável, o Raspberry Pi[20], onde os dados serão então tratados e armazenados para posterior utilização. Os dados capturados serão armazenados em um contexto espacial (geolocalizados) e temporal.

O Raspberry funcionará neste projeto como dispositivo principal, cabendo a ele o papel de receber os dados através de uma conexão serial, tratar os dados e persisti-los. Caberá também ao Raspberry Pi a tarefa posterior ao uso da bicicleta: a transmissão de dados capturados.

No Raspberry Pi a linguagem de programação escolhida foi o Python, por ser uma linguagem de script que possui uma gama de bibliotecas para diversas funções já disponíveis. O Python também possui ótima compatibilidade com o dispositivo, já vindo pré instalada na distribuição GNU/Linux adotada, o Raspbian. Foi utilizado também um pequeno script em linguagem shell para inicializar os scripts em Python durante a inicialização do sistema.

### *5.1. Microcontrolador*

O desenvolvimento do dispositivo será feito com o Arduíno Mega 2560 por este modelo possuir melhor desempenho e recursos em comparação aos outros modelos da plataforma. Ele possui uma memória maior, possibilitando, assim, maior flexibilidade na captura de maiores volumes de dados sem comprometer o desempenho do dispositivo, além de contar com um número maior de portas digitais e analógicas possibilitando, que o dispositivo seja flexível para receber novos sensores e capturar dados diferentes de acordo com a necessidade local.

O Arduíno Mega conta ainda com 3 portas seriais adicionais, o que é altamente desejável neste projeto visto que o *shield* GPS conta com um microcontrolador integrado que envia suas informações através de uma porta serial.

### *5.2. Alimentação elétrica*

A alimentação deste conjunto formado pelo Arduíno com seus sensores e o Raspberry Pi, é feita através de uma bateria com capacidade nominal de 30000 MAH, e capacidade real de 16000 MAH.

Esta bateria pode ser alimentada por uma fotocélula ou por energia elétrica convencional. Neste projeto utilizamos a recarga solar enquanto o ciclista se locomove em ambientes iluminados, com isso o projeto ganha maior autonomia, e ainda conseguimos usar uma tecnologia verde[21] não poluente, e que se utiliza de uma fonte renovável.

### *5.3. Montagem do protótipo*

A montagem do dispositivo segue é feita conectando os dispositivos usando cabos e fios para fazer ligações entre os componentes elétricos. Por contar com vários dispositivos, é necessário atenção para não trocar ou inverter os pinos durante a montagem.

Primeiramente é necessário conFigurar o *shield* GPS, nele foi necessário mudar a chave de tensão para a 5V, e inserir os *jumpers* das portas RX e TX nos valores 3 e 4 respectivamente, depois disso foi

necessário inserir a antena no *shield* e inseri-lo no Arduíno. Os pinos de comunicação serial (RX, TX) foram ligados aos pinos digitais D0-D7 do Arduíno. Neste projeto, no entanto, eles estão ligados aos pinos 3 e 4 do Arduíno além disso, os pinos 3 e 4 foram ligados aos pinos 18 e 19 do próprio Arduíno, fazendo uma espécie de ponte entre eles, pois nos pinos 18 e 19 encontram-se a porta serial1.

Depois foi feita a montagem dos módulos numa *protoboard* os pinos de alimentação e terra devem ser ligados de acordo com a documentação de cada módulo. Os pinos de dados dos módulos devem ser ligados na seguinte ordem, o DHT11 deve ser ligado na porta digital 2, os pinos Echo e Trigger do HC-SR04 devem ser ligados na nas portas digitais 12 e 13 respectivamente, os pinos RX e TX do GPS devem fazer uma ponte até as portas digitais 18 e 19, o componente LDR precisa de um resistor de 10k ohm, e deve ser ligado na porta analógica 0, o módulo de som deve ser ligado na porta analógica 1, e por fim, o módulo acelerômetro deve ter os pinos X, Y e Z ligados nas portas analógicas 2, 3 e 4 respectivamente.

