



SENAI CIMATEC

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM  
COMPUTACIONAL E TECNOLOGIA INDUSTRIAL  
Mestrado em Modelagem Computacional e Tecnologia Industrial

Dissertação de mestrado

**Um Modelo de Cloud Gaming para Jogos Digitais**

Apresentada por: Alberto Vianna Dias da Silva  
Orientador: Dra. Lynn Rosalina Gama Alves  
Co-orientador: Dr. Josemar Rodrigues de Souza

Março de 2014

Alberto Vianna Dias da Silva

## Um Modelo de Cloud Gaming para Jogos Digitais

Dissertação de mestrado apresentada ao Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial, Curso de Mestrado em Modelagem Computacional e Tecnologia Industrial do SENAI CIMATEC, como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Área de conhecimento: Interdisciplinar

Orientador: Dra. Lynn Rosalina Gama Alves  
*SENAI CIMATEC*

Co-orientador: Dr. Josemar Rodrigues de Souza  
*SENAI CIMATEC*

Salvador  
SENAI CIMATEC  
2014

---

## Nota sobre o estilo do PPGMCTI

---

Esta dissertação de mestrado foi elaborada considerando as normas de estilo (i.e. estéticas e estruturais) propostas aprovadas pelo colegiado do Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial e estão disponíveis em formato eletrônico (*download* na Página Web [http://ead.fieb.org.br/portal\\_faculdades/dissertacoes-e-teses-mcti.html](http://ead.fieb.org.br/portal_faculdades/dissertacoes-e-teses-mcti.html) ou solicitação via e-mail à secretaria do programa) e em formato impresso somente para consulta.

Ressalta-se que o formato proposto considera diversos itens das normas da Associação Brasileira de Normas Técnicas (ABNT), entretanto opta-se, em alguns aspectos, seguir um estilo próprio elaborado e amadurecido pelos professores do programa de pós-graduação supracitado.

# SENAI CIMATEC

Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial

Mestrado em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, leram e recomendam a aprovação [com distinção] da Dissertação de mestrado, intitulada “Um Modelo de Cloud Gaming para Jogos Digitais”, apresentada no dia (dia) de (mês) de (ano), como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Orientador:

---

Prof. Dra. Lynn Rosalina Gama Alves  
SENAI CIMATEC

Co-orientador:

---

Prof. Dr. Josemar Rodrigues de Souza  
SENAI CIMATEC

Membro externo da Banca:

---

Prof. Dr. Esteban Walter Gonzalez Clua  
Universidade Federal Fluminense

Membro interno da Banca:

---

Prof. Dr. Renelson Ribeiro Sampaio  
SENAI CIMATEC

Dedico este trabalho a madrugada, que não me verá mais elaborando esta dissertação!

---

## Agradecimentos

---

Agradeço a Deus, que me guiou nesse árduo caminho do término da dissertação.

Aos meus orientadores, Lynn e Josemar, pela paciência e orientação, me conduzindo de forma precisa para o sucesso do meu trabalho.

A minha mãe Ana Teresa e a meu pai Mário, por todo o suporte e amor em minha vida.

A minha namorada Cláudia, por toda a paciência e compreensão quando tivemos que nos privar dos nossos encontros, pois estava focado na dissertação, e por ela ter se dedicado à leitura deste trabalho inúmeras vezes estando sempre disposta a me ajudar.

A todos os meu amigos e familiares, pelo apoio e compreensão, principalmente quando estive ausente nos momentos de confraternização.

Aos meus colegas de trabalho, que precisaram resolver os problemas enquanto me reunia com meus orientadores.

Salvador, Brasil  
10 de Março de 2014

Alberto Vianna Dias da Silva

---

## Resumo

---

O maior mercado de entretenimento na atualidade é o de jogos, superando até Hollywood, posicionando o desenvolvimento destes artefatos culturais como um dos empreendimentos mais lucrativos para investimento global. Outra área em constante crescimento é a de dispositivos móveis. Hoje, já existem mais cartões SIMs - *Subscriber Identity Module* habilitados do que pessoas no Brasil. Consequentemente, o desenvolvimento de jogos para celulares e smartphones também cresceu, sendo hoje um dos principais segmentos de desenvolvimento de jogos no Brasil. Dentro deste contexto, e considerando que as políticas públicas educacionais visam disponibilizar tablets para alunos e professores da rede pública, torna-se fundamental a busca de estratégias viabilizando acesso aos conteúdos que serão produzidos para estes âmbitos tecnológicos. Na realidade brasileira hoje, até o jogo com os gráficos mais simples, dificilmente poderia ser utilizado em uma escola pública ou privada, devido ao fato de possuírem máquinas de baixo recurso computacional e devido à falta de máquinas para todos os alunos. Então, como proporcionar aos alunos uma experiência educacional, com jogos, dentro da escola? Este trabalho apresenta, como objetivo, propor uma solução computacional em que o processamento dos jogos educativos seja deslocado do dispositivo móvel ou dos computadores e direcionado para o cluster de desenvolvimento do SENAI-CIMATEC (Serviço Nacional de Aprendizagem Industrial - Centro Integrado de Manufatura e Tecnologia), atuando assim como uma nuvem nos conceitos de *Cloud Computing* e descartando a estrutura computacional das escolas. Esta tecnologia possibilita ao usuário do dispositivo móvel a execução de um jogo em seu aparelho, em que o processamento é realizado na nuvem e o dispositivo é responsável somente pelo envio de comandos de interação e pelo recebimento das imagens para gerar o vídeo. O desenvolvimento do modelo de arquitetura utilizou a UML - Unified Modeling Language como planejamento e neste trabalho foram criados o diagrama de caso de uso, os diagramas de classes e os diagramas de sequência. Para validar esse modelo, um jogo de memória foi desenvolvido executando na nuvem e sendo jogado de forma transparente por usuários utilizando dispositivos móveis com o sistema operacional Android.

Palavras chave: Computação em Nuvem, Jogos Digitais, Dispositivos Móveis.

---

## Abstract

---

The biggest entertainment market today is digital game, surpassing even Hollywood, positioning the development of these cultural artifacts as being one of the most profitable ventures for global investment. Another area of huge growth are mobile devices. Today, there are more SIMs cards - Subscriber Identity Module enabled than people in Brazil. Consequently, the development of games for mobile phones and smartphones has also grown, and is now one of the main segments of game development in Brazil. Within this context, and considering the educational policies that aim to provide tablets for students and teachers of the public school, it is crucial to find strategies that allow access to content that will be produced for these technological devices. In the brazilian reality today, even the game with simpler graphics, could hardly be used in a public or private school, due to the fact that machines have low computational resources and lack of machines for all students. So, how to provide students an educational experience with games inside the school? This work has the objective to propose a computacional solution in which the educational games processing is displaced from the mobile device or computer and is delegated to the development cluster of SENAI-CIMATEC, acting as a cloud on the concepts of cloud Computing, discarding the computational structure of schools. This technology enables the mobile device user to run a game on your device, in which the processing of the game is performed in the cloud and the device would be responsible only for sending interaction commands and receiving images to generate the video. The development of the architecture model used UML as planning and in this work were created the use case diagram, class diagrams and sequence diagrams. To validate this model, a memory game was developed running in the cloud and being played in a transparently way for users with mobile devices using the Android operating system.

Keywords: Cloud Computing, Digital Games, Mobile Devices.



---

# Sumário

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Organização da Dissertação de mestrado . . . . .	3
<b>2</b>	<b><i>Cloud Computing</i> (Computação em Nuvem) para <i>Games</i></b>	<b>5</b>
2.1	Histórico . . . . .	5
2.2	Definições . . . . .	6
2.3	TIaaS - <i>TI as a Service</i> (TI como Serviço) e <i>Cloud Computing</i> . . . . .	8
2.4	Modelos de Implementação . . . . .	9
2.5	<i>Cloud Computing</i> e <i>Grid Computing</i> . . . . .	10
2.6	<i>Cloud Gaming</i> . . . . .	12
<b>3</b>	<b>O Modelo</b>	<b>19</b>
3.1	Linguagens e Ambiente de Desenvolvimento . . . . .	19
3.2	EduCloudGaming - Modelos . . . . .	21
3.2.1	Arquitetura . . . . .	21
3.2.2	Diagramas de Caso de Uso e Colaboração . . . . .	23
3.2.3	Diagramas de Classes . . . . .	26
3.3	Implementação . . . . .	31
3.4	O Jogo . . . . .	36
3.5	Comunicação entre o Jogo e a Nuvem . . . . .	39
<b>4</b>	<b>Testes e Resultados</b>	<b>42</b>
<b>5</b>	<b>Considerações finais</b>	<b>46</b>
5.1	Conclusões . . . . .	46
5.2	Contribuições . . . . .	46
5.3	Atividades Futuras de Pesquisa . . . . .	47
	<b>Referências</b>	<b>49</b>

---

## Lista de Tabelas

---

3.1	Implementações Criadas para a Nuvem . . . . .	24
4.1	Tempos de resposta de uma ação do jogador. . . . .	44
4.2	FPS do jogo de memória executando na nuvem. . . . .	44
4.3	Elementos essenciais de uma nuvem na EduCloudGaming . . . . .	45

---

## Lista de Figuras

---

2.1	Um modelo de servidor qualquer que usa uma nuvem como metáfora da internet (TANENBAUM; STEEN, 2007). . . . .	7
2.2	Conceitos da <i>Cloud Computing</i> (VERAS, 2012). . . . .	10
2.3	Comparação de <i>Grids</i> , <i>Clouds</i> e outras tecnologias, onde o eixo y se refere a escala, e o x separa a orientação para aplicações ou serviços (FOSTER et al., 2008). . . . .	12
2.4	Microconsole preto distribuído pela OnLive para jogar em televisões em geral (ONLIVE, 2013). . . . .	14
2.5	NVIDIA Shield (NVIDIASHIELD, 2013b). . . . .	15
2.6	Arquitetura disponibilizada do GamingAnywhere (GAMINGANYWHERE, 2013). . . . .	16
2.7	Arquitetura desenvolvida por um grupo de pesquisadores da UFF. (BARBOZA et al., 2010). . . . .	17
3.1	Arquitetura Cloud Gaming deste trabalho. . . . .	22
3.2	Diagrama de Caso de Uso deste trabalho. . . . .	23
3.3	Diagrama de Colaboração deste trabalho. . . . .	25
3.4	Diagrama de Classes da biblioteca EduCloudGaming. . . . .	27
3.5	Diagrama de Classes do <i>Server Manager</i> . . . . .	28
3.6	Diagrama de Classes do <i>Server Worker</i> . . . . .	29
3.7	Diagrama de Classes do Cliente Android. . . . .	30
3.8	Diagrama que contém os primeiros passos quando um jogador abre a aplicação cliente. . . . .	32
3.9	Diagrama que contém os passos seguintes depois que o jogador escolhe o jogo. . . . .	34
3.10	Diagrama que contém os passos de interação entre os sistemas durante o jogo. . . . .	35
3.11	Diagrama que contém a preparação inicial da nuvem. . . . .	36
3.12	Figuras do jogo de memória. . . . .	38
3.13	Diagrama de Classes do jogo de memória desenvolvido. . . . .	39
3.14	Diagrama que contém a comunicação da nuvem com o Jogo. . . . .	40
4.1	Utilização dos recursos computacionais. . . . .	43
4.2	Tamanho dos pacotes entrando e saindo. . . . .	43

---

## Lista de Siglas

---

PPGMCTI ..	Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial
SENAI .....	Serviço Nacional de Aprendizagem Industrial
CIMATEC ..	Centro Integrado de Manufatura e Tecnologia
SIM .....	Subscriber Identity Module
MIT .....	Massachusetts Institute of Technology
DARPA .....	Defense Advanced Research Projects Agency
WWW .....	World Wide Web
CERN .....	European Center for Nuclear Physics
HTTP .....	HyperText Transfer Protocol
HTML .....	HyperText Markup Language
NIST .....	National Institute of Standards and Technology
LAN .....	Local Area Network
PDA .....	Personal Digital Assistant
TlaaS .....	TI as a Service
TI .....	Tecnologia da Informação
SaaS .....	Software as a Service
PaaS .....	Plataform as a Service
IaaS .....	Infrastructure as a Service
HaaS .....	Hardware as a Service
SOA .....	Service Oriented Architecture
E3 .....	Electronic Entertainment Expo
GDC .....	Game Developers Conference
GaaS .....	Game as a Service
RPG .....	Role Playing Game
FBI .....	Federal Bureau of Investigation
LER .....	Lesão por Esforço Repetitivo
SNES .....	Super Nintendo Entertainment Systems
SETI@home	Search for Extraterrestrial Intelligence at home
WLCG .....	Worldwide LHC Computing Grid
LHC .....	Large Hadron Collider
PC .....	Personal Computer
JVM .....	Java Virtual Machine
JDK .....	Java Development Kit
XML .....	eXtensible Markup Language
UML .....	Unified Modeling Language
SDK .....	Software Development Kit
ADT .....	Android Developer Tools
UFF .....	Universidade Federal Fluminense
DTO .....	Data Transfer Object
IP .....	Internet Protocol
SO .....	Sistema Operacional
CM .....	Cloud Manager
RAM .....	Random Access Memory
GPU .....	Graphics Processing Unit
API .....	Application Programming Interface
PNG .....	Portable Network Graphics

---

GHz .....	Gigahertz
GB .....	Gigabyte
Mbps .....	Megabit por segundo
Wi-Fi .....	Wireless-Fidelity
Kbps .....	Quilobit por segundo
FPS .....	Frames por segundo
MEC .....	Ministério da Educação
CSII .....	Centro de Supercomputação para Inovação Industrial

## Introdução

---

O mercado de entretenimento mais lucrativo na atualidade é o de jogos, superando até Hollywood. Segundo Machado (2013), em 2012, os estúdios de cinema de Hollywood faturaram US\$ 50,6 bilhões, contra US\$ 52 bilhões da indústria de jogos, posicionando o desenvolvimento de jogos como um dos empreendimentos mais lucrativos para investimento global.

