

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

APLICATIVO PARA ESTABILIZAÇÃO DE TELA EM
DISPOSITIVOS ANDROID

ADRIAN VOLKMANN

BLUMENAU
2017

ADRIAN VOLKMANN

**APLICATIVO PARA ESTABILIZAÇÃO DE TELA EM
DISPOSITIVOS ANDROID**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Mauro Marcelo Mattos, Doutor – Orientador

**BLUMENAU
2017**

APLICATIVO PARA ESTABILIZAÇÃO DE TELA EM DISPOSITIVOS ANDROID

Por

ADRIAN VOLKMANN

Trabalho de Conclusão de Curso aprovado
para obtenção dos créditos na disciplina de
Trabalho de Conclusão de Curso II pela banca
examinadora formada por:

Presidente: _____
Prof. Mauro Marcelo Mattos, Doutor – Orientador, FURB

Membro: _____
Prof. Francisco Adell Péricas, Mestre – FURB

Membro: _____
Prof. Miguel Alexandre Wisintainer, Mestre – FURB

Blumenau, 4 de julho de 2017

Dedico este trabalho a minha família que sempre me apoiou e incentivou nos estudos.

AGRADECIMENTOS

À minha família que sempre me apoiou e incentivou para concluir esta etapa.

Aos meus amigos que me ajudaram na execução deste trabalho.

Ao meu orientador Mauro Marcelo Mattos, por todo apoio prestado durante o desenvolvimento deste trabalho.

A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.

Arthur Schopenhauer

RESUMO

Este trabalho apresenta o projeto de um protótipo de um aplicativo que realiza a estabilização de um texto na tela de dispositivos móveis conforme a movimentação do mesmo. O aplicativo tem como objetivo melhorar a legibilidade do texto em situações em que ocorrem trepidações, como por exemplo quando um usuário está caminhando ou está em um carro em movimento. A detecção dos movimentos é feita utilizando o acelerômetro do smartphone. Os dados do acelerômetro são filtrados usando um filtro passa-baixa e após isso os dados são utilizados no algoritmo do sistema massa-mola-amortecedor. Então os resultados deste algoritmo são utilizados como coordenadas para atualizar a posição do texto na tela do smartphone. Como resultados obteve-se um protótipo que permite realizar ações de estabilização até um certo limiar pois em situações de grande trepidação ainda não é possível realizar a estabilização adequadamente.

Palavras-chave: Acelerômetro. Estabilização de tela. Leitura em movimento. Dispositivos móveis.

ABSTRACT

This work presents a prototype design of an application that accomplishes text stabilization on mobile devices screen according your movement. The application aims to improve the text readability in situations where trepidation occurs, such as when a user is walking or in a moving car. Motion detection is done using the smartphone's accelerometer. The accelerometer data is filtered using a low-pass filter and after that the data is used in the mass-spring-damper algorithm. Then the algorithm results are used as coordinates to update the text position on the smartphone screen. As results, we obtained a prototype that allows stabilization actions up to a certain threshold because in great trepidation situations it is still not possible to carry out the stabilization properly.

Key-words: Accelerometer. Screen stabilization. Reading on the move. Smartphone.

LISTA DE FIGURAS

Figura 1 - O labirinto Vestibular	15
Figura 2 - Acelerômetro conceitual	16
Figura 3 - Acelerômetro em um iPhone 3	17
Figura 4 – Circuito de um acelerômetro	18
Figura 5 - Medição capacitância.....	18
Figura 6 - Sistema de coordenadas Android.....	19
Figura 7 - Filtro passa-baixa.....	21
Figura 8 - Resultados filtro passa-baixa	22
Figura 9 - Sistema massa-mola-amortecedor	23
Figura 10 - Classificação amortecimento	24
Figura 11 - Câmera montada na cabeça do usuário de testes	26
Figura 12 - Modelo físico massa-mola-amortecedor.....	27
Figura 13 - Implementação do sistema NoShake	28
Figura 14 - Texto estacionário e estabilizado	30
Figura 15 - Diagrama de casos de uso	33
Figura 16 - Diagrama de classes.....	34
Figura 17 - Diagrama de atividades.....	35
Figura 18 - Dados brutos e filtro passa baixa	37
Figura 19 - Tela inicial	39
Figura 20 - Exibição Texto.....	40

LISTA DE QUADROS

Quadro 1 – Implementação acelerômetro.....	19
Quadro 2 - Taxas de leitura do acelerômetro	20
Quadro 3 - Aquisição dados acelerômetro	20
Quadro 4 - Implementação filtro passa -baixa.....	21
Quadro 5 - Fórmula coeficiente do filtro.....	21
Quadro 6 - Lei de Hooke	23
Quadro 7 - Fator de amortecimento.....	24
Quadro 8 - Fórmula coordenadas	25
Quadro 9 - Comparação trabalhos correlatos	31
Quadro 10 - Implementação acelerômetro	36
Quadro 11 - Implementação filtro passa baixa.....	36
Quadro 12 - Algoritmo amortecedor	37
Quadro 13 - Algoritmo estabilização.....	38
Quadro 14 - Comparação trabalhos	42

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

PID - Proporcional Integral Derivativo

RF – Requisito Funcional

RNF – Requisito não Funcional

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS.....	12
1.2 ESTRUTURA.....	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 LEITURA EM MOVIMENTO	14
2.2 ACELERÔMETRO.....	16
2.3 FILTRO PASSA-BAIXA.....	20
2.4 SISTEMA MASSA-MOLA-AMORTECEDOR	23
2.5 TRABALHOS CORRELATOS	25
2.5.1 Walking with your Smartphone: Stabilizing Screen Content	25
2.5.2 NoShake: Content Stabilization for Shaking Screens of Mobile Devices	27
2.5.3 Comparativo entre os trabalhos correlatos	31
3 DESENVOLVIMENTO	32
3.1 REQUISITOS.....	32
3.2 ESPECIFICAÇÃO	32
3.2.1 Diagrama de casos de uso	32
3.2.2 Diagrama de classes	33
3.2.3 Diagrama de atividades	34
3.3 IMPLEMENTAÇÃO	35
3.3.1 Técnicas e ferramentas utilizadas.....	35
3.3.2 Operacionalidade da implementação	38
3.4 ANÁLISE DOS RESULTADOS	40
4 CONCLUSÕES	43
4.1 EXTENSÕES	43
REFERÊNCIAS	44

1 INTRODUÇÃO

Atualmente os dispositivos móveis estão presentes na vida da maioria das pessoas. Segundo Schäfer (2009) “Nos dias de hoje, encontrar um adolescente que não tenha um celular é tão improvável quanto achar um menino de 13 anos que seja fã de ópera ou uma menina de 15 que não se preocupe com a aparência”.

Os dispositivos móveis possuem grande participação no mercado mundial. Conforme Corpbusiness (2015), o mercado de aplicativos movimentou US\$27 bilhões em 2013 e pode atingir US\$ 77 bilhões em 2017. Além disso, conforme Computerworld (2016) estima-se que 349 milhões de dispositivos móveis foram comercializados no primeiro trimestre de 2016.

A característica de mobilidade traz como consequência que o uso do equipamento em movimento geralmente provoca desconforto ao usuário em função da trepidação natural causada pelo deslocamento.

Quando você consegue ler no ônibus ou carro, basicamente você manda duas mensagens distintas para o seu cérebro. Do ponto de vista da leitura você está parado, já que sua visão está concentrada em um objeto plenamente estático e você está parado, normalmente sentado. Mas graças à sua visão periférica o seu cérebro entende que você está em movimento. A própria sensação de trepidação natural do veículo também leva o seu cérebro a entender que, na verdade, você está se movimentando. Como é mais difícil focar os estímulos cerebrais em uma só tarefa também há mais exigências para a visão. Como resultado, você faz um esforço maior para ler o que está escrito, o que leva à fadiga ocular e frequentemente a dores de cabeça. (HOSPITAL DE OLHOS DE SÃO PAULO, 2016, p1).

Neste trabalho propõe-se o desenvolvimento de um aplicativo para dispositivos móveis que permite a estabilização dos movimentos da tela do dispositivo no sentido de minimizar o desconforto de usuários em deslocamento. Durante o levantamento bibliográfico preliminar foi possível identificar na literatura proposta de soluções para este problema. Diante deste contexto pretende-se desenvolver um aplicativo utilizando algum dos algoritmos propostos que permita validar a eficácia da solução.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um protótipo de um aplicativo para dispositivos móveis baseados em Android que permita a estabilização do conteúdo da tela conforme os movimentos do usuário.

Os objetivos específicos são:

- a) identificar o grau de trepidação do dispositivo;
- b) realizar o cálculo da correção necessária para minimizar o efeito da trepidação;
- c) aplicar a correção na tela do smartphone.

1.2 ESTRUTURA

Este trabalho está estruturado em quatro capítulos. O primeiro capítulo apresenta a introdução do trabalho e seus objetivos. O segundo capítulo apresenta a fundamentação teórica sobre os efeitos da leitura em movimento, acelerômetro, filtro passa-baixa e o sistema massa-mola-amortecedor além da apresentação dos trabalhos correlatos. O terceiro capítulo apresenta os requisitos, a especificação em forma de diagramas, detalhamento da implementação e os resultados obtidos. Por fim, no quarto capítulo são apresentadas as conclusões e as sugestões de extensões do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

A seção 2.1 expõe os efeitos quando um usuário realiza uma leitura enquanto estando em movimento. Na seção 2.2 é apresentado o funcionamento do sensor acelerômetro. Na seção 2.3 é descrito o filtro passa-baixa. Na seção 2.4 é apresentado o sistema massa-mola-amortecedor e suas principais características. Por fim, na seção 2.5 são apresentados os trabalhos correlatos.

2.1 LEITURA EM MOVIMENTO

Muitas pessoas possuem o hábito de ler no trânsito enquanto estão se deslocando para o trabalho ou escola, o que pode causar um mal-estar para algumas pessoas. Segundo Lottenberg e Mattos (2011), “Ler no ônibus ou no carro exige mais esforço e, por isso, pode ser cansativo ou dar dor de cabeça. Contudo, o hábito não traz nenhum problema para a visão.”.

Basicamente, essa tontura vem do fato de que quem lê no ônibus está mandando duas mensagens conflitantes para o sistema nervoso. Do ponto de vista central, concentrada no texto à sua frente, você está parado. Mas a visão periférica capta a paisagem correndo ao seu lado e manda avisar que você está em movimento. “Isso gera uma confusão entre o cérebro e a visão que pode causar um mal-estar”, explica Paulo Mello Filho, membro do Conselho Brasileiro de Oftalmologia. (SUPER INTERRESANTE, 2010, p1).

Conforme Campbell (1957), é importante notar que nem todas as agitações são as mesmas. Considerando que existe uma linha direta que liga o olho e o objeto em que está focado, ou a linha de visão, o olho humano tem uma profundidade de campo relativamente grande, pelo menos $\pm 0,3$ dioptrias¹ a um diâmetro da pupila de 3 mm, o que corresponde a mais de 5 cm a uma distância de visão de 30 cm.