Com o Arduíno montado, basta agora conectá-lo ao Raspberry Pi com um cabo de dados serial em uma entrada USB do Raspberry. Por fim, basta conetar o Raspberry Pi na fonte de alimentação com o auxílio de um cabo micro USB (no Raspberry Pi), USB (na bateria solar). Com esta montagem o sistema já estará energizado, e em funcionamento.

## 6. Artefatos de software

Como parte da solução desenvolvida, foi necessário também o desenvolvimento de alguns artefatos de *software* que serão apresentados nas sub-sessões abaixo.

### *6.1. Implementação no Arduíno*

A linguagem de programação adotada, oficialmente, na plataforma Arduíno, é a *Arduino programming language*[22] que nada mais é do que um conjunto mais restrito da linguagem C++ com algumas bibliotecas específicas. O projeto Arduíno mantém também uma *Integrated Development Environment* ou IDE, onde é possível seguir todo o fluxo de desenvolvimento para a plataforma, desde a criação de bibliotecas, codificação da aplicação, compilação da aplicação e o processo de embarcar o binário para a placa além de possuir versão para os sistemas operacionais Windows, GNU/Linux e Mac OS X. A escolha da linguagem se deu pela *Arduino programming language*, por ela ser uma linguagem de nível médio, possuir boa documentação, uma comunidade ativa de desenvolvedores que mantém um fórum para troca de informações e por já possuir domínio com a mesma.

A aplicação embarcada possui 3 funcionalidades distintas, a leitura dos dados dos sensores, a formação de um arquivo JSON (*Javascript Object Notation* melhor explicado na seção 6.4. Armazenamento de dados) com os dados obtidos, e o envio desse JSON através da comunicação serial. Essa sequência de instruções é repetida indefinidamente desde o momento em que o dispositivo é ligado, até o momento em que o dispositivo é desligado.

### *6.2. Implementações no Raspberry Pi*

A implementação feita no Raspberry Pi foi escrita em Python versão 2.7. Para ter acesso ao sistema operacional do Raspberry Pi temos algumas opções, podemos ligá-lo diretamente em um monitor, acessá-lo via rede, caso esteja conectado e possua um ip válido, e por fim usar a saída serial padrão com um adaptador USB/serial para acessá-lo pelo console do GNU/Linux.

Foi escolhida a terceira opção por praticidade em demonstrar e trabalhar com a aplicação em qualquer computador com uma entrada USB e uma aplicação para ler e escrever através da conexão serial.

Foi criado um script em Python que possui duas funções principais, a primeira interage com o banco de dados, inserindo em uma tabela um valor incremental que funciona como o identificador das medições, visto que o sistema gerenciador de banco de dados usado não suporta tal funcionalidade. Em seguida esse script abre a conexão serial e recebe os dados, faz alguns ajustes necessários nos dados e os salva no banco de dados. Nesta aplicação, se fez necessário a manipulação da inicialização do GNU/Linux a fim de inserir esse script junto com a inicialização, para que ele não necessite que algum usuário faça login no sistema operacional e o execute.

### 6.3. *Licenciamento de códigos*

Os códigos desenvolvidos para este projeto estão disponibilizados numa plataforma online de versionamento colaborativa chamada github através do link `https://github.com/Ladeia/bduino`.
[1]

### 6.4. *Armazenamento de dados*

Para o armazenamento de dados foi feito um estudo e um teste de desempenho entre duas tecnologias amplamente utilizadas para armazenamento de dados, a *eXtensible Markup Language* ou XML e o *JavaScript Object Notation* ou JSON. As duas tecnologias armazenam os dados em um arquivo de texto plano, mas possuem diferenças que trazem características únicas a cada um deles.

Os testes incluíam testes de leitura, gravação e tráfego de dados entre o Arduíno e o Raspberry Pi. Tendo o JSON uma vantagem em relação ao XML no quesito desempenho sendo, então escolhido o JSON como formato padrão.

