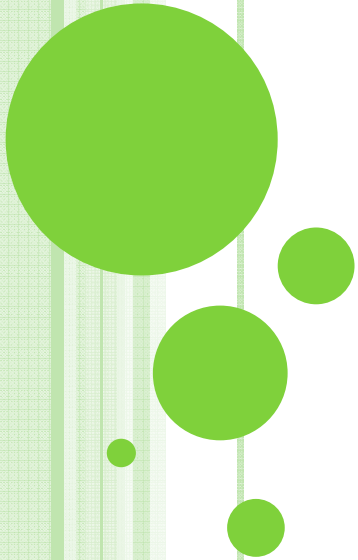
IFURBOT – MIGRAÇÃO DO FRAMEWORK FURBOT PARA A PLATAFORMA DO IPHONE



Aluno: Marco Antônio Corrêa

Orientador: Mauro Marcelo Mattos

SUMÁRIO

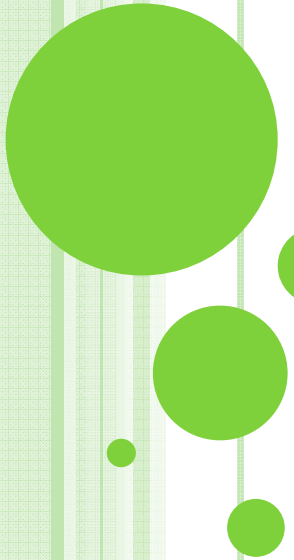
- Introdução
- Objetivos do trabalho
- Furbot
- Fundamentação teórica
- Trabalhos correlatos
- Desenvolvimento
- Conclusão
- Extensões

Introdução

- Problemas relacionados ao ensino
  - Falta de interesse
  - Complexidade
- Criação de framework
  - Interesse dos alunos por jogos
  - Oferecer desafios divertidos
- Utilização de dispositivos móveis
  - Smartphones e funcionalidades
  - iPhone é ideal para jogos

Objetivos do trabalho

- Converter framework desenvolvido em Java para Objective-C
- Mapear funcionalidades existentes no Furbot em Java para Objective-C
- Criar aplicações para iPhone utilizando o framework convertido

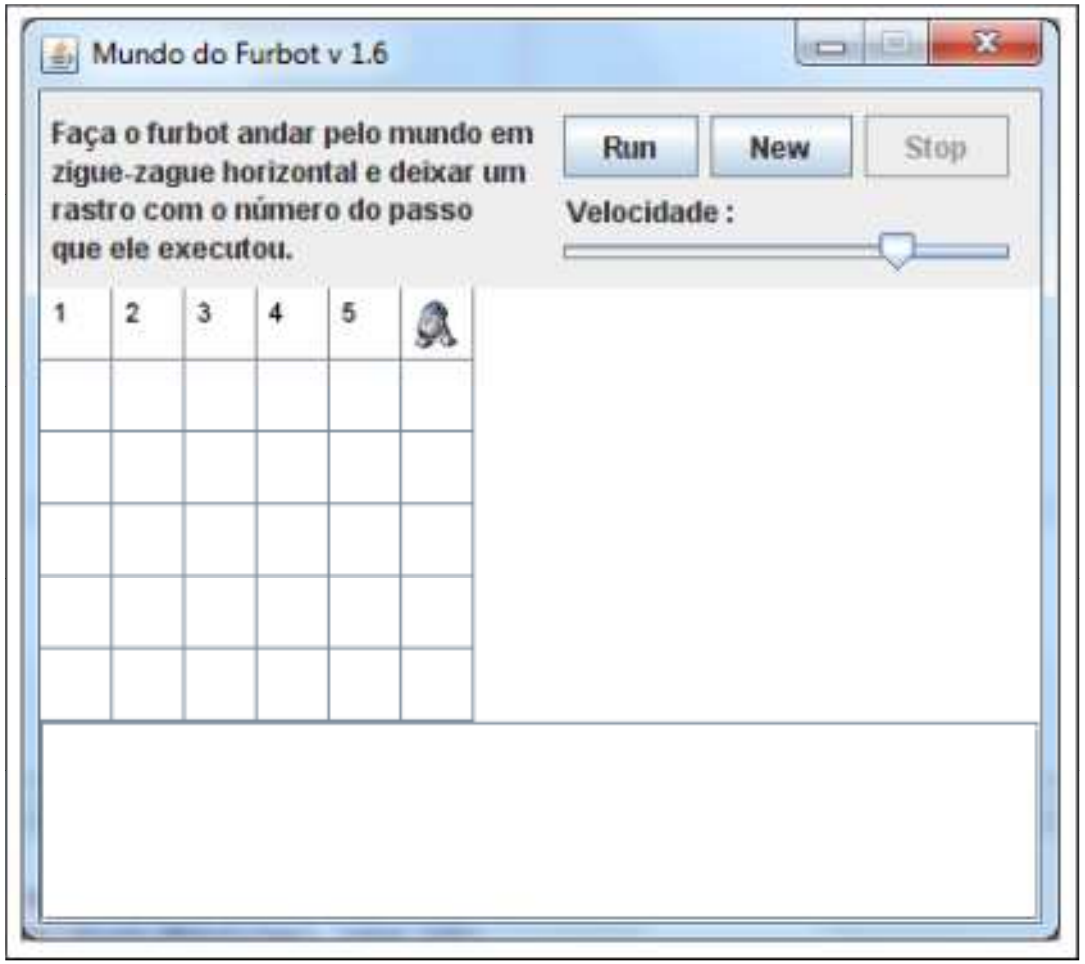


FURBOT

FURBOT

- O que é ?
  - Framework voltado para ensino de algoritmos
  - Permite a criação de aplicações com poucos comandos
  - Desenvolvido por professores da Furb
- Como funciona?
  - Interface consiste na representação de um mundo bidimensional
  - Aluno codifica objetos do mundo
  - Exercícios são repassados através de arquivos XML

FURBOT – INTERFACE DO MUNDO



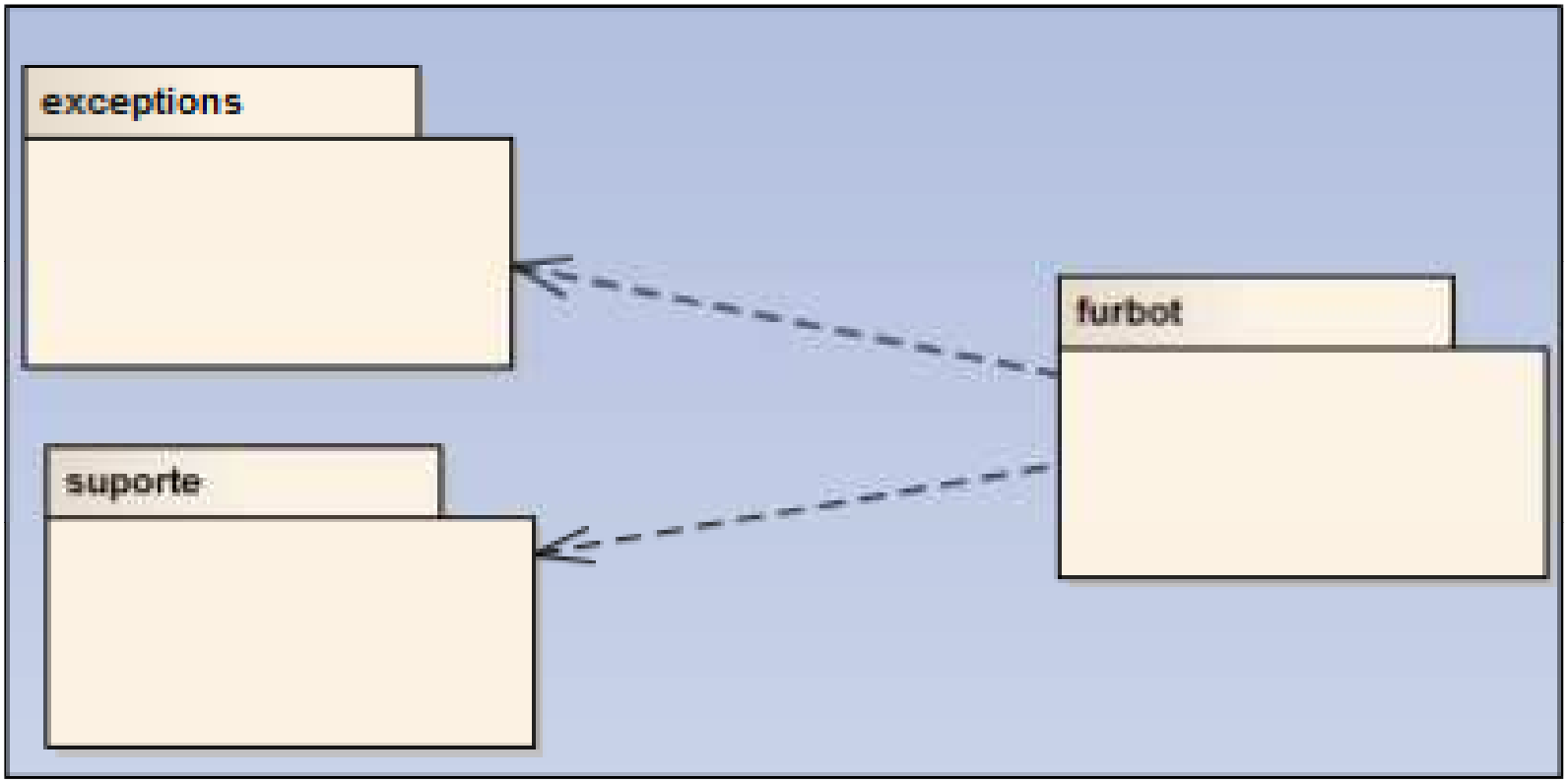
FURBOT – CODIFICAÇÃO DA LÓGICA DE CONTROLE

```
public void inteligencia() throws Exception {
    andarDireita(); // Movimenta o robo para direita
    diga("Andei para direita."); // Imprimi mensagem na console
    andarAbaixo(); // Movimenta o robo para baixo
    diga("Andei para baixo."); // Imprimi mensagem na console
}
```

