

MTAlert – Rastreador Veicular

Eder Antonio da Silva¹, Rodolfo Francisco de Oliveira¹

¹Instituto Federal de São Paulo – Campus Hortolândia (IFSP-HTO) Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistema. Caixa Postal 13183-250 – Hortolândia – SP – Brasil

ederanton@gmail.com, rodolfo.oliveira@ifsp.edu.br

Abstract. *Every year thousands of people suffer traffic accidents in the world. So, the subject of traffic safety occupies a prominent place for society and researchers in this segment, that looking for a way to minimize the impacts generated by such accidents. With the emergence of the Internet Concept of Things driven by the evolution of several areas, Different research and new technologies has been emerging with the aim increasing the safety of drivers and pedestrians. One proof of this is the breakthrough in Intelligent Vehicle studies and Standalone Vehicle. One of the problems that can even lead the victim of traffic accident to death or to have irreversible sequel is the time spent to identify the occurrence or even the location of this accident. The work in question developed a Transport Monitor with an accident alert MTAlert, aim to monitor the fall of motorcycle and bicycle in question so that if any accident happen, the monitor sends a alert the people next to this driver about the event and th current location.*

Resumo. *Todos os anos milhares de pessoas sofrem acidente de trânsito no mundo. Com isto, o assunto segurança no trânsito ocupa um lugar de destaque para a sociedade e pesquisadores deste segmento, que procuram uma forma de minimizar os impactos gerados por esse tipo de acidentes. Com o surgimento do conceito de Internet das Coisas impulsionado pela evolução de diversas áreas, diversas pesquisas e novas tecnologias estão surgindo, com o intuito de aumentar a seguranças dos condutores e pedestres. Uma prova disto é o avanço nos estudos de Veículos Inteligentes e Veículo Autônomo. Um dos problemas que podem até levar a vítima de acidente de trânsito à morte ou a ter sequela irreversível, é o tempo gasto para se identificar a ocorrência ou mesmo a localização deste acidente. O trabalho em questão desenvolveu um Monitorador de Transporte com alerta de acidente MTAlert, visa monitorar quedas em motocicletas e bicicletas em questão para que, em caso de acidente, seja capaz de alertar as pessoas próximas a este condutor sobre o ocorrido, enviando a localização atual do motorista.*

1. Introdução

O presente trabalho aborda um assunto de grande importância para a mobilidade e a segurança dos usuários de motocicletas e bicicletas, relacionado a acidentes envolvendo quedas de seus condutores.

Com apenas 27% da frota nacional, a motocicleta foi o veículo com maior número de acidentes em 2017 no Brasil, liderando o *ranking* comparado com outros veículos. Do total de acidentes, 79% acarretam invalidez permanente, 7% por mortes e 14% DAMS - despesas médicas [Trânsito 2017], conforme ilustra a Figura 1.

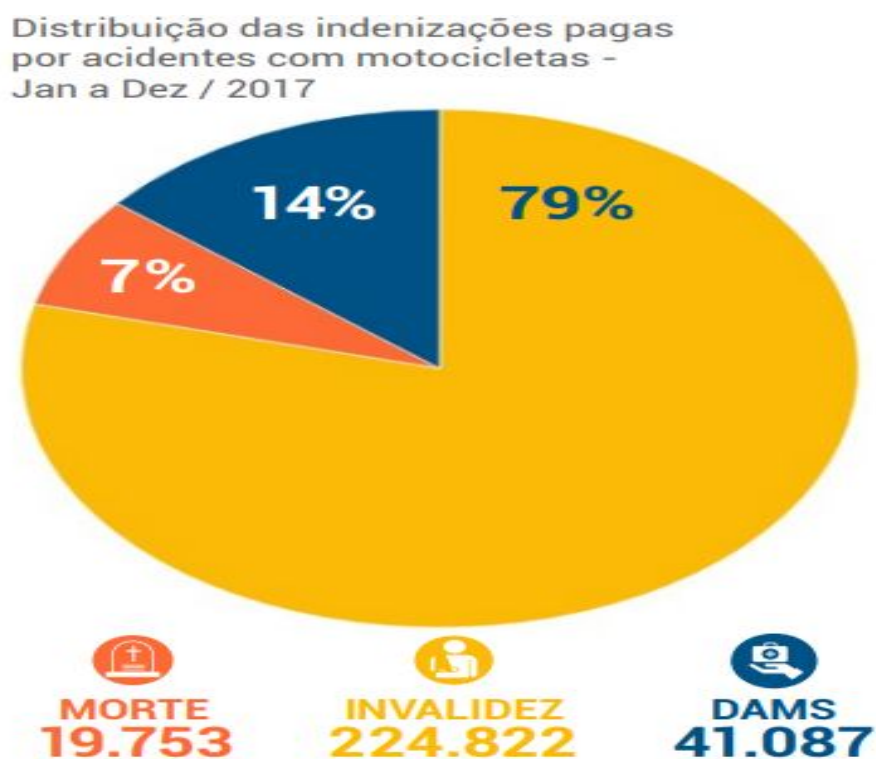


Figura 1- Relatório Anual: Distribuição das indenizações Janeiro a Dezembro 2017 (TRÂNSITO, 2017).

Partindo do pressuposto de que quanto mais rápido uma equipe de socorro for comunicada da queda (acidente), maior a chance da vítima ser socorrida com vida ou com o menor número de sequelas possíveis. Este trabalho tem como objetivo apresentar uma ferramenta tecnológica baseada no conceito de *Internet das Coisas* e em serviços de geolocalização para motocicletas e bicicletas, a fim de localizar o local da queda do condutor, contribuindo para uma melhor agilidade no socorro da vítima. Existem várias aplicações com o intuito de localizar pessoas/veículos, porém poucas são voltadas ao âmbito de detecção de queda, sendo essa a proposta deste trabalho.

O sistema em questão será composto por um módulo de Interface *Web*, onde os usuários condutores poderão se cadastrar, bem como as pessoas que o mesmo deseje que

recebam a mensagem. Além deste, um módulo denominado Nó Sensor será instalado na motocicleta ou bicicleta do usuário. Este terá como função detectar a queda e a localização geográfica da mesma (baseado no sistema GPS - *Global Positioning System*), enviando uma mensagem via rede de dados para o servidor da aplicação a fim de que este possa enviar uma mensagem de alerta aos contatos cadastrados.

Este trabalho é organizado da seguinte maneira: na Seção 2 será abordada uma breve introdução aos conceitos de Internet das Coisas e de Veículos Inteligentes, onde este trabalho se apoia. Na Seção 3 serão apresentados os Trabalhos Correlatos. Na Seção 4, será apresentada a Metodologia utilizada no trabalho. Na Seção 5 será a Apresentação da Proposta. Na Seção 6 serão apresentadas as ferramentas (tanto no nível de *hardware* quanto *software*) utilizadas na implementação da proposta. Na Seção 7 serão apresentados os testes e resultados obtidos com a implementação, bem como os respectivos comentários. Por fim, na Seção 8 será apresentada a conclusão do trabalho e as perspectivas de trabalhos futuros.