Depois dos dados serem recebidos no Raspberry Pi, eles são gravados em um sistema gerenciador de banco de dados chamado SQLite. O SQLite foi escolhido por ser amplamente utilizado em sistemas embarcados, por seu tamanho e facilidade de uso. O SQLite possui algumas limitações inerentes, como a inexistência de relacionamentos entre as tabelas, por isso uma simulação de *software* se fez necessária.

O modelo de dados é bem enxuto e grava apenas um numero incremental que armazena o identificador da atividade, a data e hora de cada checagem, e os dados armazenos em formato JSON.

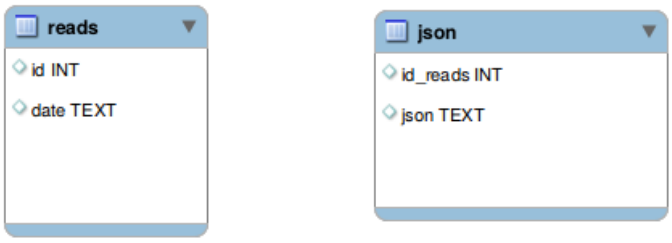


Figura 21: Diagrama de dados

[1]Todos os códigos desenvolvidos estão sobre a licensa *Generel Public License* versão 3.

A Figura 21 representa o modelo de como estão organizados os dados na tabela, na tabela reads, são gravados um identificador que representa uma coleta de dados, juntamente com a hora do seu início, a tabela JSON, possui um campo chamado reads, onde ele referencia o identificador da tabela reads juntamente com os dados de uma medição completa de todos os sensores.vNão existe relacionamento entre as tabelas pois este recurso é inexistente no sistema gerenciador de banco de dados, mas um tipo de relacionamento lógico foi criado através da aplicação. A Figura 22 representa modelo do JSON gerado pelo Arduíno, recebido pelo Raspberry e gravado no banco de dados:

```
{
"light":"light_value",
"temperature":"temperature_value",
"humidity":"humidity_value",
"distance":"distance_value",
"longitute":"longitute_value",
"latitude":"latitude_value",
"sound":"sound_value",
"shake":"shake_value"
}
```

Figura 22: Estrutura de dados gerado pelo Arduíno

## 7. Testes e validação

Os testes foram feitos com o protótipo em ambiente urbano, em sessões de aproximadamente 30 minutos, por localidades diversas na cidade de Salvador.

Este protótipo foi montado numa bolsa de tecido, a qual dobra-se ao meio no quadro ou na garupa da bicicleta, e conta com dois cordões para dar sustentação e equilíbrio.

A imagem 23 exibe o protótipo do Bduíno com seus componentes expostos em cima da armação de tecido criada para sustentá-lo.

A imagem 24 exibe o protótipo montado numa bicicleta.

O teste contou com dois ciclistas voluntários para a coleta dos dados. Os dados foram coletados em momentos variados e em localidades distintas, a fim de tentar capturar contextos urbanos diferenciados.

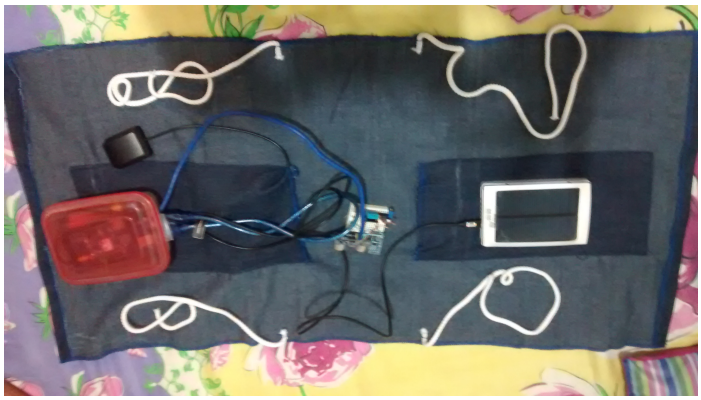

Figura 23: Protótipo do dispositivo


Figura 24: Protótipo do dispositivo

Após os dados coletados, foi possível a manipulação e leitura dos mesmos, a fim de uma análise inicial. Esta primeira análise, apesar de possuir os dados de todos os sensores, levou em consideração apenas dois fatores mensurados, a distância que o veículo passa por uma bicicleta e a iluminação da via, para a criação destes gráficos. Cabe citar que, de acordo com o artigo 201 do Código de Trânsito Brasileiro - CTB [36], a distância mínima segura ao ciclista é de 1,5 metros ao fazer uma ultrapassagem.