Outra área em constante crescimento é a de dispositivos móveis. Hoje, já existem mais cartões SIMs - *Subscriber Identity Module* habilitados do que pessoas no Brasil. Segundo o IBGE (2013), no censo aplicado em 2010 foram contabilizadas mais de 190 milhões de pessoas, enquanto existem mais de 227 milhões de cartões SIMs habilitados (ITU, 2012). Conseqüentemente, o desenvolvimento de jogos para celulares e *smartphones* também cresceu, sendo hoje um dos principais segmentos de desenvolvimento de jogos no Brasil (MURNO; DRSKA, 2013). Em uma pesquisa realizada em 2013, presente em ESPM, SIOUX e BLEND (2013), cerca de 81% dos entrevistados jogam algum *game* no celular e 73% dos entrevistados jogam no seu *smartphone*, o que pode ser considerado como uma fração significativa, já que somente 66% jogam no console. Além disso, outra informação importante dessa mesma pesquisa é que desses 73% que jogam no *smartphone*, 71% jogam online.

Para tecnologias tão exploradas, porque não unir essa popularidade com a possibilidade de utilizá-las para o ensino e disseminação do conhecimento? Tecnologias estas que favorecem a criação de jogos educativos, permitindo a esses *games* serem jogados de forma pervasiva ou, até mesmo, retirando o processamento dos dispositivos, usando, assim, uma nuvem como responsável por esse processamento.

Na realidade brasileira hoje, até o jogo com os gráficos mais simples, dificilmente poderia ser utilizado em uma escola pública ou privada, devido ao fato de possuírem máquinas de baixo recurso computacional. Ademais, excluindo o fato de haver um baixo desempenho computacional das máquinas, neste contexto, existe outro problema que não é exclusivo do Brasil: a falta de máquinas para os alunos. Segundo o G1 (2013), apenas 12% das escolas possuem computadores instalados nas salas de aulas. Normalmente, os computadores permanecem em laboratórios de informática e são em uma quantidade muito menor quando comparados ao número de alunos.

É importante salientar que alguns jogos desenvolvidos com fins educacionais podem ser executados nas máquinas das escolas, ou também podem ser executados no próprio dis-

positivo móvel. A nuvem, nesses casos, seria mais uma possibilidade para essa execução.

Com a popularização e a baixa dos preços dos *smartphones* e *tablets*, o número de dispositivos vendidos no Brasil aumentou e, aproximadamente, 3 em cada 10 brasileiros já possuem um *smartphone* (RUIC, 2013). Outro fato de destaque, nessa realidade, é que o MEC - Ministério da Educação forneceu aos professores 382.317 *tablets* no Programa Nacional de Tecnologia Educacional, de forma gratuita (MEC, 2012).

Então, como proporcionar aos alunos uma experiência educacional, com jogos, dentro da escola? Uma solução em que a estrutura da escola fosse descartada e somente *tablets* e *smartphones* dos alunos e professores fossem usados, seriam de grande valia, se tornando mais uma forma de execução dos jogos. Isto pode ser justificado pelo fato de o jogo e de o conhecimento serem disseminados com maior eficiência, o que permite ser até cogitado o uso dos computadores da escola, sendo reposicionado o processamento.

Unir educação e tecnologia é sempre um desafio pelas grandes diferenças conceituais das áreas, e ao mesmo tempo, pela sua união na prática. Neste contexto, nasce esse trabalho de pesquisa de título: Um Modelo de *Cloud Gaming* para Jogos Digitais, com a premissa de criar uma alternativa para mais uma utilização de jogos voltados para educação, acreditando na presença do lúdico na aprendizagem.

Este trabalho apresenta, como objetivo, uma solução alternativa em que o processamento dos jogos educativos é deslocado do dispositivo móvel e dos computadores e é direcionado para o *cluster* de desenvolvimento do CSII (Centro de Supercomputação para Inovação Industrial) alocado no SENAI-CIMATEC (Serviço Nacional de Aprendizagem Industrial - Centro Integrado de Manufatura e Tecnologia), atuando assim como uma nuvem nos conceitos de *Cloud Computing*.

Esta tecnologia possibilita ao usuário do dispositivo móvel a execução em seu aparelho de um jogo, em que seu processamento será realizado na nuvem e o dispositivo será responsável somente pelo envio de comandos de interação e pelo recebimento das imagens para gerar o vídeo.

Vale salientar que a solução proposta não está limitada ao *cluster* de desenvolvimento do CSII: ela funciona em qualquer outra estrutura computacional. O *cluster* de desenvolvimento do CSII foi escolhido por estar disponível aos alunos da instituição.

Para validar esse modelo de arquitetura, um jogo de memória foi desenvolvido executando na nuvem e sendo jogado de forma transparente por usuários utilizando dispositivos móveis com o sistema operacional Android. É importante ressaltar que o jogo de memória foi escolhido devido ao seu estímulo nas habilidades físicas, mentais e sensoriais (COSTA;

CARVALHO, 2005).

Existem alguns trabalhos em desenvolvimento na área. Exemplos dos mesmos seriam um projeto ainda em criação na Universidade Federal Fluminense (BARBOZA et al., 2010), a OnLive (ONLIVE, 2013) nos EUA e o BSNL (BSNL, 2013) na Índia e EUA. Contudo, todos envolvem jogos de diversos tipos ou estritamente comerciais, sem uma especificidade, ou seja, não incluem jogos com fins educacionais por exemplo. Assim, este trabalho propõe a primeira menção a jogos voltados para educação como jogos sob demanda ou *Cloud Gaming*.

O planejamento e desenvolvimento desse trabalho envolveram uma metodologia específica e relacionada a engenharia de *software*. Primeiramente, foi realizada uma análise de viabilidade do projeto, verificando projetos semelhantes e analisando tecnologias. Depois de ter assegurado que o projeto era viável, foi elaborada uma análise de tecnologias e linguagens que seriam necessárias para o desenvolvimento do projeto. Em seguida, um projeto para a solução foi feito e seu desenvolvimento foi iniciado. Após a conclusão do desenvolvimento, a etapa de testes foi implementada, seguida das análises dos resultados e para realizar os testes foi criado um jogo com a finalidade de validar o ambiente.

É preciso considerar que estão impostos alguns limites na execução deste trabalho. O principal está relacionado ao custo que um dispositivo móvel, capaz de ser submetido ao modelo proposto, possui atualmente. Custo esse que a maioria dos alunos das escolas públicas brasileiras ainda não dispõe. Desse modo, seria necessária uma redução nos preços desses aparelhos, bem como nos planos de telefonia e dados afim de disseminar essa solução para todos. O segundo limite está relacionado a conexão com a internet, pois para qualquer dispositivo móvel se conectar com a nuvem é necessária a conexão 3G ou 4G de internet móvel ou uma conexão a uma rede com internet wi-fi. Por fim, o terceiro limite está relacionado às tecnologias envolvidas no desenvolvimento dos jogos, na versão atual proposta, já que somente jogos desenvolvidos na linguagem Java são aceitos. Entretanto, é possível em futuras versões o desenvolvimento em outras linguagens.

## **1.1 Organização da Dissertação de mestrado**

Este documento apresenta 5 capítulos e está estruturado da seguinte forma:

- **Capítulo 1 - Introdução:** Contextualiza o âmbito, no qual a pesquisa proposta está inserida. Apresenta, portanto, a definição do problema, objetivos, metodologia e justificativas da pesquisa e como esta dissertação de mestrado está estruturada;
- **Capítulo 2 - *Cloud Computing* (Computação em Nuvem) para *Games*:**



Aborda a revisão da literatura sobre o tema de *Cloud Computing* e *Cloud Gaming*;

- **Capítulo 3 - O Modelo:** Este capítulo abrange o modelo proposto nesta dissertação, detalhes do desenvolvimento, assim como, alguns esclarecimentos do uso da nuvem;
- **Capítulo 4 - Testes e Resultados:** Demonstra como foram realizados os testes para validar a solução, também aborda detalhes sobre o jogo criado para esses testes e detalhes do ambiente/ hardwares usados;
- **Capítulo 5 - Considerações Finais:** Apresenta algumas sugestões de trabalhos futuros e conclusões desse trabalho.

---

## *Cloud Computing* (Computação em Nuvem) para *Games*

---

O tema *Cloud Computing* é vasto e pode ser descrito utilizando diferentes abordagens, contendo informações de rede, de computação paralela, de segurança, de tolerância a falhas, dentre outros. Nesta revisão bibliográfica, o tema é adequado aos conceitos de *Cloud Computing* em jogos, ou *Cloud Gaming*, focando nos serviços de jogos sob demanda.

Este capítulo aborda a revisão de literatura inerente a *Cloud Computing* ou Computação em Nuvem. Na primeira seção, é dissertado um breve histórico sobre o tema (seção 2.1), depois alguns conceitos que abrangem *Cloud computing* são mencionados (seções 2.2, 2.3, 2.4 e 2.5) e, por fim, uma abordagem a *Cloud Gaming* é feita para trazer os conceitos da Computação em Nuvem para os jogos digitais (seção 2.6).

### **2.1 Histórico**

A ideia de integrar computadores e disponibilizar serviços de software e hardware não é nova. Em 1966 um grupo de pesquisadores do MIT - *Massachusetts Institute of Technology* já publicava trabalhos (MARILL; ROBERTS, 1966) que originariam a primeira rede de computadores por comutação de pacotes: a ARPAnet. Esta se tornaria uma ancestral direta da internet que conhecemos hoje (KUROSE; ROSS, 2003).

Entretanto, antes mesmo do trabalho do MIT, ainda na década de 50, uma grande escala de Mainframes já estavam disponíveis em instituições de ensino e grandes corporações, sendo acessadas por clientes *thin* (definidos na seção 2.2) da época. Mas, devido ao alto custo dos Mainframes, a flexibilidade para o usuário posicionava-se em segundo plano.

Na década de 70, a ARPAnet evoluiu e muitas outras redes surgiram: a ALOHAnet, uma rede de micro-ondas que interligavam as universidades localizadas nas ilhas do Havaí; a Telenet; a Tymnet; a Transpac, dentre outras. Em 1973, surgiu os princípios da Ethernet, precursora das LANs, existentes hoje.

Já, em 1974, um trabalho financiado pela DARPA - *Defense Advanced Research Projects Agency* (Agência de Projetos de Pesquisa Avançada de Defesa) criou uma rede de redes (CERF; KAHN, 1974). Então, na descrição do trabalho, o termo "*interneting*" foi utilizado pela primeira vez e os princípios que fundamentaram o projeto da internet basearam-

se no mesmo trabalho (KUROSE; ROSS, 2003).

Com a criação e distribuição do sistema operacional Windows e também de sistemas baseados em baixa plataforma, começaram a aparecer soluções cliente/servidor e soluções distribuídas, que aumentaram a flexibilidade necessária ao usuário, diminuindo o custo com recursos (VERAS, 2012).

Na década de 90, a WWW - *World Wide Web*, começou a ser usada por milhares de pessoas e uma série de aplicações foram lançadas, como serviços de mídia, serviços bancários, etc. A *Web* foi inventada pelo CERN - *European Center for Nuclear Physics* (Centro Europeu para Física Nuclear), quando criaram versões dos protocolos HTTP - *HyperText Transfer Protocol*, HTML - *HyperText Markup Language*, um servidor *Web* e um *browser* (navegador) (KUROSE; ROSS, 2003).

Ainda no final da década de 90, antes mesmo das publicações da Amazon e da Google sobre *Cloud Computing*, um dos primeiros marcos dos princípios da nuvem foi o Salesforce.com, em 1999, uma empresa que disponibilizava serviços e aplicações corporativas por um simples *site* na internet (MOHAMED, 2009).

Hoje, a Computação em Nuvem é uma realidade, utilizada por diversas empresas de TI - Tecnologia da Informação, ou que usam a TI, para disponibilizar ou consumir seus serviços e aplicações na *Web*.