Para que uma pessoa possa estimar sua localização, o cérebro combina informações de uma variedade de fontes, como visão, o tato, a posição da articulação, o ouvido interno e suas próprias expectativas. O ouvido interno é particularmente importante porque contém sensores tanto para o movimento angular (os canais semicirculares) como para o movimento linear (os otólitos). Estes sensores são chamados de sistema vestibular (HAIN; OMAN, 2010, tradução nossa). A Figura 1 apresenta a anatomia do sistema vestibular localizado no ouvido interno.

Conforme Meldau (2017), o sistema vestibular tem a função da manutenção do equilíbrio. Segundo a autora o “sistema é composto pelos três canais semicirculares que se

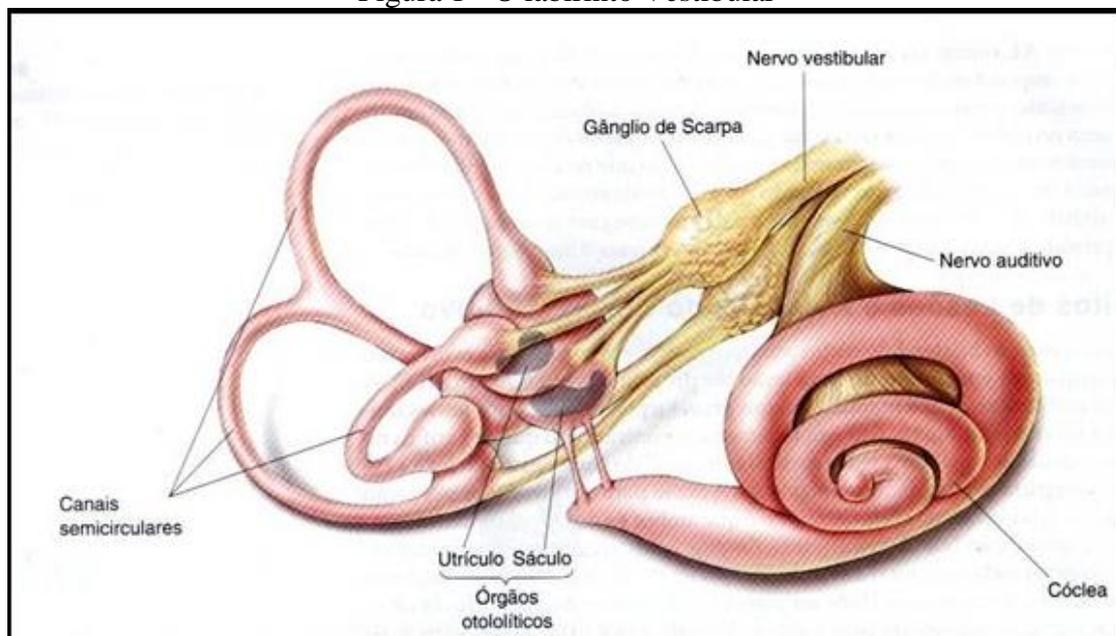
¹ Dioptria: “A Dioptria é uma unidade de medida que se refere ao poder de refração das lentes em um sistema óptico, ou seja, o famoso “grau” dos óculos popularmente conhecido. Assim temos que 1 grau é igual a 1 dioptria”. (INFOESCOLA, 2017).

fundem numa região central denominada vestíbulo, que apresenta duas outras estruturas vesiculares: o sáculo e utrículo, também conhecidos como órgãos otolíticos.”.

O utrículo apresenta maiores dimensões quando comparado ao sáculo, localizando-se na região superior do vestíbulo. Existe uma diminuta área de revestimento do sáculo e do utrículo que se diferencia em órgão sensorial, conhecida como mácula. A mácula localizada no utrículo recebe o nome de lápilus, e a do sáculo de rágata. A primeira encontra-se na porção inferior do utrículo, quase em posição horizontal, apresentando importante função na orientação da cabeça quando o indivíduo se encontra ereto, enquanto que a rágata localiza-se verticalmente apresentando papel importante no equilíbrio quando o indivíduo está deitado (em decúbito). (MELDAU, 2017, p1).

Meldau (2017) explica que o labirinto, composto da cóclea que é ligado ao sáculo através dos canalículos de Hensen, é dividido em labirinto ósseo e labirinto membranoso. Entre eles é encontrado um líquido chamado perilinfa que faz a função de amortecimento das vibrações ósseas. Ainda segundo a autora, no interior das estruturas membranosas encontra-se a endolinfa onde, em situações de movimentação, a mesma caminha em direção oposta ao movimento da cabeça. Quando ocorre uma parada abrupta da rotação a endolinfa continua em movimento, neste momento o disparo das células ciliares é cessado e conforme Meldau (2017) “ há a transmissão, pelos canais semicirculares, de um sinal de polaridade quando a cabeça começa a girar e de polaridade contrária quando esse movimento é interrompido.”.

Figura 1 - O labirinto Vestibular



Fonte: Bear, Connors e Paradiso (2002).

Hain e Oman (2010) exemplificam a situação da seguinte forma: considerando uma situação onde alguém está lendo no banco de trás de um carro. Nesta situação, os olhos da pessoa estão fixos no conteúdo sendo lido, a visão periférica desta pessoa percebe o interior do carro indicando que a mesma está imóvel, mas o sistema vestibular discorda pois registra

os solavancos, o ruído do deslocamento e outras percepções de movimento. Esta situação de ambiguidade pode provocar sensações de mal-estar em algumas pessoas. Um exemplo apresentado por Hain e Oman (2010) e que caracteriza bem a situação ocorre com frequência em barcos. O mal-estar geralmente acontece quando a pessoa está abaixo dos *decks* e não tem visão para fora do barco. Esta mesma pessoa, ao ficar no convés geralmente possui uma boa visão do horizonte e, nesta situação, a pessoa não se sente tão mal. Segundo os autores quando o sistema de equilíbrio aprende a antecipar os movimentos do barco a suscetibilidade para o mal-estar desaparece. Quando a pessoa desembarca ela pode sentir seu corpo antecipando os movimentos do barco por algumas horas, e isto pode causar o mal-estar nas pessoas mesmo elas estando paradas em terra firme.

Estes efeitos também ocorrem em astronautas pois, conforme Hain e Oman (2010), os mesmos estão acostumados a vida inteira com o efeito da gravidade na terra, e repentinamente a falta da gravidade cria muitos conflitos sensoriais. Quando os astronautas movimentam a cabeça os sensores do sistema vestibular não acompanham mais estes movimentos, o sistema vestibular não consegue localizar a noção de “para baixo”, pois sem a gravidade não existe mais esta orientação. Os autores citam que alguns astronautas se sentem sempre como estando de cabeça para baixo, não importando a posição que estejam.

2.2 ACELERÔMETRO

Segundo Laboratório Imobilis (2013) “um acelerômetro é um dispositivo eletromecânico que mede as forças de aceleração. Estas forças podem ser estáticas, como força da gravidade ou dinâmicas, causadas pela movimentação ou vibração do acelerômetro”. Na Figura 2 é exibido um acelerômetro conceitual o qual é composto de uma mola fixa em uma base que sustenta um peso, a mola esta calibrada para marcar o equivalente a força da gravidade quando estiver em repouso (LABORATÓRIO IMOBILIS, 2013).

Figura 2 - Acelerômetro conceitual



Fonte: Laboratório Imobilis (2013).

Ao movimentar todo o sistema o peso se movimenta fazendo com que a mola se comprima ou expanda. Desta forma é possível medir a força elástica aplicada na mola e assim finalmente retornar o valor da aceleração que, segundo Kircher (2017a), é medida em m/s^2 . A Figura 3 apresenta o modelo de um acelerômetro presente em um Apple iPhone 3.

Figura 3 - Acelerômetro em um iPhone 3

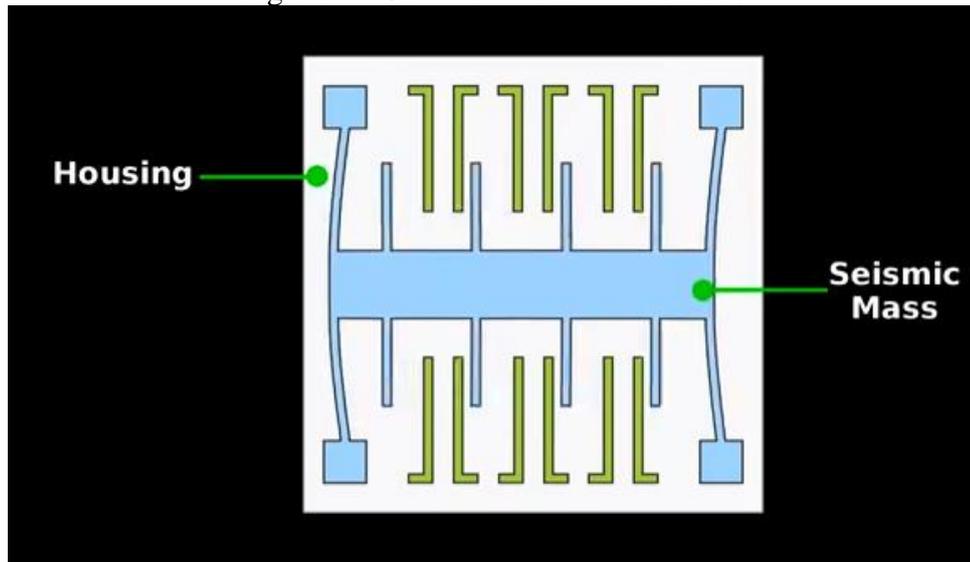


Fonte: Laboratório Imobilis (2013).

Segundo Laboratório Imobilis (2013) existem vários tipos de acelerômetros: aqueles baseados em efeito piezelétrico e aqueles baseados em medir variações na capacitância. Nos sensores de efeito piezelétrico, existem estruturas microscópicas de cristal que são afetadas pelas forças e movimentos que por sua vez geram uma tensão elétrica proporcional a agitação. Os acelerômetros capacitivos funcionam medindo a capacitância entre microestruturas eletrônicas próximas umas das outras.

Na Figura 4 é apresentado um circuito o qual é composto de um elemento móvel, o Seismic Mass, que vibra sobre a base (Housing) em caso de movimentação. Em cada vibração, as aletas do Seismic Mass se aproximam dos capacitores, alterando a capacitância. O circuito mede o diferencial da tensão e retorna o valor proporcional a movimentação na saída do acelerômetro (LABORATÓRIO IMOBILIS, 2013).

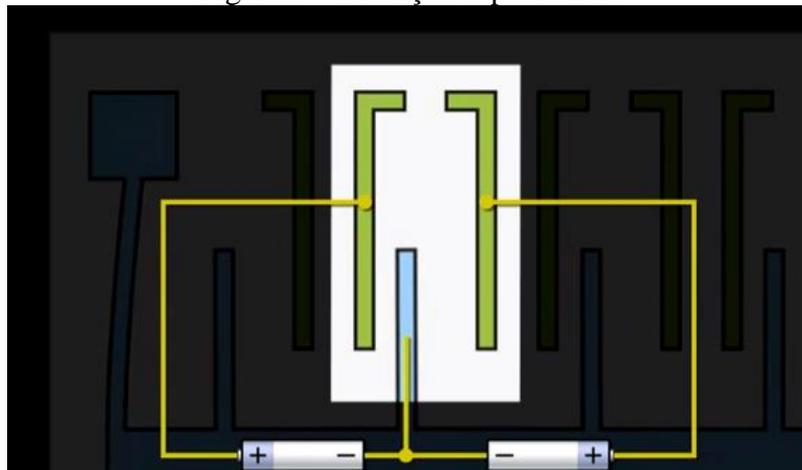
Figura 4 – Circuito de um acelerômetro



Fonte: Laboratório Imobilis (2013).

