

Sistema Computacional para Apoio na Configuração de Suspensões em Mountain Bikes

Jefferson Mendes Carvalho¹, Sylvio Andre Garcia Vieira¹

¹Curso de Ciência da Computação – Centro Universitário Franciscano (UNIFRA)
97010-491 – Santa Maria – RS – Brasil

jeffersoncarvalho89@hotmail.com, sylviovieira@gmail.com

Abstract. *This work proposes the development of a system for data acquisition and representation from the bicycle suspension. Through the use of sensors (devices that are able to transform physical values into electrical signals) and Arduino prototyping platform, the data obtained from the ground and the suspension travel, are available to the graphical application in order to provide visual information for helping professionals in the suspension setup process.*

Resumo. *Este trabalho consiste no desenvolvimento de um sistema para aquisição e representação de dados provenientes de um sistema de suspensão de bicicleta. Com a utilização de sensores (que são dispositivos capazes de transformar grandezas físicas em sinais elétricos) e da plataforma de prototipagem Arduino, os dados obtidos referentes ao terreno e ao curso da suspensão são manipulados pela aplicação de representação de dados, provendo informações gráficas que sirvam para auxiliar na tomada de decisão pelo profissional responsável pela configuração da suspensão.*

1. Introdução

Os sistemas de suspensão veiculares são mecanismos que trazem benefícios, pois buscam melhorar o desempenho do veículo em contato com o solo, oferecendo mais estabilidade ao mesmo e, também, trazendo maior conforto ao condutor. Nas bicicletas utilizadas para prática de trilhas, o ajuste da suspensão é essencial para o conforto do ciclista, pois todas as imperfeições desses ajustes, refletem no seu corpo, aumentando sua fadiga e prejudicando sua performance. O ajuste de suspensão é, normalmente, realizado por mecânicos especialistas nesse tema, sendo que a tomada de decisão para o acerto específico pode estar baseada em três pontos-chaves: sistemas de telemetria, objetivos para uma prova de competição e a experiência do mecânico [Oliveira et al. 2015].

Juntamente com a evolução dos sistemas de suspensões, vem se destacando as aplicações computacionais projetadas para essa área e uma das possíveis estratégias que podem ser utilizadas em projetos desse âmbito são os microcontroladores ou plataformas que contemplem esses dispositivos, tais como o Arduino. Este por sua vez tornou possível e facilitada a integração entre diferentes componentes, como sensores e atuadores que permitem realizar a leitura das modificações do terreno e também o trabalho realizado pela suspensão. Com isso, o Arduino possibilita a aquisição de dados por meio de variáveis do ambiente e, conseqüentemente, a manipulação desses dados por um sistema computacional com interface gráfica de usuário.

As aplicações gráficas desenvolvidas para melhoria de performance de suspensão das bicicletas trazem resultados bastante significativos, pois após a aquisição de dados e análise dos mesmos, será possível otimizar o ajuste da suspensão. Uma vez que se conheça os dados relativos ao terreno e também o peso do ciclista é possível estabelecer condições favoráveis para o melhor desempenho da bicicleta, seja por lazer ou até mesmo em competições esportivas.

1.1 Objetivos Gerais

Desenvolver um sistema para aquisição e representação de dados de suspensões veiculares, contemplando o uso da plataforma Arduino e sensores para a obtenção de dados relativos à dinâmica de funcionamento de suspensões de bicicletas e a variação do relevo em um terreno.

1.2 Objetivos Específicos

- Prototipar o sistema de aquisição e de representação de dados;
- Realizar a aquisição de dados por meio do Arduino, do potenciômetro deslizante linear e do sensor ultrassônico;
- Gravar as leituras obtidas pelos sensores no SD Card, para posteriormente serem enviadas para a aplicação gráfica;
- Implementar a aplicação computacional para representação gráfica dos dados obtidos pelos sensores;

2. Referencial Teórico

Esta seção aborda conceitos, tecnologias e trabalhos relacionados inerentes ao tema escolhido para a elaboração do projeto. Serão apresentadas algumas ideias que servem como base para a elaboração da proposta.

2.1 Sistemas de Aquisição de Dados

Atualmente os sistemas de aquisição de dados estão presentes em diversos setores da sociedade e suas aplicações abrangem áreas da engenharia, automação industrial, medicina, robótica, entre outras. Esses sistemas possibilitam obter informações da grandeza física a ser mensurada e gerar dados manipuláveis por circuitos eletrônicos, auxiliando na tomada de decisão [Baptista 2015].

Considerando que a maior parte dos eventos do mundo real e a sua medição são de natureza analógica, no processo de aquisição de dados surge a necessidade da utilização de dispositivos capazes de converter os parâmetros medidos em sinais elétricos. Os dispositivos utilizados para converter essas variações dos fenômenos físicos são denominados sensores e transdutores [Serrano et al. 2016].

2.2 Sistemas Supervisórios

Os sistemas supervisórios são sistemas utilizados nas indústrias e em outros ramos de atividades de produção, e permitem através de tecnologias de computação e comunicação de dados, monitorar os dados relacionados aos processos de produção, bem como os dispositivos conectados aos equipamentos de produção. Por meio de equipamentos de aquisição, os dados obtidos são manipulados e armazenados para apresentar informações ao usuário [Gonçalves da Silva e Salvador 2004].

Os sistemas supervisórios representam um grande avanço nos processos industriais, pois além de automatizar a monitoração e o controle desses processos, agregam qualidade na produção, pois permitem identificar eventuais problemas no processo produtivo e com isso intervir para garantir que o resultado final tenha as características esperadas [Santos 2016].

2.3 Arduino

Arduino é uma plataforma de prototipagem eletrônica que integra *hardware* e *software*, e possibilita realizar processamentos dos dados através de um microcontrolador capaz de oferecer opções de entrada e saída de dados [Arduino 2015]. Seu ambiente de desenvolvimento pode ser inserido em diversas áreas que podem incluir robótica, automação residencial, protótipos industriais ou amadores como sistemas de obtenção de dados acerca de obstáculos em terrenos acidentados [Vasiljević 2013].

Dentre os tipos de Arduino comercializados, existem várias arquiteturas, tais como o Arduino Uno, o Arduino Mega ou Arduino Nano, sendo que cada uma apresenta suas configurações específicas, que se aplicam conforme as necessidades da realização do projeto [Vasiljević 2013].

2.4 Sensores

Sensores são dispositivos capazes de ler variáveis do ambiente, relacionando informações sobre uma grandeza física que precisa ser mensurada e convertida em dados que façam sentido para os circuitos eletrônicos. O uso desses dispositivos tem uma representatividade importante nos contextos em que estão incluídos, pois possibilitaram beneficiar a forma de obter informações para análise e tomada de decisão em diversas situações, bem como suprir várias necessidades de áreas que envolvem tecnologia, indústria, robótica, entre outras [Wendling 2010].

