

FACULDADE INTEGRADA DA GRANDE FORTALEZA  
Pós-Graduação em Engenharia de Software



MARCOS AMARAL RODRIGUES

METODOLOGIA DE DESENVOLVIMENTO DE JOGOS  
COMPUTACIONAIS PARA PROGRAMADORES EM JAVA

SALVADOR  
2011

MARCOS AMARAL RODRIGUES

METODOLOGIA DE DESENVOLVIMENTO DE JOGOS  
COMPUTACIONAIS PARA PROGRAMADORES EM JAVA

MARCOS AMARAL RODRIGUES

**METODOLOGIA DE DESENVOLVIMENTO DE JOGOS  
COMPUTACIONAIS PARA PROGRAMADORES EM JAVA**

Monografia Apresentada à FACULDADE  
INTEGRADA DA GRANDE FORTALEZA  
como requisito parcial para obtenção do  
título de especialista em Engenharia de  
Software. Área de Concentração:  
Tecnologia da Informação.

ORIENTADORA:  
Ludmilla Flôres Meneses Lima.

SALVADOR  
2011

MARCOS AMARAL RODRIGUES

METODOLOGIA DE DESENVOLVIMENTO DE JOGOS  
COMPUTACIONAIS PARA PROGRAMADORES EM JAVA

Monografia Apresentada à FACULDADE  
INTEGRADA DA GRANDE FORTALEZA  
como requisito parcial para obtenção do  
título de especialista em Engenharia de  
Software. Área de Concentração:  
Tecnologia da Informação.

ORIENTADOR:  
Ludmilla Flôres Meneses Lima.

Monografia julgada e aprovada:

Prof(a). Orientadora:  
Ludmilla Flôres Meneses Lima.

## DEDICATÓRIA PARA AGORA E PARA O FUTURO.

Dedico este trabalho à todos que desejam que o Brasil se torne um grande concentrador de programadores para gerar empresas de desenvolvimento de softwares, trazendo maior prosperidade a esta nação e a meu filho Marcos David para que ele seja um programador de linha de frente na sua geração.

## **Agradecimentos**

Agradeço ao Senhor Jesus porque tem me dado sabedoria e inteligência no meu caminhar...também a minha esposa Ana Lúcia, a meu filho Marcos David e a meus pais e irmãos.

## Resumo

A metodologia serve como uma bússola para a construção de um sistema na Engenharia de Software. Podendo ser aplicada em várias equipes de desenvolvimentos de softwares. Otimizando os processos de criação de sistemas. E no ambiente de desenvolvimento de jogos computacionais, a regra é a mesma. Para melhor entendimento em desenvolvimento de Jogos Computacional (JC), pontos como: conceito, história, características, linguagens de programação usadas para o desenvolvimento de jogos, motores de jogos, Inteligência Artificial (IA), JC como forma educacional. É abordado o assunto, Orientação à Objetos (OO) com ênfase a linguagem Java. Na parte de Gestão de Projeto de Software, aparece a figura do engenheiro de software para que as atividades sejam implementadas em tempo hábil.

Palavras-chave: jogos, computador, metodologia.

## Tabela de Figura

FIGURA 2.1: Atari 2600.....	16
FIGURA 2.2: Ace Hack.....	17
FIGURA 2.3: Xbox, Microsoft.....	18
FIGURA 2.4: Playstation, Sony.....	18
FIGURA 2.5: Imagem do motor de jogo Doom 3 Engine.....	22

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	12
1.1 Título.....	12
1.2 Tema.....	12
1.3 Problema.....	13
1.4 Justificativa da Pesquisa.....	13
1.5 Objetivos .....	14
1.5.1 Geral.....	14
1.5.2 Específicos.....	14
<b>2 REVISÃO DA LITERATURA</b> .....	16
2.1 Relatar a história dos jogos computacionais.....	16
2.1.1 Conceito de Jogos.....	16
2.1.2 História.....	16
2.1.3 Indústria Bilionária.....	18
2.1.4 Característica.....	19
2.1.5 Jogos por Rede.....	19
2.1.6 CompuServe.....	19
2.1.7 Internet.....	20
2.1.8 Linguagens Computacional.....	20
2.1.9 Motores de Jogos.....	21
2.1.10 Inteligência Artificial em Jogos Eletrônicos.....	22
2.1.11 A Evolução dos Jogos Computacionais de Entretenimento para Educacional.....	23
2.1.12 Não Perdendo o Foco.....	23
2.2 Conceituar a Programação Orientada à Objetos (POO), utilizando a linguagem Java.....	24
2.2.1 História da OOP.....	24
2.2.2 Conceito da OOP.....	24
2.2.3 Conceitos que compõem a POO.....	25
2.3 Explicar a Gestão em Projetos de Softwares em Jogos Computacionais.(JC)...	31

2.3.1 O Gestor do Projeto de Software em JC.....	31
2.3.2 Relacionar a aplicação das metologias de produção de softwares à produção de games informáticos.	32
2.3.3 A figura do Game Design.....	34
2.4 Entrevista com um programador sobre o desenvolvimento de “Asteroides”.....	35
2.4.1 Introdução .....	35
2.4.2 Relato de um programador e professor do Professordelphi, Salvador-BA, (18 Dezembro 2010 19:12:57 ).....	35
2.4.3 Sobre a entrevista.....	37
<b>3 CONSIDERAÇÕES FINAIS.....</b>	<b>37</b>
<b>4 METODOLOGIA .....</b>	<b>38</b>
<b>5 REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>39</b>

## 1 INTRODUÇÃO

### 1.1 Jogos Computacionais em Java.

O desenvolvimento de jogos é uma tarefa desconhecida de muitos programadores até mesmo entre os experientes. Isso se deve ao escasso material didático sobre o assunto. E também ao grau de complexidade na montagem de um jogo. Sendo assim, nesta monografia será abordada a metodologia em desenvolvimento de jogos para ser aplicada em Jogos Computacionais (JC) na linguagem Java. Linguagem que tem um grau de complexidade, porém multi-plataforma, ou seja, permite a sua aplicação em vários sistemas operacionais. Sendo um fator de atração para vários programadores para desenvolver em JC, pois eles sabem que o software pode rodar em várias plataformas.

### 1.2 Metodologia de Desenvolvimento de Jogos Computacionais para programadores em Java.

Existem metodologias para desenvolvimento de softwares, usadas principalmente para sistemas corporativos, como é o caso do *Rational Unified Process* (RUP) e *Extreme Programming* (XP). No caso de sistemas informáticos no perfil de jogos, existe uma insegurança por parte da maioria dos programadores de qual metodologia aplicar. Na verdade, jogo computacional é um software e por isso deve obedecer os padrões de desenvolvimento para sistemas computacionais.

