

Biblioteca de componentes de interface para iPhone OS

Izabel Cristina da Silva Carmo – Acadêmico

Dalton Solano dos Reis - Orientador



Roteiro

- Introdução / Objetivos
- Fundamentação teórica
 - Contextualização
 - Tecnologias
 - Trabalhos correlatos
- Desenvolvimento
 - Requisitos principais
 - Implementação
 - Resultados e discussão
- Conclusão / Extensões



Introdução

- Dispositivos Móveis
 - Smartphones (Internet 44%, iPhone 50%)
- App Store
 - 100 mil aplicações
 - 2 bilhões de downloads
 - 10 mil aplicações por semana



Objetivos do trabalho

- Disponibilizar uma biblioteca de componentes de interface para o iPhone OS
- Componentes:
 - Joystick
 - Barra de Vida
 - Barra de Tempo
 - Pedal de Aceleração



Fundamentação teórica



Interface Humano-Computador

- Conceito
 - “É uma disciplina preocupada com o design, avaliação e implementação de sistemas computacionais interativos para uso humano e com o estudo dos principais fenômenos ao seu redor” (HEWETT et al., 1992).
- Fatores de aceitação:
 - Utilidade
 - Confiabilidade
 - Usabilidade
 - Eficiência



iPhone 3G

- Definição
 - Dispositivo que combina a interface multi-toque com recursos sofisticados como e-mail, mensagem instantâneas, câmera e celular (APPLE, 2009).
- Características
 - Tela de 3,5 polegadas (480 x 320 pixels)
 - Memória RAM de 128MB
 - Teclado Virtual
 - Armazenamento (16 ou 32 GB)
 - Tela multi-toque
 - Global Positioning System (GPS)
 - Acelerômetro



Fonte: GSMFans (2007)



Objective-C

- Linguagem de programação
 - Orientada a Objetos
 - Incorpora a linguagem C
 - Aplicativos para iPods e iPhones
- Peculiaridades:
 - Classes abstratas
 - Modificadores de acesso para métodos



Trabalhos correlatos



Minigore

- Jogo de tiro para iPhone
- *Joystick* inferior esquerdo controla o movimento
- *Joystick* inferior direito usado para atirar



Fonte: Ellis (2009).

Pocket God

- Jogo para iPhone com uso dos recursos de acelerômetro e tela multi-toque
- Barra de Vida no canto inferior direito



Fonte: Ellis (2009).

Desenvolvimento da biblioteca



UNIVERSIDADE REGIONAL DE BLUMENAU

Desenvolvimento da biblioteca

Requisitos

- Permitir ao usuário decidir onde posicionará o componente *joystick* na tela (RF)
- Permitir que imagens sejam adicionadas para o componente *joystick* (RF)
- Permitir que o componente Barra de Vida tenha seu valor de taxa de atualização modificada a qualquer momento (RF);
- Permitir modificar o valor para a quantidade restante de vida do componente Barra de Vida a qualquer momento (RF);



Requisitos

- Permitir ao usuário optar pela desaceleração usando o componente pedal de aceleração (RF)
- Ser desenvolvido em Objective-C (RNF)
- Utilizar o ambiente de programação Xcode (RNF)
- Executar na plataforma iPhone (RF)