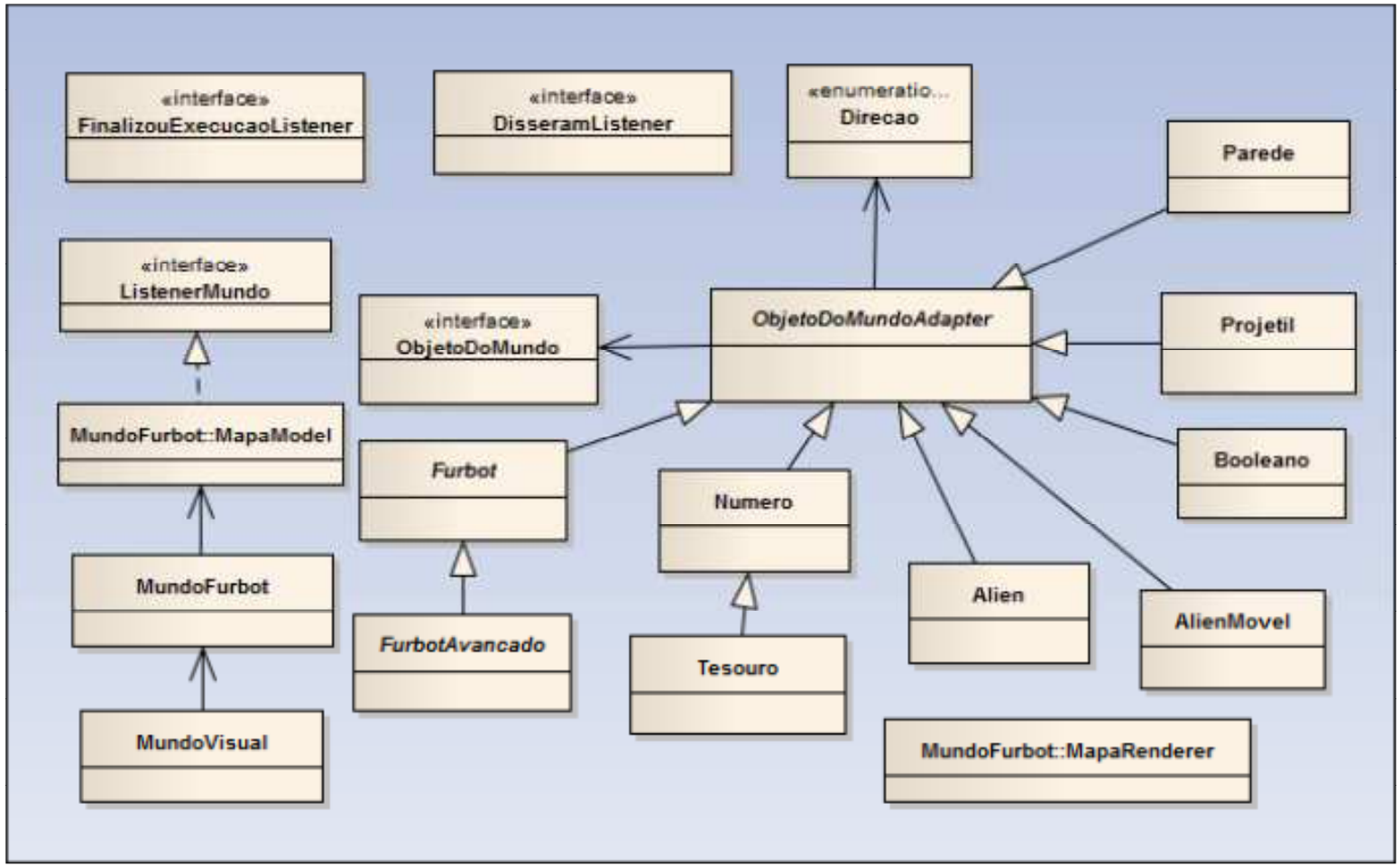
FURBOT – ESPECIFICAÇÃO DE EXERCÍCIO EM XML

```
<furbot>
   <enunciado>
    Exercicio 1. &lt;br&gt;
         Faca o robo andar ate a ultima posicao da linha. &lt;br&gt;
         Lembre-se de que as coordenadas sempre serao fornecidas como (x,y).&lt;br&gt;
         A primeira coluna e linhas sao a de numero ZERO.
         </enunciado>
         <mundo>
                  <qtdadeLin>8</qtdadeLin>
                  <qtdadeCol>8</qtdadeCol>
                  <explodir>true</explodir>
         </mundo>
         <robo>
                  <x>0</x>
                  <y>0</y>
         </robo>
</furbot>
```