## 2.2 Definições

*Cloud Computing* ou Computação em Nuvem é um conceito que ainda está em aprimoramento. Contudo, o nome *Cloud Computing* surgiu devido a inúmeros autores usarem uma nuvem para representar a internet em imagens, diagramas e gráficos. Portanto, esse nome é uma metáfora da internet atual (VELTE; VELTE; ELSENPETER, 2011)(TAURION, 2009)(MARTINS, 2010) e um exemplo de uma imagem contendo essa metáfora está abaixo, na Figura 2.1, imagem da conhecida referência de Tanenbaum e Steen (2007).

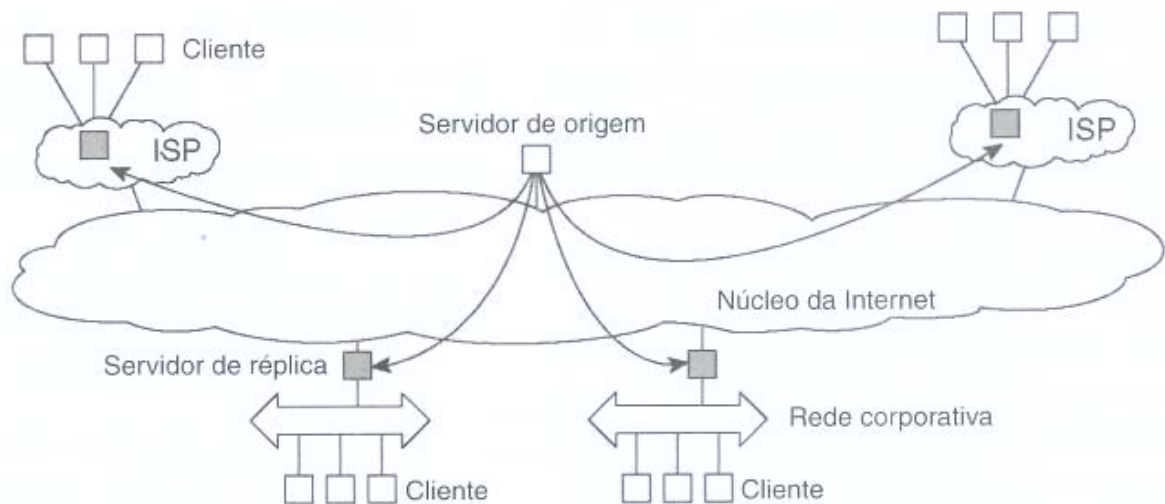


Figura 2.1: Um modelo de servidor qualquer que usa uma nuvem como metáfora da internet (TANENBAUM; STEEN, 2007).

Segundo Armbrust et al. (2009) e Armbrust et al. (2010), a *Cloud Computing* se refere ao *hardware* e ao *software* que está no *datacenter* distribuído como um serviço na internet. Já Mell e Grance (2011), afirmam que Computação em Nuvem é um modelo para habilitar uma rede de acesso ubíqua, conveniente e sob-demanda para compartilhar recursos computacionais configuráveis, tais como: redes, servidores, armazenamento, aplicações e serviços.

Velte, Velte e Elsenpeter (2011) apontam que *Cloud Computing* é uma ideia que permite utilizar diversas aplicações na internet, ao invés de tê-las instaladas no próprio computador. Taurion (2009) define Computação em Nuvem como um ambiente de computação baseado em uma imensa rede de servidores, sejam estes virtuais ou físicos. Existem autores, como Barboza et al. (2010) que definem *Cloud Computing* como um novo paradigma da computação.

Um modelo de Computação em Nuvem deve possuir algumas características essenciais para sua implementação. Segundo Mell e Grance (2011), o NIST - *National Institute of Standards and Technology* define essas características como:

- **Autoatendimento sob Demanda:** serviços computacionais podem ser consumidos automaticamente, sem requerer interação humana aos servidores;
- **Largo acesso aos serviços de rede:** recursos computacionais estão disponíveis na internet através de mecanismos padronizados, provendo assim o acesso para di-

ferentes tipos de clientes;

- **Pool de Recursos:** O provedor dos recursos computacionais os disponibiliza dinamicamente, servindo a múltiplos consumidores de forma transparente, em que o consumidor não sabe onde está o servidor que está consumindo;
- **Elasticidade Rápida:** Os recursos computacionais devem ser rápidos e elasticamente providos, em alguns casos até automaticamente. Assim, é importante que transmitam a sensação para o usuário de que os recursos que ele utiliza são ilimitados e podem ser consumidos em qualquer quantidade e a qualquer momento;
- **Serviços Mensuráveis:** Ambientes em *Cloud* podem controlar e otimizar o uso dos recursos medindo e mensurando a capacidade do tipo de serviço. Dessa maneira, os usos dos recursos são monitorados, controlados e reportados provendo transparência para o provedor e o consumidor do serviço.

Um dos principais componentes em um ambiente em *Cloud Computing* são os clientes. Os clientes estão em uma arquitetura de Computação em Nuvem, da mesma forma que em uma LAN - *Local Area Network* (Rede de Área Local) (VELTE; VELTE; ELSENPETER, 2011). Eles são os notebooks, tablets, smartphones que os usuários finais utilizam para interagirem com a nuvem. Velte, Velte e Elsenpeter (2011) definem os cliente nas seguintes categorias:

- **Dispositivos Móveis:** são compostos dos antigos PDAs - Personal Digital Assistant (Assistente Digital Pessoal), dos *smartphones* e *tablets*;
- **Clientes *thin*:** são computadores que não tem discos rígidos internos, e todo o trabalho/processamento é realizado no servidor;
- **Clientes *thick*:** é considerado como o computador pessoal normal, usualmente utiliza um *browser* para acessar a nuvem pela *Web*.

### 2.3 *TIaaS - TI as a Service (TI como Serviço) e Cloud Computing*

A TI, nos conceitos de *Cloud Computing*, é fornecida como serviço. Serviço é o conceito de poder utilizar componentes reutilizáveis, inserido em uma rede de vendas (VELTE; VELTE; ELSENPETER, 2011). Esse conceito é conhecido como "*as a service*" (como serviço). Além disso, serviço também pode ser considerado como um benefício que uma organização entrega/disponibiliza para outra organização (VERAS, 2012).

Dentro desse cenário, a Computação em Nuvem destaca três principais formas de serviços (MARTINS, 2010)(MELL; GRANCE, 2011)(VELTE; VELTE; ELSENPETER, 2011)(VERAS, 2012)(SOBRAGI, 2012) que estão melhores descritas abaixo:

- **SaaS - *Software as a Service* (Software como Serviço):** é o modelo em que uma aplicação é disponibilizada na internet para clientes diversos. Com isso elimina-se a necessidade de ter essa determinada aplicação instalada no computador do usuário. Essas aplicações são acessadas pelo usuário utilizando algum tipo de cliente, como por exemplo um *browser*;
- **PaaS - *Plataform as a Service* (Plataforma como um Serviço):** é um modelo que oferece os recursos necessários para desenvolver aplicações e serviços diretos da internet, não sendo necessário assim instalar o *software* no computador do usuário. Um exemplo de PaaS é a famosa Google Engine, da própria empresa Google, que pode ser utilizada para desenvolver aplicações a fim de serem disponibilizadas em sua nuvem;
- **IaaS - *Infrastructure as a Service* (Infraestrutura como um Serviço):** também conhecida como HaaS - *Hardware as a Service*, é outro modelo que diferente da SaaS e PaaS que provêem software, a IaaS oferece recursos físicos computacionais para o usuário. Esses recursos podem ser processamento, memória, armazenamento, etc. Todas essas opções são transparentes para o usuário final.

## 2.4 Modelos de Implementação

Existem quatro modelos de implementações, nas quais é possível projetar e configurar uma determinada nuvem. Esses quatro modelos são (VERAS, 2012)(SOBRAGI, 2012)(MELL; GRANCE, 2011):

- **Nuvem Privada (*Private CLOUD*):** a infraestrutura da nuvem é operada somente pela organização cliente. Pode ser gerenciada pela própria empresa ou via terceirização. Nela serviços são disponibilizados para serem consumidos pela própria organização. A sua localização pode ser dentro da organização ou fora dela;
- **Nuvem Pública (*Public CLOUD*):** a infraestrutura da nuvem está disponível para o público em geral, organizações e indústria, normalmente sob o modelo pague-por-uso. É oferecida por entidades que possuem grande capacidade de processamento e armazenamento;
- **Nuvem Comunitária (*Community CLOUD*):** a infraestrutura da nuvem é compartilhada por várias organizações que possuem interesses e objetivos comuns.

Pode ser gerenciada pelas organizações ou por terceiros e a sua localização pode ser dentro das organizações ou fora delas;

- **Nuvem Híbrida (*Hybrid CLOUD*)**: a infraestrutura da nuvem é formada por duas ou mais nuvens, sejam elas privadas, públicas ou comunitárias, que permanecem únicas. Elas são padronizadas e unidas por tecnologias que permitem a portabilidade de dados e aplicações.

Uma melhor ilustração dos conceitos mencionados acima podem ser visto na Figura 2.2, uma proposta gráfica do NIST (MELL; GRANCE, 2011) e de Veras (2012) para união dos conceitos.

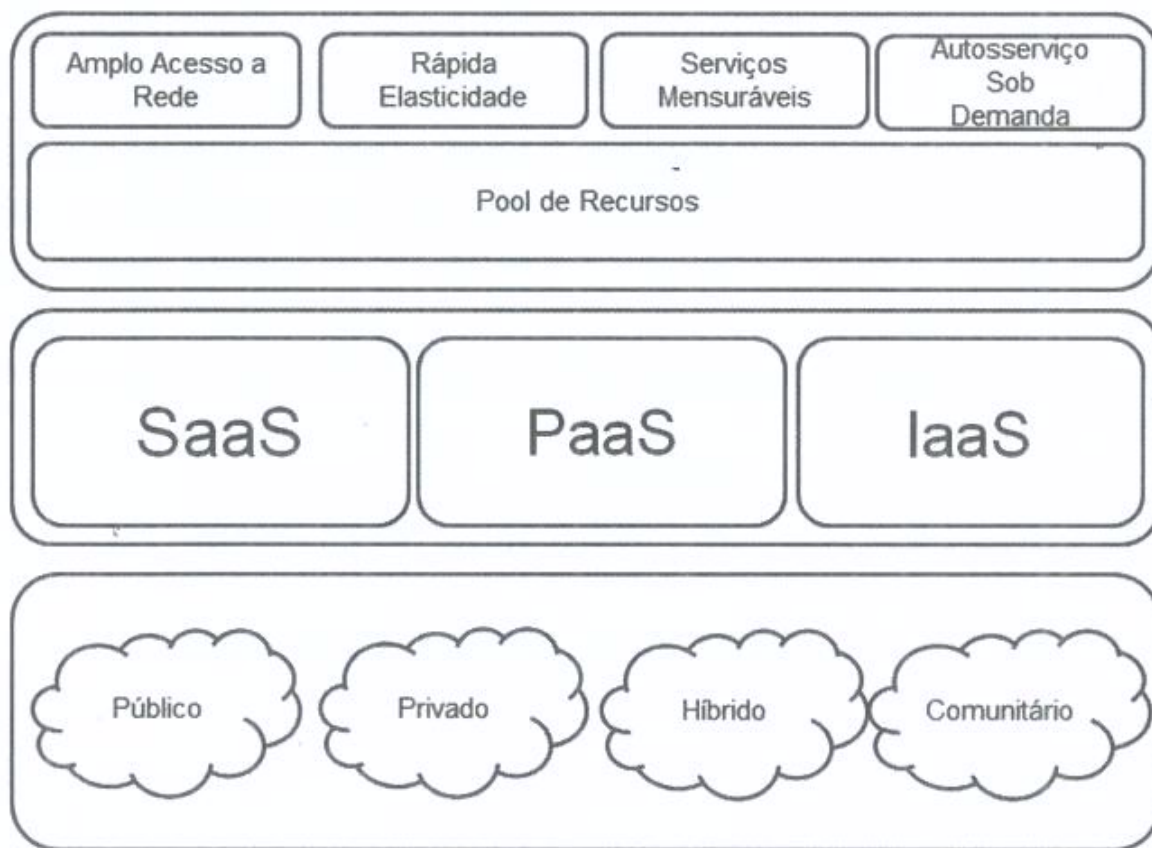


Figura 2.2: Conceitos da *Cloud Computing* (VERAS, 2012).

## 2.5 *Cloud Computing e Grid Computing*

Muitos autores, a exemplo de Taurion (2009), Armbrust et al. (2010), Velte, Velte e Elsenpeter (2011) e Foster et al. (2008), divergem quando o assunto é *Cloud Computing* e

*Grid Computing* (Computação em Grade). Alguns afirmam que compartilham os mesmos conceitos, enquanto outros referem que são conceitos totalmente diferentes.