2.4.1 Potenciômetro Deslizante Linear

Os potenciômetros são componentes eletrônicos que permitem manualmente limitar ou aumentar o fluxo da corrente elétrica que passa em um circuito eletrônico [Vasiljević 2013]. A Figura 1 apresenta uma imagem do potenciômetro deslizante linear.

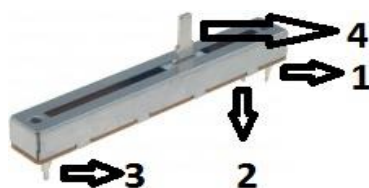


Figura 1 – Potenciômetro Deslizante Linear [Mauser 2015]

Esse componente geralmente apresenta três pinos em sua parte inferior e um cursor na parte superior, conforme a Figura 1. A imagem indica em 1 o pino pelo qual entra um valor de tensão, em 2 o pino que retorna a variação da resistência a partir do movimento do cursor localizado na parte superior, em 3 o pino que é conectado ao GND (terra) e no 4 o cursor que possibilita variar manualmente a resistência do potenciômetro.

A partir do movimento da haste do potenciômetro será possível obter a alternância de valores que indicam a variação da resistência. A resistência de um potenciômetro é

medida em ohms, que é a unidade de medida de resistência elétrica e normalmente a resistência informada em um potenciômetro é a sua resistência máxima [Patsko 2006].

2.4.2 Sensor Ultrassônico

Esses sensores podem ser utilizados em diversas aplicações industriais e também em outros campos de atividade, sendo capazes de detectar diversos tipos de objetos. Eles operam por radiação não sujeita a interferência eletromagnética e sua utilização pode ser realizada de forma eficiente para a detecção de objetos que vão de pequenas a largas distâncias [Braga 2015].

Esses sensores trabalham emitindo pulsos sonoros e quando é propagada uma onda sonora a partir dele, é realizada a medida do intervalo de propagação de emissão e retorno dessa onda. Essa reflexão reproduz um eco convertido em um sinal elétrico. Além disso, a sua emissão se dá em forma de cone devido ao seu modelo de construção e característica do sensor [Thomsen 2011]. A Figura 2 ilustra um exemplo de sensor ultrassônico, em que se pode observar em 1 a alimentação do módulo com +5V (VCC), em 2 o gatilho de disparo do pulso ultrassônico, em 3 o gerador de pulso com a duração do tempo necessário para o eco do pulso ser recebido pelo sensor. E em 4 o pino de aterramento.

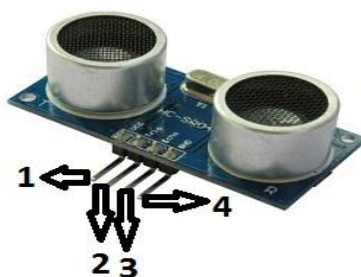


Figura 2 – Sensor Ultrassônico [Zarelli 2013]

2.5 Java

Java é uma linguagem de programação, orientada a objetos [Soares 1996], cuja proposta inicial era oportunizar uma linguagem mais simples e eficiente que as linguagens conhecidas até então, com foco no desenvolvimento de *software* para a área de produtos eletrônicos.

Java proporciona a redução de instruções que eram mais passíveis a erros, tais como ponteiros e também gerenciamento de memória que era realizado por meio do próprio código na concepção de outras linguagens. A linguagem Java também apresenta vantagens para o desenvolvimento de *software* se comparado com linguagens estruturadas, pois oferece várias bibliotecas que permitem ao programador fazer reuso de funções pré-definidas [Mengue 2002], como a *JFreeChart*.

2.5.1 Biblioteca *JFreeChart*

JFreeChart é uma biblioteca utilizada para a criação de gráficos, que tem ampla aceitação por desenvolvedores de *software*. Essa biblioteca possibilita a criação de uma grande variedade de gráficos com características diversas, sendo assim bastante flexível conforme sua necessidade de utilização [Rodarte 2016].

2.6 Trabalhos Relacionados

Nesta subseção serão apresentados alguns trabalhos, em que os conceitos e tecnologias utilizadas são aplicados de forma semelhante à proposta deste trabalho buscando contribuir no desenvolvimento do mesmo.

2.6.1 Aplicação Para Análise Gráfica de Exercício Físico a Partir da Plataforma Arduino

Esse trabalho de Oliveira (2014) contempla a implementação de um sistema, que realiza a coleta dos dados, por meio de sensores adaptados nos equipamentos de ginástica, sendo implementado um *software* para informar se o programa de treino proposto ao praticante está adequado conforme suas características físicas. Para a implementação do *software* foi escolhida a linguagem Java, pois foi a que melhor atendeu os requisitos do projeto, além de disponibilizar as bibliotecas *RXTX* e *JfreeChart* para a leitura dos dados obtidos pelo controlador (Arduino) e plotagem do gráfico.

Após a simulação das atividades pelo praticante, as informações relativas ao esforço físico de cada praticante foram enviadas para uma aplicação computacional de representação gráfica de dados, que apresenta a curva do esforço realizado pelo praticante em tempo de execução.

2.6.2 Utilização da Plataforma de Força para Aquisição de Dados Cinéticos Durante a Marcha Humana

A proposta desse trabalho [Barela & Duarte 2010] é de fornecer subsídios para a utilização da plataforma de força na aquisição de dados cinéticos durante a marcha humana e, também, apresentar informações referentes aos tipos de plataformas. Dentre as características relacionadas às plataformas de força utilizadas para a aquisição de dados, sua estrutura se caracteriza por duas superfícies rígidas, uma superior e uma inferior interligadas por sensores.

Para obter os dados, na maioria das vezes, a plataforma é disposta no chão fazendo com que sua superfície superior esteja nivelada. Assim, através dos componentes que medem as forças exercidas sobre a plataforma e, também, dos componentes que registram a força de reação do solo, é possível obter uma grandeza mecânica relevante para a análise do movimento humano, denominado centro de pressão (CP). Após os dados serem coletados, o computador que contém o *software* que gerencia a aquisição de dados gera resultados para análise de informações da simulação realizada.

2.6.4 Considerações sobre os trabalhos relacionados

Dentre as abordagens apresentadas nos trabalhos relacionados é possível identificar que todos possuem como característica estratégias para a aquisição e representação de dados obtidos através de algum tipo de monitoramento. Diante disso, a proposta deste trabalho se diferencia das demais por se tratar de um sistema de aquisição e representação de dados voltado, especificamente, para o apoio à configuração de suspensões em bicicletas, o que inclui a plataforma Arduino como estratégia de aquisição de dados e a linguagem Java como estratégia para o desenvolvimento da aplicação de representação gráfica desses dados. Outro fator a ser considerado, é que o trabalho de Oliveira (2014), utiliza a biblioteca *RXTX* para comunicação serial entre o Arduino e a aplicação computacional, enquanto que na proposta deste trabalho, a leitura dos dados será realizada previamente à

execução da aplicação gráfica, dispensando o uso da biblioteca RXTX para a manipulação dos dados em tempo de execução.