Nesta monografia a ênfase é para programadores da linguagem de programação Java. Sendo uma Linguagem Orientada à Objeto (OO), que pode trabalhar em vários sistemas operacionais como: Linux, Windows e outros. Teve o seu lançamento na década de 90, com o propósito de atender a demanda da plataforma Web. Embora que, a sua utilização pode ser aplicada no ambiente *Desktop*, ou seja, em um ambiente local. Com o avanço da tecnologia no ambiente *mobile*, que envolve os celulares, *palm-tops* e todo aparelho que tem como característica a mobilidade, a Sun, fabricante do Java, desenvolveu uma biblioteca específica, chamada plataforma J2ME, *Java Mobile Edition*. Atualmente, esta linguagem está sendo usada também para desenvolvimento de jogos.

### 1.3 Qual a metodologia em desenvolvimento de software indicada para um projeto de jogo computacional?

Uma boa indicação seria a *Extreme Programming*, pois permite visualizar o desenvolvimento do jogo com algo que pode ser testado. Embora que, haja a existência de outras metodologias, como a já citada, RUP e também a *Team Software Process* (TSP) .

No desenvolvimento de um software, o engenheiro de software precisa escolher qual o método que irá aplicar na construção do sistema, pois isso irá interferir no ciclo de vida do sistema, mesmo sendo, um jogo computacional. Pois, programadores, por falta de experiência na área de desenvolvimento de JC, mostram ter dificuldade na aplicação de técnicas na construção de JC.

A experiência dos programadores está mais ligada a construção de um ERP (*Enterprise Resource Planning*) ou SIGE (Sistemas Integrados de Gestão Empresarial, no Brasil), sistema que visa a informatização completa de uma empresa, do que a criação de JC.

Com as suas peculiaridades, principalmente, no que diz respeito à equipe que compõe o desenvolvimento do jogo. Pode-se afirmar que, a pessoa que vai projetar o jogo é o grande diferencial em relação a uma equipe de desenvolvimento de sistemas tradicionais. Dando destaque, para o *game design*, o profissional que vai idealizar o jogo e que dará jogabilidade ao *game*.

As similaridades com a metodologia de desenvolvimento de sistema e as diferenças entre, desenvolvimento de um software comum e um jogo computacional, servirão para melhor esclarecer o leitor sobre o assunto.

## 1.4 Justificativa

A motivação deste trabalho acadêmico, deve-se ao fato de que está sendo implantado o curso de Jogos Computacionais, em Java, no Professordelphi, empresa de capacitação e desenvolvimento de softwares, localizada em Salvador, na Bahia. Unida ao fato de que nesta área o material de consulta em metodologia de desenvolvimento de jogos não é abundante. Portanto essa monografia permitirá que programadores tradicionais apliquem as metodologias de desenvolvimento de software em desenvolvimento de JC com mais segurança. Na parte do Ensino, pode ser consultado este trabalho científico, pelos alunos da área de concentração da Tecnologia da Informação e simpatizantes em desenvolvimento de jogos para também conhecer as metodologias aplicadas ao desenvolvimento de JC.

## 1.5 Objetivo

### 1.5.1 Geral

Discutir os métodos aplicados no desenvolvimento de JC. Na verdade, os mesmos conceitos de Ciclo de Vida de Sistemas e metodologias de desenvolvimento, serão vistos para aplicá-los no desenvolvimento do *game*. Essa compreensão poderá aumentar o número de engenheiros de software para gerir esse segmento de programação.

### 1.5.2 Objetivos Específicos

Neste trabalho acadêmico será visto, em cinco capítulos, a vantagem da utilização de técnicas de desenvolvimento de sistemas. Tendo o objetivo de conhecer os seus benefícios e também os perigos de não utilizar um modelo de desenvolvimento de software. O objeto de estudo será focado na linguagem Java, devido ser orientada à objeto e multi-plataforma, pois trabalha tanto no Linux como Windows, além de poder operar em rede local, na Web e em aparelhos móveis, como celular.

Como objetivos específicos a serem alcançados nesta pesquisa, podem ser citados:

RODRIGUES, Marcos Amaral. **METODOLOGIA DE DESENVOLVIMENTO DE JOGOS COMPUTACIONAIS PARA PROGRAMADORES EM JAVA**. Salvador: Professordelphi, 2011.

- Relatar a história dos jogos computacionais, identificando as suas características.

A história dos primeiros jogos computacionais será abordada para que o leitor possa perceber a evolução deste mercado bilionário, ultrapassando inclusive a indústria do cinema.

- Conceituar a Programação Orientada à Objetos (POO), utilizando a linguagem Java.
- Explicar a Gestão em Projetos de Softwares em Jogos Computacionais.
- Entrevista sobre o desenvolvimento de “Asteroides” com um programador de JC.

## 2 REVISÃO DA LITERATURA

### 2.1 Relatar a história dos jogos computacionais.

#### 2.1.1 Conceito de Jogos

O conceito de jogo difere do conceito de esporte. Segundo Marcelo e Pescuite(2009, p.01), um jogo tem regras para um ambiente restrito podendo ser aplicadas de imediato, já o esporte tem regras universais. Um jogo também é uma competição possuindo regras simples ou até mesmo complexas. Dentro dos segmentos de jogos, existe segmento dos eletrônicos que podem ser divididos em: videogames ou console e os jogos de computador.

#### 2.1.2 História

A década de 80, foi um período de milhões de pessoas que jogaram o Atari, figura 2.1. Sendo um console para jogos com cartucho. Neste mesmo período, foi lançado no mercado os sistemas operacionais da Apple e da Microsoft, dando início a revolução do computador pessoal (PC). Nestes ambientes, os jogos para PC começaram a ser desenvolvidos, o que ficaram bastante limitados à quantidade de PC existentes nas residências e em empresas, que começaram a adotar a era PC. Pois, ainda não existia a Internet com é conhecida atualmente. Com o advento da rede mundial, os jogos computacionais deram um salto em termo de tecnologia, utilizando o realismo como característica, motivando os jogadores, que podem participar de partidas a nível planetário.



Figura 2.1: Atari 2600 (Fonte: Wikipédia, 2010).



Figura 2.2: Ace Hack (Fonte: Iplay.com, 2010)

A empresa Atari inc. está investindo na âmbito da Internet, isso é conferido no site, <http://www.atari.com/> . No Brasil, a existência de um site que traz jogos do Atari, agora a nível computacional, <http://www.iplay.com.br/> , trabalha com a nostalgia do público amante dos jogos do Atari, na figura 2.2, o Ace Hack.

A história dos jogos para computador, nasceu em meio ao crescimento dos jogos de console. Segundo Marcelo e Pescuite(2009,p.31), na década de 80 os jogos começaram a passar para o ambiente computacional. Nesta mesma época, o Atari, um jogo de console, que era ligado ao televisor e utilizava joysticks, controle de mão para controlar o movimento necessário no jogo, crescia de forma exponencial em todo mundo, batendo recordes de vendas. Embora que, o jogo de console foi forte concorrente para os jogos computacionais nos seus primórdios, a história mostrou que havia espaço para ambos crescerem paralelamente. Atualmente o XBOX, figura 2.3, da Microsoft e o Playstation, figura 2.4, da Sony, são líderes de vendas no ambiente console, convivendo com milhares de jogos para computador.