É importante frisar que os dados coletados representam uma amostra, visto que por algumas vezes o sensor pode captar dados que não sejam de objetos em movimento ultrapassando-os. No entanto, este universo é muito pequeno, visto que os ciclistas seguiram a norma de trafegar apenas pela via da direita, a de menor velocidade, com o sensor voltado para o lado esquerdo.

Foram criados dois gráficos, ilustrando o resultados das medições efetuadas. O gráfico da Figura 25 mostra a relação entre o número de veículos aproximados que ultrapassam o ciclista, pela distância que eles o fazem em uma determinada medição (900 pontos analisados):

Uma medição é feita a partir da leitura dos dados de um determinado ponto em sua trajetória. Cada medição relativa a distância pode variar entre os objetos que passam pela bicicleta, como um carro ou um ônibus.

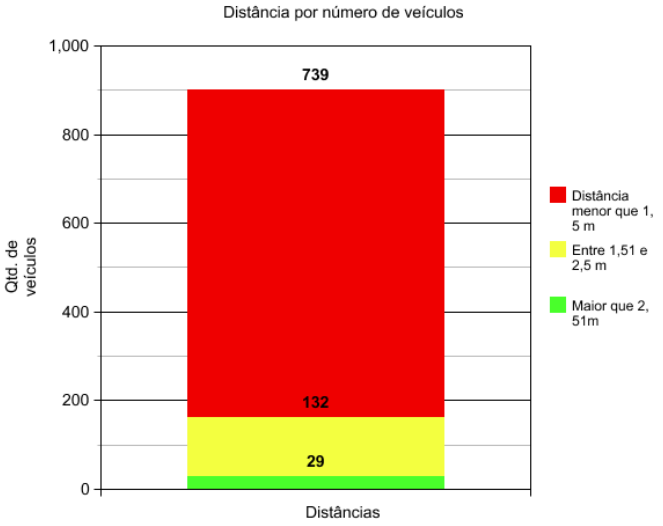


Figura 25: Gráfico de distância por quantidade de veículos em uma medição

O gráfico da Figura 26 exibe a mesma relação do gráfico anterior, mas totalizando as 06 medições efetuadas (5400 pontos analisados).

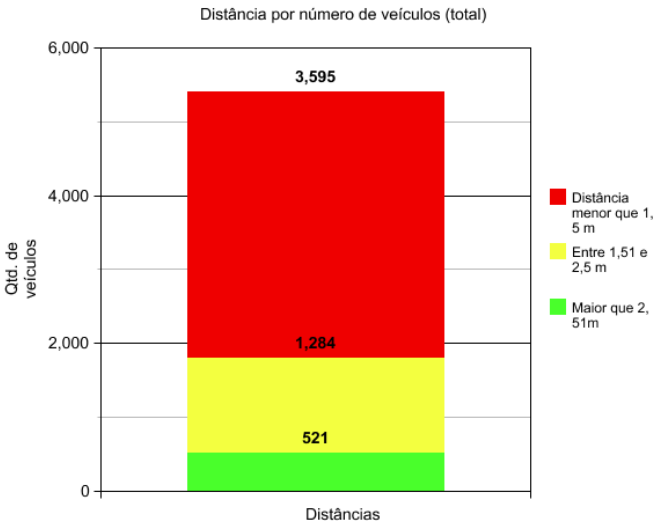


Figura 26: Gráfico de distância por quantidade de veículos totais

Como fica evidenciado pelos gráficos, a maioria das medições de distância apontaram para objetos ultrapassando a bicicleta a menos de 1,5m de distância, logo, desrespeitando o artigo 201 do CTB. Isso reforça um dos resultados da pesquisa de nosso questionário, mostrando que a questão do compartilhamento das pistas com automotores é uma dos aspectos que mais influencia negativamente no uso da bicicleta. Fica reforçada aqui a realidade de falta de incentivos e ações concretas e substanciais que realmente tornem as ruas compatíveis com as leis de trânsito.