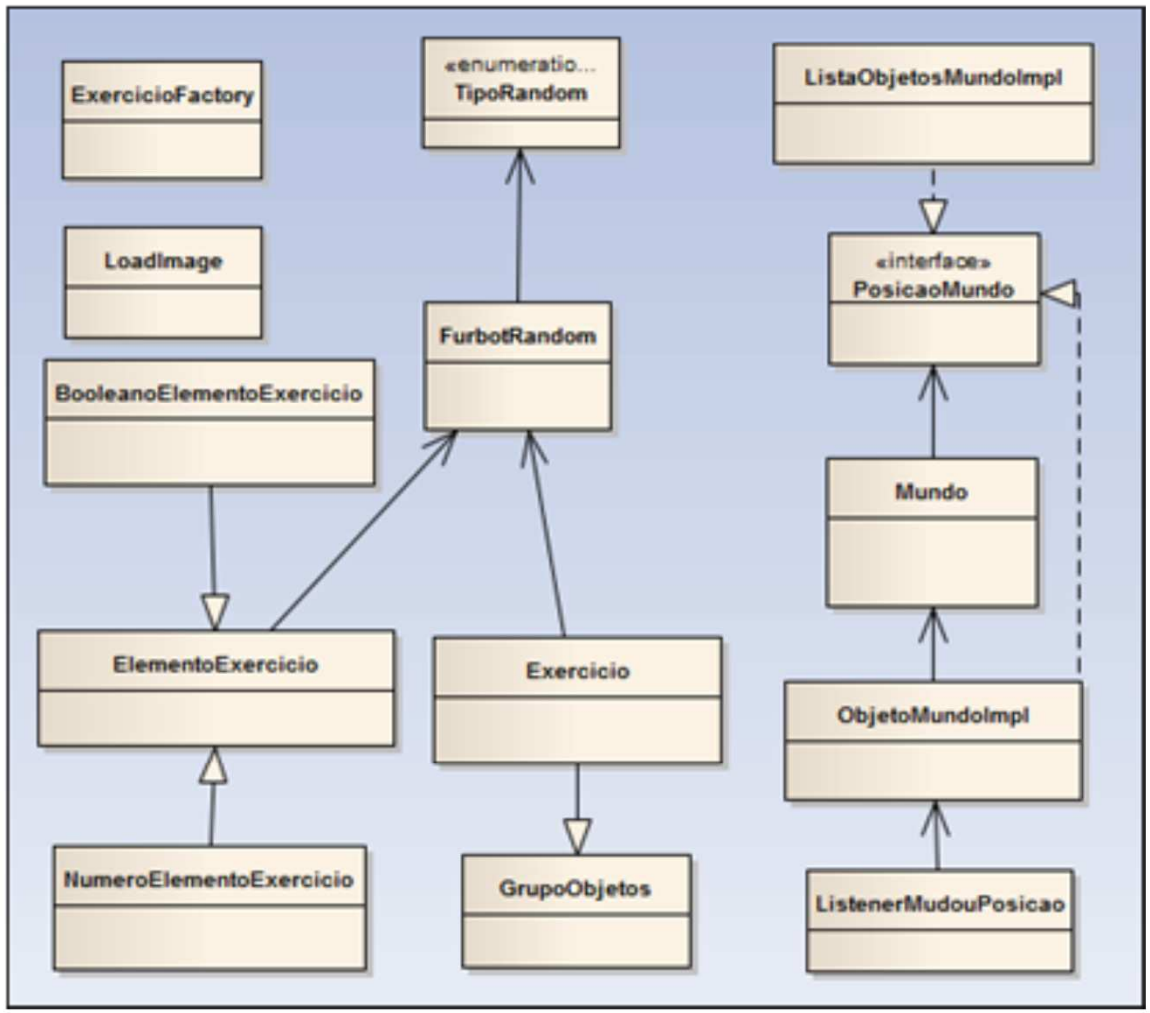
FURBOT - PACOTES



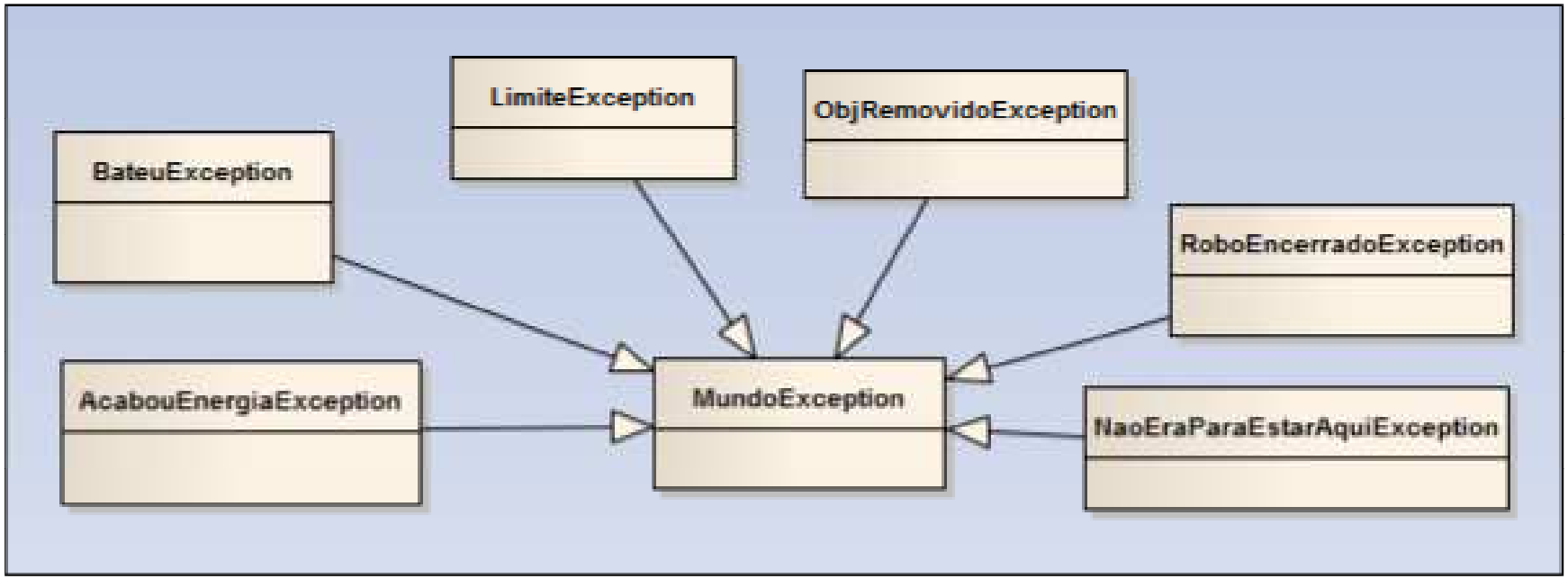
FURBOT – PACOTE FURBOT

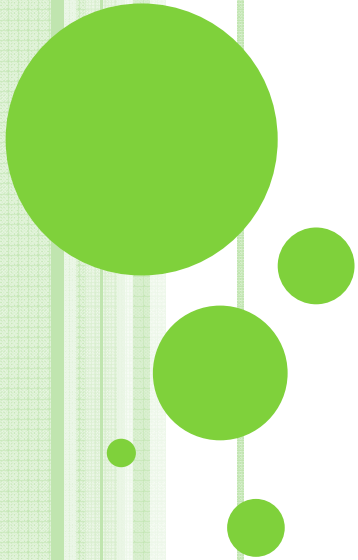


FURBOT – PACOTE SUPORTE



FURBOT – PACOTE EXCEPTION

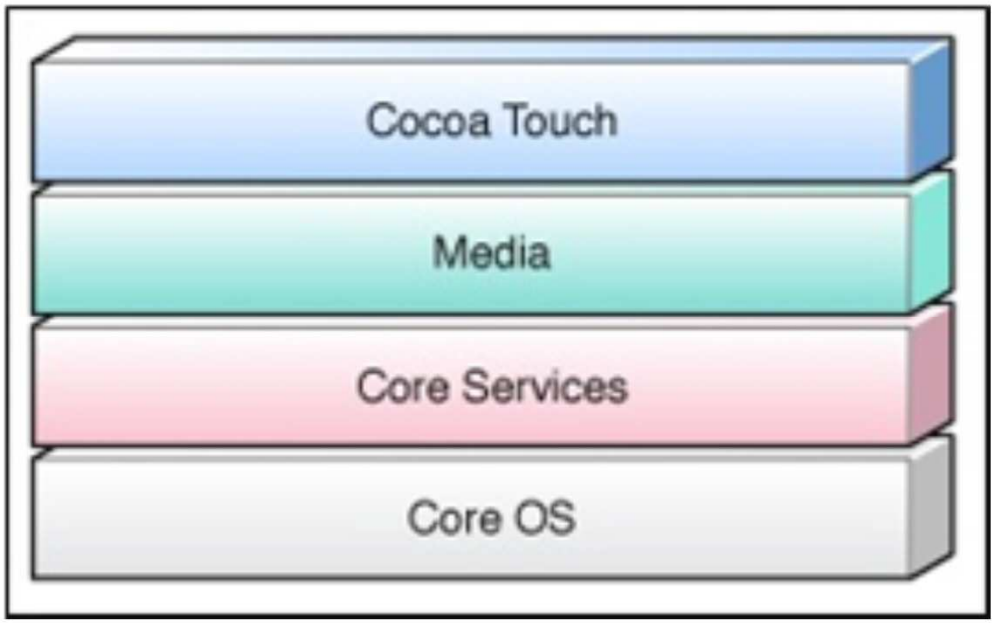


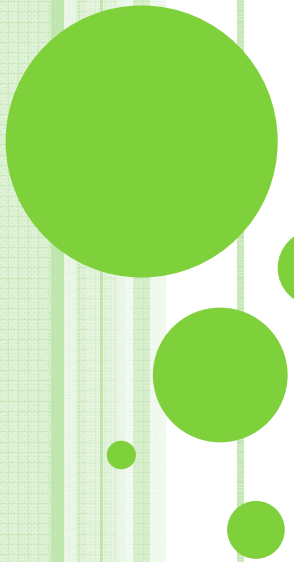


iPhone OS

iPhone OS

- Sistema operacional
- Utilizado pelos dispositivos iPhone e iPod Touch
- Fornece SDK para desenvolvimento
- Possui tecnologias divididas por camadas de abstração