2. Revisão Bibliográfica

2.1 Internet das Coisas ou *Internet of Things* (IoT)

A *Internet das Coisas* ou *Internet of Things* (IoT) nasceu do avanço de diversas áreas, tais como Sistemas Embarcados, Microeletrônica, Tecnologia da Informação e Comunicação. Seu conceito se baseia em um grande número de objetos usados no cotidiano, como lâmpadas, eletrodomésticos e veículos, conectados a Internet, proporcionando comunicação entre estes (M2M – *Machine to Machine*) e seres humanos [Chen et al. 2014]. A comunicação *Machine-to-Machine* (M2M) refere-se à interconexão e interoperabilidade entre máquinas, sendo normalmente obtida por troca de dados por meio de redes sem fio e servidores com conteúdo *back-end*. Do ponto de vista da comunicação M2M, a máquina pode realizar tal procedimento sem a necessidade de intervenção humana [Wang et al. 2015].

O conceito de IoT está revolucionando a forma com que interagimos com os equipamentos em geral. Um exemplo clássico é uma geladeira conectada à Internet. Esta geladeira pode descobrir quando um determinado alimento está para acabar e solicitar automaticamente a compra junto a uma central de distribuição. A mesma também pode ser capaz de pesquisar por receitas que usaria os produtos que nela estejam contidos, ou até mesmo pesquisar por ofertas de produtos que estão acabando ou mesmo os que normalmente o usuário costuma comprar [Alecrim, 2016]. O exemplo apresentado diz respeito a uma aplicação de *Internet das Coisas* nos lares. No entanto, a mesma pode

estar presente em outros ambientes, como nos hospitais e nas clínicas, onde os pacientes podem utilizar dispositivos conectados a fim de medir os batimentos cardíacos e a pressão sanguínea, podendo ser enviado em tempo real para o sistema de monitoração. Outras áreas de impacto são: automóveis inteligentes (o qual serão abordados com mais detalhes em um capítulo posterior), cidades inteligentes, natureza, etc. [Santos et al. 2016].

Os dispositivos a serem conectados podem ser itens grandes, como, por exemplo, a geladeira citada anteriormente, e, até mesmo, objetos pequenos como lâmpadas e relógios. O que importa é que tais itens devem ser equipados com componentes adequados, proporcionando comunicação através de uma rede de dados. Com o avanço da tecnologia, a diminuição do tamanho desses componentes e a redução do consumo de bateria dos mesmos, diversos dispositivos estão aptos para tal, como, por exemplo, o Arduino, NodeMCU, Beaglebone e Raspberry Pi [Google, 2017] e [Santos et al. 2016].

As tecnologias relacionadas a redes de comunicações vêm crescendo a cada ano que se passa. Entre as tecnologias mais atuais pode-se citar: *Wi-Fi*, Bluetooth e *NFC* (*Near Field Communication*), bem como as redes móveis como 2G, 3G, 4G/LTE, e 5G [Alves e Mendes 2015]. Outro fator importante é a migração do IPv4 para o IPv6, pois oferece um número extremamente elevado de endereços para os dispositivos [Barreto 2015].

2.2 Veículos Inteligentes

Nas últimas décadas, Sistemas de Transporte Inteligentes ou *Intelligent Transportation System (ITS)*, têm sido expressivamente pesquisados para fornecer serviços inovadores e pró ativos relacionados a gerenciamento de tráfego e segurança de condutores. Estes veículos conectados não necessariamente compartilham apenas informações sensoriais, tais como velocidade, presença (veículos ou obstáculos), mas enviam alertas de perigo para veículos próximos ou a um servidor localizados na Internet. O objetivo final é formar uma rede veicular ainda maior, composta por veículos conectados, que, por sua vez, pode contribuir para uma melhor rota ao usuários e auxiliar na gestão de acidentes [Chen et al. 2014].

Os *ITS* vêm se tornando uma alternativa viável na diminuição dos números de acidentes e identificação do mesmo. Tais conceitos se estendem desde veículos com assistência ao condutor (*ADAS*) até veículos totalmente autônomos. Eles são equipados com sensores capazes de coletar informações do ambiente e do próprio veículo. Essas informações são tratadas mediante dispositivos microprocessados e, como consequência, os veículos são capazes de tomar decisões no nível de navegação dependendo do sistema nele contido [Barbosa 2017].

Contudo, o objetivo deste trabalho não é a criação de um veículo autônomo, mas trabalhar com os dados de sensoriamento para detecção de quedas em motocicletas e bicicletas, e, de certa forma, tais dados poderiam contribuir para a formação desta rede veicular.

3. Trabalhos correlatos

Atualmente, a maioria das pesquisas tende a fazer referências a *Veículos Autônomos*. Por esse motivo, poucos trabalhos relacionados a veículos com assistência ao condutor (*ADAS - Advanced Driver-Assistance Systems*) são encontrados na literatura. Abaixo estão alguns dos mais relevantes encontrados com esse tema.

Connected Vehicle Safety em tradução literal: Segurança de Veículo Conectado, é sistema proativo baseando-se em M2M (máquina a máquina), ou, sendo mais específico, Veículo a Veículo proposto em [Chen et al. 2014]. A proposta foi o desenvolvimento de um aplicativo para *smartphone* com o intuito de coletar os dados do usuário com o auxílio dos sensores e da câmera contido no mesmo, a fim de modelar o comportamento do motorista e tentar prever os futuros comportamentos perigosos deste condutor, podendo alertar o próprio condutor ou aos veículos próximos.

O Sistema de Segurança Veicular com o uso de GPS baseado em Arduíno proposto em [Bentes 2013] tem como finalidade o rastreamento via *web* de veículo, podendo bloquear ou desbloquear o mesmo através de uma aplicação. O Arduíno, por sua vez, está configurado para coletar o sinal de um emissor que deverá estar de posse do condutor do veículo. Desta forma, se o veículo se distancia deste emissor, o mesmo irá após um tempo desligar.

Ambos os trabalhos apresentados tem como finalidade aumentar a segurança de seu condutor. A proposta do MTAAlert vem com o mesmo intuito, mas com o diferencial de ter a possibilidade de detectar uma queda.

4. Metodologia

A metodologia adotada para o desenvolvimento do trabalho em questão foi o modelo Incremental, com o uso de protótipos de *hardware* e *software*. Esse modelo intercala as atividades de especificação, desenvolvimento e validação. Sendo desenvolvido em versões (incrementos), a cada versão adicionando funcionalidade à versão anterior. Onde a versão inicial é frequentemente o núcleo do produto (a parte mais importante), a evolução acontece quando novas características são adicionadas [Sommerville 2010]. Conforme ilustra a Figura 2.