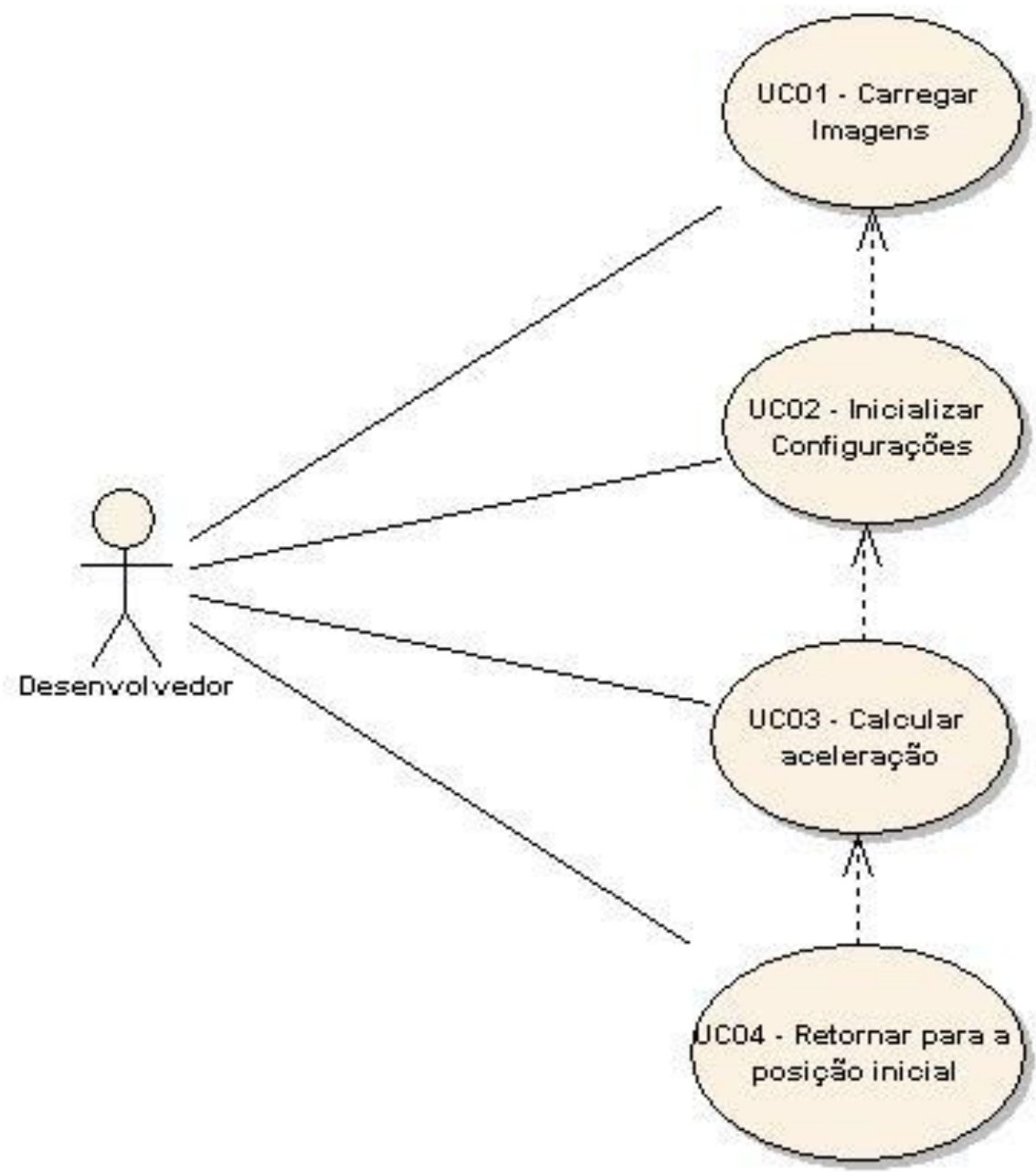