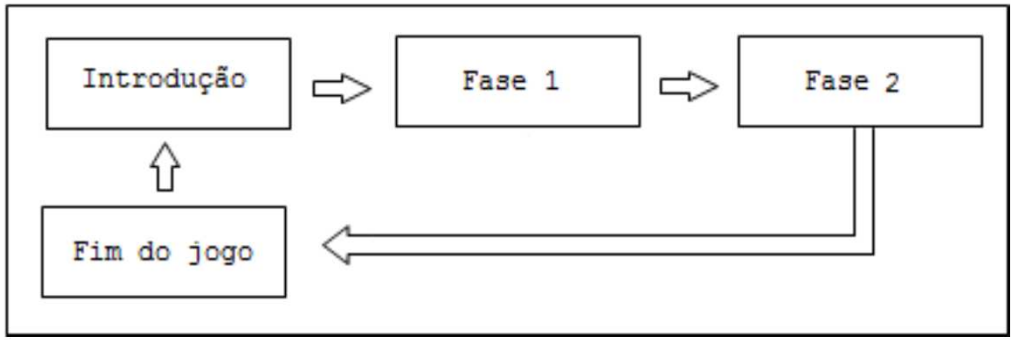
cocos2D

cocos2D

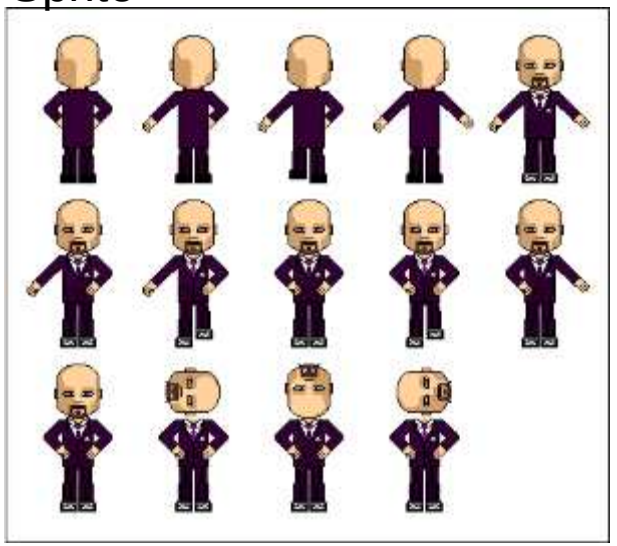
- O que é ?
- Projeto open-source
- Baseado em versão na linguagem Python
- Segue o padrão de desenvolvimento MVC
- Não permite multi-thread

COCOS2D – PRINCIPAIS CLASSES

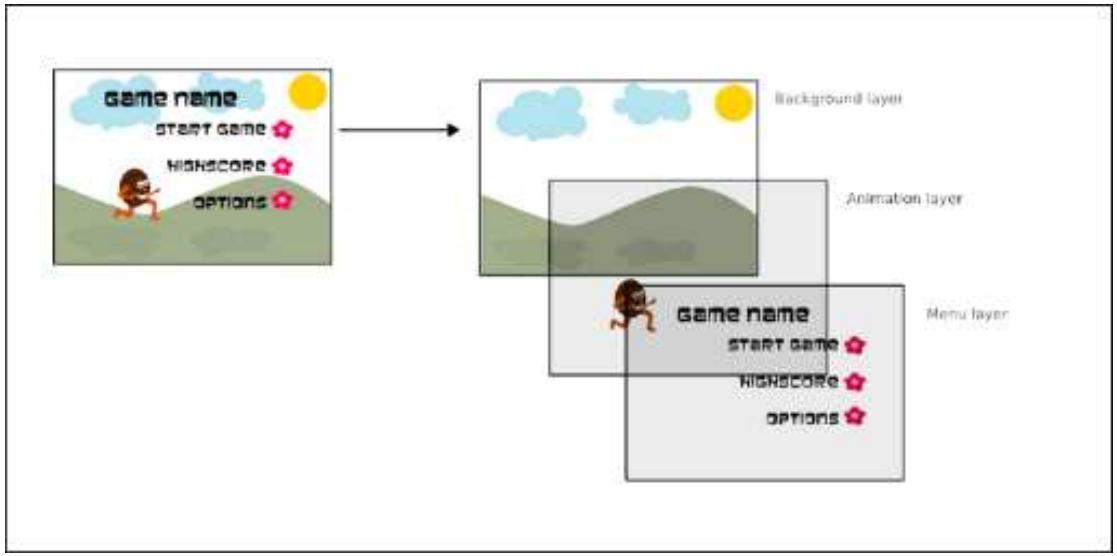
Scene



Sprite

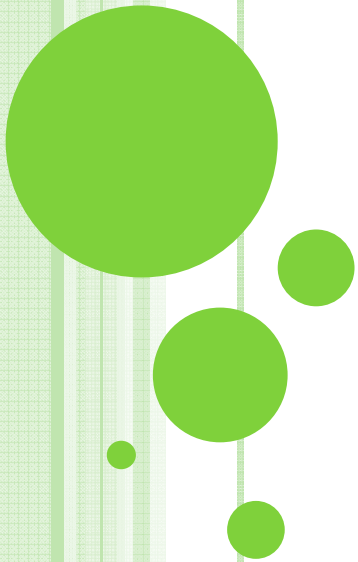


Layer



JAVA X OBJECTIVE - C

- Semelhanças
  - Baseadas em ANSI C
  - Orientação a objetos
  - Garbage collection
- Diferenças
  - Sintaxe
  - Construtores



Trabalhos correlatos

TRABALHOS CORRELATOS

Aplicação	Voltado para o ensino	Aprendizado de algoritmos	iPhone	Jogo
Greenfoot	X	X		
Gorillas		X	X	X
M-Geo	X		X	
Skattjäkt	X			X