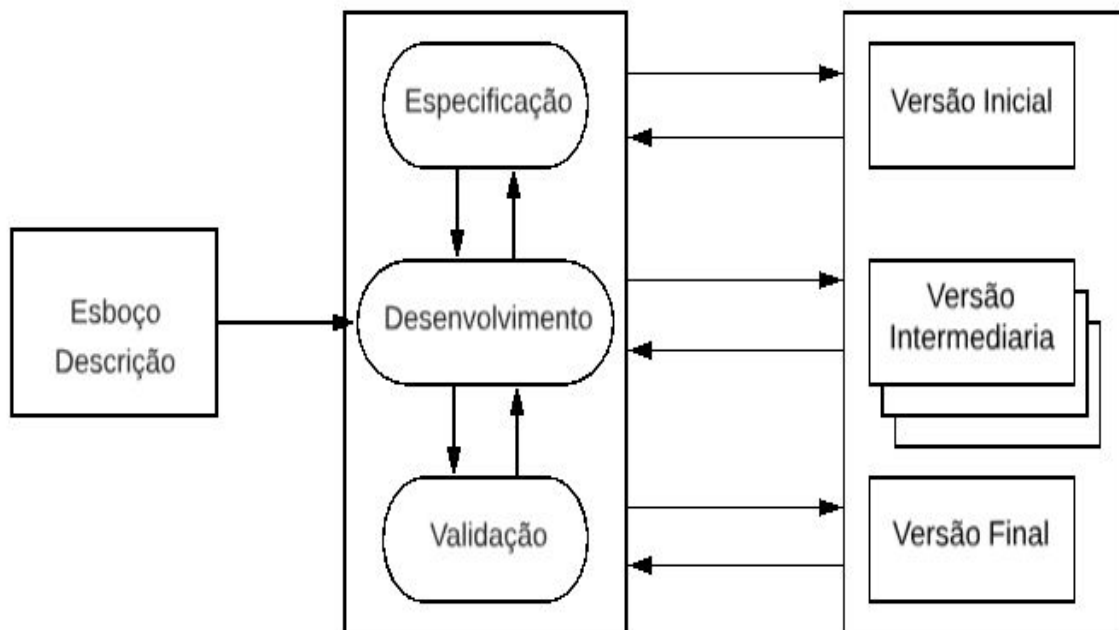


Figura 2- Ciclo Incremental (Fonte: redesenhado com base em Sommerville 2010)

5 Apresentação do Sistema

A Figura 3 apresenta os elementos que compõem o sistema proposto neste trabalho. O mesmo é composto, basicamente, por dois módulos principais: O Nó Sensor (localizado na Bicicleta / Motocicleta) e o módulo Servidor (equipamento constantemente conectado a Internet).

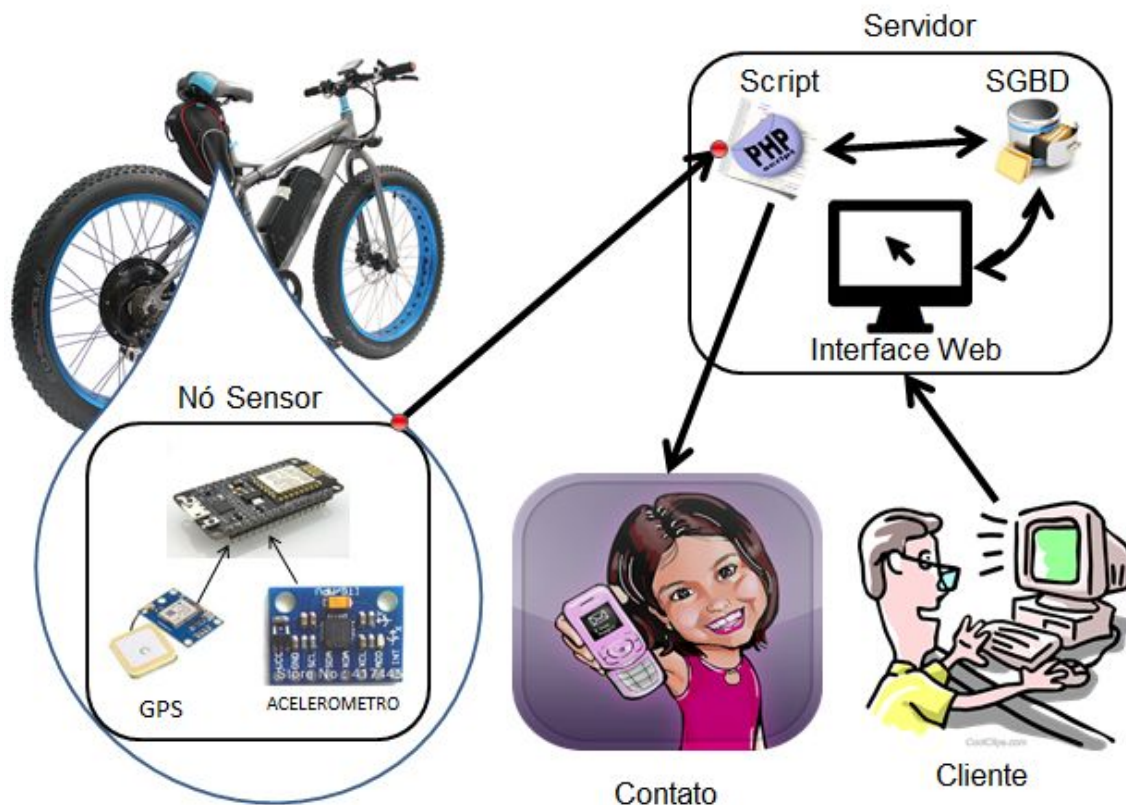


Figura 3- Arquitetura do sistema proposto (Fonte: elaboração própria)

O Módulo Nó Sensor é composto por um módulo de conexão com uma rede de dados, um Módulo GPS e um Sensor de Acelerômetro. O mesmo tem como função coletar os dados em relação ao eixo Y, converter os mesmos em ângulo e detectar possíveis quedas laterais com base no mesmo. Uma motocicleta ou bicicleta em pé se encontra no ângulo 0° . À medida que a mesma se inclina, o ângulo da mesma também se altera. Quando o Nó Sensor detecta uma queda, envia a ocorrência da mesma, juntamente com os dados coletados pelo módulo GPS (latitude, longitude, altura, data e hora) através do módulo de conexão com a rede de dados para um determinado Servidor. A Figura 4 apresenta um fluxograma completo de todas as atividades realizadas pelo Módulo Nó Sensor.