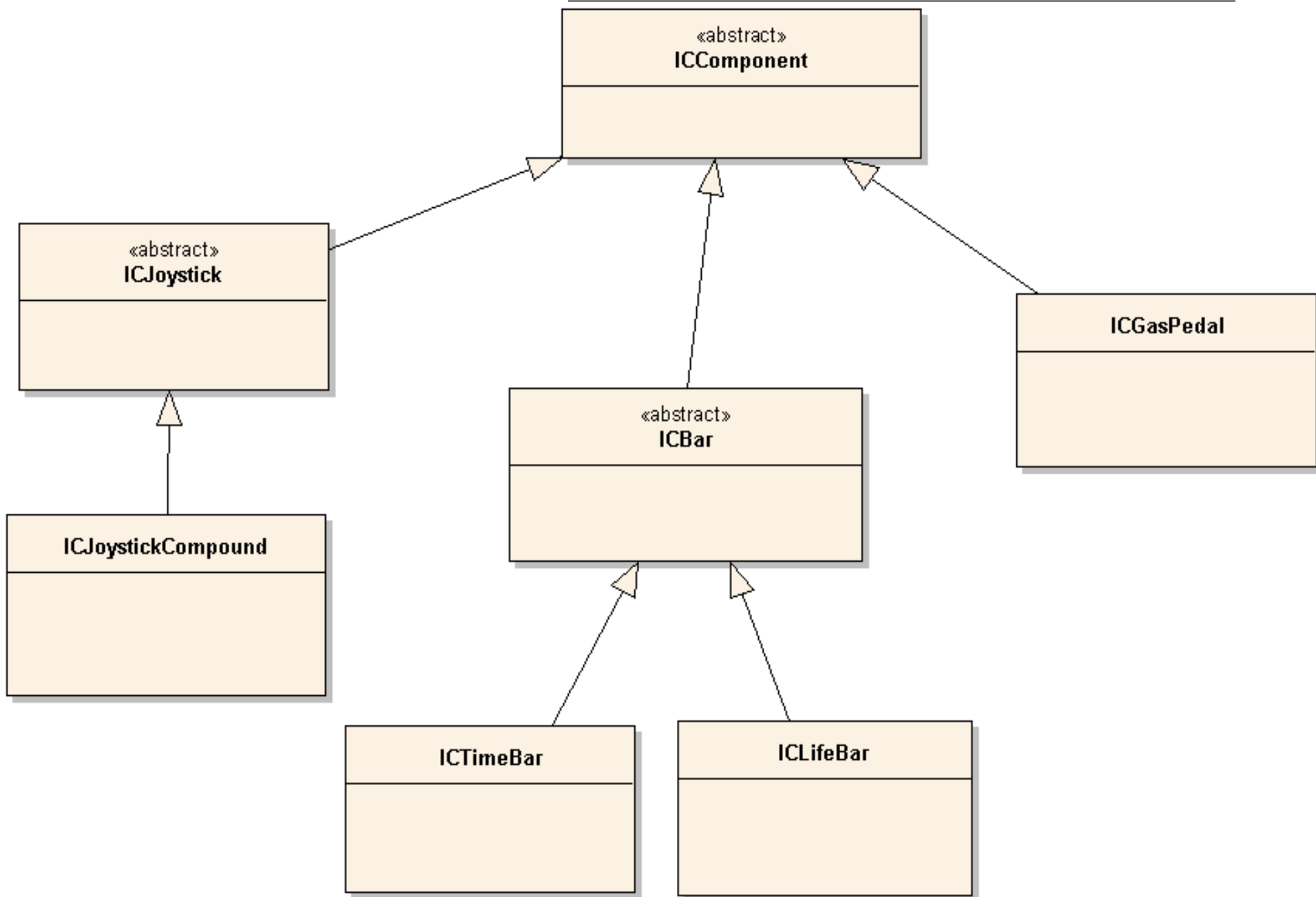