Na Figura 5 é possível verificar o modelo de medição.

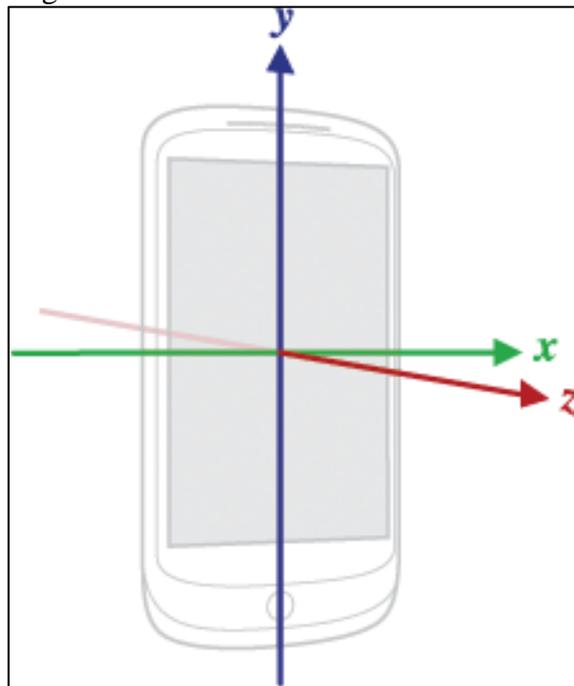
Figura 5 - Medição capacitância



Fonte: Laboratório Imobilis (2013).

Conforme Android Developers (2017a), o sensor acelerômetro do Android usa um sistema de coordenadas de três eixos que é definido em relação a tela do dispositivo quando o mesmo é mantido em sua orientação padrão (Figura 6). Conforme Android Developers (2017b), para medir a aceleração real do dispositivo é necessário eliminar a força da gravidade. A API do Android possui o Acelerômetro linear que retorna os valores da aceleração já excluindo o valor da gravidade (ANDROID DEVELOPERS, 2017a).

Figura 6 - Sistema de coordenadas Android



Fonte: Android Developers (2017a).

O trecho de código abaixo demonstra como utilizar o recurso de acelerômetro em um programa em Java (Quadro 1) na plataforma Android.

Quadro 1 – Implementação acelerômetro

```

1 public class Main extends Activity implements SensorEventListener {
2     private SensorManager senSensorManager;
3     private Sensor senAccelerometer;
4
5     /** Called when the activity is first created. */
6     @Override
7     public void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.main);
10        senSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
11        senAccelerometer = senSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
12        senSensorManager.registerListener(this, senAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);
13    }
14
15    @Override
16    public void onSensorChanged(SensorEvent sensorEvent) {
17    }
18
19
20    @Override
21    public void onAccuracyChanged(Sensor sensor, int accuracy) {
22    }
23
24 }

```

Fonte: Govender (2014).

Conforme Govender (2014) a classe que utiliza o sensor deve implementar a interface `SensorEventListener` (linha 1), isto implica em implementar os métodos `onSensorChanged` e `onAccuracyChanged`. O método `onSensorChanged` é invocado toda vez que o sensor detecta alteração no valor. Ainda segundo o autor, deve ser declarado as variáveis `senSensorManager` e `senAccelerometer` e inicializadas no método `onCreate`,

que é executado quando a classe é criada. O `senSensorManager` é usado para acessar os sensores do sistema e o `senAccelerometer` é usado para referenciar o acelerômetro. Em seguida (linha 12), o acelerômetro é registrado para fazer as leituras. O autor conclui que este método possui três parâmetros que são respectivamente o contexto da Activity, o sensor, e a taxa de leitura do sensor.

Segundo Android Developers (2017c) existem quatro taxas de leituras distintas que são adequadas para usar em situações diferentes, estas taxas são apresentadas no Quadro 2.

Quadro 2 - Taxas de leitura do acelerômetro

Taxa de Leitura	Recomendação de uso
SENSOR_DELAY_FASTEST	Leitura mais rápido possível
SENSOR_DELAY_GAME	Adequada para jogos
SENSOR_DELAY_NORMAL	Orientação da tela
SENSOR_DELAY_UI	Interface de usuário

Fonte: elaborado pelo autor.

No Quadro 3 é apresentado um exemplo de implementação do método `onSensorChanged` onde na linha 4 onde é feita uma verificação se os valores são referentes ao sensor correto. O próximo passo é obter o valor de cada eixo, onde os valores são passados através da variável `sensorEvent` que possui um vetor com estes dados (GOVENDER, 2014). Os valores dos eixos X, Y e Z são obtidos respectivamente nas linhas 5, 6 e 7.

Quadro 3 - Aquisição dados acelerômetro

```

1  public void onSensorChange(SensorEvent sensorEvent) {
2      Sensor mySensor = sensorEvent.sensor;
3
4      if (mySensor.getType() == Sensor.TYPE_ACCELEROMETER) {
5          float x = sensorEvent.values[0];
6          float y = sensorEvent.values[1];
7          float z = sensorEvent.values[2];
8      }
9  }

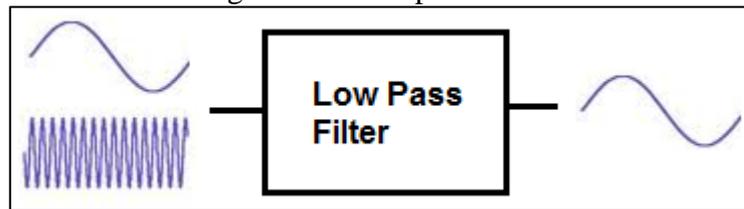
```

Fonte: Govender (2014).

2.3 FILTRO PASSA-BAIXA

Kircher (2017b) explica que um filtro passa-baixa é um tipo de filtro eletrônico que deixa passar sinais de baixa frequência sem alterá-los enquanto reduz a amplitude dos sinais com a frequência acima da frequência de corte. Segundo Learning About Electronics (2014), os sinais de baixa frequência passam muito mais fácil pelo filtro enquanto os sinais de alta frequência passam com muito mais dificuldade. É possível verificar o modelo deste sistema na Figura 7.

Figura 7 - Filtro passa-baixa



Learning About Electronics (2014).

Kircher (2017b) ainda comenta que os filtros passa-baixa podem ser usados para várias aplicações como por exemplo, processamento de áudio, processamento de imagens, processamento de sinais e filtros digitais.

No Quadro 4, Kircher (2017b) apresenta uma implementação utilizando a linguagem de programação Java do filtro passa-baixa baseado no algoritmo da Wikipedia (2017). Neste algoritmo o coeficiente do filtro, que segundo o autor determina o peso que é aplicado no filtro, é definido pela variável alpha a qual, conforme o autor, deve possuir valor entre 0 e 1 ($0 < \alpha < 1$). Segundo o autor o valor de alpha pode ser obtido através da fórmula presente no Quadro 5, onde RC é a constante de tempo e dt é a taxa de amostragem.

Quadro 4 - Implementação filtro passa -baixa

```

1  public class LowPassFilter
2  {
3      private float alpha;
4      private float[] acceleration;
5
6      public float[] lowPassFilter(float[] input)
7      {
8          // Update the Wikipedia filter
9          // y[i] = y[i] + alpha * (x[i] - y[i])
10         acceleration[0] = acceleration[0] + alpha
11             * (input[0] - acceleration[0]);
12         acceleration[1] = acceleration[1] + alpha
13             * (input[1] - acceleration[1]);
14         acceleration[2] = acceleration[2] + alpha
15             * (input[2] - acceleration[2]);
16
17         return acceleration;
18     }
19 }

```

Fonte: Kircher (2017b).

Quadro 5 - Fórmula coeficiente do filtro

$$\alpha := dt / (RC + dt)$$

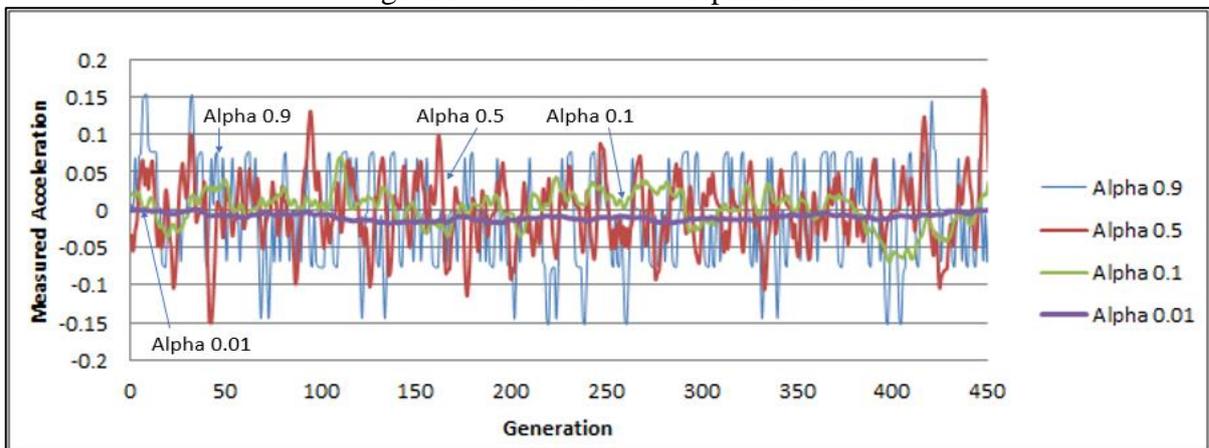
Fonte: Kircher (2017b).

Kircher (2017b) demonstra a utilização de um filtro passa-baixa para filtrar sinais digitais baseando-se no domínio do tempo. Com este filtro o autor está preocupado com a

amplitude do sinal ao longo do tempo, querendo deixar sinais com período de tempo longos passar pelo filtro enquanto elimina sinais que ocorrem em pequenos períodos de tempo.

Neste contexto, na Figura 8 são apresentados os resultados de testes feitos por Kircher (2017b) para ajustar o valor de α em cada teste. Kircher (2017b) comenta que o algoritmo usa o valor anterior e então adiciona a diferença ponderada entre o novo valor e o valor anterior. Ainda segundo o autor, quando o valor de α é muito pequeno faz com que o filtro registre as mudanças muito lentamente, enquanto o valor de α for próximo de 1 registra as mudanças muito rapidamente. Ainda conforme Kircher (2017b), é possível verificar que os efeitos de suavização ocorrem quando o α é menor que 0.1, o autor ainda conclui que quanto menor o valor de α maior será a suavização.

Figura 8 - Resultados filtro passa-baixa



Fonte: Kircher (2017b).

Kircher (2017b) ainda comenta um filtro com α de valor 0.5 não seria uma boa escolha. O autor demonstra utilizando a fórmula $T = dt * ((1-\alpha)/\alpha)$, com o valor 0.5 para α se obtém $T = dt$. Um acelerômetro fazendo leituras a 100Hz traria um $dt=0.01$. Desta forma ele conclui que qualquer coisa que exceda 0,01 de um segundo passaria pelo filtro, ou seja, quase tudo passaria pelo filtro.