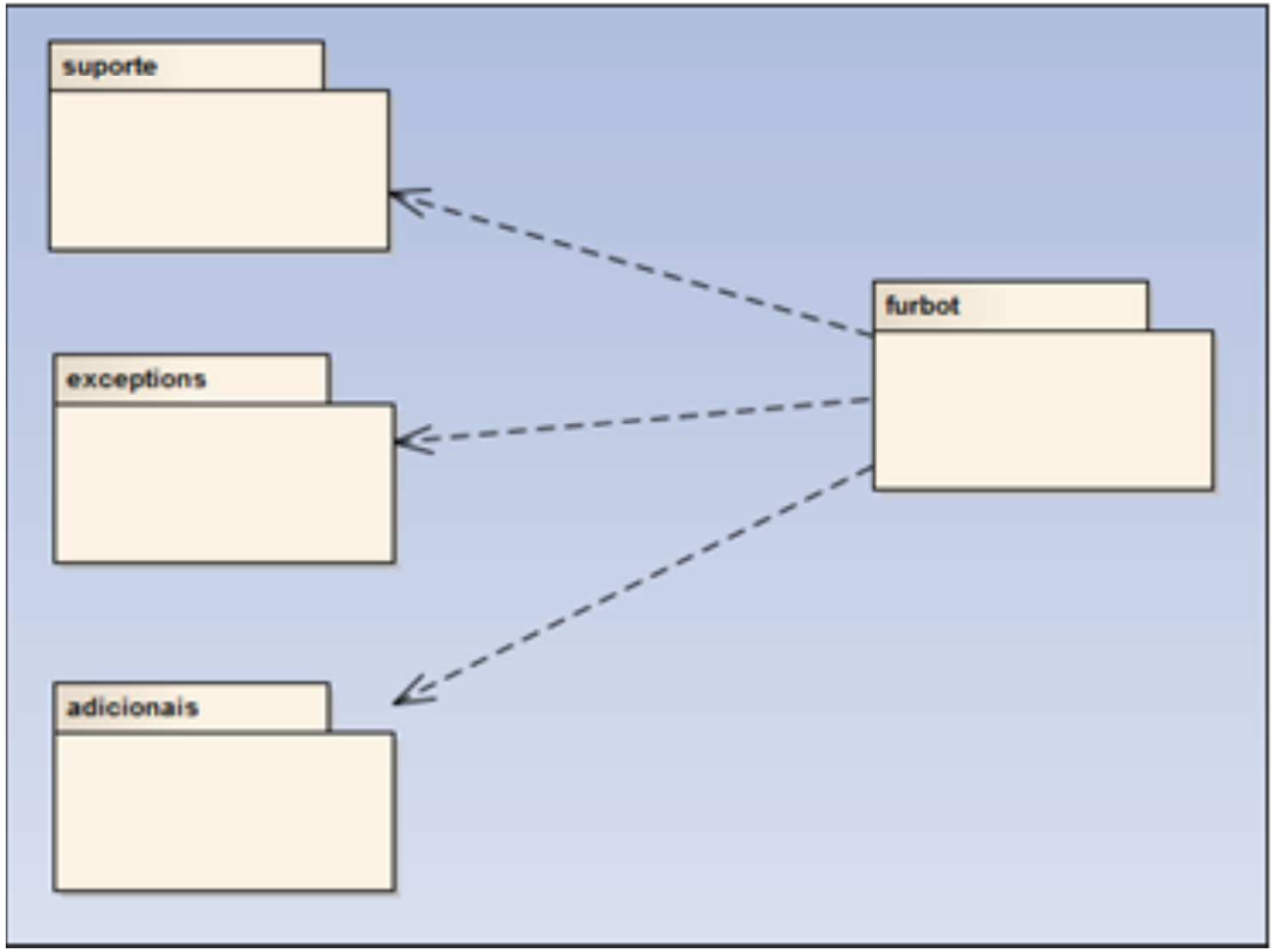
DESENVOLVIMENTO

DESENVOLVIMENTO

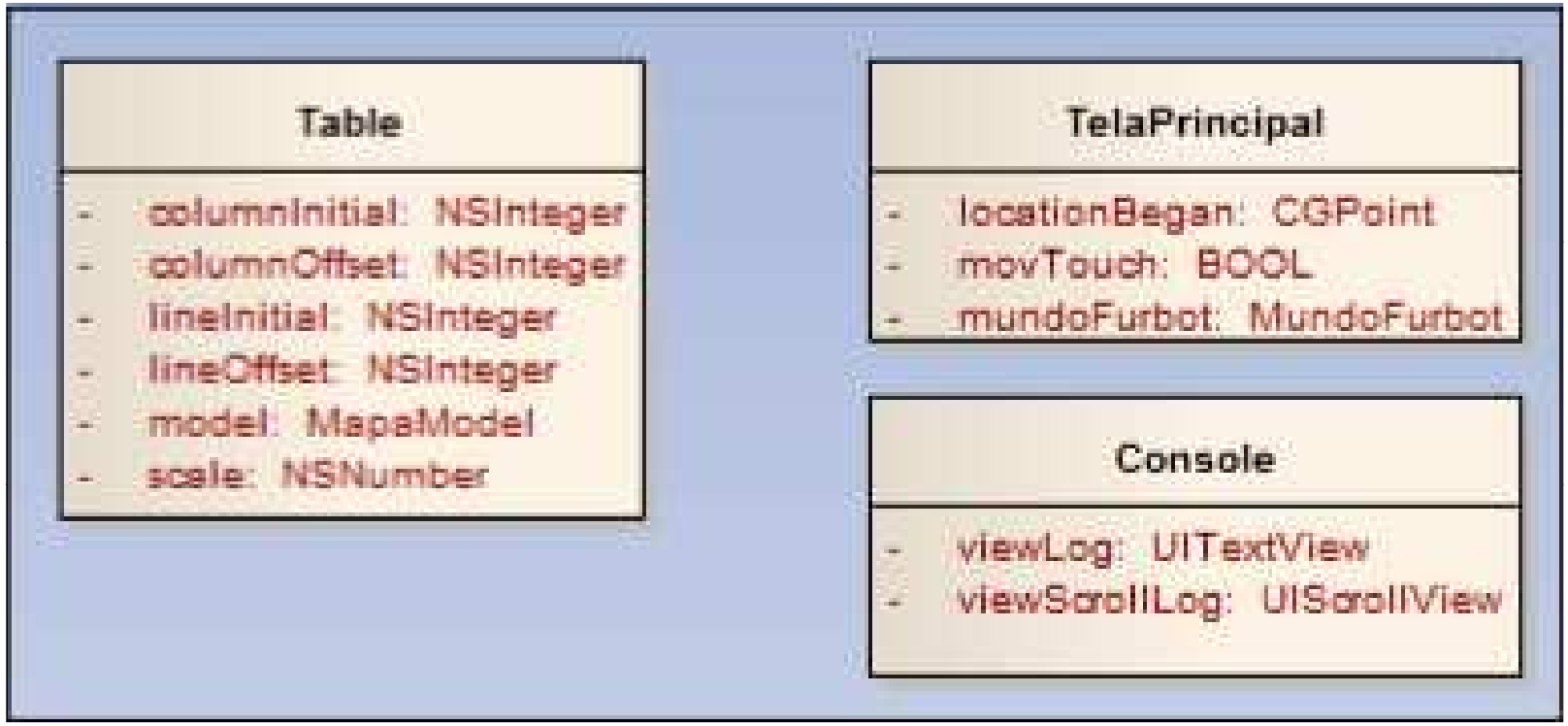
- Criar aplicações bidimensionais para iPhone
- Utilizar arquivos XML para armazenar configurações
- Mapear interações via teclado para interação de interface multi-touch
- Converter framework de Java para Objective-C
- Permitir executar aplicações desenvolvidas usando o iFurbot no iPhone

- Ferramentas utilizadas
  - XCode / Instruments / iPhone Simulator
  - cocos2d
  - Cocoa Touch / Core OS