Diagrama de casos de uso

Diagrama de Classes



iPhone Simulator

- Simulador oficial
- Janela no formato do iPhone
- Permite simular:
 - Toque
 - Multi-toque
 - Movimento de arrastar
 - Problemas com memória



Implementação - Joystick

```
187 - (int) getAcceleration : (UITouch *) touch {
188     CGPoint location = [touch locationInView:self];
189
190     // Find the center point (X,Y) from top image joystick
191     // Same as: xcenter = ((xmax - xmin) / 2) + xmin where xmax is equals to: xmin+largura
192     float x2 = [self topImageJoystick].center.x;
193     float y2 = [self topImageJoystick].center.y;
194
195
196     //Same as: (x2-x1)^2 + (y2-y1)^2
197     float distance = ((x2 - x1) * (x2 - x1)) + ((y2 - y1) * (y2 - y1));
198
199
200     //Test using only the distance
201     if([self isInside: distance]){
202         [self loadNewPosition: location];
203         [self setLastAcceleration: [self calculateAcceleration: distance]];
204         return [self lastAcceleration];
205
206     }else { // If its outside of the big image, keep the last acceleration
207         return [self lastAcceleration];
208     }
209
210 }
```



Utilizando a biblioteca

- **Inclusão do componente Joystick**

- Configuração das funcionalidades básicas
- Retorno ao usuário via console

```
11 @implementation ICExampleJoystick
12 @synthesize backgroundImageJoystick;
13 @synthesize topImageJoystick;
14 @synthesize joystickCompound;
15
16
17
18 - (id)initWithFrame:(CGRect)frame {
19     if (self = [super initWithFrame:frame]) {
20
21         joystickCompound = [ICJoystickCompound alloc];
22
23         // load joystick background image
24         backgroundImageJoystick = [joystickCompound loadImageJoystickBackground:30 :50 :100 :100 :
25             @"/Users/bell_cristina/Desktop/Joystick/ballOutImage.png"];
26
27         // load joystick top image
28         topImageJoystick = [joystickCompound loadImageJoystickTop:45 :45 :
29             @"/Users/bell_cristina/Desktop/Joystick/ballInImage.png"];
30
31         [joystickCompound initialSettings];
32     }
33     return self;
34 }
```



Utilizando a biblioteca

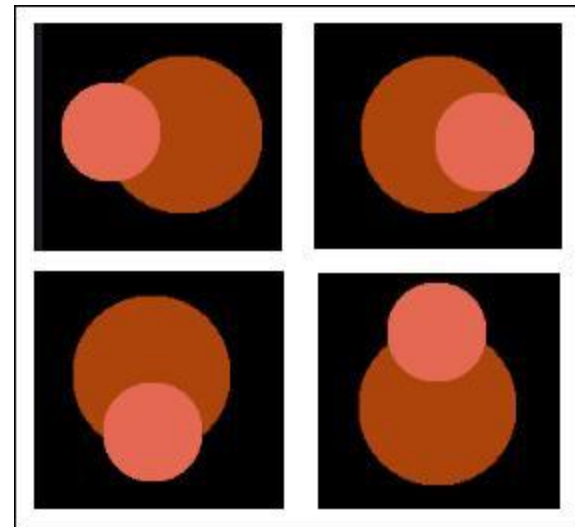
```
37 - (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event{
38     UITouch *touch = [[event allTouches] anyObject];
39
40     if([touch view] == topImageJoystick){
41         [joystickCompound touchesEndedJoystick: touch];
42     }
43 }
44
45 - (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event {
46     //detect the location of the user's touch
47     UITouch *touch = [[event allTouches] anyObject];
48
49     // Enable User Interaction for top image of the joystick
50     topImageJoystick.userInteractionEnabled = YES;
51
52     //Check if user touched the top image of the joystick (topImageJoystick)
53     if([touch view] == topImageJoystick){
54         int acceleration = [joystickCompound getAcceleration: touch];
55         NSLog(@"Aceleracao: %d", acceleration);
56     }
57 }
```