Como o objetivo de Kircher (2017b) era bloquear sinais que ocorrem em pequenos períodos de tempo, um α com valor 0.5 faz apenas o bloqueio de sinais abaixo de 0,01 segundos fazendo com que o objetivo não seja alcançado.

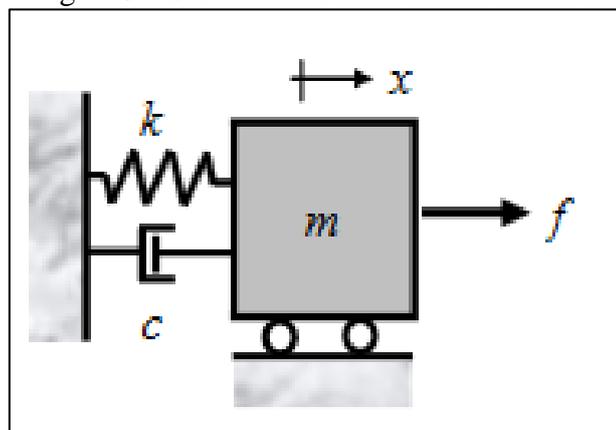
Ainda segundo Kircher (2017b) se colocar o valor de 0.1 em α e obtendo valores a uma taxa de 50Hz, obtém-se uma constante de tempo de 0.18 segundos, que seria um tempo bem mais realista. O autor comenta que os valores maiores que este tempo passam pelo filtro enquanto os valores menores são filtrados. A medida que o α se torna menor o filtro aumenta o efeito da suavização, mas também muda a frequência medida do sensor tornando-a mais lenta. Para concluir, o autor explica que diferentes dispositivos Android tem diferentes

sensores acelerômetro com frequências diferentes e que o filtro passa-baixa não terá o mesmo comportamento em todos os dispositivos.

2.4 SISTEMA MASSA-MOLA-AMORTECEDOR

Mucheroni (2012) demonstra o sistema massa-mola-amortecedor (Figura 9) o qual é composto por uma massa m , uma mola com coeficiente de rigidez k , um amortecedor com coeficiente de amortecimento c . f é a força externa aplicada na massa m na direção do movimento e x é o deslocamento da massa m na direção do movimento.

Figura 9 - Sistema massa-mola-amortecedor



Fonte: Mucheroni (2012).

Silva (2017) define uma mola como “uma peça que possui flexibilidade elástica relativamente alta, isto é, que apresenta grandes deformações quando solicitada.”. O autor ainda acrescenta “a mola opõe-se à força que a ela está aplicada, armazenando energia potencial elástica.” Khan Academy (2017) define a mola como “um objeto que pode ser deformado por uma força e que volta a sua forma original quando essa força é removida.”

Conforme Khan Academy (2017), o físico Robert Hooke notou que, dentro de certos limites, a força requerida para deformar um objeto elástico era diretamente proporcional a deformação da mola. Este comportamento das molas é definido como a Lei de Hooke, a qual é apresentada no Quadro 6, onde F é a força, x é a deformação e k é a constante de elasticidade.

Quadro 6 - Lei de Hooke

$$F = -kx$$

Fonte: Khan Academy (2017).

O amortecedor é descrito por Silva (2017) como “o componente do sistema mecânico que dissipa energia mecânica do mesmo”. Por sua vez, Silva (2017) explica que amortecimento é “processo pelo qual a energia é retirada do sistema elástico. A energia é

consumida por atrito entre as peças móveis do sistema e/ou pelo atrito interno entre as moléculas das peças do sistema, havendo uma dissipação de energia mecânica sob forma de calor e/ou som. ”

Conforme Silva (2017) o amortecimento pode ser classificado em três tipos que são amortecimento viscoso, amortecimento seco e amortecimento estrutural. O Amortecimento viscoso é o que mais ocorre na prática. O mesmo acontece entre uma peça e um fluido viscoso, como por exemplo um óleo lubrificante, interposto entre as partes móveis de um sistema mecânico. O amortecimento seco ocorre com a ausência de um fluido, onde os sólidos do sistema se atritam entre si sem nenhum tipo de lubrificação. O amortecimento estrutural é quando a energia é dissipada pelo próprio material, como exemplo uma prensa mecânica onde logo após a pancada do martelo a energia é dissipada pela estrutura molecular da máquina.

O fator de amortecimento, conforme Mucheroni (2012), é exposto no Quadro 7, onde $\zeta < 1$ para representa sistemas sub-amortecidos, $\zeta = 1$ para sistemas criticamente amortecidos e $\zeta > 1$ para sistemas sobre amortecidos.

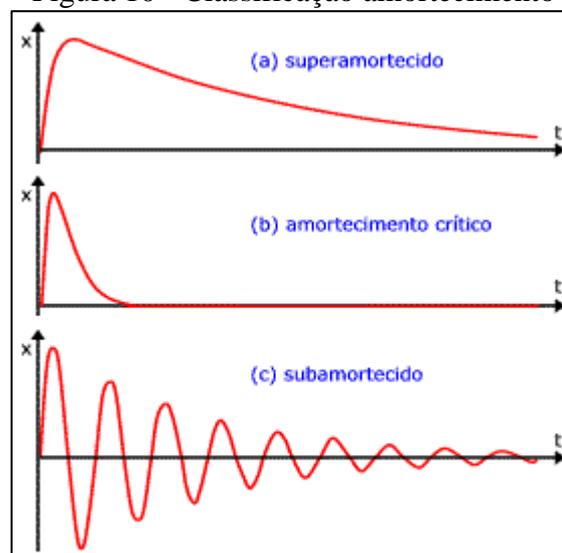
Quadro 7 - Fator de amortecimento

$$\zeta = \frac{c}{2\sqrt{km}}$$

Fonte: Mucheroni (2012).

Segundo MSPC (2017) o sistema super-amortecido não é oscilante, o amortecimento crítico também não é oscilante e chega ao ponto de equilíbrio no menor tempo possível e no sistema sub-amortecido ocorre oscilação, mas sua amplitude decai com o tempo. É possível verificar este o comportamento na Figura 10.

Figura 10 - Classificação amortecimento



Fonte: MSPC (2017).

Lungu (2011) demonstra que utilizando a fórmula do Quadro 8 é possível obter as coordenadas do sistema massa-mola-amortecedor utilizando duas coordenadas passadas x_{-1} e x_{-2} e os valores m da massa, k coeficiente de rigidez da mola, d_r coeficiente de amortecimento do amortecedor e dt que é o intervalo de tempo.

Quadro 8 - Fórmula coordenadas

$$x_0 = x_{-1} \cdot \left(1 - \frac{k}{m} \cdot dt^2\right) + (x_{-1} - x_{-2}) \cdot \left(1 - 2 \cdot DR \cdot \sqrt{\frac{k}{m}} \cdot dt\right)$$

Fonte: Lungu (2011).

2.5 TRABALHOS CORRELATOS

A seguir serão apresentados dois trabalhos correlatos. O trabalho de Jeisy (2014) apresenta uma ferramenta para estabilizar o conteúdo da tela de um smartphone de uma pessoa que está caminhando. Já o trabalho de Rahmati, Shepard e Zhong (2009) apresenta o sistema NoShake para estabilização de conteúdo em uma tela utilizando o acelerômetro. Estes trabalhos são melhor detalhados nas seções que seguem.

2.5.1 Walking with your Smartphone: Stabilizing Screen Content

Jeisy (2014) apresenta um estudo preliminar para identificar alternativas para estabilizar a tela de um dispositivo móvel que está sendo usado por uma pessoa que está caminhando. O propósito da estabilização é aumentar a legibilidade da pessoa que está segurando o dispositivo.

Na pesquisa, foram usadas quatro abordagens diferentes de estabilização: retorno direto, estimativa de posição, estimativa de posição melhorada com a ajuda de um controlador proporcional integral derivativo (PID) e usando modelo oculto de Markov.

A abordagem do retorno direto consiste em fazer o movimento contrário e de igual intensidade ao do movimento do dispositivo. Jeisy (2014) verificou que este tipo de estabilização não funciona pois a curva dos movimentos não se aproxima de uma curva senoidal. Com esta abordagem a estabilização só funcionaria se o usuário movesse o dispositivo conforme uma onda senoidal. Constatou também que, para movimentos diferentes, a solução apresentava atrasos na estabilização. Assim Jeisy (2014) concluiu que é necessário usar um histórico de valores para obter uma boa estabilização.

Usando a abordagem de posição estimada, Jeisy (2014) tentou estabilizar o movimento calculando continuamente os deslocamentos com base em um ponto de referência. Neste caso são armazenados alguns valores de aceleração e então é usado um algoritmo que calcula a velocidade e a distância absoluta. Jeisy (2014) concluiu que os sensores não são

suficientemente precisos para utilização em tempo prolongado e o fato de usar dois dados (aceleração e distância) para calcular os movimentos gerou a necessidade de fazer a calibração no início da utilização e fazer também as calibrações durante o uso.

A abordagem de controlador PID tenta manter a abordagem de estimativa de posição, mas utiliza um controlador PID para evitar a propagação dos erros de leitura dos sensores. Ao ajustar os parâmetros do PID, Jeisy (2014) observou que esta abordagem não era apropriada para o propósito do projeto, pois estava estabilizando o movimento absoluto do dispositivo ao invés de estabilizar o movimento em relação aos olhos do usuário. Concluiu que seria necessária uma abordagem onde os valores de entrada e os valores de saída não estivessem diretamente conectados.

Para finalizar, Jeisy (2014) testou a abordagem de modelo oculto de Markov. Para trabalhar com este modelo, precisou encontrar uma relação entre os dados de aceleração e o movimento relativo entre o dispositivo e os olhos do usuário. Para isso, Jeisy (2014) montou uma câmera na cabeça do usuário de testes (Figura 11) e gravou os movimentos produzidos durante o uso.

Figura 11 - Câmera montada na cabeça do usuário de testes



Fonte: Jeisy (2014).

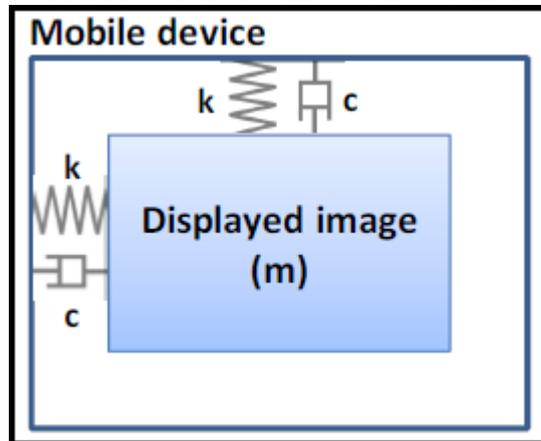
Analisando os quadros do vídeo gravado foi possível medir o deslocamento e com estes dados foi possível criar o modelo oculto de Markov. Jeisy (2014) utilizou o software MATLAB para testar se o modelo estava funcionando, mas não chegou a criar uma aplicação para testar a solução devido a limitações de tempo. Contudo, demonstrou em teoria que o modelo funciona.