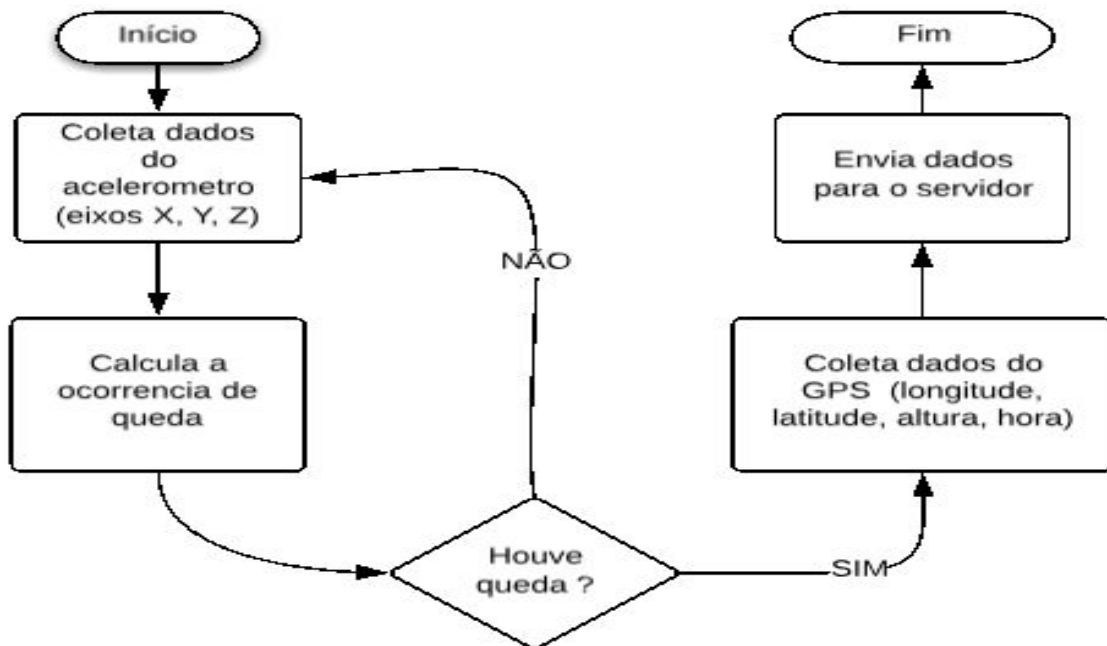


Figura 4 - Fluxograma lógico do Nó Sensor (Fonte: elaboração própria)

Quando a mensagem enviada pelo Nó Sensor chega ao Servidor, ela será processada por um *script* que terá como função contatar o usuário cadastrado no Banco de Dados, alertando o mesmo da queda do motorista. Além disso, um histórico da ocorrência será registrado no sistema.

5.1 Modelo Entidade Relacionamento

O ER (Modelo Entidade Relacionamento) é um modelo conceitual utilizado normalmente em projeto conceitual de banco de dados, para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si [Elmasri et al. 2005]. A Figura 5 apresenta a modelagem utilizada na proposta.

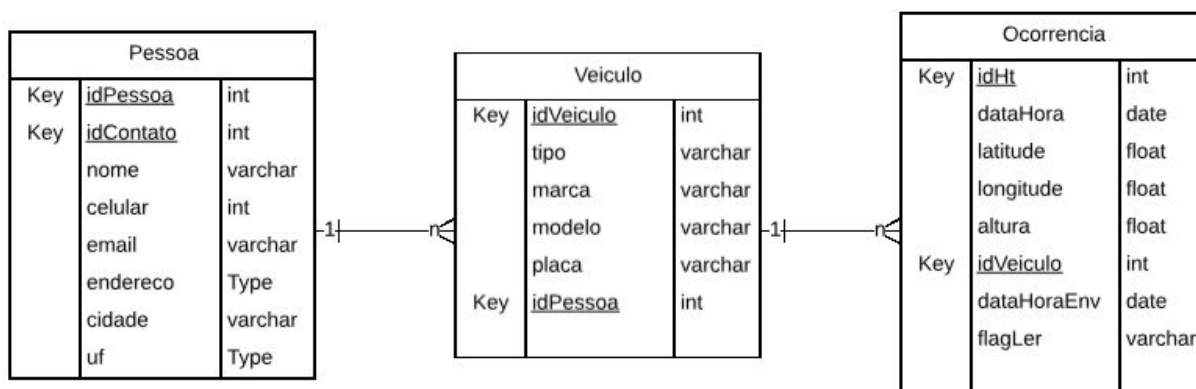


Figura 5- Diagrama Entidade Relacionamento - DER (Fonte: elaboração própria)

A entidade Pessoa faz referência ao condutor e seu contato, onde os dados mais relevantes são: nome para identificação, celular e e-mail, sendo estes dois utilizados para contatar o usuário em casos de acidente. Já a entidade Veículo tem como função registrar os diferentes veículos que o condutor poderá conduzir. A relação dos mesmos é dada pela chave estrangeira idPessoa em um relacionamento com a entidade Pessoa de 1 para n (muitos). Por fim, a entidade Ocorrência tem como objetivo registrar as ocorrências (quedas) que o motorista e o veículo possam sofrer. Nela, os principais atributos são a data e hora do registro, a latitude, longitude e altura (para a localização) e data e hora do envio pelo módulo Nó Sensor. Já o atributo flagLer é uma referência para a mensagem de alerta que fica disponível sempre que houver uma nova ocorrência. Quando seu valor é *True*, significa que o usuário já leu a mensagem no sistema. Já quando seu valor é *False*, o usuário ainda não a leu e o sistema a exibirá para o mesmo.

5.2 Diagrama de Classe

O diagrama de classe descreve a estrutura de um sistema modelando as suas classes, seus atributos, operações entre seus objetos [Elmasri et al. 2005]. A Figura 6 apresenta a integração entre as classes da proposta.