Figura 2.3: Xbox, Microsoft (Fonte: Wikipédia, 2010).



Figura 2.4: Playstation, Sony (Fonte: Wikipédia, 2010).

### 2.1.3 Indústria Bilionária

O que era no início, uma simples diversão de universitários passou para uma escala industrial. Sendo uma indústria bilionária, de acordo com Juliano Spyer(2009), o custo de um jogo computacional muitas vezes passa a de uma produção hollywoodiana e a receita arrecadada já supera a de cinema(em bilheteria). Os jogos começaram a ser associados em propagandas e inclusive gerando filmes, isso devido a sua popularidade. A tendência é uma supremacia de jogos computacionais e de console em relação ao entretenimento cinematográfico.

RODRIGUES, Marcos Amaral. **METODOLOGIA DE DESENVOLVIMENTO DE JOGOS COMPUTACIONAIS PARA PROGRAMADORES EM JAVA**. Salvador: Professordelphi, 2011.

#### 2.1.4 Características

As características dos jogos eletrônicos podem ser divididos, com alguns exemplos:

- Aventura - Colossal Caves e Myst.
- Luta - Street Fighter.
- Labirinto - PAC-MAN da Namco Japonesa (1980).
- Plataforma - Mario Bros da Nintendo.
- Shooter* (Tiro) - DOOM.
- Simulação em geral - Flight Simulator, Sim City.
- Role Playing ou RPGs - série Ultima.
- Estratégia - Age of Empires.
- Adultos - jogos com tema sexual.
- Musical - Guitar Hero.

#### 2.1.5 Jogos por Rede

Jogos por Rede foi o grande avanço na história dos jogos computacionais, a partir da década de 90. O conceito abordado pelos autores Marcelo e Pescuite(2009, p.32), sobre Jogos em Rede é, jogos de computador que usam uma rede, que através do protocolo TCP/IP, os computadores se comunicam. Essa interligação favoreceu ao aumento da emoção de jogar com duas pessoas ou mais, antes era o jogador e a máquina. De acordo com Jeannie Novak(2010, p.29), o Plato foi o precursor dos jogos em rede, no ano de 1961, desenvolvido por Steve Russell no MIT (Instituto de Tecnologia de Massachusetts).

#### 2.1.6 CompuServe

A CompuServe foi o provedor para Internet que iniciou os jogos on-line. Segundo Jennie Novak(2010, p.30), o Islands of Kesmai e o Megawars I, foram jogos desenvolvidos em texto ASCII, do estilo RPG, por John Taylor e Kelton Flinn, da Kesmai Corporation, que a CompuServe utilizou para cobrar 12 dólares por hora jogada dos seus associados.

### 2.1.7 Internet

A Internet foi uma grande diferencial nesta evolução histórica. Segundo Marcelo e Pescuite(2009,p.32), em 1997, com a popularização da banda larga os jogos passaram à um novo patamar, aparecendo os MMURPGs(Massive Multiplayers Online Role Playing Games), como exemplo, a *Ultima Online*, que chegou a 50 mil participantes em 90 dias e *World of Warcraft*. Estes jogos tem como característica, uma maior interação entre os participantes, que até fizeram amizade entre si e continuaram no mundo real. Foram inspiradores, de certa forma da criação do mundo virtual, chamado *Second Life*. Esta dimensão foi possível com a criação de jogos para Rede Mundial.

### 2.1.8 Linguagens Computacional

As linguagens de computador é um ponto importante na história de JC. A linguagem assembly é a grande estreante no mundo dos games para CPU, sendo uma linguagem de baixo nível, ou seja, uma linguagem que se aproxima da linguagem de máquina, diferente da linguagem de alto nível, que se aproxima do vocabulário do homem. Sendo poucos programadores tiveram o privilégio de desenvolver para o ambiente computacional nesta linguagem, chamada assembly. Um exemplo, segundo a autora Jeannie Novak (2010, p.24), o jogo Colossal Cave, conhecido também como Adventure, foi desenvolvido por Willian Crowther, em assembly. Depois, as linguagens C, C++, Java, Flash começaram a dominar o ambiente computacional, por serem linguagens de alto nível.

Marcelo e Pescuite(2009,p.90), cita em um livro sobre algumas linguagens, de uma forma bem resumida, mas que serve para o leitor ter uma ideia sobre o assunto. Segue-se abaixo:

a) **C** - Linguagem de alto nível, oriunda da linguagem B, foi criada por [Dennis Ritchie](#) (1972), do Laboratório Bell, americano. E tem como característica, ser bastante veloz como o assembly. Foi utilizada no desenvolvimento do sistema operacional Unix. Sendo muito utilizada no desenvolvimento de jogos computacional.

b) **C++** - Criada por [Bjarne Stroustrup](#) dos [Bell Labs](#), durante a década de [80, com a meta de](#) implementar uma versão distribuída do [núcleo Unix](#). Um exemplo de jogo, inclusive de sucesso, é DOOM 3, um jogo de tiro de primeira pessoa. Chamando à atenção, que não é linguagem em si que garante o sucesso, e sim, a jogabilidade e a clareza das regras.

c) **Java** - muito conhecida no mercado de desenvolvimento, Orientada à Objeto (OO), trabalha em ambientes *Desktop*, *Web* e *Mobile*. Na linha de jogos, há uma crítica na sua utilização quanto à velocidade, todavia isso na verdade pode estar atrelada à falta de domínio da linguagem pela parte do programador, já que a linguagem citada é de uma complexidade maior. Um site que tem uma lista de jogos on-line em Java é o <http://www.funorb.com/l=3/g=dungeonassault/gamelist.ws> , como exemplo o *Ace of Skies*, que também aparece nesta lista.

Java, pode ser escrita em várias plataformas. *Standard Edition* (SE), *Enterprise Edition*(EE), *Mobile Edition* (ME). Segundo o fabricante do Java, agora é a Oracle (2011), a linguagem permite que um jogador atuando *on-line*, troque conversa com pessoas de várias partes, faça cálculos e veja imagens tridimensional. Permitindo que o jogo faça parte dos aplicativos de uma rede interna e também de Internet.

d) **Flash** - Na verdade, a linguagem é chamada de *Action Script* (AC), Flash é uma ferramenta que serve para 'rodar' o AC. O AC tem como fonte de inspiração o java, sendo também uma linguagem OO. No caso de JC, um exemplo é o *Plants vs. Zombies*, um jogo que a brincadeira é se livrar das horríveis ondas de zumbis que atacam.

Em qualquer linguagem de programação, você pode desenvolver um jogo. O que vai diferenciar, basicamente, é o domínio do programador em relação à linguagem.