2.5.2 NoShake: Content Stabilization for Shaking Screens of Mobile Devices

No trabalho de Rahmati, Shepard e Zhong (2009) é apresentado um sistema para estabilização de conteúdo chamado NoShake que utiliza o acelerômetro para fazer as leituras dos movimentos. A solução se baseia em compensar o movimento do dispositivo mudando o conteúdo da tela na direção oposta ao do movimento.

O NoShake foi inspirado em um modelo físico, demonstrado na Figura 12, onde a tela representa uma massa que está suspensa por um sistema de mola e amortecedor viscoso independentes em cada direção. A mola permite que a tela se mova em direção oposta ao movimento e os amortecedores são usados para evitar oscilação da massa.

Figura 12 - Modelo físico massa-mola-amortecedor

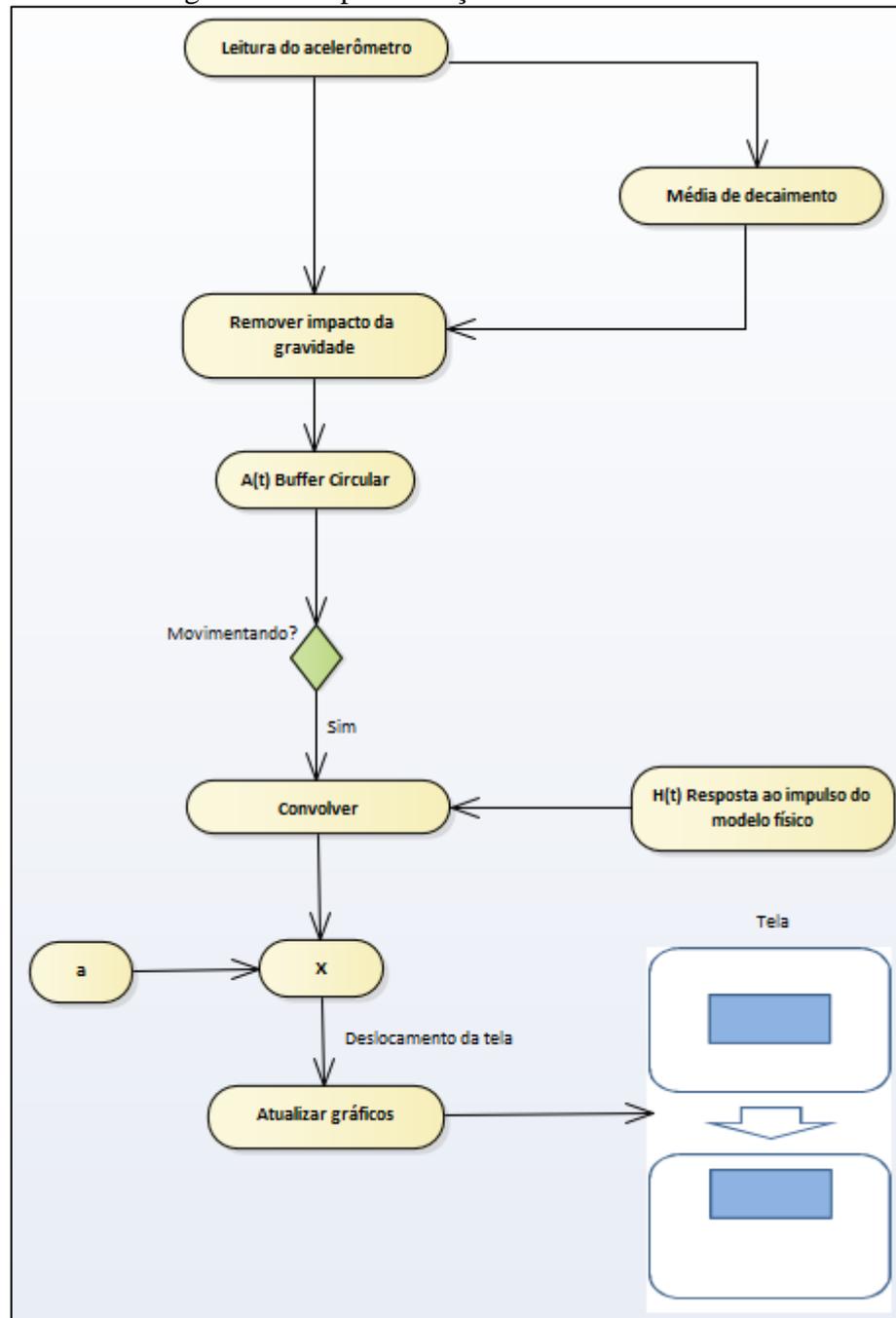


Fonte: Rahmati, Shepard e Zhong (2009).

Através de uma pesquisa com usuários confirmou-se que o software pode melhorar a experiência do usuário em um ambiente com movimentações. O NoShake pode ter um impacto maior para usuários que não conseguem segurar um dispositivo parado, como pessoas idosas ou pessoas com a doença de Parkinson (Rahmati, Shepard e Zhong, 2009, tradução nossa).

O diagrama de atividades da Figura 13 apresenta o fluxo de funcionamento do NoShake:

Figura 13 - Implementação do sistema NoShake



Fonte: Rahmati, Shepard e Zhong (2009).

Para compensar o efeito da gravidade o NoShake utiliza um filtro passa-alta. O mesmo calcula a média decadente nas leituras brutas anteriores do acelerômetro e a subtrai da leitura bruta atual para obter somente a verdadeira aceleração do dispositivo. Isto permite que o aplicativo se adapte rapidamente as mudanças de gravidade em qualquer um dos eixos.

Após remover a gravidade dos valores brutos do acelerômetro os dados são armazenados em um buffer circular para uso posterior. O tamanho do buffer circular é definido para quatro segundos, a resposta do impulso do sistema massa-mola-amortecedor da

equação utilizada se aproxima rapidamente de zero à medida que o tempo aumenta, portanto um histórico mais curto pode render uma boa aproximação. Os Autores ainda comentam que quanto maior for o buffer circular maior será o custo computacional.

NoShake atualiza a tela somente quando detecta agitações, isto faz com que a imagem fique completamente imóvel em ambientes estáveis. Para determinar se está ocorrendo as agitações o NoShake analisa as amostras dos últimos 1.8 segundos agregando os valores absolutos da aceleração em cada eixo e comparando a soma com um limite determinado empiricamente. Caso NoShake detecte a agitação ele convolve as leituras da aceleração registradas pela resposta de impulso do sistema massa-mola-amortecedor para calcular a posição. O mesmo desloca então o conteúdo da tela conforme o valor calculado a partir do centro da tela.

Existem vários parâmetros que foram empiricamente determinados, como o coeficiente de elasticidade k do sistema massa-mola-amortecedor, o limiar para detectar as agitações e o valor da escala a . Os autores citam que coloram os valores conforme suas experiências, mas fizeram que esses parâmetros fossem customizáveis para que os usuários pudessem ajustar os mesmos conforme os seus cenários de uso.

O protótipo desenvolvido para Apple iPhone usou em média 30% do processador ARM de 620MHz durante o uso ativo, ou seja enquanto havia agitações. Apenas 1 a 3 % foi associado a leitura do acelerômetro, ao cálculo do deslocamento e aplicação do algoritmo de compensação. O resto é consumido para renderizar os gráficos, esta sobrecarga da atualização da tela foi atribuída a falta de acesso a unidade de processamento gráfico. Os autores acreditam que o cálculo pode ser aplicado de forma mais eficiente se for processado diretamente pelo processado gráfico.

Para avaliar a eficácia e facilidade de uso do NoShake foi realizado um estudo com dez participantes com idades entre 18 e 29 anos. O protótipo permitiu aos usuários ativar e desativar dinamicamente a funcionalidade do NoShake assim como justapor o texto estacionário e o texto estabilizado (Figura 14).

Figura 14 - Texto estacionário e estabilizado



Fonte: Rahmati, Shepard e Zhong (2009).

Conforme os autores os testes preliminares mostraram que os olhos tendem a rastrear todo o dispositivo para manter o foco. Este hábito não é indicado para usar o NoShake já que o texto não se move com o dispositivo. Deste modo os participantes foram aconselhados a se concentrar no texto ao invés de todo o dispositivo.

Após perguntar para os participantes se eles se recordam de situações que a agitação da tela incomodava foi apresentado o protótipo do NoShake. Em seguida foi solicitado para os participantes leiam um parágrafo de texto com e sem NoShake em quatro caso de testes que são: lendo a tela enquanto caminha normalmente, lendo a tela de um dispositivo segurado por outra pessoa enquanto caminham normalmente, lendo a tela enquanto se locomove em um carro andando a 30mph, lendo a tela que está montado no painel de um carro que está se locomovendo a 30mph. Durante os testes foram coletadas as opiniões dos participantes, pedindo que comparassem o texto estabilizado com o texto estacionário. As respostas foram baseadas em uma escala quantitativa entre 1 e 5 (1 para significativamente pior, 3 para sem diferença e 5 para significativamente melhor).

A metade dos participantes relataram que as situações onde aconteciam agitações os irritavam. Estes participantes geralmente tiveram atitudes positivas em todos os casos de teste. A outra metade que não relatou o incômodo nestes ambientes se mostrou ser mais neutro em relação ao protótipo.

A pesquisa revelou que não só diferentes tipos de agitação afetam os parâmetros ótimos do NoShake, mas cada usuário parece ter suas próprias preferências pessoais em relação aos parâmetros.

NoShake provou ser mais efetivo nos casos de testes veiculares, especialmente quando o participante estava segurando o dispositivo. Os participantes relataram que a NoShake os ajudou a acompanhar a posição de leitura dentro do texto. Essa capacidade de rastrear a posição de leitura no texto é uma das vantagens mais importantes que os usuários encontraram ao usar o NoShake, especialmente nos

testes veiculares; Um usuário notou: "Era basicamente impossível para mim rastrear o texto estacionário, mas era fácil rastrear o texto estabilizado. Mesmo que não fosse legível, isso me ajudou a não perder minha posição ao ler. Isso seria ótimo ao ler um parágrafo completo em um livro ". Outro participante deu um comentário semelhante: "Se passamos por um grande buraco, não consigo ler o texto estacionário, e eu perco completamente a posição [no texto]. O texto estabilizado é mais fácil de ler e eu não perco a posição ". De um lado, um usuário disse que teria desistido de ler o texto estacionário, mas achou o texto estabilizado relativamente fácil de ler. (RAHMATI; SHEPARD; ZHONG, 2009, p5, tradução nossa).

2.5.3 Comparativo entre os trabalhos correlatos

No Quadro 9 é apresentada uma comparação entre os trabalhos correlatos, onde se pode verificar que ambos usaram os sensores presentes nos smartphone para fazer as leituras dos movimentos, porém usaram abordagens diferentes para a solução do problema. Rahmati, Shepard e Zhong (2009) usaram a abordagem do modelo físico massa-mola-amortecedor para implementar um aplicativo para Apple Iphone e o submeteram a uma pesquisa com usuários testes para avaliar o aplicativo. Jeisy (2014) implementou protótipos com as abordagens de retorno direto, estimativa de posição e estimativa de posição melhorada, porém não obteve um resultado satisfatório. Para finalizar Jeisy (2014) criou o modelo oculto de Markov usando os dados que obteve através da câmera montada na cabeça de usuário de testes, Jeisy (2014) utilizou o software MATLAB para testar o modelo oculto de Markov mas não implementou um aplicativo para demonstrar o seu funcionamento, como não implementou o aplicativo não fez nenhuma pesquisa com usuários para avaliar o resultado obtido.