3. Proposta

A proposta do trabalho consiste no desenvolvimento de um sistema capaz de realizar leituras prévias de um terreno de competição e do curso de amortecimento da suspensão de uma bicicleta sobre esse mesmo terreno.

Com base nas informações de leitura realizadas a aplicação de representação de dados irá exibir os gráficos de variação de terreno versus variação do curso de amortecimento oferecendo, ao responsável pela configuração da suspensão, informações gráficas que sirvam como subsídio para a tomada de decisão. Consequentemente, o objetivo é melhorar o desempenho da bicicleta em detrimento a um possível conforto - suspensões menos rígidas podem oferecer maior conforto. A Figura 3 apresenta uma visão geral com a ideia da proposta do trabalho.

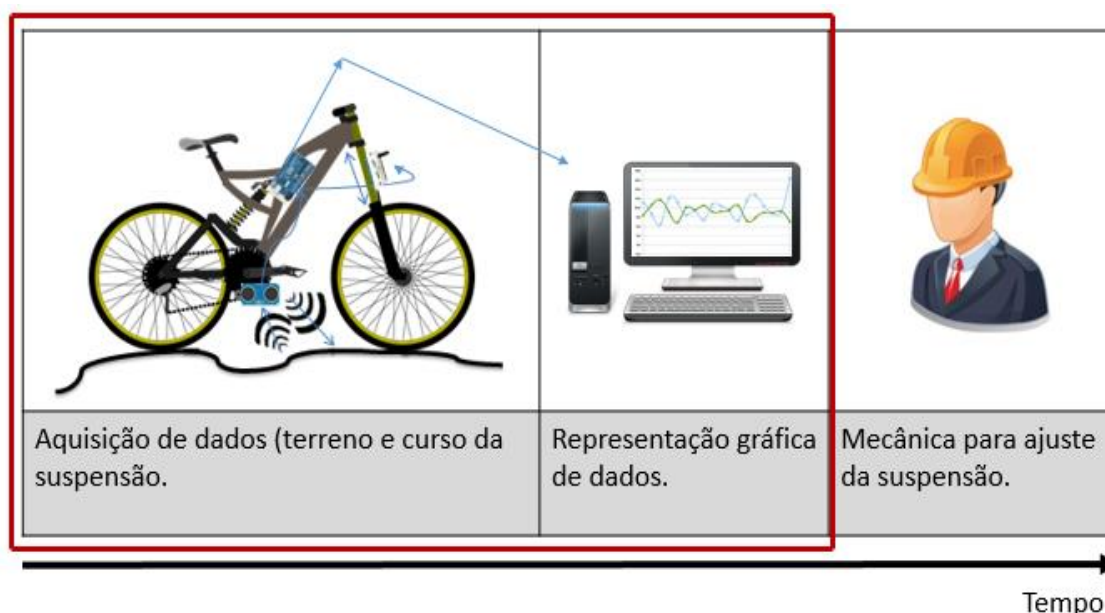


Figura 3 - Visão geral da proposta

A imagem que representa a visão geral da proposta (Figura 3), está dividida em três etapas que são executadas no decorrer do tempo para contemplar o propósito do sistema proposto. É possível verificar que a imagem foi separada por um retângulo de cor vermelha, que especifica a realização da aquisição de dados do sistema e a ilustração gráfica dos dados pela aplicação desenvolvida.

Dessa forma, esse trabalho está baseado na metodologia *Feature Driven Development* (FDD). Essa metodologia foi escolhida pelo fato de atender as necessidades para a realização do projeto, fornecer uma estrutura que facilita o desenvolvimento e oferecer transparência do progresso realizado para o desenvolvedor [Pressman 2011].

A aplicação da metodologia FDD se caracteriza por dividir o desenvolvimento do projeto em etapas [Castro 2013]. A metodologia é aplicada seguindo cinco etapas conforme a ilustra a Figura 4.



Figura 4 – Processos FDD [Palmer e Felsing 2012].

3.1 Projeto do Sistema

Nessa subseção serão descritas as etapas envolvendo a concepção e planejamento do sistema proposto seguindo os padrões da metodologia FDD.

3.1.1 Processo Desenvolver Modelo Inicial

Consiste na etapa inicial da aplicação da metodologia FDD, em que todos os envolvidos na equipe de construção do projeto fazem a abstração do sistema solicitado pelo cliente. Nesse processo é criado o diagrama de domínio e também o diagrama de classes, que contém os atributos e métodos das classes a serem implementadas [Castro 2013]. Assim, seguindo a metodologia, é apresentada na Figura 5 o diagrama de domínio referente ao sistema proposto.

Primeiramente é construída a aplicação responsável por realizar a aquisição de dados do terreno e do curso da suspensão. Através do *integrated development environment* (IDE) do Arduino, são implementados os métodos que possibilitam a aquisição de dados por meio dos sensores instalados na bicicleta. Após a conclusão dessa etapa, o projeto contempla o desenvolvimento do *software* que realiza a exibição gráfica dos dados. Sua execução será através da classe GUI, que é responsável por instanciar objetos para leitura de arquivos e plotagem das informações de leitura via interface gráfica de usuário.