Para Taurion (2009), a Computação em Nuvem é evolução natural da convergência de outros conceitos, como: Virtualização, *Grid Computing*, *Utility Computing* e *Autonomic Computing*. Ela é uma evolução que segue tecnologias e tendências em uso no momento, como a *Web 2.0*, SOA - *Service Oriented Architecture* (Arquitetura Orientada a Serviço) e o modelo de SaaS.

Já Armbrust et al. (2010), relatam que o termo *Grid Computing*, relacionado a comunidade de computação de alta performance, sugere protocolos para oferecer computação compartilhada e armazenamento em longas distâncias. Entretanto, esses protocolos não suportam o ambiente de *software* que cresceu paralelo a essa comunidade.

Velte, Velte e Elsenpeter (2011) afirmam que Computação em Nuvem e Computação em Grade são conceitos completamente diferentes. Os autores afirmam que enquanto na *Grid Computing* um projeto é dividido em computadores múltiplos para empregar seus recursos, na *Cloud Computing* ocorre exatamente o contrário, já que permite o funcionamento de múltiplos aplicativos ao mesmo tempo.

Algumas aplicações reais da Computação em Grade, como explicam Velte, Velte e Elsenpeter (2011), são a SETI@home - *Search for Extraterrestrial Intelligence at home* e a WLCG - *Worldwide LHC Computing Grid*.

A SETI@home é um grupo científico de pesquisa que tem como objetivo detectar formas de vida inteligentes fora da terra (SETI@HOME, 2013). O grupo criou uma *Grid* para interpretar sinais de rádio vindos do espaço. A WLCG é um projeto global, com o objetivo de usar os recursos da grade para analisar os dados provenientes do LHC - *Large Hadron Collider* (WLCG, 2013). Ambas as grades foram criadas para um objetivo em comum, enquanto na nuvem da Google, por exemplo, é possível adicionar diferentes aplicações em sua arquitetura (GOOGLE, 2013), como aplicações pessoais, corporativas, educacionais, etc.

Uma excelente comparação é feita por Foster et al. (2008, p. 1), os quais fazem e respondem a seguinte pergunta: "Então Computação em nuvem é só um novo nome para a Grade?". Eles mesmos respondem:

**Sim:** a visão é a mesma. Para reduzir os custos da computação, aumentar a confiabilidade, e aumentar a flexibilidade transformando computadores que compramos e operamos nós mesmos para algo que seria operado por uma terceira entidade.

**Mas não:** as coisas são diferentes agora comparado a 10 anos atrás. Nós temos uma nova necessidade de analisar dados massivos, devido a motivada e alta demanda de computação. Tendo notado os benefícios das mudanças de *mainframes* para *clusters*, nós achamos que os



*clusters* são custosos para operar. *Cloud Computing* tem virtualizações de baixo custo, e, acima de tudo, existem múltiplos bilhões de dólares sendo gastos pelas maiores empresas de TI, Amazon, Google, Microsoft, etc.

**Ainda assim, sim:** Os problemas são na maioria os mesmos na nuvem e na grade. Existe uma necessidade comum de ser capaz de gerenciar grandes ambientes computacionais; implementar computações paralelas que possam ser executadas nesses ambientes; definir métodos pelos quais consumidores do serviço descubram, solicitem e usem os recursos disponibilizados por esses ambientes.

Na mesma publicação, outras tecnologias são discutidas, como *Utility Computing*, *Services Computing*, e sistemas distribuídos em geral. Maiores informações a respeito disso podem ser obtidas em Foster et al. (2008). Contudo, há uma comparação interessante exposta em uma imagem, na qual é difundida a ideia de que a *Web 2.0* abrange quase a maioria do espectro das aplicações orientadas a serviço. Essa imagem pode ser vista na Figura 2.3.

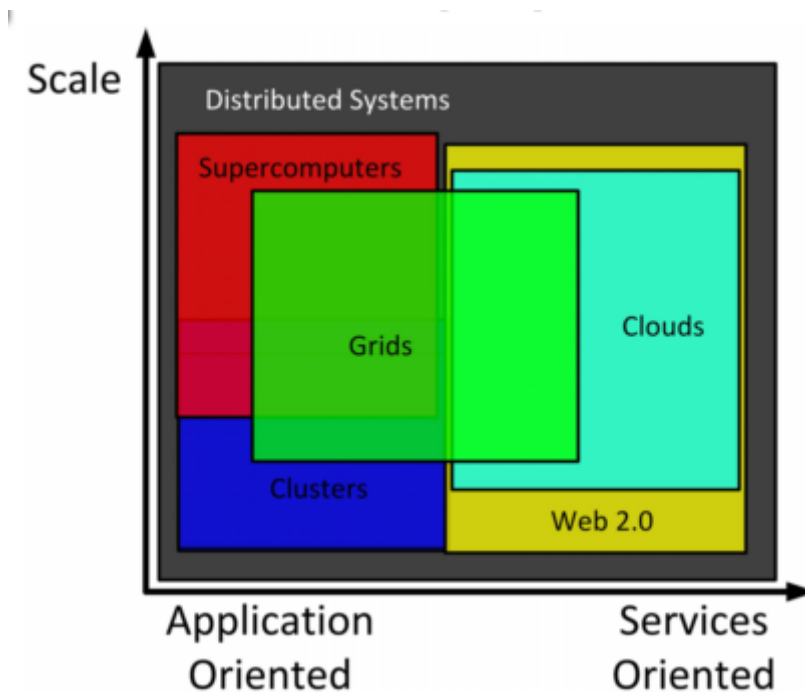


Figura 2.3: Comparação de *Grids*, *Clouds* e outras tecnologias, onde o eixo y se refere a escala, e o x separa a orientação para aplicações ou serviços (FOSTER et al., 2008).

## 2.6 *Cloud Gaming*

A ideia de pessoas usarem clientes *thin* para processar jogos que exijam grande performance computacional não é totalmente recente. A primeira aparição do conceito não foi com a OnLive (ONLIVE, 2013) em 2009, pois em 2000 já existiam pesquisas em desenvolvimento sobre o tema.

Em 2000, uma empresa chamada G-cluster Global (GCLUSTERA, 2013) foi a pioneira em serviços de jogos em uma nuvem. Em 2001 em Los Angeles, nos Estados Unidos, na E3 - *Electronic Entertainment Expo*, esta empresa realizou uma demonstração pública de jogos sendo enviados usando *streams*, dentro de uma rede Wi-Fi (GCLUSTERB, 2013).

Outra empresa que realizou pesquisas na área, antes do aparecimento da OnLive, foi a Crytec (CRYTEC, 2013). A empresa foi fundada em 1999 e é desenvolvedora de jogos famosos como *Crysis*, *Far Cry*, *Warface*, dentre outros. As pesquisas da Crytec começaram em 2005 e foram interrompidas em 2007, pois duvidaram da escalabilidade da solução, já que na época a internet não era tão rápida e era muito mais cara comparada aos dias de hoje (DOBRA, 2013).

Então, em 2009, na GDC - *Game Developers Conference*, a OnLive fez uma demonstração básica sobre serviços de jogos sob demanda. Na apresentação foi mencionado que era necessária uma internet de 1,5 megabytes por segundo para vídeos com resolução de 480p e de 5 megabytes por segundo para HD (720p) (1UP, 2009). No mesmo evento foi divulgado outro serviço de jogos sob demanda, O Gaikai (que hoje pertence a Sony) (GAIKAI, 2013). O que destacou a OnLive e a Gaikai das outras empresas foi o imediato apoio de grandes desenvolvedoras de jogos como: Ubisoft, Electronic Arts, Eidos, THQ, Codemasters, Warner Bros, dentre outros.

Com todo esse apoio, em 2010, a OnLive lançou o seu serviço cobrando uma taxa mensal de 4,95 dólares mais os custos dos jogos e do microconsole do OnLive. Uma melhor noção do que seria esse microconsole, pode ser obtida na Figura 2.4.



Figura 2.4: Microconsole preto distribuído pela OnLive para jogar em televisões em geral (ON-LIVE, 2013).

A NVIDIA criou uma solução arrojada para *Cloud Gaming* chamada de "portfólio NVIDIA GRID" (NVIDIAGRID, 2013). Este portfólio abrange soluções em *software* e *hardware* a fim de prover acesso, para diferentes usuários, a jogos com intensos usos de gráficos usando GPUs - Graphics Processing Unit (Unidade de Processamento Gráfico) compartilhados. Sua estrutura é composta por 20 servidores, com um total de 240 NVIDIA GPUs, o que equivale a aproximadamente 700 Xbox 360s (BURNS, 2013). Naturalmente, a NVIDIA também está trabalhando em clientes para acessar a nuvem. Um de seus produtos, o NVIDIA Shield, está sendo preparado para essa função, ele utiliza um SO Android e possui um controle acoplado para os jogos. Contudo, ainda está na versão beta e só pode ser utilizado na Califórnia (NVIDIASHIELD, 2013a). Uma imagem do NVIDIA Shield pode ser visualizada na Figura 2.5.



Figura 2.5: NVIDIA Shield (NVIDIASHIELD, 2013b).

Uma publicação recente, divulgada no decorrer desse trabalho de pesquisa, foi o GamingAnywhere (GAMINGANYWHERE, 2013), a primeira nuvem especializada em jogos disponível no formato *open source* (código aberto) (HUANG et al., 2013).

A GamingAnywhere possui uma arquitetura muito parecida com a proposta por este trabalho de dissertação. Os usuários utilizam aplicações clientes em PCs - *Personal Computers* (Computadores Pessoais) e dispositivos móveis e se conectam no ambiente que a equipe criadora do GamingAnywhere denomina "Servidores portais". Depois, a responsabilidade e configurações são direcionadas para os "Servidores de jogos" que estabelecem a conexão direta do jogo com o aplicativo/sistema cliente. Essa arquitetura pode ser vista na Figura 2.6.

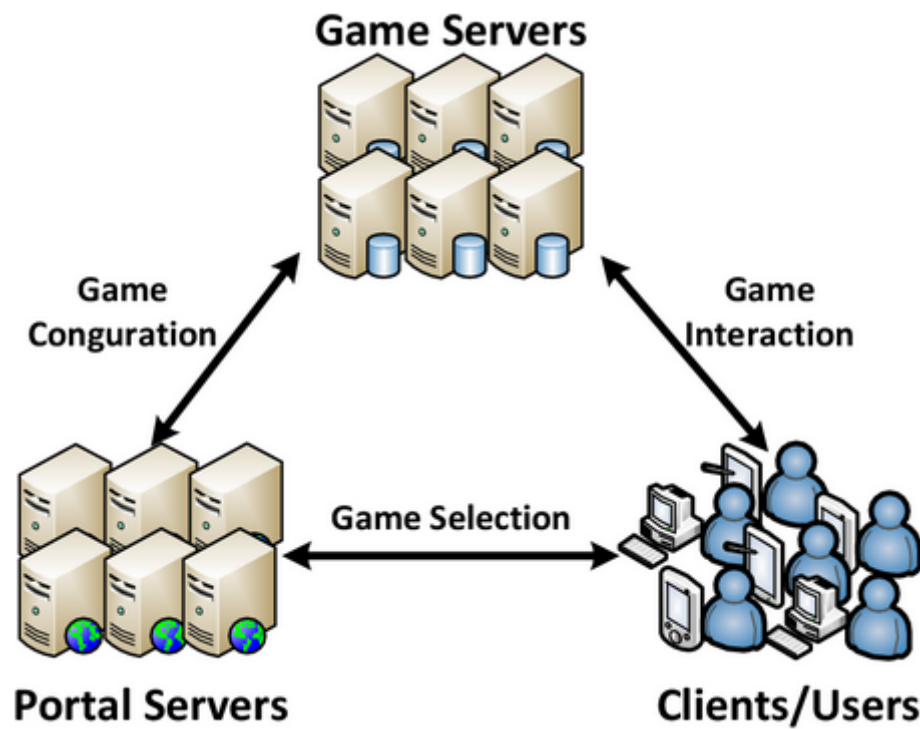


Figura 2.6: Arquitetura disponibilizada do GamingAnywhere (GAMINGANYWHERE, 2013).

Outra arquitetura importante foi desenvolvida por um grupo de pesquisadores da UFF - Universidade Federal Fluminense e pode ser vista na Figura 2.7. Esta arquitetura, assim como a proposta pela GamingAnywhere, possui uma aplicação cliente e um *Cloud Manager* (chamado de *Portal Manager* pela GamingAnywhere), mas difere da anterior por possuir um *Host Manager*, que gerencia cada máquina virtualizando recursos para a nuvem.