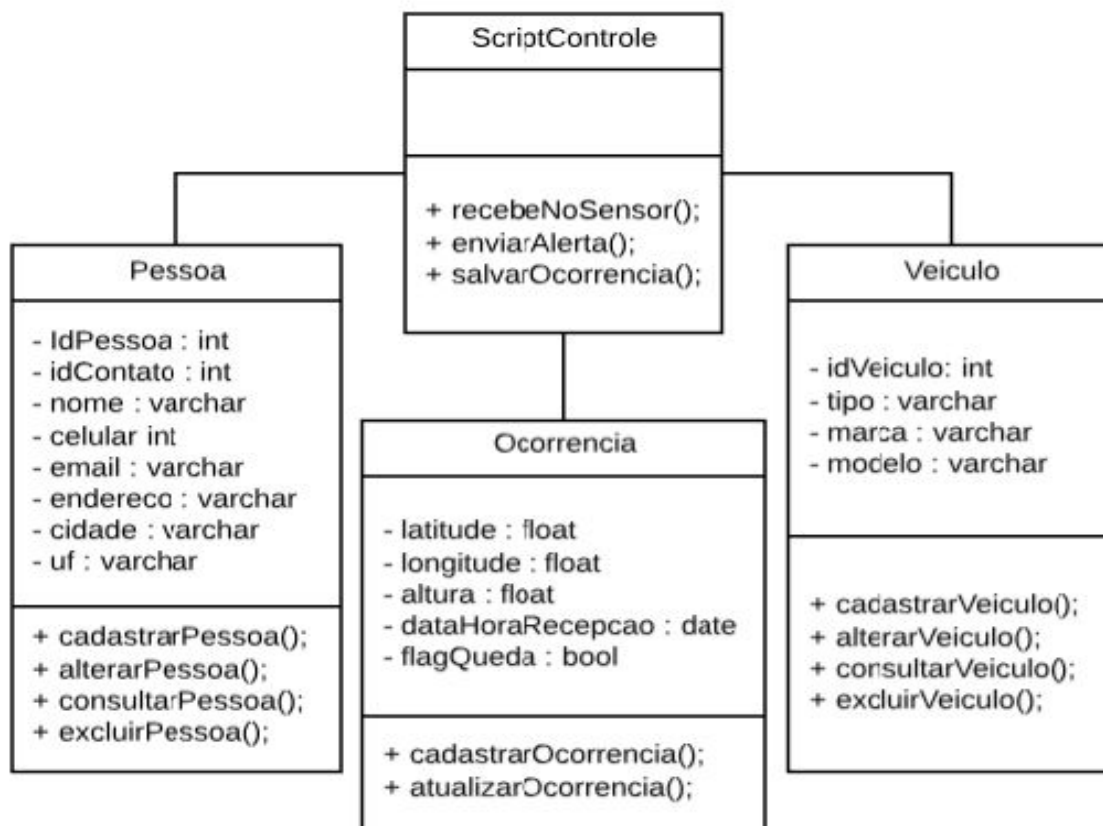


Figura 6- Diagrama de classe (Fonte: elaboração própria)

A Classe Ocorrência é uma representação responsável pela manutenção dos dados referentes ao registro de queda, assim, seus atributos são privados, ou seja, apenas a classe ocorrência poderá manipular os mesmos. As demais classes poderão utilizar apenas os métodos por ela disponibilizado, sendo eles: `cadastrarOcorrencia()`, e `atualizarOcorrencia()`. O método `cadastrarOcorrencia()`, é executado sempre que o Nó Sensor enviar novos dados ao Servidor. Já o método `atualizarOcorrencia()` irá cuidar dos dados já cadastrados no Banco de dados (SGBD), realizando alterações quando necessário.

A classe `ScriptControle` é responsável pela recepção dos dados enviados pelo Nó Sensor utilizando - se do método `recebeNoSensor()`, para garantir a integridade dos dados recebidos, o método `enviarAlerta()`, fica responsável pela buscas dos dados de contato cadastrados ao banco de dados, para usá-lo no envio da mensagem de alerta, e o método `salvarOcorrencia()`, é responsável pela criação de uma nova ocorrência com os dados recebidos pelo Nó Sensor. Já a classe `Pessoa` é quem gerencia os dados do usuário e seus contatos. Os métodos necessários para tal tarefa são: `cadastrarPessoa()`, `alterarPessoa()`, `consultarPessoa()`, e `excluirPessoa()`.

5.3 Algoritmo de Detecção de Quedas

Para o Nó Sensor detectar a ocorrência de quedas, ele utiliza dados gerados por um sensor de aceleração (Acelerômetro). O Acelerômetro é capaz de medir a aceleração de um determinado objeto por 3 eixos, e existe a possibilidade de converter os valores deste eixos em ângulo, como mostra a Figura 7.

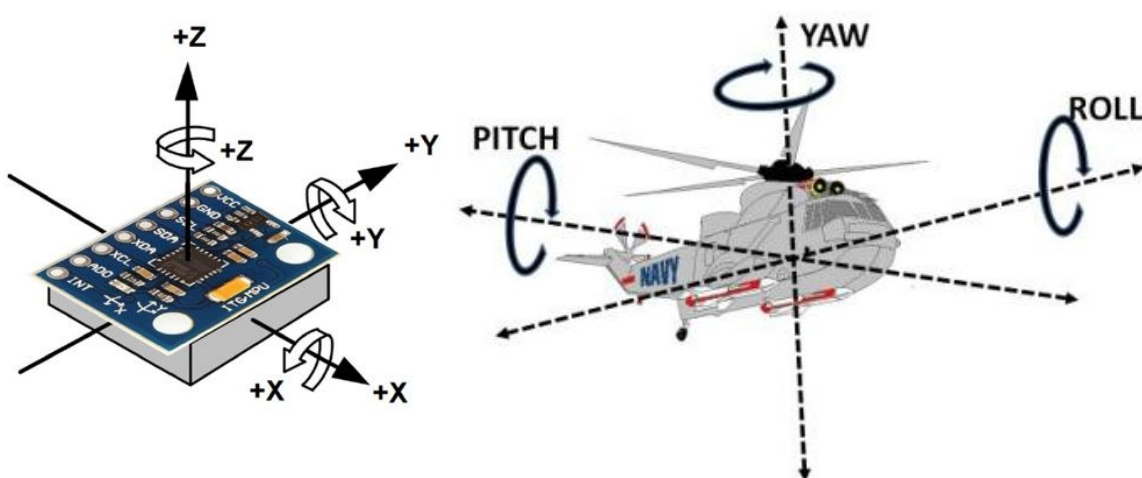


Figura 7 - Eixos de Aceleração (Fonte: <http://www.mecatronizando.com.br/2017/02/controle-de-2-eixos-com-acelerometro.html>)

Inicialmente, os valores referente aos seus eixos (X, Y, Z) se encontram na forma bruta sendo necessário transformá-los em ângulos. O método utilizado para transformação é o Ângulo de Euler, o qual é um conceito utilizado em aeronaves para reprodução dos ângulos de atitude - rolagem, arfagem e guinada (respectivamente *Roll*, *Pitch*, e *Yaw*), e deslocamento linear - Lateral, Vertical e Longitudinal [Obeid et al. 2014]. A detecção da queda propriamente dita é realizada através da análise da alteração referente ao ângulo Y (*Pitch* - deslocamento linear Vertical ou inclinação da motocicleta). É considerada uma queda caso a inclinação seja superior ou igual a 60° graus.

6. Implementação

Para a implementação da proposta em questão, foram utilizadas exclusivamente tecnologias *open-source* e *open-hardware* a fim de que fosse possível serem implementadas novas funcionalidades e modificações na mesma em trabalhos futuros. Além disso, o baixo custo dos componentes de *hardware* e *software* também foram levados em consideração para que a proposta fosse viável economicamente. A seguir é apresentada as implementações dos módulos Nó Sensor e o Servidor.

6.1 Implementação do Módulo Nó Sensor

O módulo Nó Sensor é composto por um NodeMCU Lolin versão 3, um GPS NEO-6M e um acelerômetro MPU-6050. A programação do NodeMCU foi realizada através do Arduíno IDE. A Figura 8 mostra a imagem do NodeMCU.