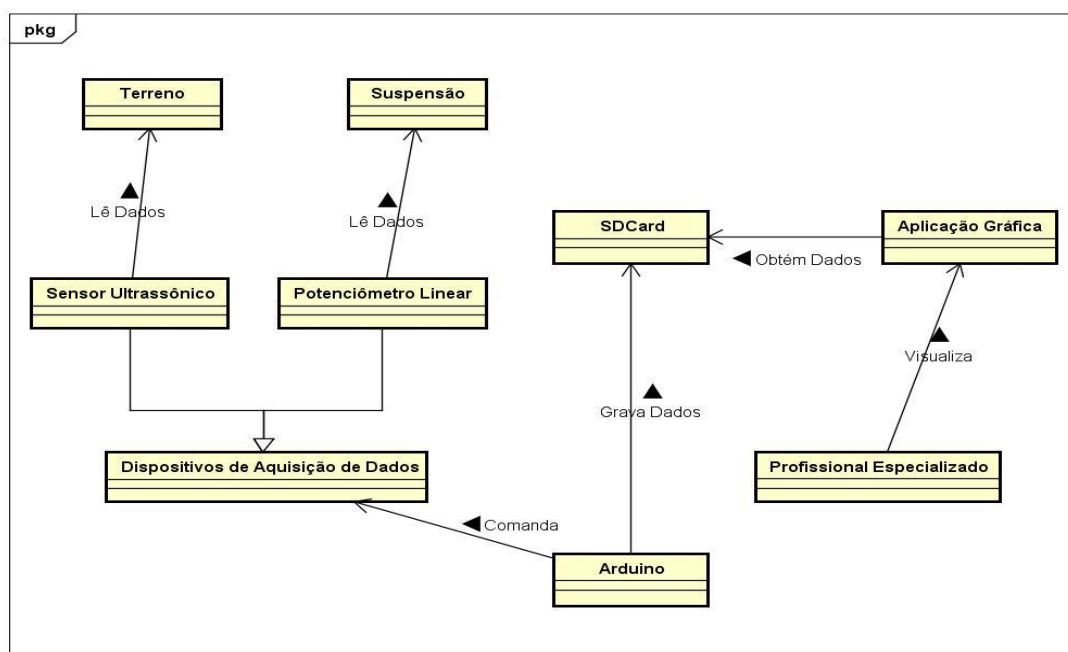


Figura 5 - Diagrama de domínio do sistema proposto

Posteriormente a elaboração do diagrama de domínio, foi construído o diagrama classes que representa o sketch (nome dado para uma aplicação Arduino) responsável pela aquisição de dados do sistema ilustrado na Figura 16 Apêndice 1, e também o diagrama de classes que representa a aplicação gráfica desenvolvida, representado pela Figura 17 Apêndice 1.

3.1.2 Processo Criar Lista de Funcionalidades

Esta etapa consiste em identificar as funcionalidades que vão estar presentes no sistema. Para chegar a essas funcionalidades é realizada a divisão do projeto em subáreas correspondentes ao negócio e, após a execução das tarefas relacionadas ao negócio, se obtém as funcionalidades que vão estar presentes no *software* [Castro 2013].

As funcionalidades previstas para o sistema proposto neste trabalho são:

- Obter os dados correspondentes à suspensão e terreno por intermédio do potenciômetro deslizante e do sensor ultrassônico;
- Gravar os dados obtidos no SD Card;
- Obter os dados do SD Card;
- Gerar o gráfico referente à leitura do curso da suspensão e terreno, para ajuste da suspensão pelo profissional especializado;

3.1.3 Processo Planejar por Cada Funcionalidade

Nesta fase é realizado um estudo da dependência entre as funcionalidades, estimativa de tempo, complexidade e prioridades relativas ao desenvolvimento do projeto. Após, é designado um conjunto de tarefas para cada um dos responsáveis das equipes envolvidas, para o desenvolvimento das funcionalidades previstas [Castro 2013].

Nesta etapa são definidas algumas preferências para a aplicação responsável pela aquisição dos dados e também para a visualização dos dados nos gráficos de linhas. Para a leitura prévia dos dados do terreno e do curso da suspensão, o sistema possibilita ao usuário a interação com o Arduino através de um interruptor. Com isso, para cada leitura realizada a aplicação disponibiliza arquivos com nomeações diferentes, oferecendo ao usuário (especialista pelo ajuste da suspensão) subsídios para uma análise mais profunda de acordo com a configuração da suspensão.

Após a execução da aplicação gráfica, o profissional responsável pelo ajuste da suspensão pode selecionar o diretório em que estão disponíveis os arquivos de texto. Ao abrir os arquivos de texto contendo os dados referentes ao curso da suspensão e do terreno, o profissional especializado consegue realizar a análise dos dados. A biblioteca *JFreeChart* também dispõe de recursos que permitem ao usuário após executar o sistema configurar as propriedades do gráfico, bem como salvá-los através de imagem, PDF ou até mesmo imprimi-los.

3.2 Implementação do Projeto

Nesta subseção serão apresentadas as etapas inseridas na fase de construção do projeto.

3.2.1 Arquitetar por Funcionalidade

Com essa fase, inicia-se a etapa de construção cujo objetivo é refinar a arquitetura do *software* e o modelo de objetos [Castro 2013].

No que tange a etapa de obter os dados referentes à leitura do terreno e do curso da suspensão, um método que testa a inicialização do programa proporciona a gravação

de múltiplos arquivos, para suprir a necessidade de analisar o desempenho da suspensão conforme a configuração que foi utilizada.

3.2.2 Construir por Funcionalidade

Nesta etapa inicia-se a implementação das funcionalidades e do *software* propriamente dito. [Castro 2013].

Seguindo a metodologia e de acordo com o projeto, a implementação está baseada em dois módulos, um de aquisição e outro de representação gráfica dos dados. O módulo que realiza a aquisição dos dados é implementado através da IDE do Arduino, que possibilita realizar as leituras do terreno e da suspensão. A Figura 6 contempla o módulo do programa responsável por realizar leituras do terreno e do curso da suspensão.

```
void lerSensores(){  
  
    long microsec = ultrasonic.timing();  
    distancia = ultrasonic.convert(microsec, Ultrasonic::CM);  
    Serial.println("Distancia sensor Ultrassonico em cm: ");  
    Serial.println(distancia);  
    gravarUltrassonico(distancia, novoCont);  
  
    potValor = analogRead(A0);  
    Serial.print("Valor do Potenciometro = ");  
    Serial.println(potValor);  
    gravarPotenciometro(potValor, novoCont);  
  
}
```

Figura 6 - Módulo de leitura do terreno e curso da suspensão

Posteriormente são realizadas chamadas sucessivas aos métodos responsáveis pela gravação dos dados nos arquivos de texto. A Figura 7 demonstra a gravação dos dados obtidos pelo sensor ultrassônico e a Figura 8 ilustra a gravação dos dados obtidos pelo potenciômetro deslizante.

```
void gravarUltrassonico(float distancia, int novocont){  
    String nomeArquivo = "ult";  
  
    String nomeCont= nomeArquivo += novocont;  
    String nomeFinal= nomeCont += ".txt";  
    Serial.println(nomeFinal);  
  
    char filename[nomeFinal.length()+1];  
    nomeFinal.toCharArray(filename, sizeof(filename));  
  
    ultFile = SD.open(filename, FILE_WRITE);  
    if (ultFile) {  
        ultFile.println(distancia);  
        ultFile.close();  
        Serial.println("gravando no arquivo");  
    }else{  
  
        Serial.println("erro ao criar ult.txt");  
    }  
  
}
```

Figura 7 – Gravação de dados do sensor ultrassônico

```

void gravarPotenciometro(float potvalor, int novocont){
    String nomeArquivo = "pot";

    String nomeCont= nomeArquivo += novocont;
    String nomeFinal= nomeCont += ".txt";
    Serial.println(nomeFinal);

    char filename[nomeFinal.length()+1];
    nomeFinal.toCharArray(filename, sizeof(filename));

    potFile = SD.open(filename, FILE_WRITE);
    if(potFile){
        potFile.println(potvalor);
        potFile.close();
        Serial.println("gravando no arquivo");
    }else{
        Serial.println("erro ao criar pot.txt");
    }
}