### 2.1.9 Motores de Jogos

Atualmente, existem motores de jogos (*games engines*). Segundo Daniel Souza(2007), Doom 3 Engine, figura 2.5, é um motor de jogos que desenvolveu jogos como: Doom 3, Quake 4, Prey e Quake Wars. Motores de jogos são ferramentas poderosas com todos os pontos necessários para desenvolvimento. Favorecendo a implantação gráfica, sonoplastia e níveis de jogabilidade com um certo nível de facilidade. Segundo Marcelo e Pescuite(2009,p.89), muitos programadores começaram a sua vida profissional com os motores de jogos. Estas ferramentas, às vezes, nem é necessário o domínio de uma linguagem de programação, pois oferece o chamado WYSIWYG (é o [acrônimo](#) da expressão em inglês "**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et"), ou seja, como você desenvolve assim é que você verá.

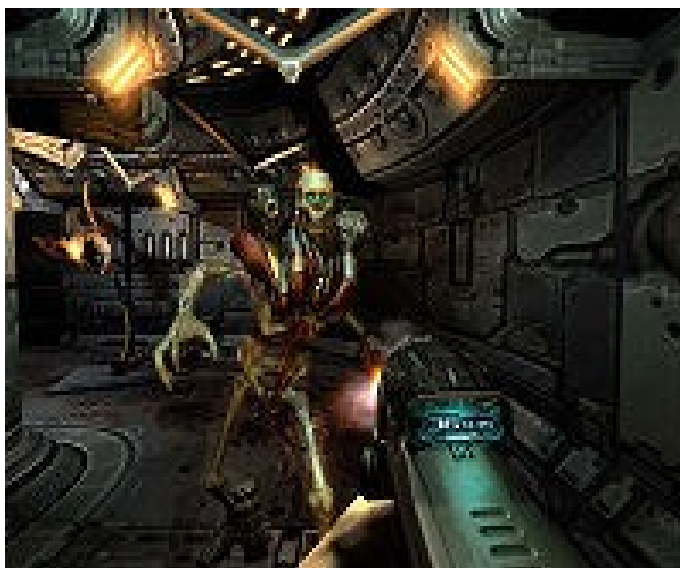


Figura 2.5: Imagem do motor de jogo Doom 3 Engine (Fonte: Wikipédia, 2010).

### 2.1.10 Inteligência Artificial em Jogos Eletrônicos

No avanço dos jogos a jogabilidade é uma forte característica do JC e esta característica está associada à Inteligência Artificial (IA), conceito que determina um certo grau de aprendizagem do software, quando jogado por alguém. Segundo André Kishimoto(2004), na evolução dos JC, em 1962, *Spacewar*, para 2 jogadores, não havia vestígio de IA, porém no ano de 1975, *Gun Fight*, um jogo com personagens em movimento, tem as primeiras utilizações de IA. A IA traz melhor qualidade para aos olhos do jogador, pois o mesmo tem a sensação de realidade. Atualmente, vários JC, tem utilizados das seguintes técnicas de AI, redes neurais, *reinforcement* e *observational learning*, que o detalhamento destes assuntos serve para geração de novas linhas de pesquisa.

### 2.1.11 A evolução dos JC de entretenimento para JC educacional.

O jogo para computador, na sua evolução, não só tem enfatizado o lado da diversão mas começa a ser usado para o aprendizado. Segundo Saulo Camarotti, Carla Denise Castanho, Et alii (2007), no artigo localizado em, <http://www.arte.unb.br/6art/textos/jeomm.pdf>, o jogo chamado *Chengo Chinese*, disponibiliza 1000 atividades de aprendizado, como situações que envolve cultura, área social, geográficas e históricas da China. Este jogo foi uma inspiração do JC, a *Ultima Online*, que gerou toda a cadeia da industria de games no ponto da realidade virtual. Agora, as autoridades chinesas estão tirando melhor proveito neste casamento de JC com a parte educacional.

### 2.1.12 Não Perdendo o Foco.

Este capítulo teve como objetivo criar um cenário de JC para proporcionar ao leitor uma abordagem conceitual e histórica de jogos para computador. Pois, a história do objeto de estudo facilita, didaticamente, o entendimento do assunto. O próximo capítulo irá tratar sobre os conceitos de Programação Orientada à Objeto.

### 2.2.1 História da OOP

A nova forma de percepção do universo pela Orientação à Objetos, trouxe para a Ciência da Computação o entendimento novo, gerando linguagens orientadas à objetos, como exemplo, o Java. Embora que, segundo Agustin Juan Ferrante e Martius Vicente Rodriguez y Rodriguez (2000, p.299), “o conceito de POO foi introduzido durante os anos de 1960 com a implementação da linguagem de programação chamada SIMULA”. Atualmente, a linguagem Java, esta sendo utilizada em várias áreas de desenvolvimento de sistemas, inclusive no desenvolvimento de jogos computacionais.

### 2.2.2 Conceito da OOP

Programação Orientada à Objetos é uma técnica de desenvolvimento de programas através de classes, que serve para representar entidades do mundo real. De acordo com o exemplo de J. Glenn Brookshear(2003, p.226), pode-se conceituar melhor o OOP, ele utilizou um jogo computacional (JC) que tem como tema, proteger a Terra através de raios laser. Cada laser é um objeto, que contem procedimentos para alterar a sua pontaria. Sendo o laser um produto de um modelo. Esse modelo é chamado de classe. Então uma classe é uma fábrica de objetos.

A técnica de POO, facilita a implementação de códigos a longo prazo pois requer uma disciplina maior e muitas vezes o conhecimento de padrões como JavaBeans (neste capítulo será utilizada a linguagem Java como exemplo).

O Javabeans é um padrão que determina o acesso de variáveis privadas através de métodos, veja no exemplo seguinte:

## Exemplo 01

```
public class Funcionario
{

private String nome; //atributo da classe funcionário encapsulado

public Funcionario()
{
super();
}

public void setNome(String nome) //método de comunicação de escrita na variável nome
que é privada.
{
this.nome=nome;
}

        public String getNome() // método de leitura da variável nome que é privada.
        {
        return nome;
        }

}
```

Observe que, os métodos determinam a comunicação com a variável que é privada. Segundo Cleuton Sampaio(2007, p.80), o *JavaBeans* é a representação do banco de dados no conceito de multi-camadas, chamada MVC, (*Model, View, Controller*). Como as ferramentas que trabalham com Java já tem mecanismos que desenvolvem automaticamente o *JavaBeans*, não é necessário decorar a sua codificação, pois o mais importante é o entendimento do assunto.

O MVC é outro padrão que é usado nas linguagens OOP. Segundo Cleuton Sampaio(2007, p.79), basicamente o padrão MVC implementa um caso de uso interativo em três componentes: Modelo, Visualização e Controlador. Este conceito, MVC, permite dividir o sistema em camadas, facilitando a distribuição das tarefas para equipe. No caso de JC, o modelo MVC, pode ser utilizado pelos engenheiros de softwares, visando facilitar a divisão das tarefas e uma melhor gestão do processo de desenvolvimento.