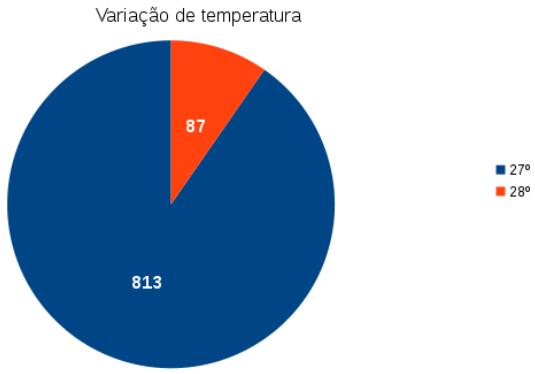


Figura 27: gráfico de variação de temperatura

O gráfico da figura 27 é referente a temperatura em uma medição, e levou em consideração os dados recebidos pelo dispositivo. Os dados recebidos pelo Arduíno e repassados ao Raspberry Pi variam numa escala de 0ºC a 50ºC aproximadamente com precisão de 2ºC para mais ou para menos. Além disso, a temperatura durante o período foi também consultada através do site do clima tempo [23], para conferir a sua corretude.

É necessário lembrar que os resultados não estão separados por horários, e isso pode impactar no resultado obtido. Ao final, apesar da maioria dos entrevistados na pesquisa acreditar que a temperatura nas vias é um fator impactante, a amostra de dados coletada, mostra que a temperatura das vias está com pouca variação durante a maior parte do tempo.

## 8. Conclusão

Ao final do projeto, o dispositivo criado suportou as operações pensadas em sua concepção. O Bduíno conseguiu medir os valores de sensores diversos, armazenando num sistema gerenciador de banco de dados com exceção da captação das velocidades dos veículos motorizados ao ultrapassar a bicicleta, os estudos feitos, mostraram que o *hardware* não oferecia uma boa solução para este problema. Assim, o Bduíno foi capaz de capturar dados ambientais e do contexto urbano para responder questões sobre mobilidade urbana, inspirado numa demanda global, adaptada para um contexto local, facilmente replicável e modificável, pois foram utilizadas tecnologias de código aberto e de arquitetura aberta que podem ser distribuídas e adquiridas a baixo custo.

De fato, a coleta de dados como distância entre bicicleta e veículos motorizados carecia de algum tipo de pesquisa / estudo para medição in loco que conseguisse mostrar que a maioria dos veículos automotores desrespeitam as leis de trânsito no Brasil (ao menos em Salvador).

Dentro do que foi pesquisado, revisado na literatura e nos projetos analisados e coletados na pesquisa desenvolvida, alguns dos aspectos eram contemplados, mas utilizando uma solução de arquitetura fechada. O estágio avançado de evolução do respeito às leis de trânsito e, por conseguinte, ao ciclista, assim como as políticas públicas para a promoção deste modal em outros países levaram a criação de ambientes urbanos menos caóticos, onde a necessidade de medições de aspectos fundamentais, como distância não fazia mais sentido.

No entanto, considerando o contexto urbano brasileiro, esta demanda é clara e evidente pelos dados de violência no trânsito já apresentados.

O Bduíno conseguiu suprir esta e outras demandas, uma vez que este projeto é pioneiro no Brasil, a partir da aquisição baseada em sensores com testes práticos realizados em ruas e avenidas e confirmar através de instrumentos e dispositivos o que muitos dos usuários de bicicleta relatam sobre seu cotidiano.

### *8.1. Trabalhos futuros*

Sendo esta solução de código aberto e flexível, é possível que o trabalho seja expandido no futuro com novas melhorias e funcionalidades, tais como:

- Inclusão de um novo módulo arquitetural responsável pelo envio de dados no momento da coleta para um servidor.