```

Figura 8 – Gravação de dados do potenciômetro deslizante

Para o desenvolvimento do módulo de representação gráfica dos dados, foi utilizado o *integrated development environment* (IDE) NetBeans. Diante desse contexto, a ideia principal da aplicação consiste em obter os dados através de uma classe Arquivo para viabilizar a exibição de dados por intermédio do modo gráfico. A classe Arquivo contém o método responsável por ler o arquivo, que foi implementado conforme mostra a Figura 9.

```

public void lerArquivo(String caminho, ArrayList<Double> distancias) {

    try {

        FileInputStream leitura = new FileInputStream(caminho);
        BufferedReader arquivo = new BufferedReader(new InputStreamReader(leitura));

        while (arquivo.ready()) {
            linha = arquivo.readLine();
            distancias.add(Double.parseDouble(linha));
        }
    }
}

```

Figura 9 – Módulo de leitura dos dados pela aplicação gráfica

Após a execução do método de leitura do arquivo, a classe *Graphical User Interface* (GUI) implementa os métodos para a visualização do esboço gráfico. A Figura 10 apresenta os métodos que criam a base de dados referente a leitura do terreno e oportunizam sua exibição pela interface gráfica. Posteriormente a Figura 11 ilustra os métodos que criam a base de dados do curso da suspensão e possibilitam sua visualização gráfica.

```

//Cria o conjunto de dados a serem exibidos no gráfico
DefaultCategoryDataset dataset = new DefaultCategoryDataset();
for (int contador = 0; contador < terrenoDados.size(); contador++) {
    dataset.addValue(terrenoDados.get(contador), "linha", (Comparable) contador);
}

//Cria o gráfico referente a leitura do terreno
JFreeChart lineChart = ChartFactory.createLineChart(Titulo2, eixoXName, eixoYNameG2, dataset,
    PlotOrientation.VERTICAL, Legenda, tooltips, URLs);

```

Figura 10 – Métodos para exibição gráfica da leitura do terreno

```

//Cria o conjunto de dados a serem exibidos no gráfico
DefaultCategoryDataset dataset = new DefaultCategoryDataset();
for (int indice = 0; indice < suspensaoDados.size(); indice++) {
    dataset.addValue(suspensaoDados.get(indice), nomeLinha, (Comparable) indice);
}

//Cria o gráfico referente a leitura suspensão
JFreeChart lineChart = ChartFactory.createLineChart(Titulo, eixoXName, eixoYNameG1, dataset,
    PlotOrientation.VERTICAL, Legenda, tooltips, URLs);

```

Figura 11 – Métodos para exibição gráfica do curso da suspensão

Após a implementação dos métodos que realizam a leitura e gravação dos dados obtidos pelos sensores e do desenvolvimento dos métodos que criam o conjunto de dados e implementam os gráficos para esses dados, é possível a execução da aplicação gráfica. A Figura 12 ilustra a imagem da aplicação gráfica desenvolvida.

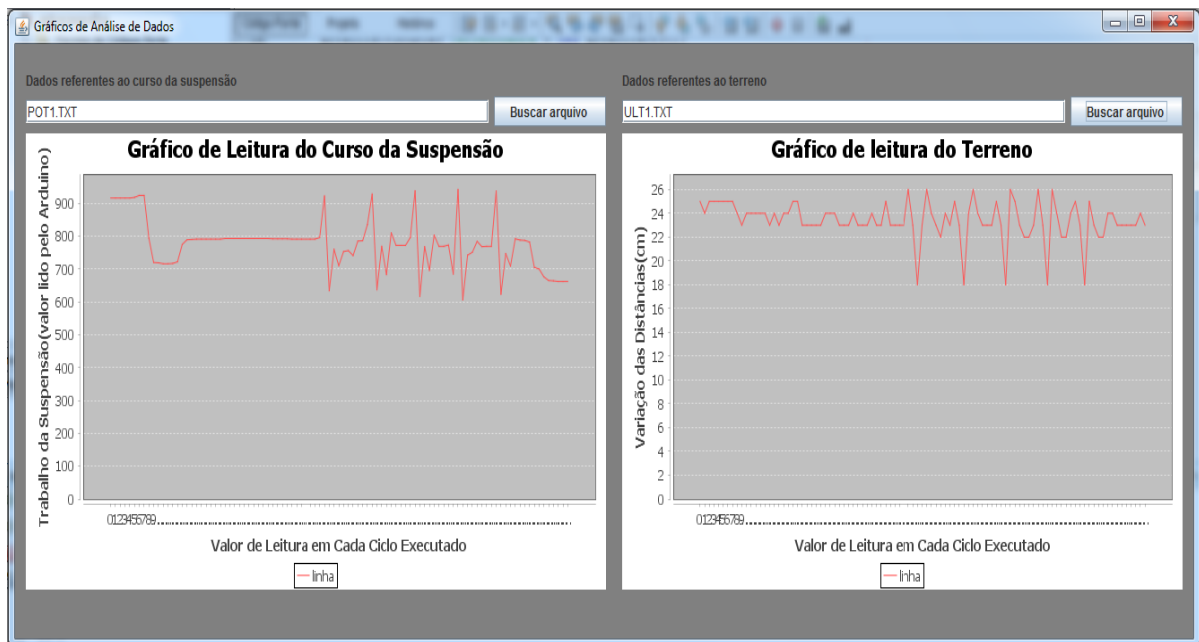


Figura 12 - Aplicação gráfica para análise de dados

4. Cenário de Validação e Testes

Dentre as possibilidades existentes para a concepção do projeto, as quais poderiam incluir a utilização de diferentes tipos de veículos com modelos de suspensões variadas, optou-se pela realização em uma bicicleta pelo fato de sua facilidade na instalação dos dispositivos de aquisição de dados.

Dentro desse contexto, os testes foram realizados com uma bicicleta, marca *Specialized*, modelo Epic Comp [Specialized 2015], e foram adaptados o potenciômetro deslizante na suspensão para realizar a leitura do trabalho realizado pela mesma e o sensor ultrassônico na base da bicicleta (abaixo do movimento central), com a finalidade de mensurar a distância entre a bicicleta e o solo, identificando a variação do terreno.

Para a realização dos testes foi escolhido um terreno plano com distância de trinta metros, onde foram instalados cinco obstáculos para simular um terreno de condições adversas, buscando obter uma variação de terreno para posteriormente reproduzir o comportamento da suspensão em função do terreno. Os obstáculos posicionados no trajeto possuíam uma altura de cinco centímetros, largura de vinte e um centímetros e estavam dispostos a três metros de proximidade entre si.

Após a preparação do cenário, foram executadas seis voltas sobre o mesmo percurso abordando diferentes tipos de configurações e a execução dos testes deu-se a partir de uma velocidade média de 6Km/h, sendo que para cada dois testes praticados, optou-se pelo ajuste da suspensão. Foi adotado para o primeiro e segundo testes o modo de suspensão macia, terceiro e quarto testes suspensão média e no quinto e sexto testes suspensão dura. Na Figura 13 é apresentada uma visão geral do cenário de testes construído.