Dentro do MVC, o *Model* é a implementação do *JavaBeans*, por isso que o conhecimento do *JavaBeans*, como padrão para uma linguagem de programação como o Java é importante. Pode-se explicar que, o *JavaBeans* representa o banco de dados que o sistema irá manipular. A *View* é a camada visual, podendo conviver com o HTML e o *Controller*, camada que representa as regras de negócio, ou seja, a parte que define as operações que irão otimizar os processos corporativos. E de acordo com Cleuton Sampaio(2007, p.79), esta camada *Controller*, determina a mudança de estado da camada *Model* e atualização das informações na camada *View*. Através desta técnica, os programadores podem ser divididos em: webdesigners, para camada visual, administradores de banco de dados para camada modelo e programadores em Java para a camada de controle.

### 2.2.3 Conceitos que compõem a POO.

A POO, é composta de cinco conceitos básicos: Herança de Classe, Polimorfismo, Encapsulamento, Sobrecarga e Abstração. No livro de Serson (2007), ele aborda estes cinco conceitos buscando esclarecer a composição da POO.

Herança de Classe: é a capacidade que uma classe tem de herdar atributos e métodos de uma classe ancestral. Segundo Cleuton Sampaio(2007, p. 25), a generalização, conhecida como "Gen-spee", estabelece um relacionamento de herança entre classes. A maneira, em Java, de relacionarmos uma classe a outra, de tal maneira, que uma delas, herde tudo da classe ancestral é utilizando a palavra *extends*. Como exemplo, funcionários de uma empresa podem ser representados através de uma classe **Funcionario** (grifo para enfatizar que é um nome de classe). Nela vai conter atributos e métodos, ou seja, características e funcionalidades que poderão ser herdados por uma classe descendente chamada **Vendedor** através da palavra reservada *extends*, o que se denomina Herança de Classe. Segue um exemplo de código em Java para melhor explicar.

Exemplo 02.

```
public class Funcionario
{

String cpf;

public Funcionario()
{
super();
}

public void setCpf(String cpf)
{

this.cpf=cpf;
}
}
```

```

public class Vendedor extends Funcionario
{
public Vendedor()
{
super();
}
public double calcularSalario(String cargo,double salario)
{
if (cargo.equals("Vendedor"))
{
salario=salario*1.1;

} } }

```

Encapsulamento: é a técnica de determinar a visibilidade de um atributo e/ou método de uma classe. Segundo Cleuton Sampaio(2007, p.67), o encapsulamento pode aumentar a confiabilidade de um sistema. O encapsulamento é sub-dividido em:

- private - a variável e o método só podem ser vistos na própria classe, no exemplo 01, a variável 'nome' é privada, ou seja, outra classe não pode acessar ela diretamente.
- protected - a variável e o método podem ser vistos na própria classe e em classes do próprio pacote (pasta de diretório). Este tipo de visibilidade é a menos usada, todavia, quando bem analisado o sistema real, através de um levantamento de requisito, pode aparecer a necessidade de usá-la.
- public - a variável e o método podem ser vistos na própria classe e por todas as classes, quer seja dentro ou fora do pacote, no exemplo 01, o método getNome() pode ser acessado por qualquer outra classe. Este método inclusive tem o propósito de retornar o conteúdo da variável privada 'nome'.

Polimorfismo: é a utilização do mesmo nome de método por classes diferentes e com funcionalidades diferentes. Segundo Serson (2007, p.242), isso facilita para os programadores pois podem incluir características a mais que não foram consideradas no início do sistema. E segundo o Cay Horstmann(2003, p.350), "todos os métodos de instância no Java são polimórficos". Segue exemplo:

Exemplo 03.

```
public class Vendedor extends Funcionario
{

public Vendedor()
{
super();
}

public double calcularSalario(String cargo,double salario)
{
if (cargo.equals("Vendedor"))
{
salario=salario*1.1;

}

}

}

}

public class Gerente extends Vendedor
{

public Gerente()
```

```

{
super();
}

public double calcularSalario(String cargo,double salario)
{
if (cargo.equals("Gerente"))
{
salario=salario*1.2;

}

}

}

}

```

Observe que o exemplo acima, trabalha com o mesmo nome de método em classes diferentes. O método atuará de acordo como a classe que instanciou o objeto.

Exemplo 04:

```

public static void main(String args[])
{
Gerente g=new Gerente(); //gera o objeto da classe Gerente
System.out.println("O Salário do gerente será acrescido de 20% , o seu valor é :
"+g.calcularSalario("Gerente",1000); //o resultado será 1200 de salário para o gerente

}

```

Se fosse um objeto da classe Vendedor o salário seria acrescido apenas de 10%.

- Sobrecarga: similar ao Polimorfismo, a diferença é que o método está na própria classe com parâmetros diferentes. O escritor Serson (2007, p.136), além de utilizar esta mesma definição, completa, a assinatura define o tipo, a ordem e a quantidade de argumento do método.

- Abstração: a utilização de interfaces e classes abstratas com métodos abstratos que deverão ser obrigatoriamente implementados nas classes descendentes. Pois de acordo com Serson (2007, p.178), “num dado objeto somente interessa ao cliente as funções que ele executa e não a implementação da mesma”. Veja o exemplo a seguir.

Exemplo 05.

```
public interface IGameEvento {  
  
    public void onFire();  
  
}  
  
public class AppletGeral extends JApplet implements IGameEvento{  
  
    public void onFire() {  
        // aqui são desenvolvidos os comandos necessários  
    }  
  
}
```

Veja que, o `onFire()` deve ser desenvolvido na classe que `AppletGeral`, caso não seja implementado método, o JVN, máquina virtual do Java, acusa um erro em tempo de projeto na classe `AppletGeral` (esta classe faz parte do jogo `asteroides`, desenvolvido pela equipe do `Professordelphi` e pode ser acessado no endereço: <http://www.professordelphi.com/novosite/novosite/asteroides.html>). Esta obrigatoriedade, é criada pela técnica da abstração. Em suma, o método que é declarado na interface será obrigatoriamente desenvolvido na classe que herdar a interface.

## **2.3 Explicar a Gestão em Projetos de Softwares em Jogos Computacionais(JC).**

### **2.3.1 O Gestor do Projeto de Software em JC.**

Um projeto é um conjunto de atividades, que são desenvolvidas por um período de tempo, gerenciadas por um profissional competente, com o objetivo de alcançar um propósito específico. Segundo José Carlos Cordeiro Martins (2007,p.2), a necessidade de negócio com um plano de solução é o início de um projeto, que deve levar em conta também as restrições de tempo e custo.

No desenvolvimento de softwares, a figura do gerente de projeto é reconhecida e valorizada. De acordo com Roque Rabechini Junior(2005, p.76) , está nas habilidades do gerente de projeto o sucesso ou a derrota do projeto. A figura do gerente de projetos torna-se central, ou seja, de grande importância. Pois, é através deste profissional, que o desenvolvimento de um sistema é conduzido. Ele desenvolve através de uma equipe que precisa ser gerenciada.