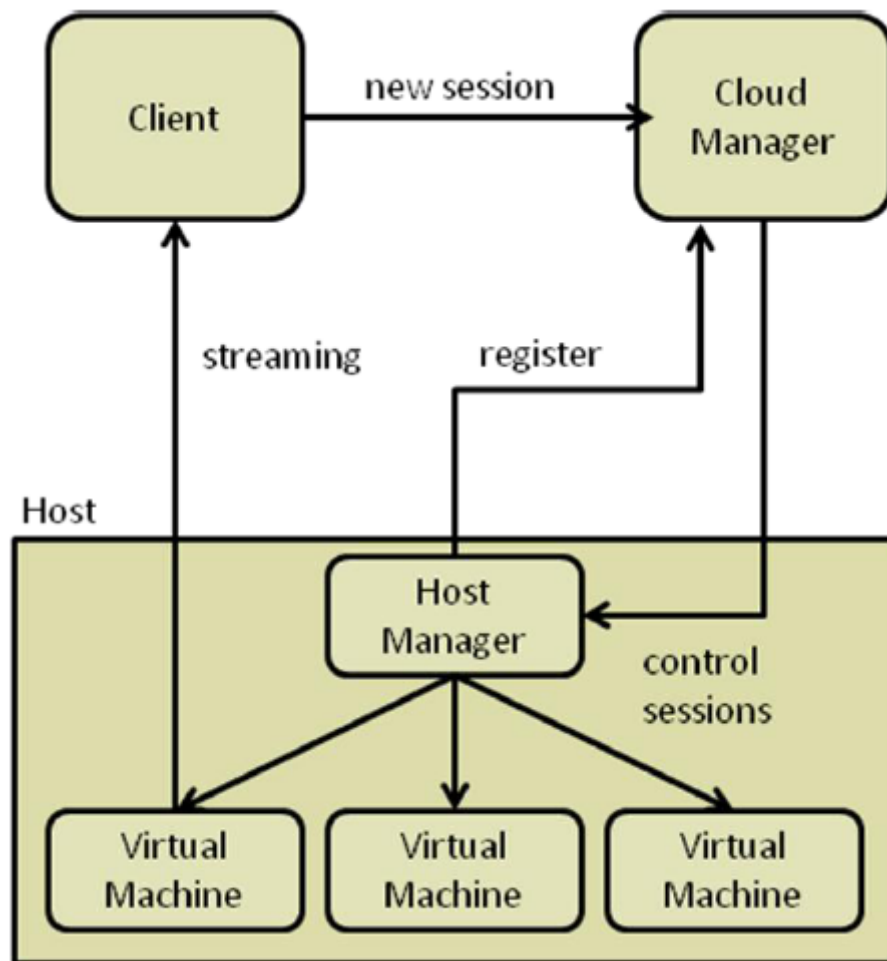


Figura 2.7: Arquitetura desenvolvida por um grupo de pesquisadores da UFF. (BARBOZA et al., 2010).

*Cloud Gaming* também é um conceito que está em aprimoramento. Até o nome ainda não está consolidado, além de *Cloud Gaming*, é denominado *game on demand* (jogo sob demanda) (LEE et al., 2012).

*Cloud Gaming* poderia ser descrito como um deslocamento da execução, armazenamento e renderização do jogo para um servidor remoto. O jogo é renderizado e enviado por *streams* através de uma conexão de internet, permitindo assim, a execução de jogos em clientes *thin* e dispositivos móveis (SUSELBECK; SCHIELE; BECKER, 2009)(BARBOZA et al., 2010)(CHEN et al., 2011)(JARSCHEL et al., 2011). Nesse contexto mais um tipo de serviço surge, o GaaS - *Game as a Service* (Jogo como um serviço), em que o próprio jogo é disponibilizado para consumo de diferentes clientes.

O CloudGamingReport (2012) previu que o mercado de jogos em *Cloud Gaming* irá aumentar 9 vezes no período de 2011 a 2017. Esta previsão demonstra o potencial que

essa solução possui, principalmente pelo fato de o jogador poder usufruir de qualquer jogo, que esteja hospedado em uma nuvem, em qualquer lugar com internet que queira, e em qualquer dispositivo desejado, com a limitação de ter um aplicativo cliente desenvolvido para este dispositivo.

Acrescenta-se também que, além das vantagens descritas no parágrafo anterior, os custos envolvidos com *hardware* e *software* para o jogador ainda são reduzido, comparado aos mesmos custos comprando jogos tradicionais, fora da nuvem. (HUANG et al., 2013).

Relacionando essa potencial tecnologia com a educação, Nolan Bushnell, fundador do Atari, em uma palestra na *Cloud Gaming Conference* nos Estados Unidos, mencionou que a *Cloud Computing* melhorará a educação (MATULEF, 2011). Ele, que trabalha com projetos educacionais na computação há mais de dez anos, afirma que a geração de crianças em evolução cognitiva, hoje, caminha junto com computadores. Bushnell está trabalhando com centenas de classes com cerca de 40000 estudantes e afirma que atualmente eles estão ensinando as crianças 10 vezes mais rápido e a Cloud Computing contribuirá na melhora dessa estatística (MATULEF, 2011).

## O Modelo

---

O modelo de arquitetura descrito nessa seção tem como objetivo a construção de uma nuvem para hospedar jogos voltados para a educação, com a finalidade de serem jogados em dispositivos móveis com o sistema operacional Android. Esse objetivo surgiu em virtude da estrutura computacional decadente das escolas, com máquinas de baixo desempenho computacional, que não suportam a execução de jogos com fins educacionais pelos alunos e professores, aliado ao fato de não haver computadores para todos.

Na versão atual da nuvem, presente neste trabalho, ainda é possível adicionar *games* de qualquer tipo, ou seja, ainda não há especificidade relacionada a jogos voltados para a educação. Entretanto, este é somente o início de um projeto que evoluirá principalmente nas questões pedagógicas e nos estudos referentes à interação humano-computador. Dessa forma, ao final destes estudos, a nuvem se tornará exclusiva para o uso de Jogos com fins educacionais.

Este capítulo abrange o modelo de arquitetura proposto nesta dissertação. Estão descritas as linguagens e o ambiente para o desenvolvimento do mesmo (seção 3.1), assim como os artefatos/modelos gerados para desenvolvimento do trabalho (seção 3.2). Em seguida, alguns elementos importantes da implementação também estão presentes (seção 3.3). Depois é demonstrado o jogo que foi desenvolvido para testar a arquitetura (seção 3.4) e, por fim, uma abordagem sobre a comunicação entre o jogo e a nuvem (seção 3.5).

### **3.1 Linguagens e Ambiente de Desenvolvimento**

Muitas tecnologias e linguagens foram usadas no desenvolvimento da nuvem proposta. As principais delas serão descritas no decorrer dessa seção. Inicialmente são abordadas as linguagens utilizadas e, a seguir, o ambiente de desenvolvimento.

A principal linguagem usada nesse projeto foi a Java (JAVA, 2013), que é uma linguagem de alto nível, e, apesar de possuir variáveis de tipos primitivos, seu desenvolvimento é obrigatoriamente orientado a objetos. Uma linguagem orientada a objetos está baseada nos conceitos do paradigma de Orientação a Objetos, que abstrai para a programação conceitos do mundo real.

Java foi escolhida por diversos motivos, dentre eles (SANTOS, 2003):



- **Java é obrigatoriamente Orientada a Objetos:** Existem linguagens que permitem a criação de variáveis e funções em diferentes partes do programa, enquanto que na linguagem Java tudo deve estar dentro de uma classe;
- **Java é portátil:** o código-fonte de um programa ou classe Java pode ser compilado em qualquer sistema operacional que possua uma Máquina Virtual Java. Em ambientes de configuração de nuvens essa característica é necessária;
- **Java é gratuita:** A JVM - *Java Virtual Machine* (Máquina Virtual Java) e o JDK - *Java Development Kit* (Conjunto de Desenvolvimento Java) são distribuídos gratuitamente pela empresa Oracle, proprietária do Java;
- **Java possui bibliotecas prontas para diversas aplicações:** As bibliotecas de classes Java já implementam diversos recursos para ajudar o programador, tais como: manipulação de strings, acesso a internet, estrutura de dados, etc.

Java foi a principal linguagem, mas não foi a única utilizada. A XML - *eXtensible Markup Language* (Linguagem de Marcação Extensiva) também foi usada, principalmente na parte de comunicação e na criação do cliente para o SO - Sistema Operacional Android.

Para o desenvolvimento da solução deste projeto uma IDE - *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado) foi utilizada: a Eclipse (ECLIPSE, 2013). Criada originalmente pela IBM, em 2001, essa IDE suporta várias linguagens, inclusive a Java que foi utilizada nesse trabalho. A Eclipse foi escolhida por ser gratuita e por ser aceita e usada largamente pela comunidade desenvolvedora Java. Uma estimativa feita por White (2012), diz que a maioria dos desenvolvedores Java, cerca de 68%, usam Eclipse como IDE.

O desenvolvimento de aplicações clientes para todos os dispositivos móveis seria inviável para este projeto. Portanto, foi necessário realizar um corte no escopo deste projeto e os escolhidos foram os *smartphones* e *tablets* com sistema operacional Android. A preferência por eles foi devido a alguns fatores, dentre eles:

- Android possui um robusto *framework* de desenvolvimento e disponibiliza, sem custos, todas as ferramentas necessárias para o desenvolvimento de aplicações. Disponibiliza também o Android SDK - *Software Development Kit*, que contém bibliotecas e ferramentas de desenvolvimento, necessárias para criar, testar e simular aplicações (LECHETA, 2011)(ANDROIDDEVELOPERS, 2013);
- Não é necessário adquirir uma licença para distribuir versões de uma aplicação desenvolvida;

- O desenvolvimento das aplicações nativas pode ser realizado em Java e agrega tudo o que foi escrito anteriormente pela linguagem;
- Para o desenvolvimento, o Eclipse pode ser usado, unido ao *plugin* ADT - *Android Developer Tools*, uma integração entre o desenvolvimento para Android e o Eclipse agregando, assim, todas as vantagens do uso da IDE.

Para a criação dos modelos presentes neste trabalho foi usada a ferramenta Astah (ASTAH, 2013), utilizada para criação de Diagramas UML - *Unified Modeling Language* (Linguagem de Modelagem Unificada). Ela possui uma versão livre chamada *Astah Community* (ASTAHCOMMUNITY, 2013), que foi usada para gerar os modelos/diagramas abaixo.

## 3.2 EduCloudGaming - Modelos

Nesta subseção são demonstrados alguns dos modelos criados para este trabalho. Entretanto, a seguir, na subseção 3.2.1, está o modelo principal deste trabalho, o modelo da arquitetura do ambiente da nuvem, chamada de EduCloudGaming.

### 3.2.1 Arquitetura

A arquitetura da solução proposta segue os padrões das nuvens já existentes. A arquitetura existente na nuvem GamingAnywhere, presente na Figura 2.6, é um exemplo desses padrões de arquitetura.

Para este trabalho, a arquitetura que pode ser vista na Figura 3.1 foi organizada seguindo a estrutura de *cluster* existente no SENAI-CIMATEC. Vale salientar que a arquitetura pode ser utilizada em qualquer estrutura física, ou seja, ela é independente do *cluster* do CSII no SENAI-CIMATEC. Contudo, como o *cluster* de desenvolvimento estava disponível para os alunos do mestrado, o mesmo foi utilizado para evitar custos ao projeto, já que no SENAI uma estrutura computacional para simular a *cloud*.

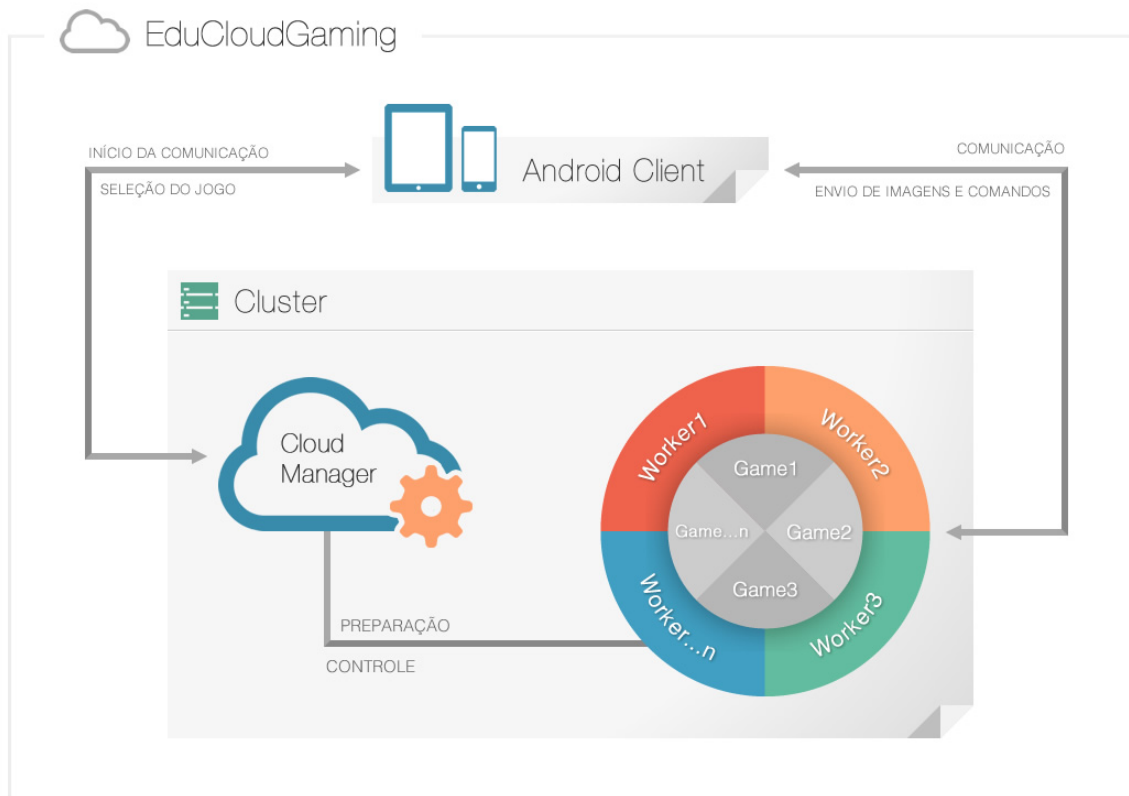


Figura 3.1: Arquitetura Cloud Gaming deste trabalho.