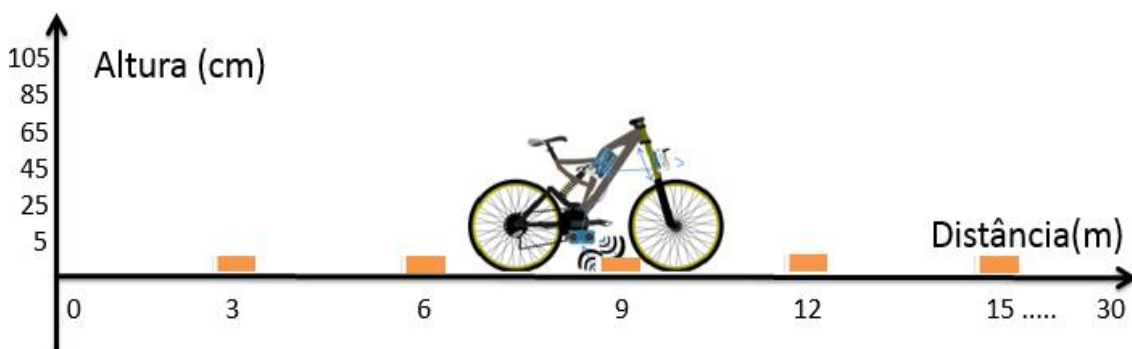


Figura 13 - Percurso realizado pela bicicleta para testes

Após as distâncias serem obtidas pelo Arduino, elas são disponibilizadas pela aplicação de representação gráfica que ilustra as linhas resultantes da variação do curso de amortecimento e variação do terreno. Em seguida o profissional especializado poderá realizar o ajuste da suspensão, que é realizado através do *cap-top* localizado na parte superior da mesma. Na Figura 14 é apresentada uma imagem mostrando o local exato aonde será realizado o ajuste.



Figura 14 – Local de ajuste da suspensão [Expressa 2015].

5. Resultados e Discussões

A concepção do projeto esteve continuamente conduzida para o desenvolvimento de um sistema que proporcionasse um auxílio para a melhor tomada de decisão no ajuste de suspensões em bicicletas.

Durante a construção do cenário de aquisição de dados, um dos gargalos encontrados foi em relação à instalação dos sensores, para que o instante de leitura de terreno fosse executado proporcionalmente ao momento do trabalho realizado pela suspensão para um mesmo obstáculo. Essa dificuldade deu-se pela impossibilidade dos sensores estarem alinhados, pois a pressão na roda dianteira que aciona o potenciômetro (suspensão dianteira) ocorre alguns milésimos de segundo antes que o sensor ultrassônico (abaixo do movimento central) seja acionado.

Devido ao fato do momento de leitura do trabalho da suspensão não ser executado ao mesmo tempo da varredura do terreno para cada obstáculo, foi realizada uma análise detalhada sobre os dados obtidos e pôde verificar-se que, a leitura dos obstáculos ajustados no terreno ocorreu em um ciclo de execução do Arduino após o momento de leitura do curso da suspensão.

Sucedendo a etapa de observação e plotagem dos dados obtidos pela ferramenta gráfica é possível analisar, como mostra a (Figura 15 – A), que nos primeiros instantes o gráfico de leitura da suspensão sofre um declive relativo a uma pressão exercida sobre a suspensão no início do teste. Em seguida é possível observar que o trabalho realizado pela suspensão se mantém constante até o momento que a bicicleta encontra o primeiro obstáculo. A partir desse momento, percebe-se que para cada obstáculo (Figura 15 – B) posto no solo o gráfico acentua um pico e um declive, que respectivamente indicam a força exercida para cima ao cruzar o obstáculo e a absorção do impacto pela suspensão no momento que o pneu encontra o solo novamente.

Em relação à leitura do terreno, nota-se que o gráfico apresenta uma variação em uma faixa de valores bem próxima, conforme a (Figura 15 – C), até o momento em que o sensor ultrassônico detecta cada um dos obstáculos dispostos no solo. Porém antes de identificar cada um dos obstáculos, é possível notar que o gráfico apresenta um pico, que indica um afastamento do sensor ultrassônico em relação ao solo no momento que o pneu transpõe cada obstáculo (momento de leitura da suspensão). No instante que o sensor percebe os obstáculos ajustados no solo, o gráfico aponta um declive para cada obstáculo,

(Figura 15 – D), que indica a aproximação do obstáculo em relação ao sensor ultrassônico. A Figura 15 apresenta a execução da aplicação gráfica exibindo as linhas resultantes de um dos testes executados.