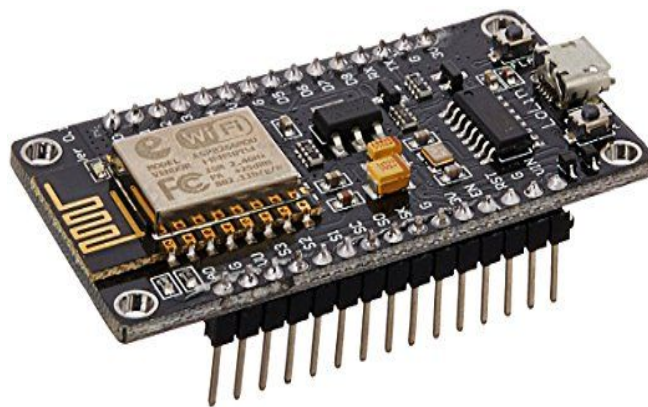


Figura 8 - NodeMCU(Fonte:

<https://www.inventelectronics.com/product/nodemcu-lua-based-esp8266-esp-12e-devkit>
/)

Para compreender um pouco mais sobre o NodeMCU, é necessário entender sua essência. O mesmo é um microcontrolador, ou seja, um circuito integrado que contém processador, memória e periféricos em contraste com processadores de propósito geral (como os utilizado nos PCs). Dessa maneira, pode-se dizer que um microcontrolador é

um tipo especial de circuito integrado, pois vem com a possibilidade de ser programado para desempenhar tarefas específicas. Eles são embarcados no interior de algum dispositivo, como, por exemplo, carros, eletrodomésticos e outros eletrônicos [Barros, 2012 e Bentes 2013].

O NodeMCU é o responsável pela gerência dos dados no Nó Sensor, que são coletados através do módulo GPS NEO-6M. Um módulo GPS tem como finalidade coletar três dados: latitude, longitude e altura em relação ao nível do mar. Para calcular a posição 2D (longitude, latitude), o receptor deve receber a posição de ao menos três satélites, já para calcular a posição 3D (longitude, latitude e altura), é necessária a recepção de dados de quatro ou mais satélites. Após o primeiro envio de dados, a unidade GPS pode calcular outras informações como: velocidade, proa, rastro, distância de viagem, distância ao destino, tempo de viagem, direção do nascer e do pôr do sol [Cláudia e Preto 2013]. Sendo representado na Figura 9 a imagem do GPS NEO-6M



Figura 9 - GPS Neo-6m (Fonte: <https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/>)

Outro equipamento que compõe o Módulo Nó Sensor é o acelerômetro e giroscópio MPU-6050. O MPU-6050 tem como função retornar a aceleração e o eixo de rotação referente aos três eixos (X, Y, Z) (Arduinosb 2017). Este projeto utiliza somente os dados do acelerômetro, mais especificamente o eixo Y, a fim de detectar quedas laterais nas motocicletas e bicicletas, conforme ilustrado na Figura 7 apresentada anteriormente.

A Figura 10 apresenta o diagrama de conexão dos elementos que compõem o Nó Sensor. Um *buzzer* (buzina) foi acrescentado ao projeto a fim de auxiliar a resposta do Nó Sensor em relação às quedas durante a fase de testes.

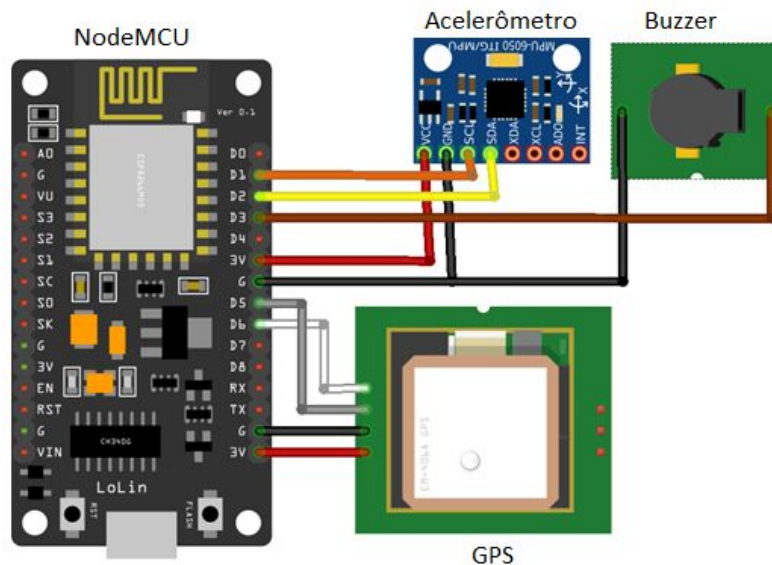


Figura 10 - Diagrama de Conexão dos componentes do Nó Sensor (fonte: elaboração própria)

A Figura 11 apresenta uma foto do protótipo implementado para os testes da aplicação. Nela é possível observar uma moto de brinquedo (com dimensões aproximadas de 30 cm de comprimento por 16 cm de altura), onde o módulo Nó Sensor foi instalado no interior da mesma.



Figura 11 - Protótipo do Nó Sensor (fonte: elaboração própria)

6.2 Implementação do Módulo Servidor

Para a implementação do Módulo Servidor (*Script Receptor* e programação da Interface *Web*), foi escolhida a linguagem PHP (*Hypertext Preprocessor*), por ser uma linguagem *open source*, de uso geral, com alta aceitação e utilização no mercado [The PHP group, 2017].

Para o desenvolvimento da interface foi utilizado o padrão HTML5, a mais recente evolução do HTML. O mesmo contém (se comparado com a versão mais antiga do HTML) novos elementos, atributos e comportamentos, assim como um conjunto

maior de tecnologias que permite o desenvolvimento de aplicações *web* diversas e poderosas [Clark et al. 2014]. A interface também se utiliza do *framework bootstrap* com o objetivo de tornar a mesma mais responsiva. O *bootstrap* foi criado por um *designer* e programador da Empresa Twitter e se tornou uma das estruturas de código aberto mais populares para o desenvolvimento de interfaces *web* [Otto et al. 2017].

Para o módulo de Banco de Dados, foi optado pelo SGBD (Sistema Gerenciador de Banco de Dados) *open source* MySQL. O mesmo, atualmente, é mantido pela *Oracle Corporation* e possui alto nível de desempenho, se comparado a outras alternativas disponíveis (como o PostgreSQL) [Pires et al. 2008].

7. Resultados Obtidos e Testes

7.1 Interface *Web* - Telas

A Figura 12 apresenta a tela inicial da aplicação. Nela é possível observar o menu da aplicação na parte superior e uma descrição do funcionamento da aplicação na parte principal.



Figura 12 - Tela inicial Site (Fonte: elaboração própria)

O formulário de cadastro do usuário é apresentado na Figura 13. Os campos obrigatórios passam por uma validação, a qual está implementada através de recursos do HTML5, a fim de evitar inconsistências ou campos em branco no banco de dados.