Existe um quadro na figura anterior que representa as máquinas do *cluster* de desenvolvimento do CSII no SENAI-CIMATEC. Inserido nele pode-se visualizar os elementos/sistemas da nuvem. Todos esses elementos são virtualizações JVM que possibilitam a utilização de vários elementos em uma mesma máquina.

A principal instância da EduCloudGaming é a *Cloud Manager*, instância/sistema responsável pela gerência da nuvem. Ela controla todos os clientes que iniciam sua comunicação com a nuvem, controla e gerencia todos os *Workers*, delegando-os a responsabilidade do fluxo do jogo do cliente.

Outra instância importante é a *Worker* (Trabalhador). Atualmente, é possível instanciar quantos *Workers* forem necessários a depender da limitação da estrutura física das máquinas da nuvem. Essa instância é responsável pela execução do jogo, que foi escolhido pelo jogador no Cliente Android, estabelecendo uma comunicação direta com o mesmo.

Outro elemento que pode-se verificar na imagem é o Cliente Android. Vale salientar que

a nuvem está preparada para qualquer cliente (desenvolvido para qualquer sistema operacional), contudo, como escopo do projeto, somente o cliente Android foi desenvolvido. Ele não executa nenhum jogo nativamente, pois todo o processamento do jogo é realizado na nuvem.

Para cada elemento descrito acima foi desenvolvido um sistema, que inclui um planejamento e uma implementação. O planejamento está descrito nas subseções 3.2.2 e 3.2.3, enquanto a implementação está descrita na seção 3.3.

### 3.2.2 Diagramas de Caso de Uso e Colaboração

Como diagrama inicial para concepção da solução foi criado o diagrama de Caso de Uso presente na Figura 3.2.

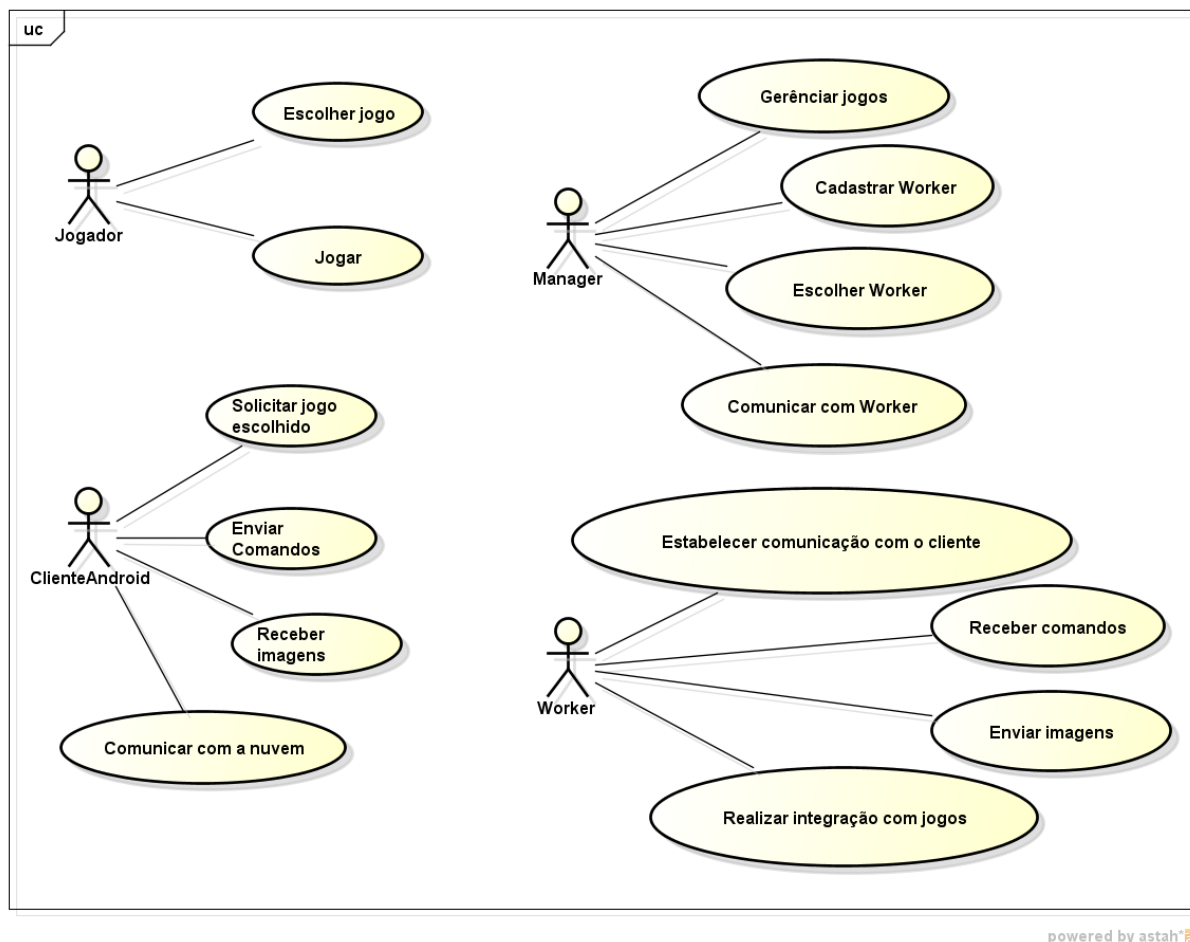


Figura 3.2: Diagrama de Caso de Uso deste trabalho.

O diagrama de Caso de Uso foi dividido em quatro visões de acordo com os atores presentes no modelo. A primeira visão, a do ator **Jogador**, contém as ações que o Jogador realiza para utilizar a nuvem. Ele, de forma transparente, pode escolher um *game* presente na nuvem a fim de jogá-lo quando escolhido.

A segunda visão, a visão do **Cliente Android**, abrange as ações que a aplicação cliente, presente nos dispositivos Android, realiza. De forma transparente para o usuário/jogador, ele se comunica com a nuvem, solicita o jogo escolhido pelo jogador, envia os comandos durante o jogo, recebe e exibe as imagens enviadas pela nuvem.

A terceira visão, a visão do **Manager**, contém as ações que o sistema gerenciador da nuvem executa. Ele deve gerenciar quais jogos existem na nuvem, além de cadastrar e gerenciar os sistemas trabalhadores que existem nela. Após a escolha do jogo pelo jogador, o gerenciador deve escolher qual trabalhador executa o jogo, e assim, estabelece comunicação com o trabalhador e define algumas configurações.

A última visão, a visão do **Worker**, inclui as ações que o trabalhador da nuvem executa. Após a troca de mensagens/configurações com o gerente da nuvem, o trabalhador deve estabelecer a comunicação com a aplicação cliente de forma transparente. Nesta lógica, o sistema trabalhador também deve realizar a integração com todos os jogos disponíveis, e assim, no momento de executar o jogo, receber os comandos do jogador e enviar as imagens do jogo.

Portanto, nesse contexto, foi definido que a nuvem possuiria 3 implementações (sendo uma biblioteca e 2 sistemas) e um cliente para acessá-la. Maiores detalhes podem ser vistos na Tabela 3.1.

Tabela 3.1: Implementações Criadas para a Nuvem

Implementação	Descrição
EduCloudGaming	Biblioteca criada para auxiliar todas as outras implementações.
ServerManager	Gerente da nuvem e porta de entrada para o cliente.
ServerWorker	Trabalhador da nuvem que executa o jogo para o cliente.
Cliente Android	Cliente que acessa a nuvem desenvolvido para o SO Android.

Os sistemas mencionados na Tabela 3.1 trocam mensagens em seu funcionamento. Essas mensagens, assim como os sistemas, podem ser vistos na Figura 3.3, que contém um diagrama de Colaboração (ou Comunicação) com os passos, em sequência, realizados pelos sistemas.

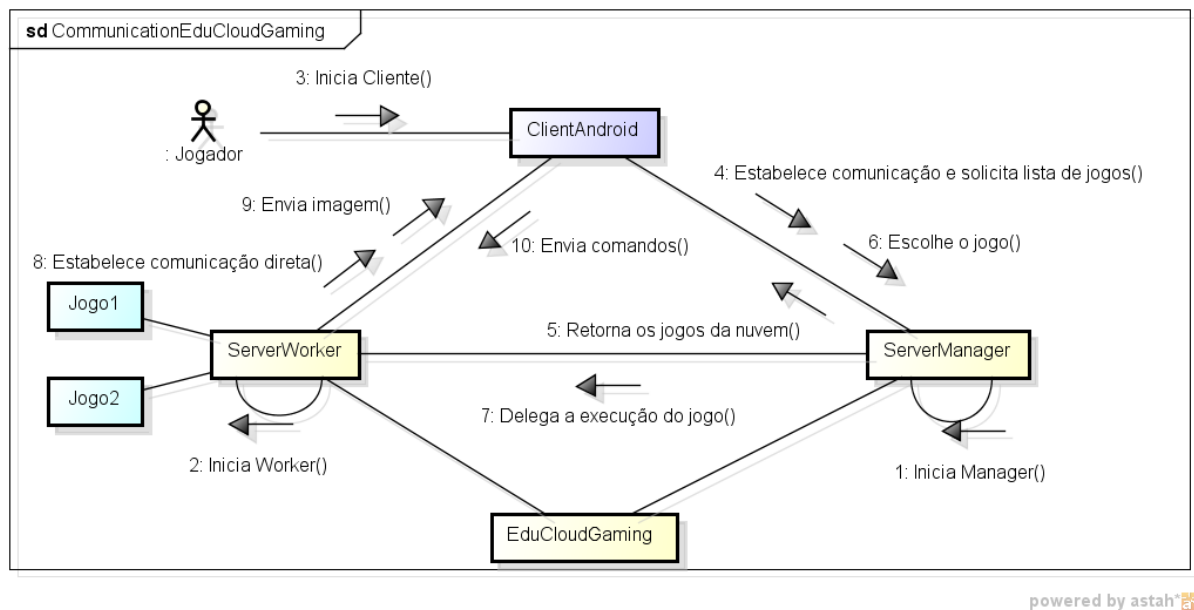


Figura 3.3: Diagrama de Colaboração deste trabalho.

Um melhor entendimento da Figura 3.3 pode ser obtido na lista numerada a seguir, com todos os passos descritos.

1. **Inicia Manager:** o sistema ServerManager deve ser o primeiro a iniciar, para guardar as informações de todos os Workers e Clientes;
2. **Inicia Worker:** o sistema ServerWorker deve ser iniciado depois do Manager, para se autenticar na nuvem;
3. **Inicia Cliente:** quando a nuvem estiver preparada, com gerente e trabalhadores ativos, o jogador pode iniciar o aplicativo cliente;
4. **Estabelece a comunicação e solicita lista de jogos:** primeiramente, o cliente estabelece comunicação com o Manager e solicita ao mesmo a lista com os jogos disponíveis na nuvem;
5. **Retorna os jogos da nuvem:** O sistema Manager verifica os jogos disponíveis e envia a lista para o cliente;
6. **Escolhe o jogo:** O jogador escolhe um jogo, dentre os disponíveis na nuvem;
7. **Delega a execução do jogo:** O sistema ServerManager delega a execução do jogo para o sistema ServerWorker, enviando ao mesmo informações de acesso do cliente;
8. **Estabelece comunicação direta:** Com as informações necessárias, o Worker estabelece comunicação direta com o cliente;

9. **Envia imagem:** O sistema ServerWorker envia imagens do jogo para o cliente;
10. **Envia comandos:** O sistema ServerWorker recebe mensagens com comandos, durante o jogo, do cliente.

### 3.2.3 Diagramas de Classes

Para o planejamento da solução foram criados os diagramas de classes dos sistemas que compõem a nuvem. No decorrer do desenvolvimento, estes diagramas sofreram modificações e evoluíram para os diagramas expostos abaixo, em sua versão final. Vale ressaltar que os diagramas abaixo não estão completos, pois possuem muitas informações que não seriam possíveis apresentá-las aqui. Informações como todos os atributos e todos os métodos referentes aos diagramas de classes não estão presentes, somente os mais importantes.