Na área de jogos computacionais, a Gestão de Projetos é indispensável. Isso devido a complexidade de um jogo informático que envolve vários profissionais. Segundo Perucia, Berthêm, et al (2005, p.27), para produzir um jogo é necessária uma equipe qualificada e multidisciplinar. Esta equipe pode ser formada por programadores, artistas, projetistas de níveis/fases, projetistas de jogos, planejador de software, arquiteto-chefe, gerente de projeto, o sonoplasta, testadores. Sendo que, o gerente de projeto, faz o papel de um engenheiro de software.

2.3.2 Relacionar a aplicação das metodologias de produção de softwares à produção de *games* informáticos.

Programadores podem utilizar as mesmas metodologias de desenvolvimento de sistemas, quando se trata de desenvolvimento de jogos informáticos . Segundo Carlos Eduardo Lé Velasquez(2009, p.7), na Engenharia de Software existe uma divisão clássica durante o ciclo de vida do software, que são: Levantamento de Requisitos, Design de Software, Construção de Software, Testes de Software e Manutenção de Software. No desenvolvimento de um JC, deve seguir este ciclo de vida, seguindo a divisão conhecida da Engenharia de Software. Portanto, um JC, na verdade deve ser tratado como outro software qualquer no que diz respeito as fases clássicas. O que vai diferenciar entre o JC e um software tradicional é diversidade de profissionais envolvidos nas etapas do JC, que neste caso é maior.

Neste capítulo, é necessário uma observação quanto à linguagem Java para desenvolvimento de JC no ponto sobre Metodologias. As metodologias podem ser aplicadas independentes das linguagens computacionais e isso está incluso a linguagem Java. Porém a linguagem Java tem como diferencial, o ser uma linguagem orientada à objetos e muito popular. Segundo o site do fabricante, Sun(2011): “O Java permite que você jogue *on-line*, converse com pessoas do mundo inteiro, calcule a taxa de juros de um financiamento e veja imagens em 3D, apenas para citar alguns exemplos.”

Porém, a linguagem Java em si, não garante o êxito do desenvolvimento do *game* informático. Por isso, que a utilização de modelos de desenvolvimento de software, já testados, deve ser praticados. Pois isso é que irá aumentar a probabilidade de eficácia no desenvolvimento do JC.

A metodologia clássica é a no formato Cascata (*waterfall*). Esta prática pode ser utilizada se o JC for simples. Segundo Carlos Eduardo Lé Velasquez(2009,p.17), este modelo é constituído por cinco fases: Identificação de Requisitos, Concepção, Codificação, Testes e Manutenção. Cada fase, só é desenvolvida após o término da anterior. Se o erro for identificado, basta recuar para a fase anterior para buscar corrigir o problema.

A Prototipação é uma metodologia aplicada no desenvolvimento de softwares comerciais. Sendo que, na construção de um jogo computacional é uma excelente estratégia, pois vai facilitar a visualização do produto que está sendo construído pelos participantes, permitindo as devidas modificações antes do término. Segundo Carlos Eduardo Lé Velasquez(2009,p.18), um dos benefícios deste modelo está relacionado a uma melhor entendimento entre cliente e programador à medida que se mostra os protótipos que foram desenvolvidos. No caso de JC, isso traz uma melhor segurança para o cliente por ter uma melhor compreensão e para a equipe uma sensação de estar no rumo certo.

No desenvolvimento de JC complexos é indicado a utilização do modelo em Espiral, principalmente quando vai utilizar a Prototipação. Segundo Matthew May(2007, p.79), quando se tem um sistema complexo a aplicação do método Espiral é melhor e deve ser associada com a Prototipação rápida. Embora que este autor está enfatizando a utilização da engenharia, não necessariamente a de software, a sua utilização pode ser aplicada no mesmo princípio de desenvolvimento de JC. Observe que este modelo em Espiral pode também ser aplicado na criação de JC.

A outra metodologia, é a *Extreme Programming (XP)*, um conjunto de técnicas que motiva a equipe a trabalhar para entregar algum resultado, o mais cedo possível, convidando o cliente a participar de uma forma ativa no desenvolvimento do software. De acordo com Carlos Eduardo Lé Velasquez(2009,p.22), a liberdade de escolha de método para o desenvolvimento do software é uma característica do XP. E no caso dos *games* informáticos, esta metodologia é sugerida, devido a sua dinâmica, para que a equipe participe de uma forma mais ativa no seu desenvolvimento. Outra característica desta metodologia é a utilização de testes. Segundo José Carlos Cordeiro Martins(2007,p.302), os testes automatizados ajudam a projetar o sistema e ter uma implementação melhor. Os testes ajudarão outros desenvolvedores a evoluir o sistema. Testes tanto na visão do programador, chamado caixa branca e testes pela visão do cliente, chamado

caixa preta. Devido os testes aplicados, o produto final terá uma quantidade de erros inferior e a satisfação do cliente será maior.

A XP está no contexto de Modelos Ágeis. A SCRUM e a RAD (*Rapid Application Development*) também estão dentro do contexto de Modelos Ágeis. Segundo José Carlos Cordeiro Martins (2007, p.270), o Scrum é método que trabalha com interação, procurando antecipar o lucro do projeto para cliente, pois o software pode ser entregue ao cliente passo-a-passo, sendo que, a cada passo o usuário final já pode ir utilizando o sistema. No caso de JC, este modelo também pode ser utilizado, lembrando que um JC pode ser tratado como um software tradicional no que tange a respeito de Metodologia de Desenvolvimento de Software. De acordo com o autor, Carlos Eduardo Lé Velasquez(2009, p.19), o conceito de RAD é a utilização de ferramentas que especificam as características do software. Isso otimiza o processo de desenvolvimento. Nesta metodologia estão enquadrados os *Games Engines* (programas computacionais que visam facilitar a vida dos programadores, por conter bibliotecas, ou seja, código fonte já desenvolvido, que agiliza o processo de desenvolvimento de JC .

### 2.3.3 A figura do Game Design.

Uma das peças chaves para a Prototipação ou qualquer outro modelo de desenvolvimento de um jogo computacional é a figura do *Game Design*(GD). Segundo Perucia, Berthêm, et al (2005, p.30), um GD descreve cada detalhe de como o jogo funcionará. O GD é um profissional que pode fazer os seguintes papéis, o do projetista de níveis/fases, o de projetista de jogos e o planejador de software. Ele pode projetar o jogo, deste do design até a sua jogabilidade. Segundo Antonio Marcelo e Julio Pescuite(2009, p.35), ele projeta o jogo para as pessoas se divertirem, o que não é tarefa fácil. Pois, o GD tem que se projetar como um jogador.