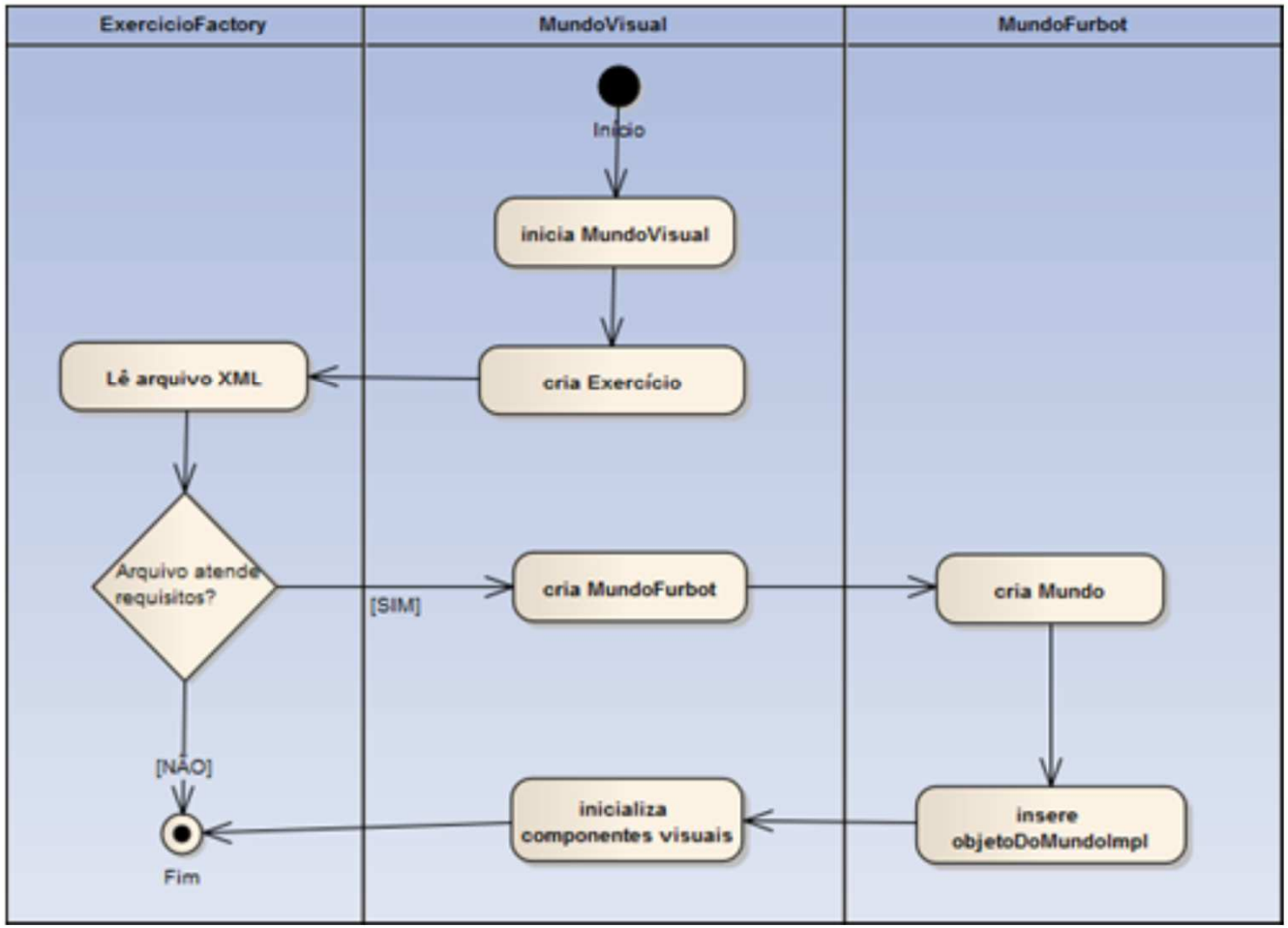
iFurbot – Pacotes



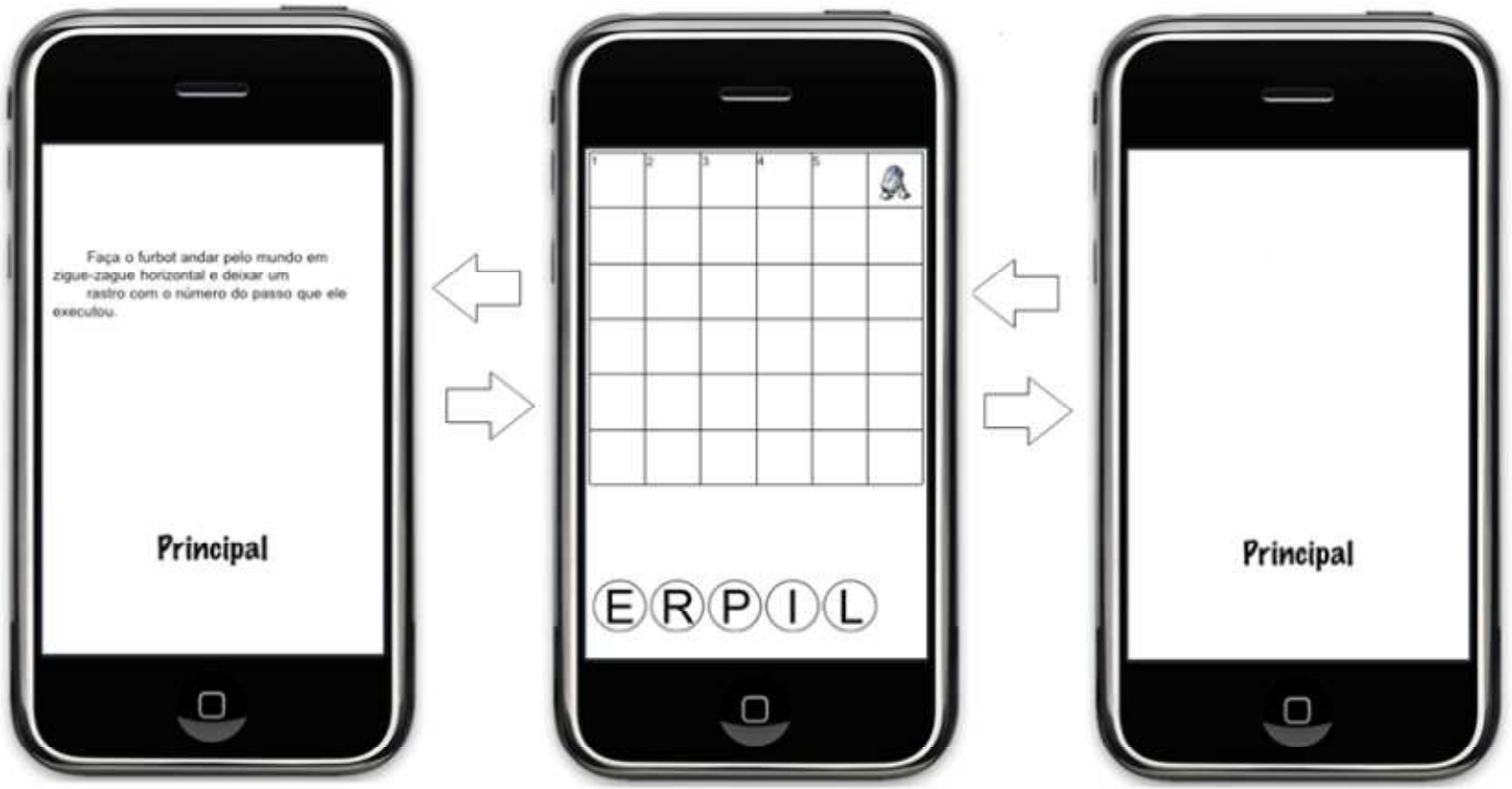
iFurbot – Pacote adicionais



iFurbot – Processo inicial



ACELERÔMETRO - INTERFACES



ACELERÔMETRO – TELA ENUNCIADO

```
-(void) atualizaPosicoesTelaEnunciado {
    CGSize dimensoes = [[Director sharedDirector] winSize];

    if ([[Director sharedDirector] deviceOrientation] == CCDeviceOrientationPortrait) {
        [menuTelaEnunciado setPosition: CGPointMake(dimensoes.width/2,100)];
        [enunciado setPosition: CGPointMake(dimensoes.width/2-50, 300)];
        [enunciado setContentSize: CGSizeMake(200,400)];
        [telaEnuciado changeWidth: dimensoes.width height: dimensoes.height];
    } else {
        [menuTelaEnunciado setPosition: CGPointMake(dimensoes.width/2,50)];
        [enunciado setPosition: CGPointMake(dimensoes.width/2+30,150)];
        [enunciado setContentSize: CGSizeMake(400,200)];
        [telaEnuciado changeWidth: dimensoes.width height: dimensoes.height];
    }
}
```

ACELERÔMETRO – TELA PRINCIPAL

```
-(void) atualizaPosicoesTelaPrincipal {

    CGSize dimensoes = [[Director sharedDirector] winSize];

    if ([[Director sharedDirector] deviceOrientation] == CCDeviceOrientationPortrait) {
        [menuTelaPrincipal alignItemsHorizontallyWithPadding: 4];
        [menuTelaPrincipal setPosition: ccp(dimensoes.width-180,50)];
        [telaPrincipal changeWidth: dimensoes.width height: dimensoes.height];
    } else {
        [menuTelaPrincipal alignItemsVerticallyWithPadding: 4];
        [menuTelaPrincipal setPosition: ccp(dimensoes.width-30,150)];
        [telaPrincipal changeWidth: dimensoes.width height: dimensoes.height];
    }
}
```