Foram utilizados Padrões de Projeto na arquitetura da nuvem e alguns deles podem ser vistos nos diagramas abaixo. Padrões de Projeto são um conjunto de informações que relatam, em um determinado contexto, um problema recorrente e sua solução. Esse conjunto de informações é relatado dentro do Padrão de Projeto para que analistas e desenvolvedores possam usá-las sempre que necessário (GAMMA et al., 1994).

Primeiramente, uma biblioteca foi desenvolvida para possibilitar o reuso de *software*. Ela é utilizada em todos os sistemas/aplicações desenvolvidos neste projeto. O diagrama de classes dessa biblioteca pode ser visto na Figura 3.4. Nele já é possível verificar o Padrão de Projeto DTO - *Data Transfer Object* (Objeto de Transferência de Dados), que é usado para transportar dados entre processos a fim de reduzir o número de chamadas a métodos e o número de parâmetros entre eles (FOWLER, 1994).

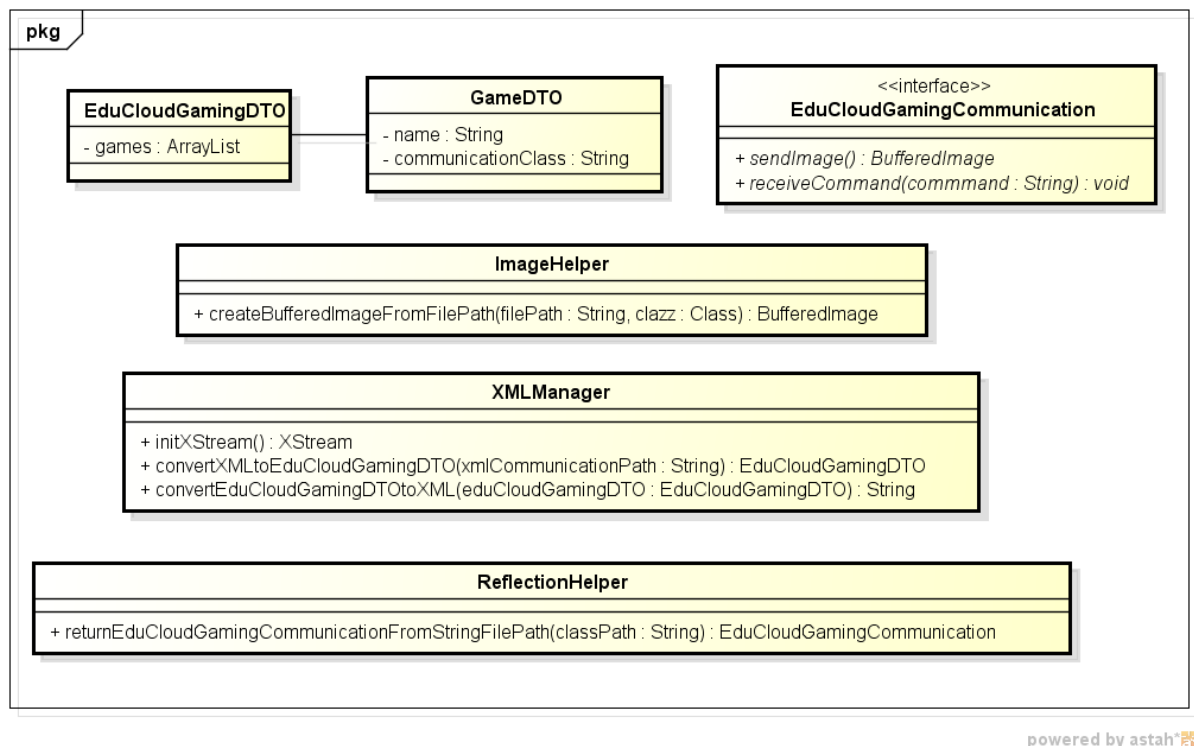


Figura 3.4: Diagrama de Classes da biblioteca EduCloudGaming.

Os objetos DTOs estão representados nas classes `EduCloudGamingDTO` e `GameDTO`. A primeira classe contém informações gerais sobre a nuvem, enquanto que a segunda possui informações gerais sobre os jogos presentes na nuvem.

A interface `EduCloudGameCommunication` é uma das ferramentas usadas para comunicação entre a nuvem, o cliente e o jogo. Mais detalhes de como os mesmos se comunicam podem ser analisados na seção 3.3. Esta interface possui dois métodos que recebem comandos do cliente e enviam imagens do jogo. Assim, todo o jogo desenvolvido para a nuvem deste trabalho deve possuir uma classe que herda desta interface.

A biblioteca também inclui classes que servem como ajuda para outras classes, como por exemplo as classes: `ImageHelper`, `XMLManager` e `ReflectionHelper`. A primeira possui funcionalidades para auxiliar a manipulação de imagens, a segunda tem métodos para ajudar na interpretação de arquivos XML e a terceira possui auxílios para a utilização de Reflection. Reflection é um recurso que habilita programas a acessarem comportamentos e estruturas internas programaticamente (SULLIVAN, 2001).

O diagrama presente na Figura 3.5 está relacionado ao sistema *Server Manager* que gerencia a nuvem e serve de porta de entrada para os clientes. No diagrama pode ser



observada uma outra classe DTO, a WorkerDTO, que possui informações importantes de configuração e comunicação com o gerente da nuvem.

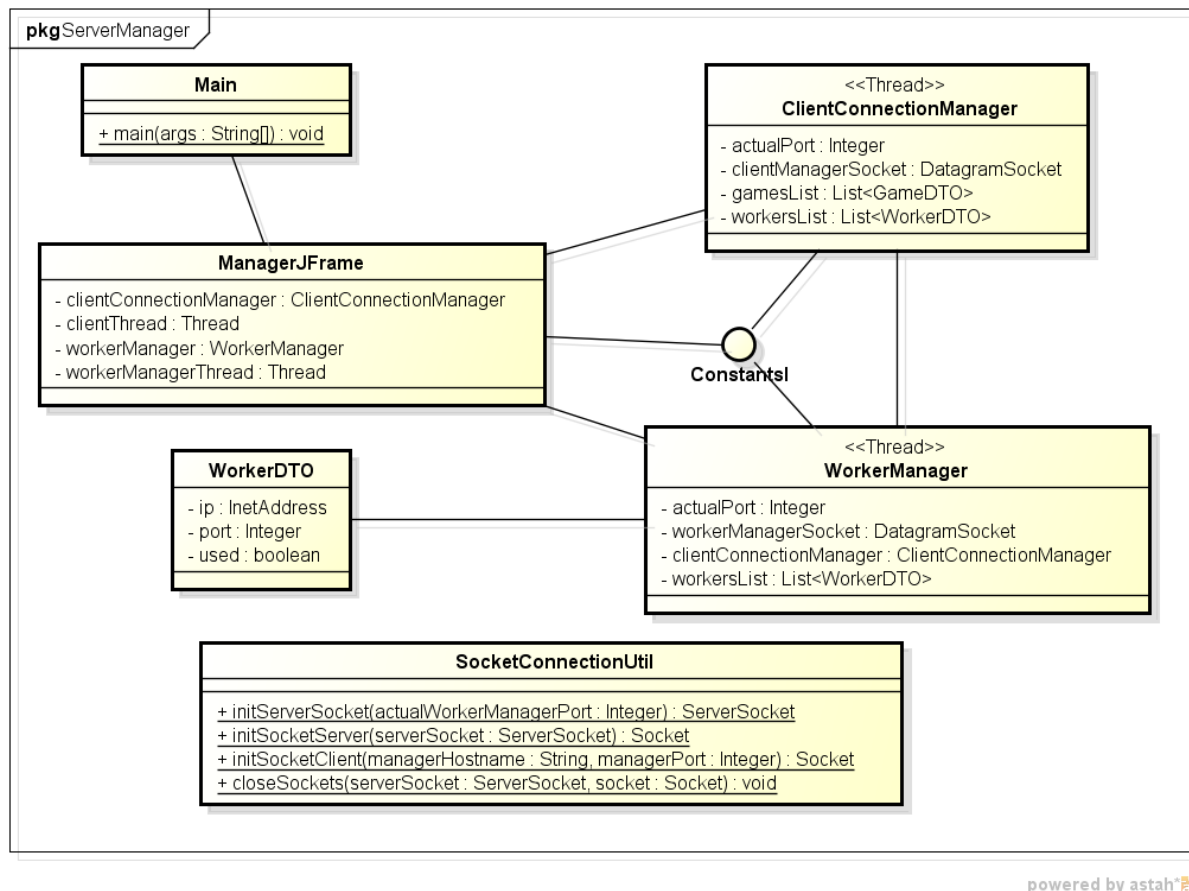


Figura 3.5: Diagrama de Classes do *Server Manager*.

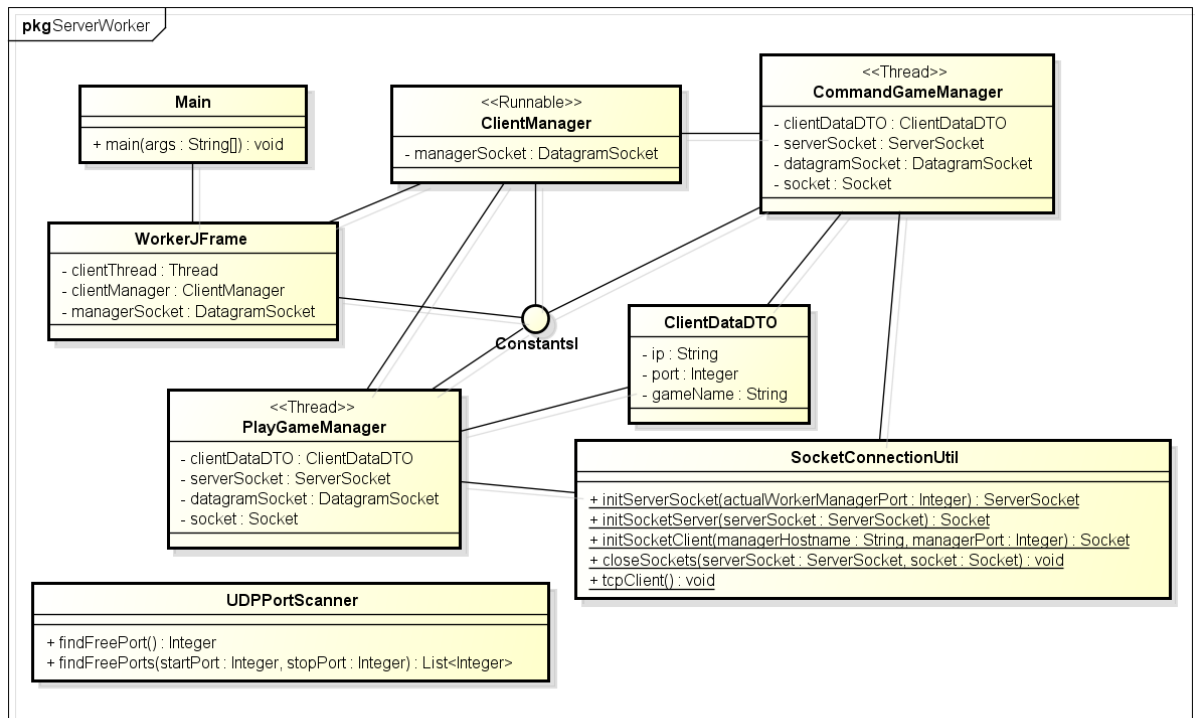
A classe Main é o ponto de início do sistema e a interface ConstantsI contém constantes de diferentes tipos que são necessárias em diversos pontos do sistema. A classe ManagerJFrame possui informações e funcionalidade de uma pequena tela representativa do sistema, já a classe SocketConnectionUtil inclui funcionalidades que ajudam a conexão e comunicação via socket. Pode-se verificar a presença desta última classe também nas Figuras 3.6 e 3.7. Elas possuem algumas diferenças na programação, mas os seus objetivos são iguais.

As classes ClientConnectionManager e WorkerManager estão representadas como *Threads*. Para conhecimento, *Threads* são partes de programas que podem ser executados simultaneamente com outras *Threads* de forma assíncrona (DEITEL; DEITEL, 2003).

A *thread* ClientConnectionManager é responsável pela primeira comunicação com a aplicação

cliente e pela sua gerência, até o cliente ser designado para algum *Worker*. Já a *thread WorkerManager* é responsável pelo primeiro contato com os *Workers* e sua gerência na nuvem.