- Inclusão de novos módulos arquiteturais de sensoreamento como coleta de gás carbônico, velocidade de deslocamento, entre outros.

- Inclusão de um novo módulo arquitetural responsável pela medição de velocidade de veículos motorizados.

- Inclusão de um novo módulo arquitetural responsável por melhorar a alimentação elétrica do dispositivo com um dínamo para coletar energia mesmo pedalando em ambientes pouco iluminados.

- Inclusão de um novo módulo arquitetural responsável por uma plataforma para visualização dos dados.

- O Uso de *threads* para tratar operações críticas como acesso a banco e leitura de dados na comunicação serial.

Algumas das modificações para o futuro podem ser facilmente implementadas, outras demandam maior esforço e podem representar outros trabalhos.

## 9. Referências


[1] Associação Nacional de Transporte Público - ANTP, Sistema de Informações da Mobilidade Urbana - Relatório Geral 2008, disponível em http://antp.org.br/_5dotSystem/download/dcmDocument/2013/04/11/4959DF25-8BE7-4DD1-8D31-64D95B5B2385.pdf. Último acesso em 20/05/2014.

[2] Estudo do Coletivo Mobicidade Salvador de 2013 sobre os investimentos do governo do estado, disponível em http://pt.slideshare.net/mobicidade/apresentao-coletivo-mobicidade-dilogos-abertos-adm-ufba-30538158. Último acesso em 20/05/2014.

[3] Gestão da velocidade: um manual de segurança viária para gestores e profissionais da área. Brasília, D.F.: OPAS, 2012. ISBN 978-92-75-31709-9, Organização Pan-Americana da Saúde.

[4] R. Mortarie, G. Luiz Euzébio, O custo do caos - Prejuízo ao bolso e ao meio ambiente - cidades não suportam mais o crescimento da frota de veículos, disponível em http://www.ipea.gov.br/desafios/index.php?option=com_content&id=1252:reportagens-materias&Itemid=39. Ultimo acesso em 16/05/2015. Instituto de pesquisa econômica aplicada - IPEA, 2009.

[5] S. Magalhães Lacerda, Precificação de congestionamento e transporte coletivo urbano, Banco nacional do desenvolvimento - BNDES, disponível em http://www.bndes.gov.br/SiteBNDES/bndes/bndes_pt/Institucional/Publicacoes/Consulta_Expressa/Setor/Logistica_de_Transporte/200603_7.html. Ultimo acesso em 24/05/2015, 2006.

[6] Literature Review on Vehicle Travel Speeds and Pedestrian Injuries, disponível em http://www.nhtsa.gov/people/injury/research/pub/HS809012.html. Ultimo acesso em 16/05/2015, U. S. Department of Transportation, National Highway Traffic Safety Administration, 1999.

[7] Manual de Medidas Moderadoras de Tráfego pela Empresa de Transportes e Trânsito de Belo Horizonte, BHTRANS.

[8] PROGRAMA BRASILEIRO DE MOBILIDADE POR BICICLETA – BICICLETA BRASIL, Caderno de referência para elaboração de Plano de Mobilidade por Bicicleta nas Cidades, Brasília: Secretaria Nacional de Transporte e da Mobilidade Urbana, ISBN: 978-85-60133-47-5, 2007.

[9] Arduíno Introduction, disponível em http://www.arduino.cc/en/Guide/Introduction. Ultimo acesso em 26/05/2014.

[10] O que é um microcontrolador, disponível em http://tecnologia.hsw.uol.com.br/microcontroladores1.htm. Último acesso em 10/01/2015.

[11] Wikihouse, disponivel em http://www.wikihouse.cc/. Ultimo acesso em 26/05/2014.

[12] Open Source Ecology, disponível em http://opensourceecology.org/. Último acesso em 26/05/2014.

[13] Machines: Global Village Construction Set, disponivel em http://opensourceecology.org/gvcs/. Último acesso em 26/05/2014.

[14] D. DePriest, NMEA data, disponivel em http://www.gpsinformation.org/dale/nmea.htm. Último acesso em 26/05/2014.