Utilizando a biblioteca



Componente joystick retornando a aceleração via console



Componente joystick arrastado em várias direções

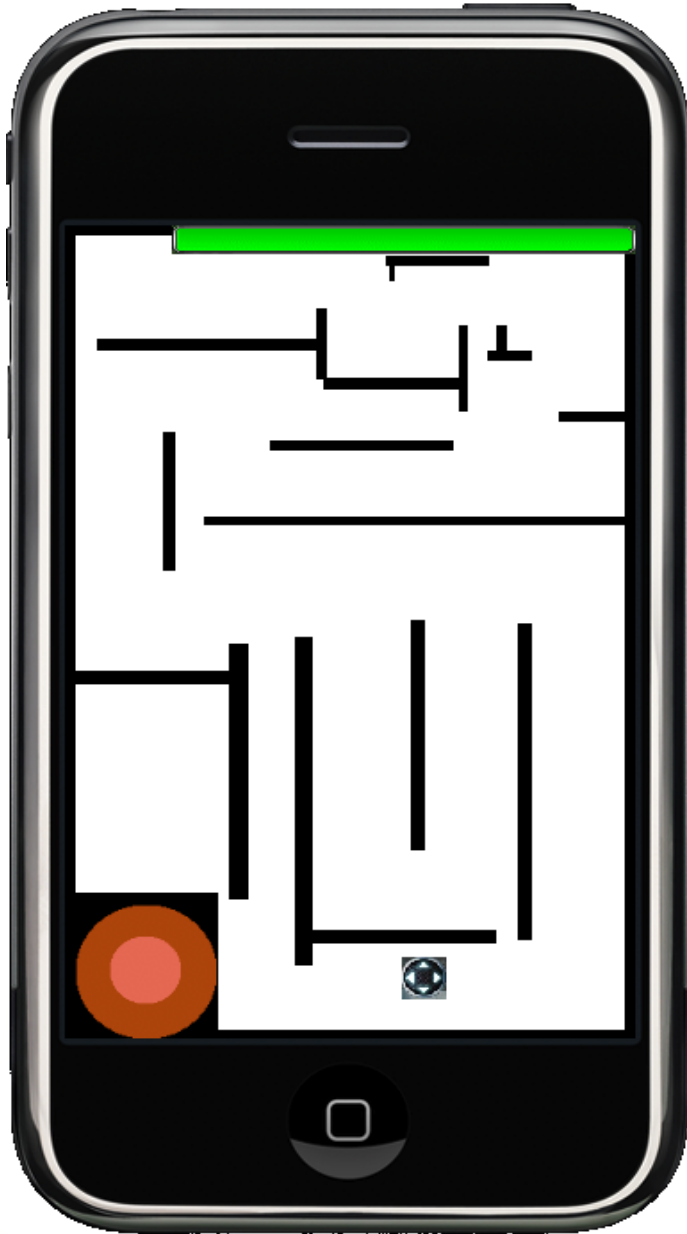


Resultados e discussão

- Preocupação com Orientação a Objetos
- Testes práticos das funcionalidades
 - *Joystick* com imagens circulares
- Validação no iPhone Simulador
- Validação no aparelho iPhone
- Testes de usabilidade