O GD é um profissional que tem como característica ser criativo, dinâmico e ampla experiência de jogos. Esta carreira está em destaque, principalmente por causa da Internet. Exerce a função de projetar-se e jogar várias vezes o jogo até chegar no ponto aceitável. Agora tem uma característica que é chave na equipe para um GD, é a facilidade de comunicação. Segundo Antonio Marcelo e Julio Pescuite(2009, p.36), no GD é necessário a capacidade de se comunicar e convencer a ideia do jogo. Pois, como um programador irá desenvolver o jogo se ele não “comprar a ideia” ?

## **2.4 Entrevista com um programador sobre o desenvolvimento de “Asteroides”.**

### 2.4.1 Introdução

Será abordado um estudo de caso sobre o desenvolvimento de um jogo computacional chamado Asteroides, criado por um professor do curso chamado Professordelphi, localizado em Salvador-BA.

Este jogo tem um propósito de capacitar novos programadores em Java para o segmento de *games*. Embora seja um jogo, a sua operação gera um efeito educacional. Pois, permite que o jogar desenvolva as suas habilidades cognitivas em relação à digitação. Isso para quem digita muito devagar. E também serve para desenvolver o aprendizado em crianças na fase da alfabetização. Este jogo pode ser conferido no <http://www.professordelphi.com/novosite/novosite/asteroides.html> . (acessado: 16 de fevereiro de 2011).

2.4.2 Relato de um programador e professor do Professordelphi, Salvador-BA, (18 Dezembro 2010 19:12:57 ).

#### **Motivação para estudar o assunto sobre JC.**

- Sempre vi o desenvolvimento de jogos como uma excelente oportunidade para praticar planejamento de desenvolvimento de software. Os jogos geralmente simulam situações reais do dia-a-dia , desta forma, ao trabalhar num jogo, estaria desenvolvendo, ao mesmo tempo trabalhando muito com a matemática e também criando animações, que é uma excelente forma de praticar geometria.

- Só iniciei as criações de jogos, quando vi a necessidade dos meus sobrinhos (6 - 12 anos) de aprender digitação. A princípio, criei um programa tutorial para prática de digitação, que foi muito rejeitado, pelo visual e complexidade. Então comecei a desenvolver o mesmo programa como um jogo de computador, o que teve uma aceitação bem melhor.

### **Sites que utilizou para estudo.**

- As pesquisas foram feitas no site da Oracle, <http://www.oracle.com>. E teve como propósito o encontrar formas de publicar o jogo em um site e também na utilização de sons nos jogos.

### **Qual a dica que você daria para uma equipe, para desenvolver um jogo computacional em java.**

- Aconselho para um equipe que irá começar a desenvolver jogos, que busquem simplicidade e beleza, em jogos 2D, já que para desenvolver um jogo 3D os custos serão muito elevados e a comparação com outros jogos do gênero será inevitável. E o mais importante para que se tenha produtividade é ter um excelente domínio dos conceitos de orientação à objeto.

### **Quantos participantes você indicaria para desenvolver um jogo profissional? e quais?**

- No mínimo quatro.
- Um Designer que um dos pontos mais importante, já que o seu jogo será avaliado a primeira vista pelas imagens.
- Um produtor musical é muito importante , porque o som é responsável pelas emoções durante o jogo .
- Um programador para desenvolver os códigos. No princípio seria muito trabalho para apenas um programador , mas se os conceitos de orientação à objeto forem bem aplicados, a maior parte do código pode ser reaproveitada para os próximos jogos, com o uso de pacotes de classes com funcionalidade padrões.
- Um analista de sistemas que resolveria a organização dos sistemas, os problemas de lógica, distribuição das entidades em classes e abstração de classes. (esta é a minha visão do analista de sistemas).

### 2.4.3 Sobre a entrevista.

A entrevista foi feita através de email. O programador domina: delphi e java. É um estudante universitário, faculdade localizada em Salvador-Ba. É professor de Jogos Computacionais em Java, pelo Professordelphi (<http://www.professordelphi.com>).

## 3 CONSIDERAÇÕES FINAIS

A história dos JC foi abordada para preparar o leitor para entender o objeto de estudo. No início o JC era pouco utilizado em comparação como os games de console como era o caso do Atari. Mas, com o advento dos sistemas operacionais da Apple e da Microsoft, os JC tiveram um grande avanço. Até chegar na atualidade, com a Internet e a capacidade dos JC serem utilizados a níveis globais. Também no ponto financeiro, os JC tem um destaque de escala industrial e bilionária. Jogos musicais como *Guitar Hero*, *Age of Empires*, *Street Fighter* são muitos conhecidos com características distintas. Na década de 90, os jogos por rede começaram a se destacar. Sendo que, a Compuserv foi a precursora dos jogos on-line, como provedor de Internet. Enfim, as linguagens de computador são a base para o desenvolvimento de JC e neste material acadêmico foi dado ênfase a linguagem Java por ser conhecida e também por ser multiplataforma.

O conceito de Orientação à Objetos abordado neste trabalho busca trazer uma compreensão desta técnica de desenvolvimento de sistema no meio informático. Por ser bastante utilizada no desenvolvimento de JC na linguagem Java. Assuntos como: Herança de Classe, Polimorfismo, Encapsulamento, Sobrecarga e Abstração, também foram abordados completando o conceito de Programação Orientada à Objetos.

A gestão de desenvolvimento de JC é um ponto importante pois trabalha diretamente com o tempo e o custo. Sendo que, a figura do gerente de projeto é uma peça-chave no êxito do projeto de JC. Relacionar a aplicação das metodologias de produção de softwares à produção de *games* informáticos deve ser observada com naturalidade pois, um JC também é um software. Sendo assim, existe um ciclo de vida de sistema para JC, como exemplo: Levantamento de Requisitos, Design de Software, Construção de Software, Testes de Software e Manutenção de Software.

Metodologias de desenvolvimento como Cascatas (*Waterfall*), Prototipação, Espiral, XP são escolhidas para desenvolvimento de JC de acordo com a necessidade. Geralmente, decidida pelo Gerente de Projeto, podendo também ser decidida pela equipe, neste caso, requer uma equipe madura no desenvolvimento de JC.

Tanto importante quanto o Gerente de Projeto é o *Game Designer*. Pois é um profissional criativo e idealiza as estratégias do jogo, trazendo a vontade de jogar aos participantes e é responsável também pela jogabilidade.

A entrevista com um programador visou trazer uma realidade vivida nos meios das empresas de desenvolvimento de JC. Trazendo uma maior motivação para esta área de desenvolvimento de JC que está em franca expansão.

#### **4 METODOLOGIA**

Foi utilizada a metodologia de pesquisa bibliográfica para a confecção desta monografia e também uma entrevista, como forma de enriquecer a pesquisa.

**Bibliografia:**

ACE OF SKIES. Disponível em:

<<http://www.funorb.com/l=3/g=dungeonassault/gamelist.ws>>. Acessado em: 15 de janeiro de 2011.

ADOBE. Disponível em: <<http://www.adobe.com/br/products/flash/?promoid=BOZRB>>. Acessado em : 14 de janeiro de 2011.