Quadro 9 - Comparação trabalhos correlatos

Característica / Trabalhos relacionados	Jeisy (2014).	Rahmati, Shepard e Zhong (2009)
Utilização sensores	Sim	Sim
Abordagem utilizada	Retorno direto, estimativa de posição, estimativa de posição melhorada com a ajuda de um controlador PID e Modelo oculto de Markov	Modelo físico massa-mola-amortecedor
Pesquisa com usuários	Não	Sim
Solução implementada	Modelo oculto de Markov não foi implementado, mas em teoria se mostrou uma boa opção.	Desenvolvido aplicativo para Apple Iphone

Fonte: elaborado pelo autor.

3 DESENVOLVIMENTO

Neste capítulo serão apresentadas as etapas do desenvolvimento do aplicativo. Na seção 3.1 são listados os requisitos funcionais e não funcionais. A seção 3.2 apresenta a especificação do aplicativo, demonstrando sua arquitetura. A seção 3.3 detalha a implementação e técnicas utilizadas. Por fim, a seção 3.4 demonstra os resultados obtidos.

3.1 REQUISITOS

A seguir estão listados os requisitos funcionais (RF) e requisitos não funcionais (RNF) que o aplicativo proposto deverá:

- a) identificar as trepidações usando o acelerômetro (RF);
- b) calcular a correção da trepidação usando o algoritmo do amortecedor (RF);
- c) aplicar a correção na janela do aplicativo monitorado (RF);
- d) usar a linguagem Java (RNF);
- e) executar no Sistema Operacional Android (RNF).

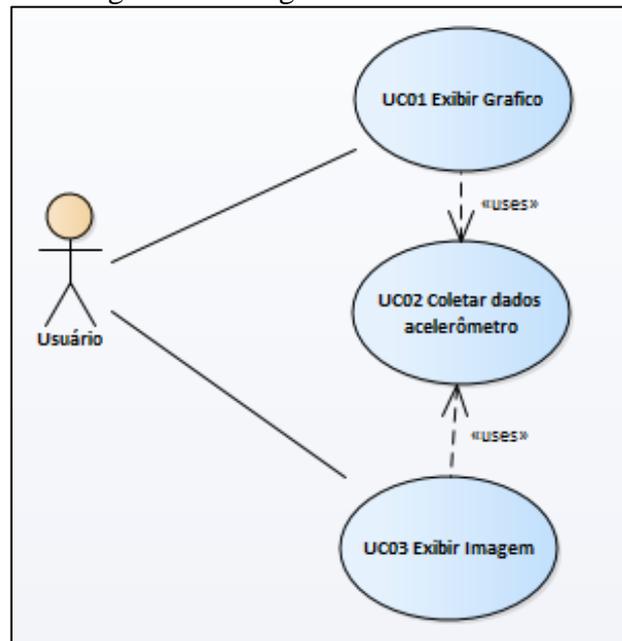
3.2 ESPECIFICAÇÃO

Esta seção apresenta os diagramas Unified Modeling Language (UML), utilizando a ferramenta Enterprise Architect. A seção 3.2.1 apresenta o diagrama de casos de usos. A seção 3.2.2 demonstra o diagrama de classes e a seção 3.2.3 apresenta o diagrama de atividades.

3.2.1 Diagrama de casos de uso

A Figura 15 exibe o diagrama de casos de uso do aplicativo. No diagrama existe apenas um ator denominado *Usuário* que utiliza todas as funcionalidades do aplicativo. No UC01 - *Exibir Gráfico* é exibida a tela que contém os gráficos com os valores obtidos do acelerômetro e do filtro passa-baixa. No UC03 - *Exibir Imagem* é exibida a tela onde ocorre a estabilização do texto. Ambos os casos de usos fazem uso do UC02 - *Coletar dados acelerômetro* que representa a coleta de dados do acelerômetro presente no smartphone.

Figura 15 - Diagrama de casos de uso

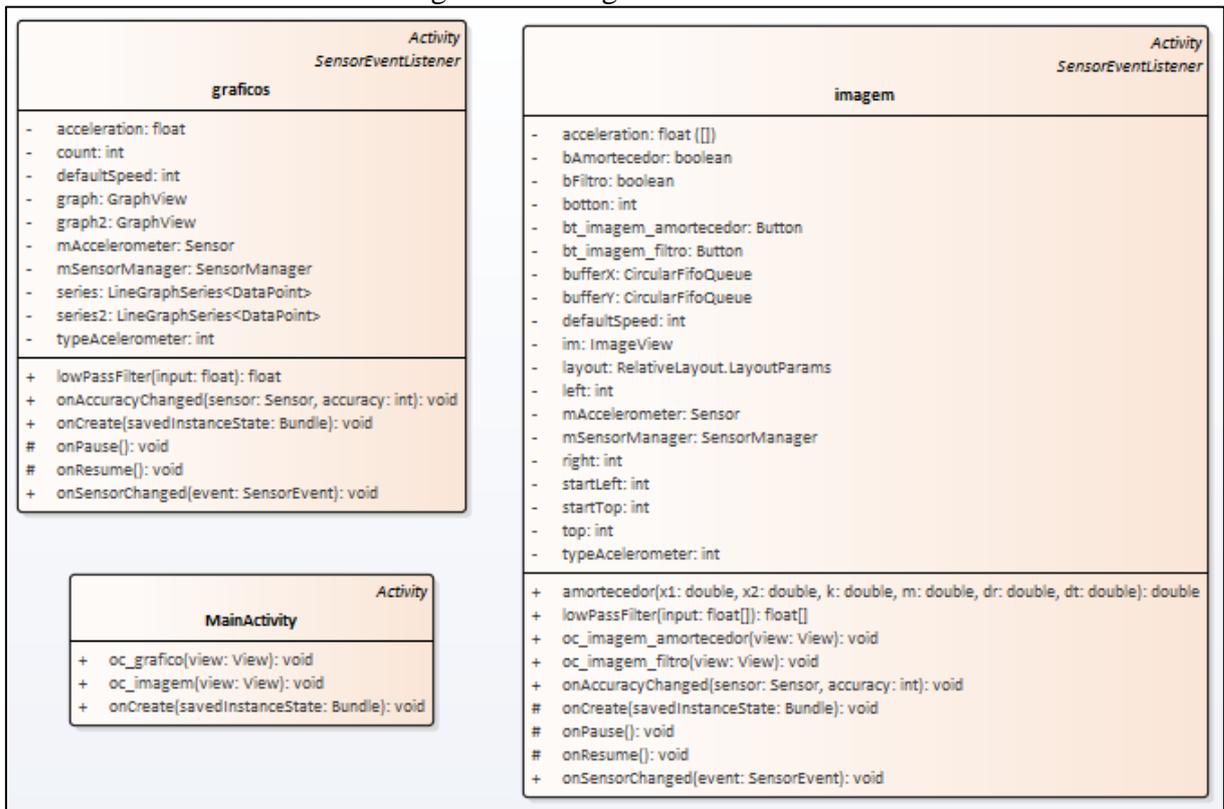


Fonte: elaborado pelo autor.

3.2.2 Diagrama de classes

A Figura 16 exibe o diagrama de classes do aplicativo. A classe `MainActivity` possui o código fonte da tela inicial do aplicativo que contém apenas as ações dos botões. A classe `gráficos` possui os métodos para fazer a leitura do acelerômetro e atualização dos gráficos. Se destaca o método `lowPassFilter` que faz a função do filtro passa-baixa (seção 2.3) e o método `onSensorChanged` que recebe os valores do acelerômetro e atualiza os dados nos gráficos. Por fim a classe `imagem` é responsável por fazer a estabilização do texto. Se destaca o método `amortecedor` que possui o algoritmo do sistema massa-mola-amortecedor (seção 2.4), `lowPassFilter` que faz a função do filtro passa-baixa, `onSensorChanged` que recebe os valores do acelerômetro e atualiza a posição do texto na tela conforme os dados processados.

Figura 16 - Diagrama de classes

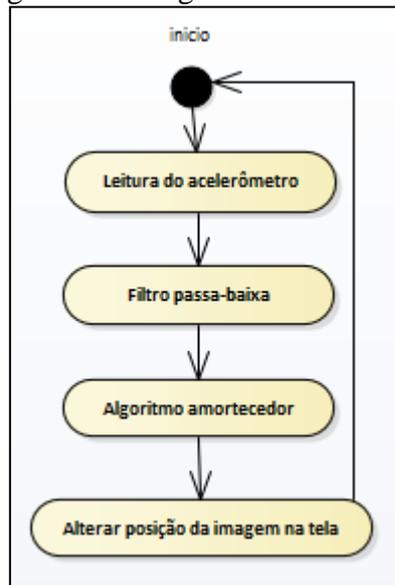


Fonte: elaborado pelo autor.

3.2.3 Diagrama de atividades

A Figura 17 exibe o diagrama de atividades do processo de estabilização do texto. O processo consiste em obter os valores brutos do acelerômetro e então passar estes valores pelo filtro passa-baixa, com os dados obtidos através do filtro é utilizado o algoritmo do sistema massa-mola-amortecedor para obter as coordenadas. Por fim a posição da imagem é atualizada na tela conforme as coordenadas do algoritmo do sistema massa-mola-amortecedor e todo o ciclo se reinicia.

Figura 17 - Diagrama de atividades



Fonte: elaborado pelo autor.

3.3 IMPLEMENTAÇÃO

A seguir são apresentadas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

O aplicativo foi desenvolvido para a plataforma Android utilizando a linguagem de programação Java e o ambiente de desenvolvimento Android Studio. Além disso foi utilizada a biblioteca GraphView para a exibição de gráficos e também a biblioteca Apache Commons Collections para a utilização de um buffer circular.

No Quadro 10 é apresentada a implementação do acelerômetro. Para obter os dados do acelerômetro do smartphone foi utilizado o `Sensor.TYPE_LINEAR_ACCELERATION` da API do Android, onde conforme Android Developers (2017c) retorna os valores da aceleração já excluindo o valor da gravidade. Os dados são obtidos usando a taxa `SensorManager.SENSOR_DELAY_GAME` da API do Android que equivale a taxa de 50Hz no smartphone Samsung Galaxy S5 New Edition SM-G903M usado nos testes do aplicativo. Esta frequência foi obtida através do aplicativo Sensor Kinetics.

Quadro 10 - Implementação acelerômetro

```

40     private SensorManager mSensorManager;
41     private Sensor mAccelerometer;
42     private int typeAccelerometer = Sensor.TYPE_LINEAR_ACCELERATION;
43     private int defaultSpeed = SensorManager.SENSOR_DELAY_GAME;
44
45     protected void onCreate(Bundle savedInstanceState) {
46         super.onCreate(savedInstanceState);
47         setContentView(R.layout.activity_imagem);
48
49         mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
50         mAccelerometer = mSensorManager.getDefaultSensor(typeAccelerometer);
51         mSensorManager.registerListener(this, mAccelerometer, defaultSpeed);
52     }

```

Fonte: elaborado pelo autor.