Testes de usabilidade



- Labirinto
 - 3 fases
 - 15 usuários
 - 0..50 e 0..100 aceleração



Testes de usabilidade

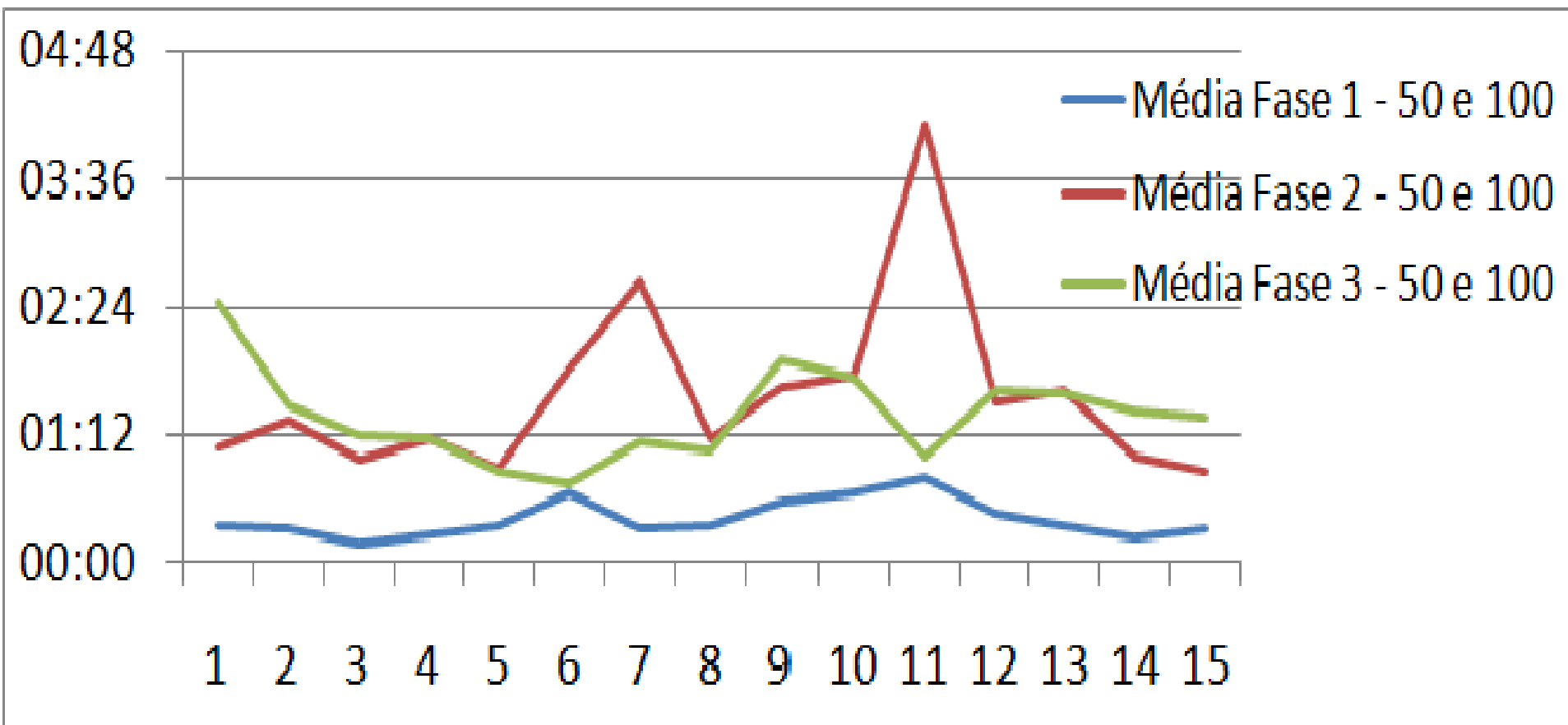
Usuário	Colisões	Tempo	Tempo Total
1	12	00:46	01:10
2	7	01:32	01:46
3	1	01:00	01:02
4	1	01:21	01:23
5	13	00:26	00:52
6	15	00:23	00:53
7	16	00:22	00:54
8	7	00:29	00:43
9	10	00:28	00:48
10	6	00:32	00:44
11	0	05:08	05:08
12	3	01:00	01:06
13	5	00:59	01:09
14	9	00:26	00:44
15	12	00:20	00:44

- Componente joystick
- Fase 2
 - Aceleração 0..50
- Dados obtidos
 - Colisões
 - Tempo Total