[15] Arc e Tec, disponível em http://www.arq-e-tec.com/2011/09/wikihouse-projetos-de-arquitetura-open-source/. Último acesso em 26/05/2014.

[16] Free Software Foundation, disponível em https://www.fsf.org/about/what-is-free-software. Último acesso em 10/01/2015.

[17] GNU Operating System, disponível em http://www.gnu.org/. Último acesso em 10/01/2015.

[18] R. Peace, Electric bikes start to come of age, disponível em http://www.bikeradar.com/news/article/electric-bikes-start-to-come-of-age--18203/. Último acesso em 10/01/2015.

[19] The RS232 Standard, disponível em http://www.camiresearch.com/Data_Com_Basics/

RS232_standard.html. Último acesso em 10/01/2015.
[20] What is a Raspberry Pi?, disponível em http://www.raspberrypi.org/help/what-is-a-raspberry-pi/. Último acesso em 10/01/2015.
[21] FERREIRA, Alisson Gonçalves. *Tecnologias da informação verdes.*. 2009. 29 f. Trabalho de conclusão de curso (Especialização em Ecologia humana) - Instituto Superior de Educação do Paraná, Maringá, 2009. [Orientador: Fábio Angeoletto.]
[22] Language Reference, disponível em http://arduino.cc/en/Reference/HomePage. Último acesso em 10/01/2015.
[23] Climatempo, disponível em http://www.climatempo.com.br/previsao-do-tempo/cidade/56/salvador-ba. Último acesso em 18/05/2015
[24] Solar Fuels and Artificial Photosynthesis, disponível em http://www.rsc.org/campaigning-outreach/global-challenges/energy/. Último acesso em 10/01/2015. Royal Society of Chemistry.
[25] Pesquisa de mobilidade, disponível em http://www.seinfra.ba.gov.br/mobilidade2012/mobilidade.html. Último acesso em 10/01/2015.
[26] S. Monk, Programando o Raspberry Pi, primeiros passos com Python, novatec, 2013.
[27] J. Boxall, Arduino workshop A hands-on introduction with 65 projects, No Starch Press, 2013.
[28] C. Outram, C. Ratti, A. Biderman, The Copenhagen Wheel: An innovative electric bicycle system that harnesses the power of real-time information and crowd sourcing, COP15 United Nations Climate Change Conference in Copenhagen, 2010.
[29] K. Brunet, T. Oliveria, Arte, DIY e Comunicação Ambiental: Estudo de caso do projeto Sensorium, do mar para o rio, 2º Encontro Interdisciplinar de Comunicação Ambiental (EICA), 2013.
[30] N. Almeida Martins, Sistemas Microcontrolados Uma Abordagem com o Microcontrolador PIC 16F84, Novatec editora, 2005.
[31] E. Schimidt Oleques, Sistemas embarcados, Acadêmico do Curso de Sistemas de Informação, Universidade da Região da Campanha – Alegrete-RS.
[32] P. Marwedel, Embedded System Design, 1ª edição, Kluwer Academic Publishers: Hardbound, outubro de 2003.
[33] D. Moura Santos, Projeto de sistemas embarcados: Um estudo de caso baseado em microcontrolador e seguindo AOSD, Universidade Federal de Santa Catarina - UFSC, Florianópolis, SC, 2006.
[34] LEI Nº 12.587, DE 3 DE JANEIRO DE 2012. disponível em https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12587.htm. Último acesso em 20/05/2015. Palácio do Planalto, 2012.
[35] Definição de Open Source Hardware (OSHW) 1.0, disponível em http://www.oshwa.org/definition/portuguese/. Último acesso em 15/05/2015. Open Source Hardware Association.
[36] Ministério das Cidades, Conselho Nacional de Trânsito, Código de trânsito Brasileiro e legislação complementar em vigor, disponível em http://www.denatran.gov.br/publicacoes/download/CTB_E_LEGISLACAO_COMPLEMENTAR.pdf. Último acesso em 24/05/2015.
[37] NMEA 0183 Standard, acessado em http://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp. Último acesso em 24/05/2015
[38] Programa de mobilidade não motorizada para o estado da Bahia, Companhia de desenvolvimento urbano do estado da Bahia - CONDER 2009, disponível em http://www.conder.ba.gov.br/arquivos/biblioteca/34/PDF_BIBLIOTECA.PDF. Último acesso em 18/06/2015