Após obter os dados do acelerômetro foi utilizado um filtro passa baixa (Quadro 11). Nesta implementação é calculada apenas os valores dos eixos X e Y do acelerômetro. Foi utilizado o valor inicial de 0.1 para a constante alpha.

Quadro 11 - Implementação filtro passa baixa

```

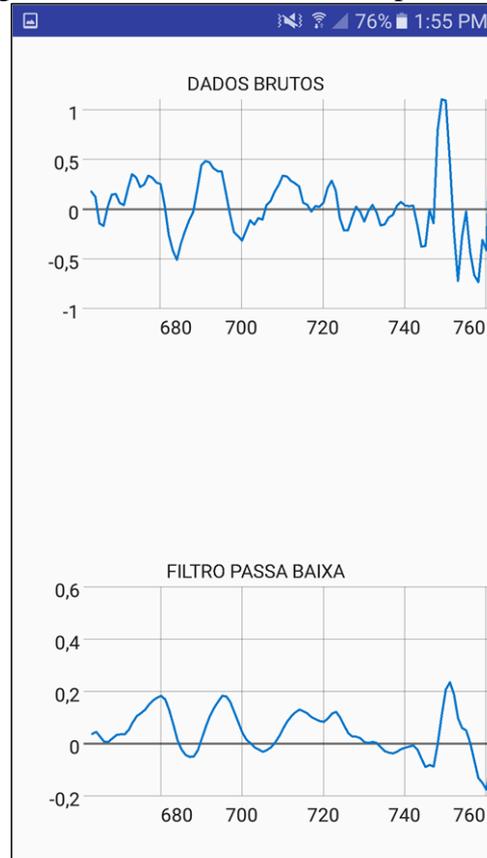
public float[] lowPassFilter(float[] input) {
    float alpha = 0.1f;
    acceleration[0] = acceleration[0] + alpha * (input[0] - acceleration[0]);
    acceleration[1] = acceleration[1] + alpha * (input[1] - acceleration[1]);
    return acceleration;
}

```

Fonte: elaborado pelo autor.

Na Figura 18 são exibidos dois gráficos os quais são atualizados em tempo real no aplicativo. O gráfico Dados Brutos mostra os valores obtidos diretamente do acelerômetro no eixo X e o gráfico Filtro Passa Baixa mostra os valores do acelerômetro no eixo X após passarem pelo algoritmo filtro passa baixa. Este gráfico foi implementado para permitir analisar em tempo real os dados que estão sendo utilizados no aplicativo.

Figura 18 - Dados brutos e filtro passa baixa



Fonte: elaborado pelo autor.

O próximo passo foi utilizar os valores do filtro passa baixa no algoritmo do sistema massa-mola-amortecedor. É possível verificar a implementação do algoritmo do amortecedor no Quadro 12. Os parâmetros do método são respectivamente o valor atual do eixo, valor anterior do eixo, coeficiente de elasticidade da mola, massa, coeficiente de amortecimento e o tempo entre as medições dos eixos.

Quadro 12 - Algoritmo amortecedor

```
public double amortecedor(double x1, double x2, double k, double m, double dr, double dt) {
    double kmdt = k * (Math.pow(dt, 2) / m);
    return x1 * (1 - kmdt) + (x1 - x2) * (1 - 2 * dr * Math.sqrt(kmdt));
}
```

Fonte: elaborado pelo autor.

No Quadro 13 está a implementação da estabilização. O primeiro passo é a obtenção dos valores do acelerômetro que é realizado nas linhas 76 e 77 para os eixos X e Y respectivamente. Em seguida os valores são processados pelo filtro passa-baixa, isto ocorre na linha 81. Como os valores do acelerômetro são muitos baixos para definir as coordenadas multiplicou-se por um valor inicial, definido como 50 conforme as linhas 83 e 84, para poder obter a visibilidade dos movimentos na tela. Além disso, os valores do eixo Y conforme a linha 84 foram multiplicados por um número negativo, isto foi feito para inverter a direção do

movimento da estabilização do texto no eixo Y. Para a utilização do algoritmo do amortecedor foi necessário implementar um buffer circular para armazenar valores passados do filtro passa baixa, isto ocorre nas linhas 86 e 87. Nas linhas 93 e 94 ocorre o cálculo da coordenada utilizando o algoritmo do amortecedor, os parâmetros k , m e dr foram usados conforme Lungu (2011) onde o mesmo traz um cenário criticamente amortecido. O valor do tempo 0.02 foi obtido dividindo um segundo pela frequência de obtenção de dados (1s/50Hz). Por fim, conforme a linha 96 e 97, é atualizado a coordenada de onde o texto será exibido.

Quadro 13 - Algoritmo estabilização

```

74  @Override
75  public void onSensorChanged(SensorEvent event) {
76      float x = event.values[0]; //positivo
77      float y = event.values[1]; //negativo
78
79      if (bAmortecedor && bFiltro) {
80          float vetor[] = {x, y};
81          float[] filtro = lowPassFilter(vetor);
82
83          float xf = filtro[0] * 50;
84          float yf = filtro[1] * -50;
85
86          bufferX.add(xf);
87          bufferY.add(yf);
88
89          if (bufferX.isAtFullCapacity()) {
90              double k = 1;
91              double m = 1;
92              double dr = 1;
93              float xa = (float) amortecedor((float) bufferX.get(1), (float) bufferX.get(0), k, m, dr, 0.02);
94              float ya = (float) amortecedor((float) bufferY.get(1), (float) bufferY.get(0), k, m, dr, 0.02);
95
96              left = startLeft - (int) xa;
97              top = startTop - (int) ya;
98          }
99      }

```

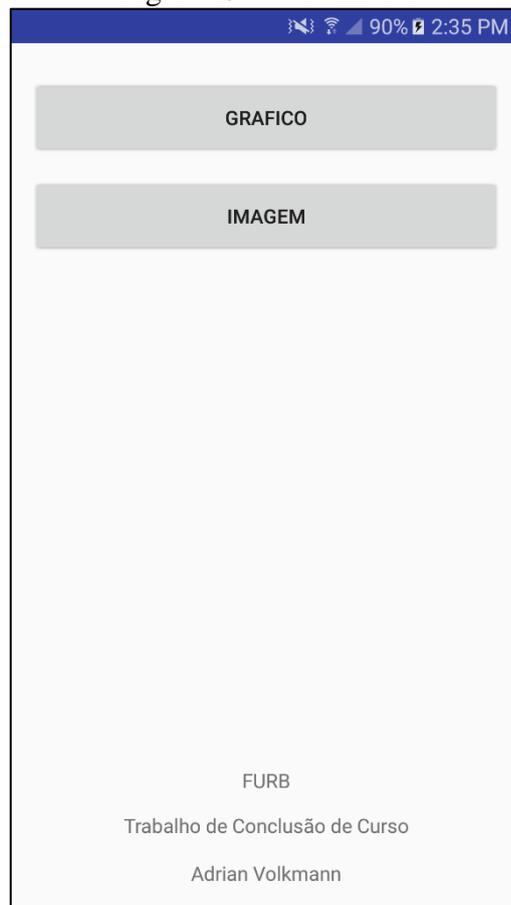
Fonte: elaborado pelo autor.

Este algoritmo retorna os dados que são usados como coordenadas para movimentar o texto na tela, os valores obtidos são usados para movimentar o texto conforme seu eixo, ou seja, os dados do eixo X do acelerômetro são usados para mover a imagem para direita ou esquerda enquanto os valores obtidos do eixo Y são usados para mover a imagem para cima ou para baixo.

3.3.2 Operacionalidade da implementação

Ao abrir o aplicativo é demonstrada a tela conforme a Figura 19 que possui dois botões GRAFICO e IMAGEM que abrem respectivamente as telas conforme a Figura 18 e Figura 20. A Figura 18 já foi apresentada na seção 3.3.1.

Figura 19 - Tela inicial

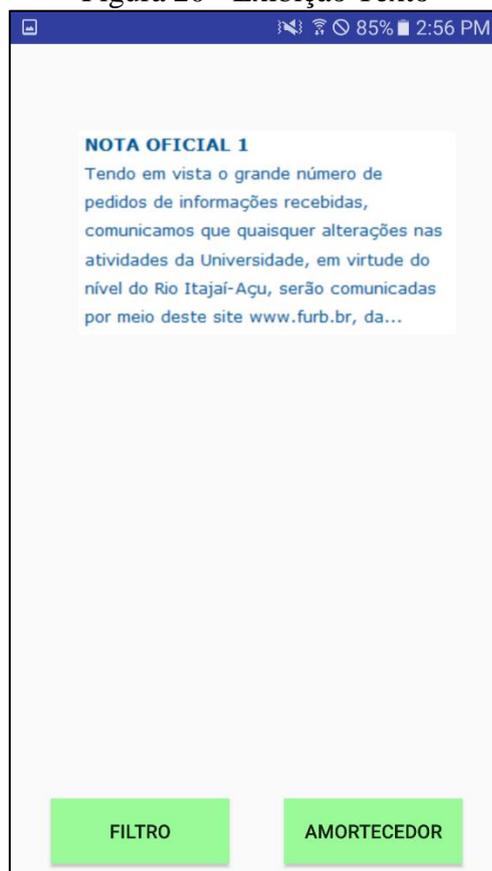


Fonte: elaborado pelo autor.

Na Figura 20 é demonstrado a tela onde é feita a estabilização do texto. Ainda nesta tela existem os botões FILTRO e AMORTECEDOR que ativam ou desativam sua respectiva função. Estes botões foram implementados para visualizar o efeito causado por cada algoritmo usado.

O texto utilizado nos testes faz referência a uma nota oficial publicada acerca de uma situação de enchente pela qual Blumenau passou recentemente.

Figura 20 - Exibição Texto



Fonte: elaborado pelo autor.

Para utilizar a funcionalidade da estabilização basta movimentar o smartphone que será possível notar o texto se movimentando na tela conforme os movimentos. Enquanto o smartphone estiver em repouso o texto permanecerá parado, a partir do momento que o acelerômetro registrar movimentação, o texto começará a se movimentar com o fim de suavizar os movimentos e melhorar a legibilidade do texto que está sendo exibido.

3.4 ANÁLISE DOS RESULTADOS

Ao término do projeto constata-se que foi possível atender aos objetivos pré-estabelecidos, pois utilizando o acelerômetro foi possível identificar a trepidação de um dispositivo móvel. A correção foi aplicada utilizando o algoritmo do amortecedor e com essas informações altera-se a disposição do texto na tela do smartphone.

Tendo em vista possibilitar uma comparação no processo de estabilização do texto foram adicionadas as opções para desativar os algoritmos implementados, possibilitando a análise de cada um separadamente.

Com o efeito do filtro passa-baixa e amortecedor desativados observa-se que o texto se move de forma abrupta dificultando a leitura, isto se deve pelos picos medidos pelo

acelerômetro. Com apenas o filtro passa-baixa ativo foi possível perceber um movimento mais suave do texto, isto ocorre pelo efeito de suavização do filtro que elimina os picos de leitura do acelerômetro. Com ambos os algoritmos ativos o texto se desloca ainda mais suavemente e com menos intensidade se comparando a implementação com apenas o filtro passa baixa.