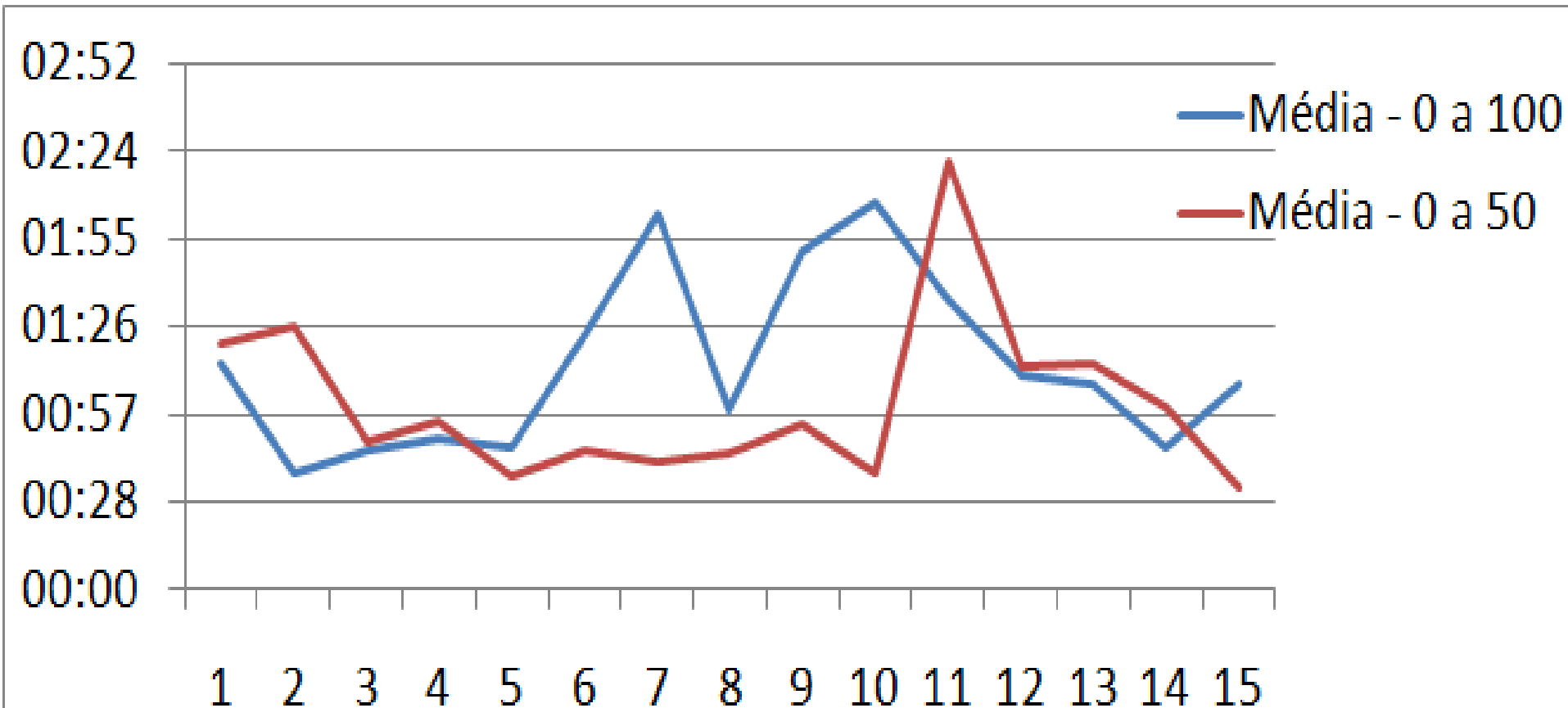
Testes de usabilidade

- Gráfico do grau de dificuldade do experimento para as fases 1, 2 e 3



Testes de usabilidade

- Gráfico da análise da utilização da aceleração 0..50 e 0..100



Conclusão

- Agilidade na implementação dos componentes
- Redução no nível de complexidade da aplicação
- Requisitos e objetivos em geral foram alcançados
- Componente joystick
 - Menor aceleração mais controle
 - Menor tempo mais colisões



Extensões

- Criação de outros componentes de interface
- Adição de novas funcionalidades
- Uso com OpenGL ES
- Testes com um grupo maior de pessoas
- Integração com o ambiente de desenvolvimento de telas Interface Builder
- Criação de um novo ambiente de desenvolvimento de telas



Apresentação Prática



Obrigado!

“Uma mente que se abre pra uma nova idéia nunca volta ao seu estado normal ”.

Albert Einstein