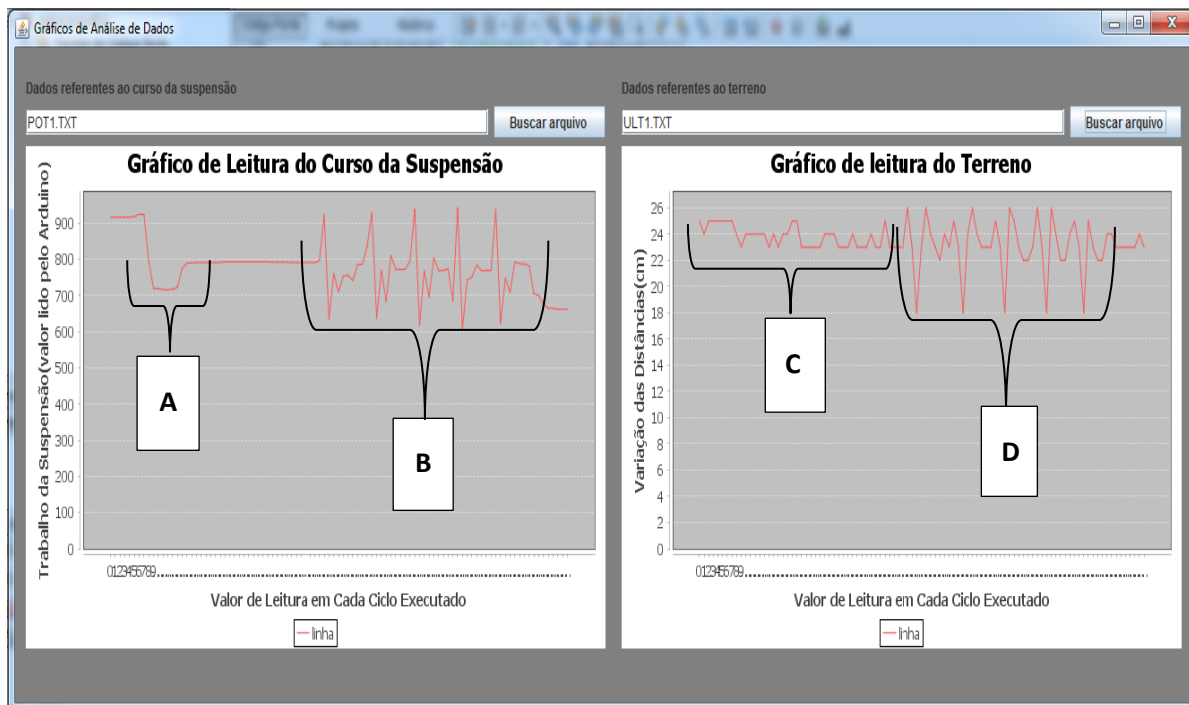


Figura 15 - Aplicação gráfica para análise de dados

6. Conclusões

Os cenários e equipamentos de competições esportivas estão cada vez mais modernos, fazendo uso de tecnologias que vão desde a comunicação de dados e ajuste remoto de veículos (telemetria) até sistemas computacionais sofisticados com predição na busca pela melhor tomada de decisão. Assim, um viés de utilização está relacionado com as suspensões veiculares, uma vez que o ajuste da suspensão em função de um terreno pode ser determinante para um melhor desempenho do veículo. Dessa forma, a proposta deste trabalho contemplou um sistema que visa auxiliar no ajuste da suspensão de bicicletas.

O fato de se ter um conjunto de suspensões conceituadas pelo mercado não garante a excelência de uma prova de competição ciclística, ou mesmo o bom desempenho de práticas de ciclismo se o ajuste da suspensão não atender um conjunto de requisitos. Daí surge a necessidade de considerar fatores como: configuração da suspensão conforme as características do terreno, particularidades individuais de cada ciclista (peso), afinal uma mesma bicicleta com um determinado conjunto de suspensões poderá ser utilizada tanto por um ciclista de 60kg quanto por um de 90kg, sendo que os ajustes especificados para cada ciclista que vão diferir.

Para satisfazer as necessidades que cada situação necessita, o software desenvolvido atende as condições necessárias que permitem ao profissional especializado analisar o comportamento antecipado da suspensão em função de um determinado terreno. O emprego do sistema desenvolvido em prol de benefícios ciclísticos faz com que seja possível incorporar segurança na transposição de obstáculos, confortabilidade para o ciclista, evitando impactos mais dolorosos e também melhoria de performance nas

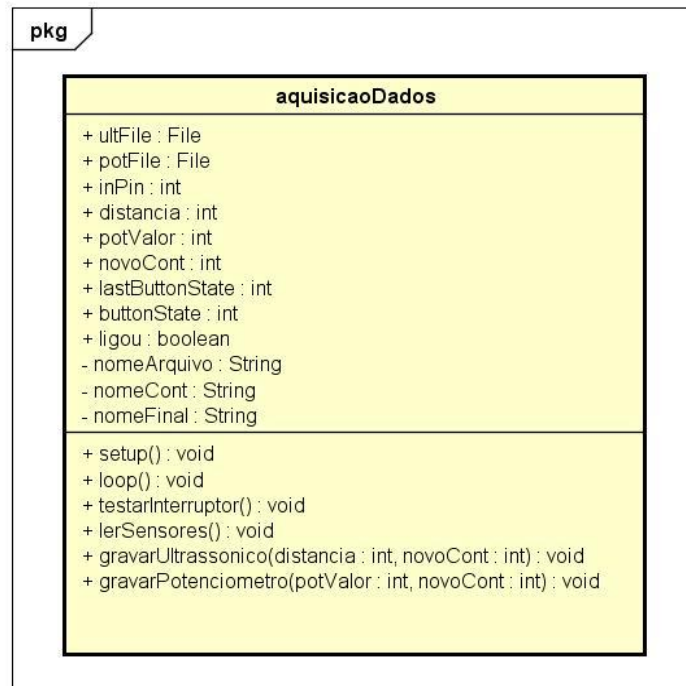
bicicletas, com isso, trazendo diferencial competitivo em provas profissionais de bicicletas ou mesmo práticas por lazer.

Referências

- Arduino. (2015), “Como identificar um Arduino falsificado”. Acesso em Outubro de 2015. Disponível em: <<https://www.arduino.cc/en/Products/Counterfeit>>
- Baptista, M. (2015), “Sistema de Aquisição de Dados”. Acesso em Agosto de 2015. Disponível em: <http://www.estgv.ipv.pt/paginaspessoais/maeb/im/Teorica_Bibliografia/Cap_E_Sistemas%20de%20Aquisi%C3%A7%C3%A3o%20de%20Dados/1-Introdu%C3%A7%C3%A3o/Texto%20de%20Estudo%20-%20Sistemas%20de%20Instrumenta%C3%A7%C3%A3o%20-%20Capitulo%204.pdf>
- Barela, A., Duarte, M. (2010), “Utilização da plataforma de força para aquisição de dados cinéticos durante a marcha humana”. Acesso em Setembro de 2015. Disponível em: <<http://www.socibracom.com/bjmb/index.php/bjmb/article/view/32/27>>
- Braga, N. (2015), “Como Funcionam os Sensores Ultrassônicos”. Acesso em Agosto de 2015. Disponível em: <<http://www.newtonbraga.com.br/index.php/como-funciona/5273-art691>>
- Castro, V. (2013), “Conhecendo o FDD”. Acesso em Outubro de 2015. Disponível em: <<http://coyoti.com.br/blog/conhecendo-o-fdd/>>
- Expressa, B. (2015), “Suspensão RST Dianteira Omega”. Acesso em Novembro de 2015. Disponível em: <<http://www.viaexpressabike.com.br/ciclismo/componentes/garfo-e-suspensao/suspensao-rst-dianteira-omega-tnl-a-set-100mm-aro-26/>>
- Mauser, R. (2015), “Potenciômetro Deslizante Linear”. Acesso em Novembro de 2015. Disponível em: <http://mauser.pt/catalog/product_info.php?products_id=59429>
- Mengue, F. (2002), “Curso de Java Básico”. Acesso em Setembro de 2016. Disponível em: <http://ftp.unicamp.br/pub/apoio/treinamentos/linguagens/java_basico.pdf>
- Oliveira, A. A. M. (2014), “Aplicação para a Análise Gráfica de Exercício Físico A Partir da Plataforma Arduino”, In.: XVIII SEPE 2014. Santa Maria.
- Oliveira, J., Cunha, L., Dias, R., Abreu, Y. (2015), “Sistema de Suspensão Veicular Otimizado Através de um Controle Eletromecânico do Conjunto Mola e Amortecedor”. Acesso em Setembro de 2015. Disponível em: <<http://www.aedb.br/wp-content/uploads/2015/05/322275.pdf>>
- Oliveira, J., Souza, J. (2013), “Monitoramento de Temperatura e Umidade de uma Pequena Lavoura de Maracujá”. Acesso em Agosto de 2015. Disponível em: <http://sombrio.ifc.edu.br/download/redes/TCC_2013/TCC_Jadson_Jhonatan.pdf>
- Palmer, S. R.; Felsing, M. (2002) “A Practical Guide to Feature-Driven Development”. 1st. ed. [S.l.]: Pearson Education. ISBN 0130676152.
- Patsko, L. (2006), “Tutorial, Aplicações, Funcionamento e Utilização de Sensores”. Acesso em Novembro de 2015. Disponível em: <http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf>