ACELERÔMETRO – TELA CONSOLE

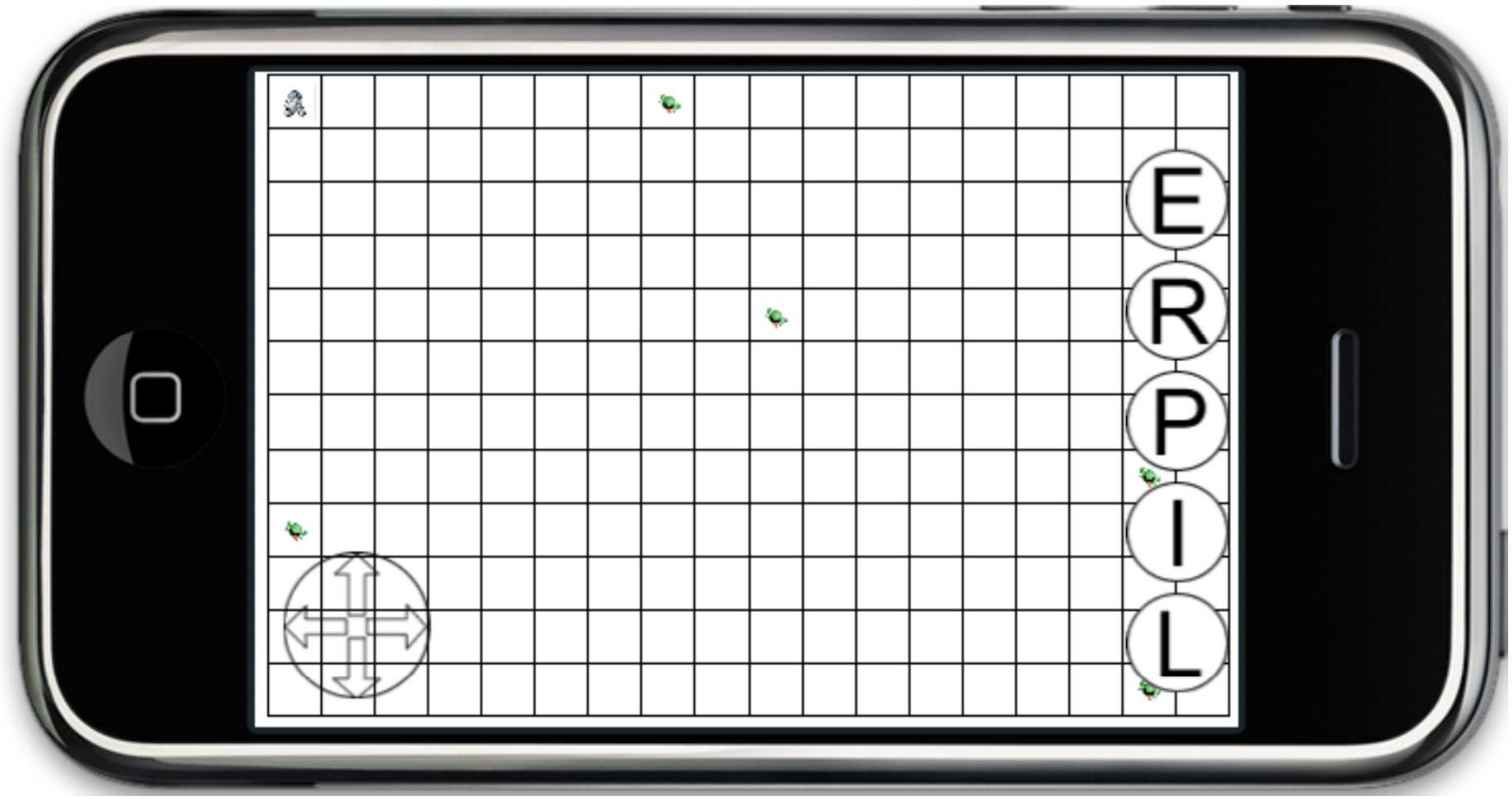
```
- (void)atualizarPosicoes {

    CATransform3D rotationTransform = CATransform3DIdentity;
    [console.layer removeAllAnimations];

    UIDeviceOrientation interfaceOrientation = [[UIDevice currentDevice] orientation];
    CGSize dimensoes = [[Director sharedDirector] winSize];

    CGRect contentRect = CGRectMake(0, 0, dimensoes.width, dimensoes.height);
    console.bounds = contentRect;

    if (interfaceOrientation == UIInterfaceOrientationPortrait)
        rotationTransform = CATransform3DRotate(rotationTransform, 3.14159*2, 0.0, 0.0, 1);
    else if (interfaceOrientation == UIInterfaceOrientationLandscapeRight)
        rotationTransform = CATransform3DRotate(rotationTransform, 3.14159/2, 0.0, 0.0, 1);
    else
        rotationTransform = CATransform3DRotate(rotationTransform, 3.14159+(3.14159/2), 0.0, 0.0, 1);

    console.layer.transform = rotationTransform;
    botaoTelaPrincipal.frame = CGRectMake(dimensoes.width/2-80, dimensoes.height-40, 150, 30);
    viewLog.frame = CGRectMake(0, 0, dimensoes.width, dimensoes.height-50);

}
```

FUNCIONALIDADE DE DIRECIONAL

```
+(void) usarTeclas {
    [[Director sharedDirector] setDeviceOrientation: UIInterfaceOrientationLandscapeLeft];
    [euMesmo orientationChanged: nil];
    [[euMesmo mundoFurbot] desativarAutoRotate];
    [euMesmo ativarDirecional];
    [[NSNotificationCenter defaultCenter] removeObserver: self];
}
```

```
-(BOOL) verificaTeclaPressionada:(NSInteger) tecla pontoPressionado:(CGPoint) ponto {
    CGRect rect;

    switch (tecla) {
        case ACIMA:
            rect = CGRectMake(47, 55, 9, 30);
            break;
        case ABAIXO:
            rect = CGRectMake(47, 15, 9, 30);
            break;
        case ESQUERDA:
            rect = CGRectMake(16, 45, 30, 9);
            break;
        case DIREITA:
            rect = CGRectMake(56, 45, 30, 9);
            break;
        default:
            break;
    }
    return CGRectContainsPoint(rect, ponto);
}
```

USANDO O DIRECIONAL



VELOCIDADE DA EXECUCAÇÃO

```
-(BOOL) ccTouchesBegan:(NSSet *)touches withEvent:(UIEvent *)event {
    UITouch* touch = [touches anyObject];
    locationBegan = [[Director sharedDirector] convertCoordinate: [touch locationInView: touch.view]];

    if (botaoDirecional != nil) {
        if ([self verificaTeclaPressionada: ACIMA pontoPressionado: locationBegan])
            [self pressionadaTecla: ACIMA];
        else if ([self verificaTeclaPressionada: ABAIXO pontoPressionado: locationBegan])
            [self pressionadaTecla: ABAIXO];
        else if ([self verificaTeclaPressionada: ESQUERDA pontoPressionado: locationBegan])
            [self pressionadaTecla: ESQUERDA];
        else if ([self verificaTeclaPressionada: DIREITA pontoPressionado: locationBegan])
            [self pressionadaTecla: DIREITA];
    }

    return kEventHandled;
}
```