Uma dificuldade encontrada para realização deste trabalho foi encontrar um meio para suavizar os movimentos, utilizando os valores brutos do acelerômetro não estava trazendo bons resultados. As movimentações sempre se mostravam de forma abrupta. Para resolver isto foi necessário estudar alguma forma de eliminar ruídos e picos de leitura do acelerômetro. Foram verificados alguns tipos de filtros de sinal digital e o que melhor se encaixou foi o filtro passa baixa. Com o tipo de filtro definido foi necessário definir qual o valor da variável α utilizar, após alguns testes analisando os resultados optou-se pelo valor 0,1 que trazia o nível de suavização desejada.

Outra dificuldade encontrada foi em relação ao funcionamento e as fórmulas do sistema massa-mola-amortecedor. Como o autor não tinha conhecimentos aprofundados sobre esta área levou-se um certo tempo até compreender todo funcionamento do sistema e também a compressão das fórmulas matemáticas. Em muitas etapas do desenvolvimento deste projeto o autor se deparou com assuntos que não possuía muito conhecimento, sendo necessário buscar informação e estudar para entender o funcionamento de certas tecnologias. Estas pesquisas e estudos demandaram boa parte do tempo no desenvolvimento deste trabalho.

No Quadro 14 é exibido um comparativo entre os trabalhos correlatos e este trabalho. Todos os trabalhos utilizaram o acelerômetro removendo sua gravidade, Rahmati, Shepard e Zhong (2009) utilizaram um filtro passa-alta para isto. Este trabalho se assemelha ao trabalho de Rahmati, Shepard e Zhong (2009) por usar o sistema massa-mola-amortecedor enquanto Jeisy (2014) utiliza outras abordagens. O único trabalho a fazer a pesquisa de usabilidade com usuários foi o de Rahmati, Shepard e Zhong (2009). Por fim, Jeisy (2014) não implementou sua abordagem final mostrando apenas o modelo teórico, este trabalho trouxe uma implementação para smartphones Android enquanto o trabalho de Rahmati, Shepard e Zhong (2009) trouxe uma implementação para Apple Iphone.

Quadro 14 - Comparação trabalhos

Característica / Trabalhos relacionados	Jeisy (2014).	Rahmati, Shepard e Zhong (2009)	Trabalho Atual
Utilização acelerômetro	Sim	Sim	Sim
Remoção da gravidade	Sim	Sim	Sim
Utilização de filtros	Não	Filtro passa-alta para remoção da gravidade	Filtro passa-baixa
Abordagem utilizada	Retorno direto, estimativa de posição, estimativa de posição melhorada com a ajuda de controlador PID, Modelo oculto de Markov	Modelo físico massa-mola-amortecedor	Modelo físico massa-mola-amortecedor
Pesquisa com usuários	Não	Sim	Não
Plataforma	Não foi implementado	Apple Iphone	Android

Fonte: elaborado pelo autor.

4 CONCLUSÕES

Este trabalho apresentou a implementação de um protótipo de um aplicativo que permite a estabilização de um texto na tela conforme a trepidação do dispositivo móvel. Foi utilizado o acelerômetro para obter os dados da trepidação, e após passarem pelo filtro passa-baixa os dados são utilizados no algoritmo do sistema massa-mola-amortecedor e por fim atualiza a posição do texto exibido na tela. Deste modo foi possível atender todos os objetivos pré-estabelecidos.

Estima-se que, após um processo de validação com usuários, o protótipo possa contribuir para uma melhor legibilidade enquanto se está em ambientes onde ocorram trepidações, como por exemplo, caminhando ou estando em um carro em movimento. O protótipo consegue estabilizar o texto até um certo limite, pois quando as trepidações são muito intensas o mesmo não é capaz de estabilizar adequadamente.

Os testes foram baseados utilizando o smartphone Samsung Galaxy S5 New Edition SM-G903M. Para a utilização deste aplicativo em outros dispositivos seria necessário avaliar novamente todos os parâmetros utilizados neste trabalho para verificar se o efeito continua igual ou semelhante.

4.1 EXTENSÕES

Sugestão de extensões para trabalhos futuros:

- a) avaliar o processo de estabilização utilizando a abordagem do Filtro de Kalman;
- b) desenvolver um navegador web baseado no processo de estabilização desenvolvido;
- c) implementar um filtro passa-baixa com α adaptável conforme cada dispositivo;
- d) realizar a implementação contemplando a possibilidade de parametrizar dinamicamente as variáveis de controle, para poder realizar a estabilização de formas diferentes quando as grandezas do acelerômetro forem muito altas.

REFERÊNCIAS

- ANDROID DEVELOPERS. **SensorEvent**. 2017a. Disponível em: <<https://developer.android.com/reference/android/hardware/SensorEvent.html>>. Acesso em: 01 jun. 2017.
- ANDROID DEVELOPERS. **SensorManager**. 2017b. Disponível em: <<https://developer.android.com/reference/android/hardware/SensorManager.html>>. Acesso em: 01 jun. 2017.
- ANDROID DEVELOPERS. **Motion Sensors**. 2017c. Disponível em: <https://developer.android.com/guide/topics/sensors/sensors_motion.html>. Acesso em: 01 junho 2017.
- BEAR, Mark F; CONNORS, Barry W.; PARADISO, Michael A.. **Neurociências: desvendando o sistema nervoso**. 2. ed. Porto Alegre: Artmed Editora S.a., 2002.
- CAMPBELL, F.w.. The Depth of Field of the Human Eye. **Optica Acta: International Journal of Optics**, [s.l.], v. 4, n. 4, p.157-164, dez. 1957. Informa UK Limited. <http://dx.doi.org/10.1080/713826091>.
- COMPUTERWORLD. **3,8 milhões de smartphones são vendidos no mundo diariamente**. 2016. Disponível em: <<http://computerworld.com.br/38-milhoes-de-smartphones-sao-vendidos-no-mundo-diariamente>>. Acesso em: 18 set. 2016.
- CORPBUSINESS. **Mobile intelligence 2.0**. 2015. Disponível em: <<http://www.corpbusiness.com.br/download/material/2015/mobintelligence/agenda.pdf>>. Acesso em: 18 set. 2016.
- GOVENDER, Sashen. **Using the Accelerometer on Android**. 2014. Disponível em: <<https://code.tutsplus.com/tutorials/using-the-accelerometer-on-android--mobile-22125>>. Acesso em: 22 jun. 2017.
- HAIN, Timothy C.; OMAN, Charles M.. **Why does reading in a moving car cause motion sickness?** [2010?]. Disponível em: <<https://www.scientificamerican.com/article/why-does-reading-in-a-mov/>>. Acesso em: 07 nov. 2016.
- HOSPITAL DE OLHOS DE SÃO PAULO. **Mito ou verdade: ler no ônibus ou carro faz mal para a visão?**. 2016?. Disponível em: <<http://blog.hospitaldeolhos.net/index.php/ler-no-onibus-ou-carro/>>. Acesso em: 18 set. 2016.
- INFOESCOLA. **Dioptria**. 2017?. Disponível em: <<http://www.infoescola.com/optica/dioptria/>>. Acesso em: 04 jul. 2017.
- JEISY, Kevin. **Walking with your Smartphone: Stabilizing Screen Content**. 2014. 22 f. TCC (Graduação) - Curso de Computação Distribuída, Grupo de Computação distribuída Engenharia de Computação e Laboratório de Redes, Instituto Federal de Tecnologia de Zurique, Zurique, 2014.
- KHANACADEMY. **O que é a lei de Hooke?** 2017. Disponível em: <<https://pt.khanacademy.org/science/physics/work-and-energy/hookes-law/a/what-is-hookes-law>>. Acesso em: 22 jun. 2017.
- KIRCHER, Kaleb. **Android Accelerometer Noise, Offset and Skew**. 2017?a. Disponível em: <<http://www.kircherelectronics.com/blog/index.php/11-android/sensors/7-android-accelerometer>>. Acesso em: 03 jun. 2017.

KIRCHER, Kaleb. **Low-Pass Filter: The Basics**. 2017?b. Disponível em: <<http://www.kircherelectronics.com/blog/index.php/11-android/sensors/8-low-pass-filter-the-basics>>. Acesso em: 03 jun. 2017.

LABORATÓRIO IMOBILIS. **Acelerômetros: uso em celulares e detecção de velocidade. Parte 1**. 2013. Disponível em: <<http://www.decom.ufop.br/imobilis/acelerometros-uso-em-celulares-e-deteccao-de-velocidade-parte-1/>>. Acesso em: 12 jun. 2017.

LEARNING ABOUT ELECTRONICS. **Low Pass Filter- Explained**. 2014?. Disponível em: <<http://www.learningaboutelectronics.com/Articles/Low-pass-filter.php>>. Acesso em: 03 jun. 2017.

LOTTENBERG, Claudio; MATTOS, Rubens Belfort. **Ler no ônibus ou no carro não faz mal para a visão**. 2011. Disponível em: <<http://g1.globo.com/bemestar/noticia/2011/06/ler-no-onibus-ou-no-carro-nao-faz-mal-para-visao.html>>. Acesso em: 31 out. 2016.

LUNGU, George. **Casual Introduction to Numerical Methods: a spring-mass-damper system model - part2**. 2011. Disponível em: <http://excelunusual.com/wp-content/uploads/2011/02/Spring_Mass_Damper_Tutorial_2.pdf>. Acesso em: 06 jun. 2017.

MELDAU, Débora Carvalho. **Aparelho Vestibular**. 2017. Disponível em: <<http://www.infoescola.com/anatomia-humana/aparelho-vestibular/>>. Acesso em: 22 jun. 2017.

MSPC. **Vibrações mecânicas I-30**. 2017?. Disponível em: <<http://www.mspc.eng.br/mecn/mvbr130.shtml>>. Acesso em: 22 jun. 2017.

MUCHERONI, Mário Francisco. **Dinâmica de Sistemas Discretizados Vibrações Mecânicas**. 2012. Disponível em: <<http://www2.eesc.usp.br/labdin/mucheroni/Vibr%20Mec%2002.pdf>>. Acesso em: 06 jun. 2017.

RAHMATI, Ahmad; SHEPARD, Clayton; ZHONG, Lin. NoShake: Content stabilization for shaking screens of mobile devices. **2009 Ieee International Conference On Pervasive Computing And Communications**, [s.l.], mar. 2009. IEEE.

SCHÄFER, Annete. **Geração celular**. 2009. Disponível em: <http://www2.uol.com.br/vivermente/reportagens/geracao_celular.html>. Acesso em: 18 set. 2016.

SILVA, Renato Molina da. **Elementos de um Sistema Mecânico**. 2017?. Disponível em: <https://www.ime.usp.br/~oda/contents/01Matem%E1tica/01Sistemas%20Din%E2micos/04_Elem_Sist_Mec.pdf>. Acesso em: 18 jun. 2017.

SUPER INTERESSANTE. **Ler no ônibus descola a retina?** 2010. Disponível em: <<http://super.abril.com.br/saude/ler-no-onibus-descola-a-retina/>>. Acesso em: 31 out. 2016.

WIKIPEDIA. **Low-pass filter**. 2017. Disponível em: <https://en.wikipedia.org/wiki/Low-pass_filter>. Acesso em: 03 jun. 2016.