AMAZONAS, Daniel Souza. **Desenvolvimento de Jogos 3D em Java com a Utilização do Motor Gráfico Irrlich**. Disponível em: <[http://www.flf.edu.br/revista-flf/monografia\\_jogo3d.pdf](http://www.flf.edu.br/revista-flf/monografia_jogo3d.pdf)>. Acessado em: 14 de janeiro de 2011.

BROOKSHEAR, J. Glenn, **CIENCIA DA COMPUTAÇÃO: UMA VISAO ABRANGENTE**. Porto Alegre: Bookman, 2003.

FERRANTE, Agustin Juan, RODRIGUEZ, Vicente Rodrigues y. Tecnologia de Informação e Gestão Empresarial. Rio de Janeiro: E-Papers, 2000.

KISHIMOTO, André. **Inteligência Artificial em Jogos Eletrônicos**. Disponível em:<[http://www.programadoresdejogos.com/trab\\_academicos/andre\\_kishimoto.pdf](http://www.programadoresdejogos.com/trab_academicos/andre_kishimoto.pdf)>. Acessado em: 06 de janeiro de 2011.

HORSTMANN, Cay. **CONCEITOS DE COMPUTAÇÃO COM O ESSENCIAL DE JAVA**. Porto Alegre: Bookman, 2003.

IPLAY. ATARI ONLINE jogos java direto no site. Disponível: <<http://www.iplay.com.br/Jogos/Atari/2600/>>. Acessado em: 16 de janeiro de 2011.

JAVA. **O que é o Java?**. Disponível em: <[http://java.com/pt\\_BR/download/index.jsp](http://java.com/pt_BR/download/index.jsp)>. Acessado em: 15 de janeiro de 2011.

JUNIOR, Roque Rabechini. **Competências e maturidade em gestão de projetos: uma perspectiva estruturada**. 1ª ed. São Paulo: ANNABLUME, 2005.

MARCELO, Antonio, PESCUITE, Julio. **Design de Jogos Fundamentos**. Rio de Janeiro: Brasport, 2009.

MARTINS, José Carlos Cordeiro. **Técnicas Para Gerenciamento de Projetos de Software**. Rio de Janeiro: Brasport, 2007.

MAY, Matthew. **TOYOTA - A Fórmula da Inovação**. Rio de Janeiro: Elsevier, 2007.

RODRIGUES, Marcos Amaral. **METODOLOGIA DE DESENVOLVIMENTO DE JOGOS COMPUTACIONAIS PARA PROGRAMADORES EM JAVA**. Salvador: Professordelphi, 2011.

MELO JR, Cleuton Sampaio. **GUIA DO JAVA ENTERPRISE: DESENVOLVENDO APLICAÇÕES CORPORATIVAS**. 5 ed. Rio de Janeiro: Brasport, 2007.

NOVAK, Jeannie. **Desenvolvimento de Games – Tradução de 2ª Edição Norte-Americana**. São Paulo: Cengage Learning, 2010.

PERUCIA, Alexandre Souza, BERTHÊM, Antônio Córdova de, BERTSCHINGER, Guilherme Lage, MENEZES, Roberto Ribeiro Castro. **Desenvolvimento de Jogos Eletrônicos – Teoria e Prática**. 2ª ed. São Paulo: Novatec, 2005.

PROFESSORDELPHI. **Curso de Jogos Computacionais**. Disponível em: <<http://www.professordelphi.com/novosite/novosite/cursojogos.htm>>. Acessado em 31 de janeiro de 2011.

SAULO, Camarotti, CASTANHO, Carla Denise, et al. **Proposta de um modelo de Jogo Online Massivamente Multiusuário aplicado à educação**. Disponível em: <<http://www.arte.unb.br/6art/textos/jeomm.pdf>>. Acessado em: 06 de janeiro de 2011.

SERSON, Roberto Rubinstein. **PROGRAMAÇÃO ORIENTADA A OBJETOS COM JAVA 6**. Rio de Janeiro: Brasport, 2007.

SHIGUE, C. Y. **Cálculo Numérico e Computacional**. Disponível em: <<http://www.alunos.eel.usp.br/numerico/notasDeAula/intro.pdf>>. Acessado em: 03 de janeiro de 2010.

SPYER, Juliano Spyer. **Para Entender a Internet: Noções, práticas e desafios da comunicação em rede**. São Paulo: Naozero, 2009.

SUN. Disponível em: <[http://java.com/pt\\_BR/download/index.jsp](http://java.com/pt_BR/download/index.jsp)>. Acessado em: 26 de janeiro de 2011.

WIKIPÉDIA. **Atari 2600**. Disponível em: <[http://pt.wikipedia.org/wiki/Atari\\_2600](http://pt.wikipedia.org/wiki/Atari_2600)>. Acessado em : 14 de janeiro de 2011.

VELASQUEZ, Carlos Eduardo Lé. **Modelo de Engenharia de Software para o Desenvolvimento de Jogos e Simulações Interactivas**. Disponível em: <[https://bdigital.ufp.pt/dspace/bitstream/10284/1361/3/DM\\_CarlosVelasquez.pdf](https://bdigital.ufp.pt/dspace/bitstream/10284/1361/3/DM_CarlosVelasquez.pdf)>. Acessado em 25 de janeiro de 2011.

WIKIPÉDIA. **C (linguagem de programação)**. Disponível em: <[http://pt.wikipedia.org/wiki/Linguagem\\_C](http://pt.wikipedia.org/wiki/Linguagem_C)>. Acessado em: 14 de janeiro de 2011.

WIKIPÉDIA. **C++**. Disponível em: <<http://pt.wikipedia.org/wiki/C%2B%2B>>. Acessado em: 14 de janeiro de 2011.

RODRIGUES, Marcos Amaral. **METODOLOGIA DE DESENVOLVIMENTO DE JOGOS COMPUTACIONAIS PARA PROGRAMADORES EM JAVA**. Salvador: Professordelphi, 2011.

WIKIPÉDIA. **Doom 3**. Disponível em:<[http://pt.wikipedia.org/wiki/Doom\\_III](http://pt.wikipedia.org/wiki/Doom_III) >. Acessado em : 15 de janeiro de 2011.

WIKIPÉDIA. **Id Tech 4**. Disponível em:<[http://en.wikipedia.org/wiki/Doom\\_3\\_Engine](http://en.wikipedia.org/wiki/Doom_3_Engine) >. Acessado em : 14 de janeiro de 2011.

WIKIPÉDIA. **Motor de Jogo**. Disponível em:<[http://pt.wikipedia.org/wiki/Motor\\_de\\_jogo](http://pt.wikipedia.org/wiki/Motor_de_jogo) >. Acessado em: 14 de janeiro de 2011.

WIKIPÉDIA. **XBOX**. Disponível em : <<http://pt.wikipedia.org/wiki/Xbox> >. Acessado em 14 de janeiro de 2011.