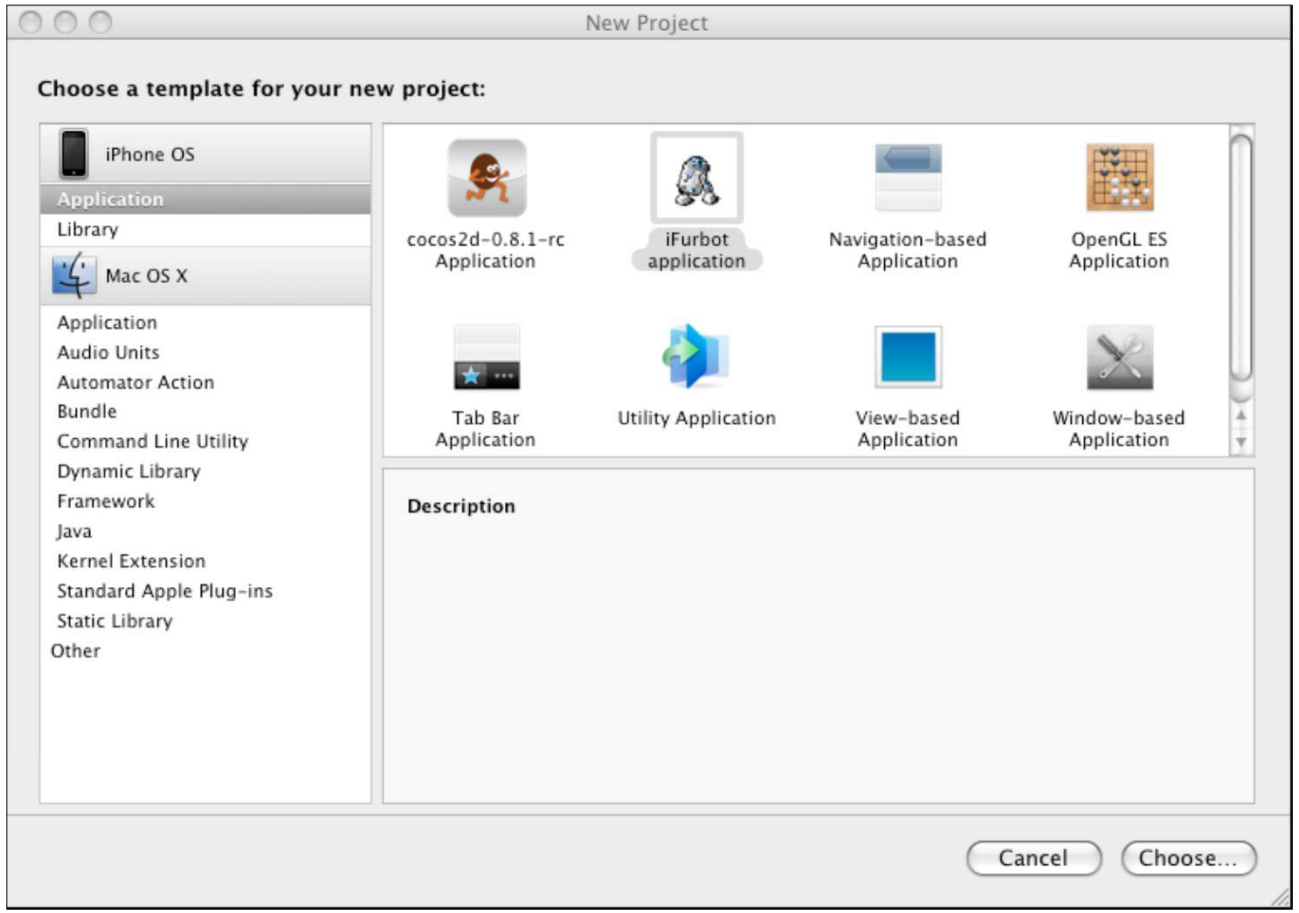
VELOCIDADE DA EXECUCAÇÃO

```
-(BOOL) ccTouchesEnded:(NSSet *)touches withEvent:(UIEvent *)event {
    UITouch* touch = [touches anyObject];
    CGPoint locationEnded = [[Director sharedDirector] convertCoordinate: [touch locationInView: touch.view]];

    if (movTouch) {
        double tempo = [mundoFurbot tempoEspera];

        if (fabs(locationBegan.y-locationEnded.y) < offSetMoveTouchY) {
            if (locationEnded.x > locationBegan.x+offSetMoveTouchX) {// right
                if (tempo-offSetTempoEspera < 0)
                    [mundoFurbot setTempoEspera: 0];
                else
                    [mundoFurbot setTempoEspera: tempo-offSetTempoEspera];
            } else if (locationEnded.x < locationBegan.x-offSetMoveTouchX) {
                [mundoFurbot setTempoEspera: tempo+offSetTempoEspera];
            }
        }
        movTouch = FALSE;
    } else {

    }
    return kEventHandled;
}
```

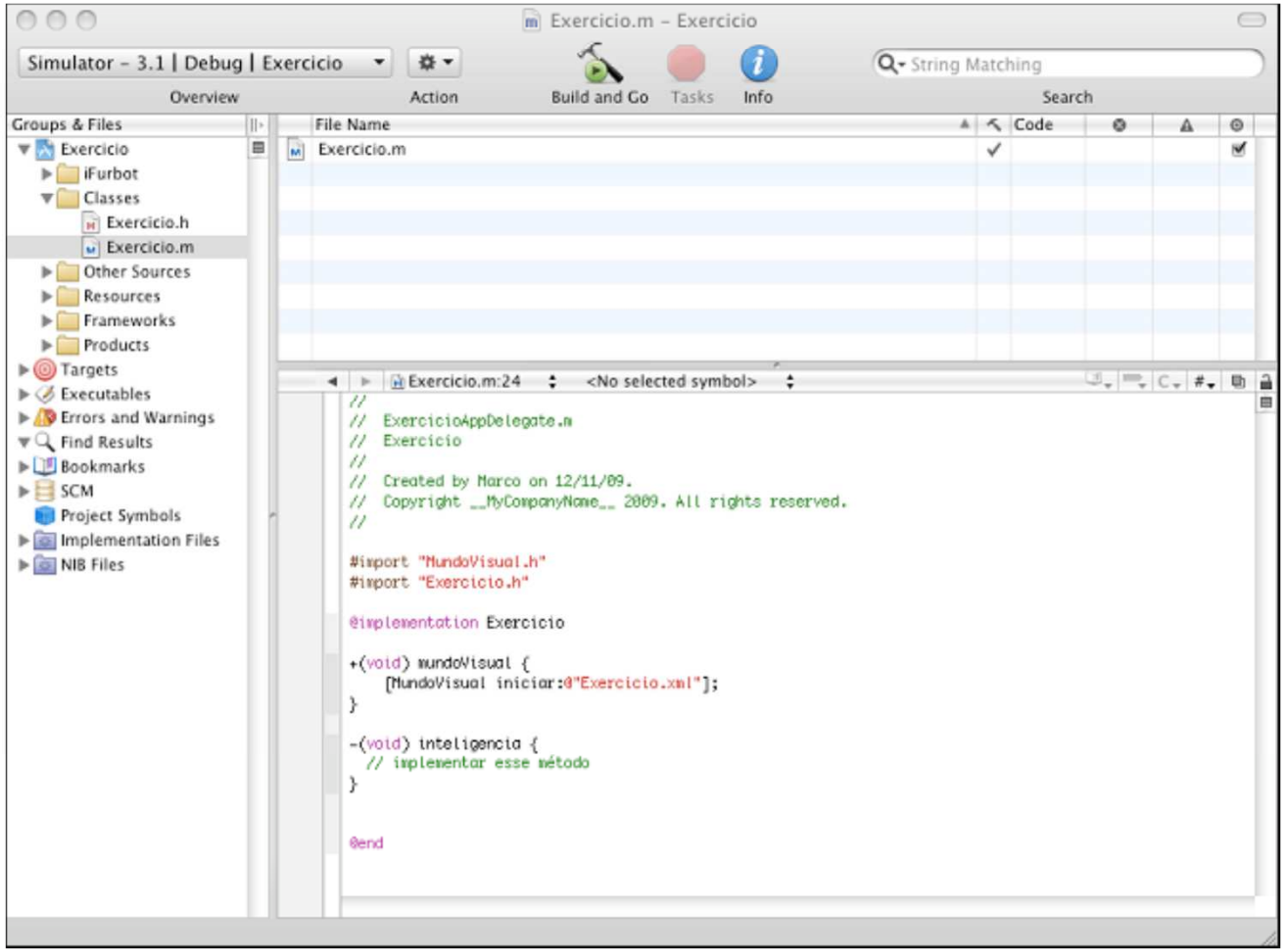
iFurbot x Trabalhos correlatos

Aplicação	Voltado para o ensino	Aprendizado de algoritmos	iPhone	Jogo
Greenfoot	X	X		
Gorillas			X	X
M-Geo	X		X	
Skattjäkt	X			X
Sist. de gerenc. de objetos de aprendi. para dispositivos móveis	X			
iFurbot	X	X	X	

TEMPLATE PARA CRIAÇÃO DE PROJETO



TEMPLATE PARA CRIAÇÃO DE PROJETO



Resultado e discussão

- Criação de novas classes para atender os requisitos
- Gerenciamento manual de memória
- Controle multi-thread
- Mantida nomenclatura das classes e métodos
- Mudança da finalidade da classe MundoVisual
- Uso de cocos2d permitiu maior abstração de classes visuais

Conclusão

- Framework totalmente convertido
- Requisitos levantados foram alcançados
- Utilizada ferramenta ObjectAlloc para corrigir problemas de vazamento de memória
- Utilizada aplicação Gorillas como fonte de informações sobre cocos2d
- Limitações que existem ao utilizar o iFurbot

Extensões

- Maior integração do iFurbot com recursos oferecidos pelo framework cocos2d
- Oferecer a funcionalidade de multi-jogadores utilizando o recurso de multi-touch
- Adicionar eventos no mundo do iFurbot como terremotos utilizando o recurso de acelerômetro
- Permitir que o aluno possa ativar e desativar as funções de acordo com o que for solicitado no exercício