MTAlert Home [Login](#)

Cadastrar Usuário

Dados Pessoais
Dados de Contato
Veículo

Nome	<input type="text" value="Nome Completo"/>		
Celular	<input type="text" value="(**) ###-####"/>	E-MAIL	<input type="text" value="e-mail"/>
Senha	<input type="text" value="Senha"/>	Confirmação de Senha	<input type="text" value="Confirmação de Senha"/>
Endereço	<input type="text" value="Endereço"/>		Número <input type="text" value="Número"/>
Bairro	<input type="text" value="Bairro"/>	Cidade <input type="text" value="Cidade"/>	Estado <input type="text" value="São Paulo"/>

[Cadastrar](#)

MTAlert © 2017

Figura 13 - Tela de Cadastro de Usuário (Fonte: elaboração própria)

A Figura 14 mostra a possibilidade de alteração dos dados dos contatos já cadastrados, sendo possível a alteração dos dados do usuário. Os campos já vêm preenchidos com os dados já cadastrados no SGBD, tendo com base o *login* atual, ou seja, quando o usuário logado quiser realizar uma alteração de dados, é enviada uma requisição ao SGBD solicitando os dados referentes à chave representada pelo usuário logado, o SGBD responde essa requisição enviando tais dados ao formulário correspondente a ser alterado.

MTAlert Home [Listar Contato](#) [Alterar Cadastro](#) [Sair](#)

Alteração dos contatos cadastrados

Nome	<input type="text" value="Nany"/>		
Celular	<input type="text" value="99999999"/>	E-MAIL	<input type="text" value="eddyjog@hotmail.com"/>
Endereço	<input type="text" value="seis"/>		Número <input type="text" value="415"/>
Bairro	<input type="text" value="Jd. das Figueira 2"/>	Cidade <input type="text" value="Hortolândia"/>	Estado <input type="text" value="São Paulo"/>

[Editar](#)

MTAlert © 2017

Figura 14 - Tela de Alteração de Cadastro Contatos (Fonte: elaboração própria)

7.2 Módulo Nó Sensor - Teste de detecção de quedas

A fim de detectar a eficiência do algoritmo de detecção de quedas do Módulo Nó Sensor, o protótipo apresentado na seção 5 foi submetido a 100 quedas consecutivas para cada lado da moto (direita e esquerda). Após cada queda, uma aferição foi realizada através do som emitido pelo *buzzer* instalado no protótipo, e uma contabilização foi realizada. A Tabela 1 apresenta os resultados obtidos.

Tabela 1 - Média de acertos na identificação de quedas

Simulação de Quedas	Direita	Esquerda
Acertos	100	100
Erros	0	0
Porcentagem Acertos	100%	100%
Média de Acertos	100%	

Com o alto nível de acerto na simulação de quedas foi realizado um novo teste para verificar se a porcentagem de quedas identificada pelo Nó Sensor é a mesma recebida pelo servidor (*script* de recepção dos dados). O teste de conexão foi realizado em uma rede local (*LAN - Local Area Network*). Inicialmente, foi realizado um teste direto, ou seja, cem vezes para cada lado. O resultado desse teste não foi considerado satisfatório, gerando 29 acertos e 71 erros na conexão ao servidor. Uma das causas possíveis para esse erro pode ser a escolha do protocolo (*HTTP*) de envio dos dados ao Servidor, pois este é um protocolo demasiadamente complexo para o simples *hardware* do *nodeMCU*.

Uma terceira bateria de teste foi realizada, dividindo em 5 sessão de 20 simulação de quedas para ambos os lados e resetado o protótipo (Nó Sensor) ao fim de cada seção. A Tabela 2 apresenta os resultados obtidos.

Tabela 2 - Média de acertos na identificação de quedas e envio ao Servidor

Simulação de Quedas	Acertos					Erros					% Acertos	Média Acertos
	1	2	3	4	5	1	2	3	4	5		
Direita	20	20	20	20	19	0	0	0	0	1	99%	19,8
Esquerda	20	20	20	19	20	0	0	0	1	0	99%	19,8

A Figura 15 comprova que a veracidade na recepção aos email enviados pelo Script de controle.

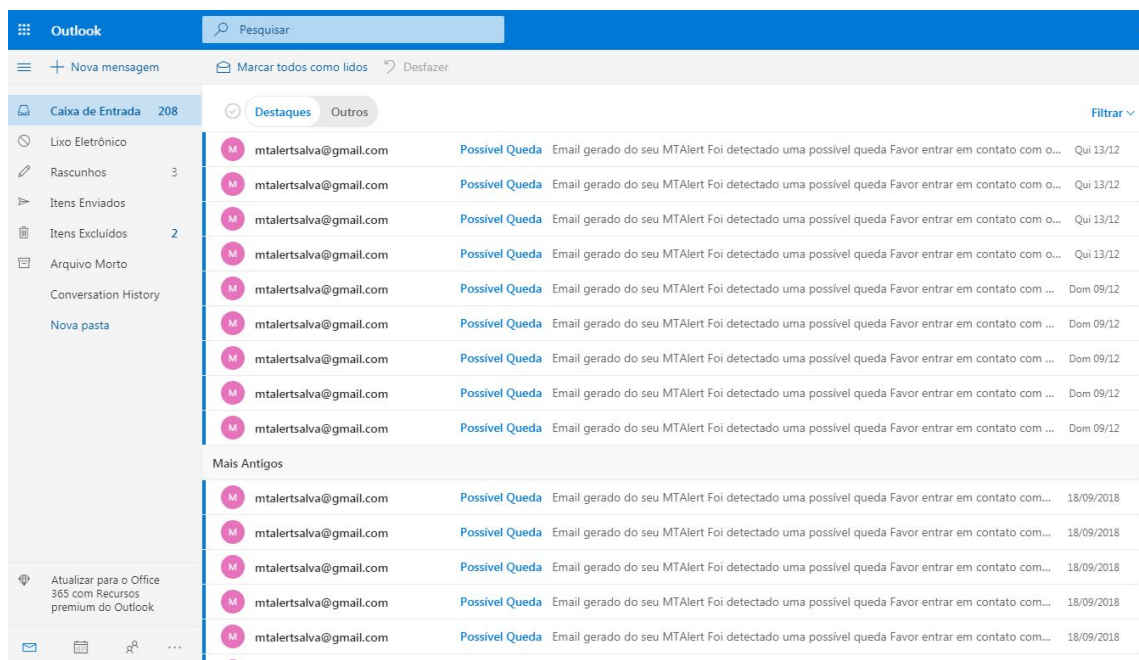


Figura 15 - Tela de Alteração de Cadastro Contatos (Fonte: elaboração própria)

Os testes demonstraram que a proposta se mostrou eficaz no cenário de testes proposto, tanto em detectar quedas quanto para enviá-las para o servidor em questão. No entanto, testes novos devem ser realizados no futuro, envolvendo conexões através de redes móveis (2G, 3G e 4G), bem como testes envolvendo motocicletas e/ou bicicletas reais, para verificar a eficácia em ambientes mais próximos da realidade.

8. Conclusão

Este artigo teve o objetivo de apresentar uma ferramenta tecnológica baseada em geolocalização para motocicletas e bicicletas, tendo como resultado o protótipo *MTAlert*, sendo este um rastreador com a função de detectar a ocorrência de uma queda, e enviar mensagens aos contatos devidamente cadastrados. Os testes realizados foram expressivamente positivos, onde foi constatado um alto nível de acerto sendo 100% em detecção de quedas e 98% de envio de mensagem.