- Pressman, R. S. (2011), “Engenharia de Software: Uma Abordagem Profissional”, 7ª ed. McGraw-Holl Interamericana do Brasil Ltda, São Paulo.
- Rocha, F. (2013), “Introdução ao FDD - Feature Driven Development”. Acesso em Setembro de 2015. Disponível em: <<http://www.devmedia.com.br/introducao-ao-fdd-feature-driven-development/27971>>
- Rodarte, C. (2016), "JfreeChart e iReport – Parte 1”. Acesso em Setembro de 2016. Disponível em: <<http://www.devmedia.com.br/jfreechart-ireport-parte-i/3163>>
- Santos, M. M. (2016), “Sistemas Supervisórios”, Acesso em Setembro de 2016. Disponível em: <http://www.cpdee.ufmg.br/~michelle/fabrai/index_arquivos/Aula%20%20-%20Sistemas%20Supervisorios%20parte%201.pdf>
- Serrano, L. M. V., Alcobia, C. J. O. P. J., Mateus, M. L. O. S., Silva, M. C. G. (2016), “Sistemas de Aquisição, Processamento e Armazenamento de Dados”. Acesso em Setembro de 2016. Disponível em: <http://www.spmet.pt/Eventos/Encontro1/Luis_Serrano.pdf>
- Silva, A., Salvador, M. (2014), “O que são Sistemas Supervisórios?”. Acesso em Agosto de 2015. Disponível em: <http://www.wectrus.com.br/artigos/sist_superv.pdf>
- Soares, L. (1996), “Linguagem Java”. Acesso em Agosto de 2015. Disponível em: <<http://www.cin.ufpe.br/~arfs/introjava.pdf>>
- Specialized. (2015), “Specialized Bicycle Components”. Acesso em Outubro de 2015. Disponível em: <<http://www.specialized.com/br/pt-br/bikes/mountain/epic/epic-comp-29>>
- Thomsen, A. (2011), “Como conectar o sensor ultrassônico HC-SR04 ao Arduino”. Acesso em Setembro de 2016. Disponível em: <<http://blog.filipeflop.com/sensores/sensor-ultrassonico-hc-sr04-ao-arduino.html>>
- Vasiljević, G. (2013), “Apostila de Arduino”. Acesso em Agosto de 2015. Disponível em: <http://escoladeverao.weebly.com/uploads/5/0/3/7/5037808/apostila_v0.5a.pdf>
- Wendling, M. (2010), “Sensores”. Acesso em Agosto de 2015. Disponível em: <<http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>>
- Zarelli, G. (2013), “Arduino - Sensor Ultrassônico HC-SR04”. Acesso em Outubro de 2015. Disponível em: <<http://helpdev.com.br/2013/02/11/arduino-sensor-ultrassonico-hc-sr04-arduino-ping-sensor-de-distancia/>>

Apêndice 1



powered by Astah

Figura 16 - Classe referente ao módulo de aquisição de dados do sistema

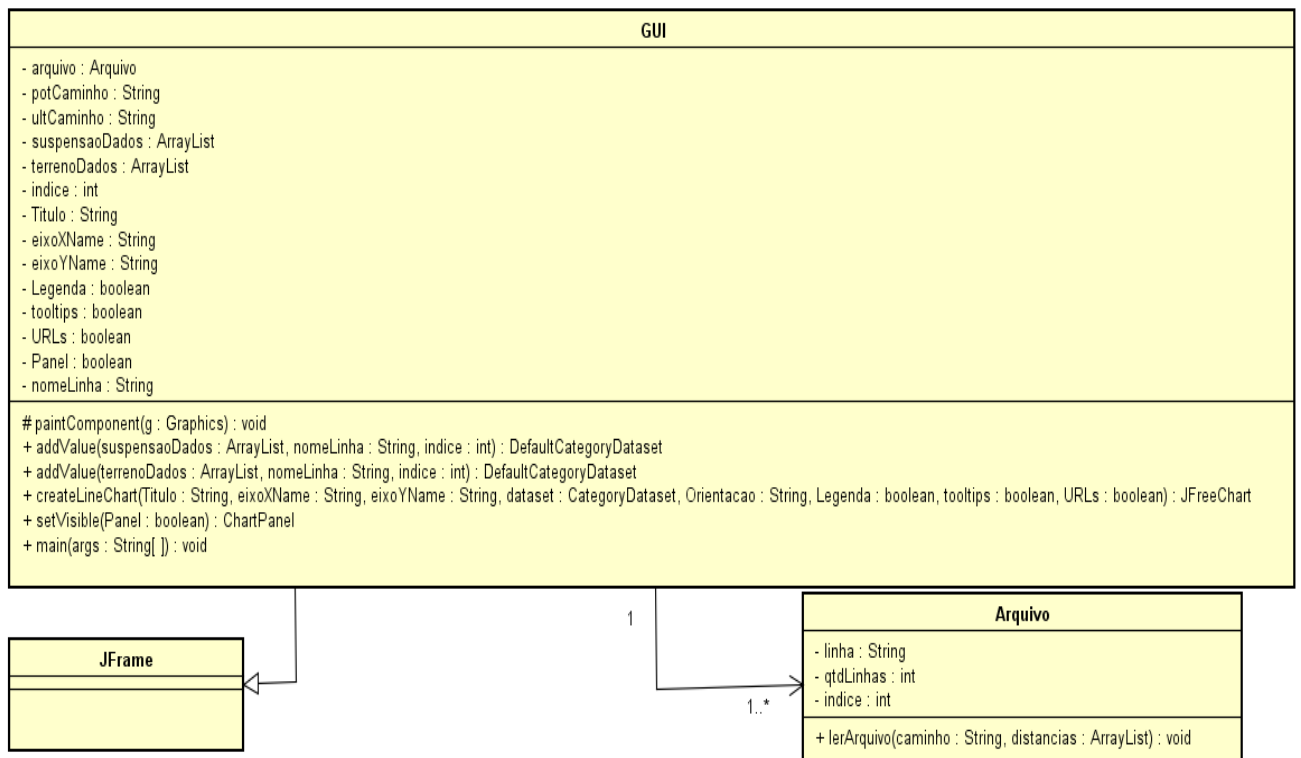


Figura 17 - Diagrama de Classes (Aplicação Gráfica)