## ApêndiceA. NMEA 0183

A NMEA 0183 é um conjunto de especificações para comunicação de dispositivos eletrônicos de navegação tais como Anemômetros, girocompassos, piloto automático, receptores GPS e muitos outros tipos de instrumentos [37].

Esse conjunto de especificações é necessário para decodificar as informações vindas do GPS.

A comunicação com o receptor GPS é definida nesta especificação. Estes dados recebidos através da especificação incluem uma solução de posição, velocidade e tempo calculada a partir do receptor de GPS.

O protocolo definido pelo NMEA funciona através do envio de dados em uma única linha independente. Cada família de dispositivo conta com sentenças padrões e existe a possibilidade de extender as sentenças para uso individual.

As sentenças padrão são formadas por um prefixo de duas letras que definem o tipo de dispositivo que utiliza este tipo de sentença, (para os receptores GPS o prefixo é GP) que é seguido por uma sequência de três letras que definem os conteúdos das sentenças.

Além disso, o padrão NMEA permite que fabricantes de hardware possam definir suas próprias sentenças. As sentenças proprietárias começam com a letra P e são seguidas por 3 letras que identificam o fabricante dessa sentença.

Toda sentença inicia-se com um "$"e termina com uma sequência de /fimdelinha e não pode possuir mais de 80 caracteres de texto visível (mais os terminadores de linha). Os dados estão contidos dentro

desta linha única com itens separados por vírgulas. Os dados em si são apenas texto ASCII e podem ser estendidos ao longo de várias sentenças, e em certos casos, especializados, mas normalmente são totalmente contidos em uma frase comprimento variável até o seu máximo visível, por exemplo, a sentença "$GPRMC,123519,A,4807.038,N,01131.000,E,022.4, 084.4,230394,003.1,W*6A."nos informa que "GP"se trata de um dado de GPS, "RMC"significa mínima sentença recomendada, "123519", significa 12 horas, 35 minutos e 19 segundos no fuso horário UTC, "A"significa que a comunicação está ativa, "4807.038,N"a latitude 48.07038 graus N e "01131.000"a longitude 11.31 leste.

Os dados podem variar na quantidade de precisão contida na mensagem. Por exemplo, o tempo pode ser indicado por uma parte decimal de segundo ou a localização pode ser vista com 3 ou até 4 dígitos depois do ponto decimal. Programas que lêem os dados só devem usar as vírgulas para determinar os limites do campo e não depender das posições das colunas.

Existe uma disposição para uma soma de controle no final de cada período, que pode ou não ser verificada pela unidade que faz a leitura dos dados. O campo *checksum*(verificação) consiste de um '*' e dois dígitos hexadecimais que representam um bit OU exclusivo de todos os caracteres entre, mas não incluindo, o '$' e '*'. A soma de verificação é necessária em algumas sentenças [14].

## ApêndiceB. Calculo de velocidade de ultrapassagem

No início do projeto foi pensado na possibilidade do uso do sensor ultrasônico, HC-SR04 para medir a velocidade que um carro passa por uma bicicleta, essa medição se tornou inviável por uma série de motivos diferentes, dentre eles:

- O tipo de sensor usado emite suas ondas ultrasônicas em propagação cônica, ao invés de linear, tendo assim mais tempo de captura de dados da mesma fonte, diminuindo o intervalo de cálculo.
- O sistema existente tende a estar em movimento no mesmo sentido e direção da fonte calculada, necessitando-se assim do cálculo da velocidade média do sistema, que hoje não é possível com módulo atual.

Para esta funcionalidade se faz necessário um pouco mais de pesquisa e um equipamento mais especializado.