A implementação do protótipo *MTAlert* comprova que as funcionalidades propostas são viáveis, podendo ser reproduzida, instalada e utilizada por condutores, abrindo caminho para novas interações e melhorias, tais como envio de mensagens via *SMS* e aplicativos de celulares, histórico de rota, integração com *API* de mapas e muitas outras funcionalidades.

Para que fosse possível chegar aos resultados adquiridos fez-se necessário vencer grandes desafios, um deles foi a conversão dos valores para ângulo, onde não se conseguia ter um valor estável, e a calibragem do módulo acelerômetro tinha que ser realizada manualmente. Após vários testes e pesquisas foi encontrada uma biblioteca

que realiza a calibração e conversão dos dados. Como citado no segundo teste ocorreu uma falha que possivelmente foi causada pela escolha do protocolo *HTTP*, ficando para um trabalho futuro a utilização de um novo protocolo apropriado para a Internet das Coisas, como por exemplo o *MQTT*, e quem sabe a possibilidade que realizar os testes em um protótipo de escala real como uma bicicleta ou mesmo motocicleta, com o auxílio de rede móvel.

O desenvolvimento deste protótipo foi possível graças aos conhecimentos adquiridos no curso Superior em Análise e Desenvolvimento de Sistemas e em especial algumas disciplinas que foram fundamentais para o desenvolvimento: Linguagem de Programação, Banco de dados, Engenharia de *Software*, Inglês Técnico, Desenvolvimento *Web*, Gestão de Projeto, Projeto de Sistemas, Qualidade de *Software*, Redes de Computadores e Serviços de Redes.

9. Referências

- ©2011 INVENSENSE, I. A. RIGHTS RESERVED. (2013) MPU-6000 and MPU-6050 Register Map and Descriptions. Disponível em: <<https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>>.
- ALECRIM, E. O que é Internet das Coisas (Internet of Things)? Disponível em: <<https://www.infowester.com/iot.php>>. Acesso em: 25 maio. 2017.
- Arduinob. MPU – 6050 Accelerometer + Gyro. (2017) Disponível em: <<https://playground.arduino.cc/Main/MPU-6050>>. Acesso em: 10 maio. Alecrim, E. O que é Internet das Coisas (Internet of Things)? Disponível em: <<https://www.infowester.com/iot.php>>. Acesso em: 25 maio. (2017).
- Alves, M. H. P. and Mendes, L. L. (2015) Levantamento dos cenários de utilização das redes 5G no Brasil. Simpósio Brasileiro de Telecomunicações, .
- Barbosa, H. C. (2017) Conceito de dispositivos registradores de dados para veículos inteligentes.
- Barreto, J. DOS S. (2015) Um modelo de migração de ambiente IPv4 para IPv6 em uma rede acadêmica heterogênea. [s.l.] Universidade de Brasília, 23 mar. .
- Barros, W. R. (2012) Sistema de Automação Veicular com Arduino e Android. p. 59, .
- Bentes, L. M. A. (2013) Sistema de Segurança Veicular Com Uso de GPS Baseado em Arduino. p. 114, .
- Brazil, S. P. Estatísticas nacionais de acidentes de trânsito Estatísticas do DENATRAN. Disponível em: <http://www.vias?seguras.com/layout/set/print/os_acidentes/estatisticas/estatisticas_nacionais>. Acesso em: 19 fev.(2017).

- Chen, K.W. et al. (2014) Connected vehicle safety science, system, and framework. IEEE World Forum on Internet of Things (WF-IoT). Anais...IEEE, mar. 2014Disponível em: <<http://ieeexplore.ieee.org/document/6803165/>>
- Clark, R. et al. Introdução ao HTML5 e CSS3 : A Evolução da Web. Altas Book ed. Rio de Janeiro: 2014.
- Cláudia, A. and Preto, F. (2013) Sistema de Aquisição de Dados para Monitorização de Transportes de Carga.
- Elmasri, R. and Navathe, S. B. Sistema de banco de dados. 4. ed. São Paulo:2005.
- GOOGLE DEVELOPERS, Google. Visão geral da Internet das coisas. 2017. Disponível em: <https://cloud.google.com/docs/tutorials?hl=pt-br#internet_of_things>. Acesso em: 19 fev. 2017.
- G1.GLOBO. Suspeita é que modelo Aline Furlan morreu em acidente, afirma delegado Globo Comunicação e Participações S.A. Piracicaba e Região: [s.n.]. Disponível em: <<http://g1.globo.com/sp/piracicaba-regiao/noticia/2016/07/suspeita-e-que-modelo-alin-furlan-morreu-em-acidente-diz-delegado.html>>. Acesso em: 1 set. (2016).
- OBEID, Ingrid Moura; NETO, Mauro Speranza. Desenvolvimento de Simuladores de Movimento para Análise e Visualização da Dinâmica de Veículos . 2014. Disponível em: <http://www.puc-rio.br/pibic/relatorio_resumo2014/relatorios_pdf/ctc/MEC/MEC-Ingrid%20Moura%20Obeid.pdf>. Acesso em: 3 jul. (2018).
- Oracle Corporation. MySQL 5.7 Reference Manual. Disponível em: <<https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>>. Acesso em: 3 jul. (2018).
- Otto, M. and Thornton, J. B. · The most popular HTML, CSS, and JS library in the world. Disponível em: <<https://getbootstrap.com/>>. Acessado em: 19 fev.(2017).
- Pires, C. E. S., Nascimento, R. O., Salgado, A. C. (2008) Comparativo de desempenho entre bancos de dados de código aberto. Centro de Informática-Universidade Federal de Pernambuco. Recife.
- Santos, B. P., Silva, L. A. M., Celes, C. S. F. S., et al. ([S.d.]). Internet das Coisas: da Teoria à Prática. Disponível em: <<http://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>> Acessado em 7 ago. (2018).
- Yick, J. et al. (2010) Wireless Sensor Networks. Wireless Sensor Network, v. 3, n. 1, p. 1–6, Wang, F. et al. A Survey from the Perspective of Evolutionary Process in the Internet of Things. International Journal of Distributed Sensor Networks, v. 11, n. 3, p. 462752, 31 mar. (2015).

Trânsito, D. O. S. DO. RELATÓRIO ANUAL SEGURADORA LÍDER-DPVAT (2017).
dpvat, 2017. Disponível em:
<https://www.seguradoralider.com.br/Documents/Relatorio-Anual/Relatorio-Anual-Seguradora%20Lider_2017.pdf>

THE PHP GROUP. PHP: What is PHP? - Manual. Disponível em:
<<http://php.net/docs.php>>. Acessado em: 19 fev.(2017).

WANG, F. et al. A Survey from the Perspective of Evolutionary Process in the Internet of Things. International Journal of Distributed Sensor Networks, v. 11, n. 3, p. 462752, 31 mar. 2015.

Zamfirescu, C.-B. PHP: What is PHP-Manual. Disponível em: <<http://php.net/my.php>>. Acessado em 19 fev.(2017).