O sistema *ServerWorker*, representado no diagrama de classes da Figura 3.6, realiza o trabalho de execução do jogo escolhido pelo jogador. Este sistema se comunica com o gerente da nuvem e diretamente com a aplicação cliente, quando delegado pelo *ServerManager*.



powered by astah

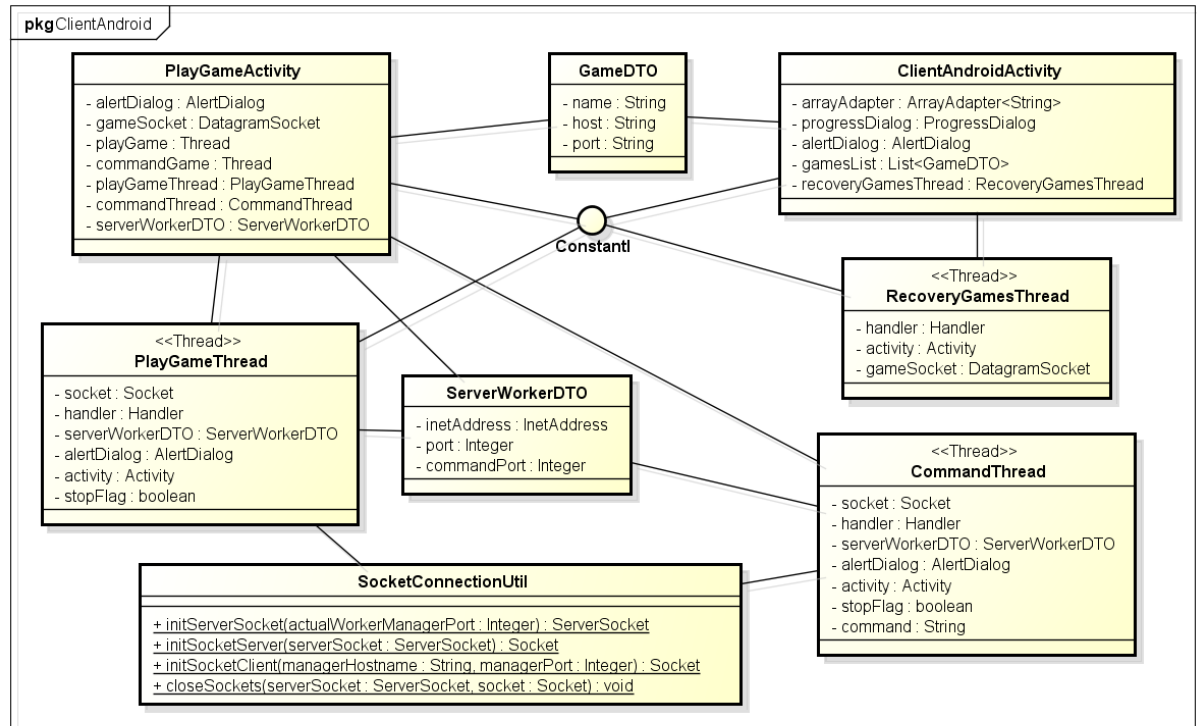
Figura 3.6: Diagrama de Classes do *Server Worker*.

Assim como na Figura 3.5, o diagrama na Figura 3.6 contém as classes *SocketConnectionUtil*, *Main* e *WorkerJFrame* com os mesmos objetivos. Observa-se também a presença de um DTO, o *ClientDataDTO*, que abrange informações do cliente que está conectado a este *worker*.

O sistema *ServerWorker* possui 3 *threads*. A primeira, *ClientManager*, gerencia o cliente com relação as funcionalidades iniciais que devem ser executadas para o começo do jogo. A segunda, a *PlayGameManager*, é responsável por manipular e enviar as imagens do jogo para o cliente e a última, a *CommandGameManager*, tem como incubência o recebimento dos comandos do jogador e a comunicação desses comandos para o jogo.

O diagrama de classes, contido na Figura 3.7, refere-se ao aplicativo cliente que é usado

pelo jogador em seu dispositivo Android. No diagrama pode-se perceber alguns conceitos que norteiam o desenvolvimento para Android, como por exemplo, as classes `PlayGameActivity` e `ClientAndroidActivity`. Elas por padrão herdam de `Activity`, que possui diversas responsabilidades, como controlar as telas, os eventos da tela, a navegação, dentre outros.



powered by astah

Figura 3.7: Diagrama de Classes do Cliente Android.

A aplicação cliente começa sempre por uma `Activity`. No caso, a `ClientAndroidActivity` gerencia o início da aplicação e a lista de jogos fornecidas pela nuvem. Já a `Activity PlayGameActivity` é responsável pela gerência da tela durante o jogo, controlando as imagens recebidas e capturando os eventos de tela como comandos do jogador (LECHETA, 2010).

No diagrama da Figura 3.7 é possível perceber 2 classes DTOs: a classe `ServerWorkerDTO` e a classe `GameDTO`. A primeira guarda informações do Worker que este cliente está acessando e o segundo DTO guarda informações referentes ao jogo escolhido pelo jogador. O diagrama também contém algumas classes cujos objetivos já foram explicados anteriormente, como a `SocketConnectionUtil` e a interface `ConstantsI`.

Dentro da aplicação cliente existem 3 *threads* diferentes. A `RecoveryGameThread` busca quais jogos estão disponíveis na nuvem e informa para a `Activity` que é responsável por

avisar ao jogador. A `PlayGameThread` e a `CommandGameThread` são *threads* que ficam em execução durante o jogo. A `PlayGameThread` recebe as imagens vindas da nuvem e informa a `Activity`, a medida que a `CommandGameThread` envia os comandos do jogador para a nuvem.

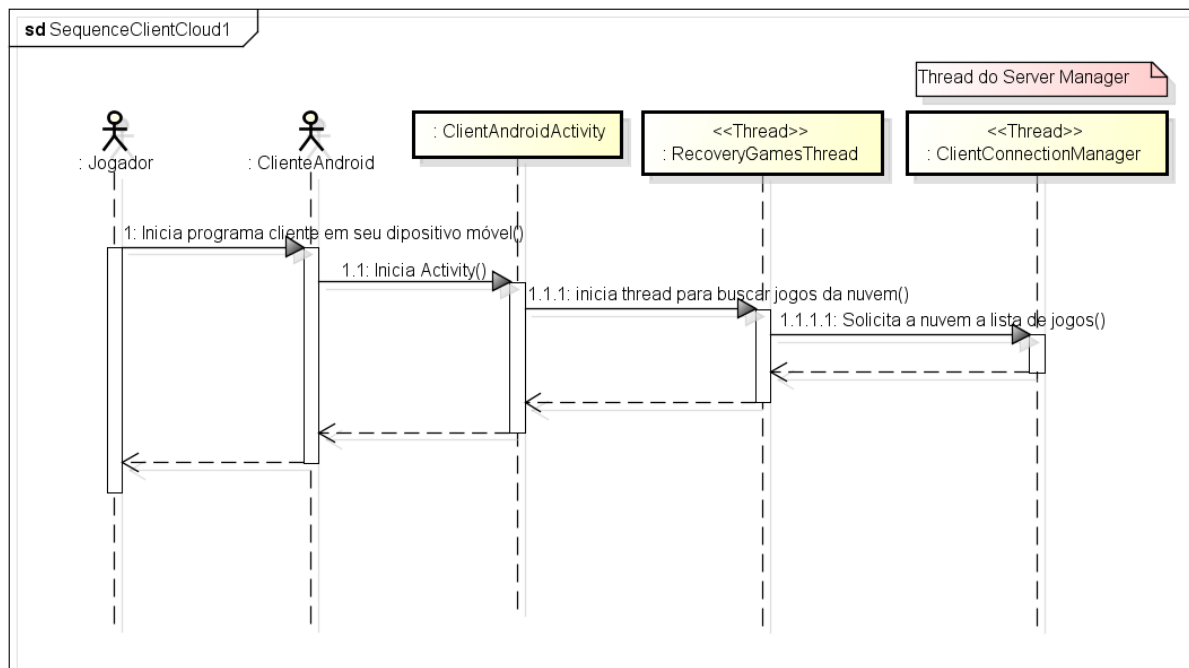
### 3.3 Implementação

A implementação da nuvem se desenvolveu em paralelo a conceitos de engenharia de *software*. Neste contexto, o modelo de processo de *software* usado foi o modelo incremental de desenvolvimento.

Existe uma intensa comunicação entre as aplicações, e ela é desenvolvida utilizando sockets. Para conhecimento, sockets são uma comunicação bidirecional entre dois processos, em que cada um deles possui um endereço IP - *Internet Protocol* e uma porta única (REZENDE, 2009). A escolha da comunicação utilizando sockets foi uma estratégia para não utilizar protocolos já existentes na internet, como o HTTP, e permitir a criação de um protocolo específico livre para o futuro.

No decorrer da seção existem diagramas de sequência que foram modificados para melhor apresentação neste trabalho. Então, as ações que deveriam ser métodos da implementação estão em formato de texto, para resumir a ação e facilitar o entendimento da sequência de execuções que as aplicações usam para seu funcionamento.

Na Figura 3.8 está um dos diagramas de sequência mencionados no parágrafo anterior. Nota-se na ação **”1.1.1: inicia thread para buscar jogos na nuvem”** que ela não é um método, mas uma descrição de uma ação que envolve vários métodos. Dessa forma, fica mais fácil compreender esta ação.



powered by astah

Figura 3.8: Diagrama que contém os primeiros passos quando um jogador abre a aplicação cliente.

No diagrama da Figura 3.8 estão contidos os primeiros passos, que a solução proposta realiza, quando o jogador inicia a aplicação cliente no dispositivo Android. Para o passo 1.1.1.1 do diagrama, o ServerManager da nuvem retorna a lista de jogos. Esta lista está disposta em um arquivo de configuração da nuvem denominado eduCloudGaming-config.xml. Este arquivo pode ser visto no quadro a seguir e deve conter todos os jogos que estão inseridos na nuvem.

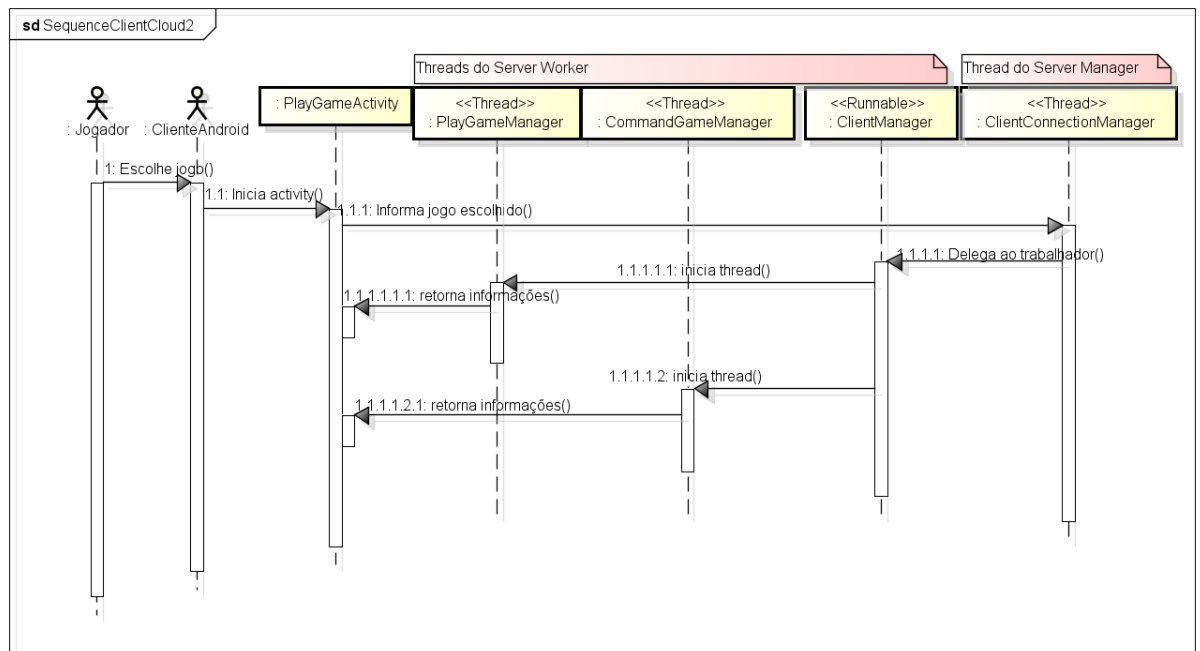
## Arquivo XML de Comunicação/Configuração

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <eduCloudGaming>
3   <games>
4     <game>
5       <name>MemoryGameTest</name>
6       <communicationClass>com.cloudgaming.communication.Communication</
       communicationClass>
7     </game>
8     <game>
9       <name>MemoryGame</name>
10      <communicationClass>com.cloudgaming.memory.communication.
        Communication</communicationClass>
11    </game>
12  </games>
13 </eduCloudGaming>
```

No quadro anterior, pode-se ver a *tag* `<game>` que contém informações de cada jogo inserido na nuvem. Esta *tag* possui duas outras *tags*, a `<name>` e a `<communicationClass>`. A *tag* `<name>` possui o nome do jogo que é exibido ao jogador na lista de jogos disponíveis na nuvem e a *tag* `<communicationClass>` informa à nuvem qual a classe que ela deve instanciar para executar o jogo selecionado.

No diagrama da Figura 3.8, após o retorno da solicitação do jogador contendo a lista de jogos disponíveis, os passos presentes na Figura 3.9 são executados quando o jogador escolhe o *game* que quer jogar.

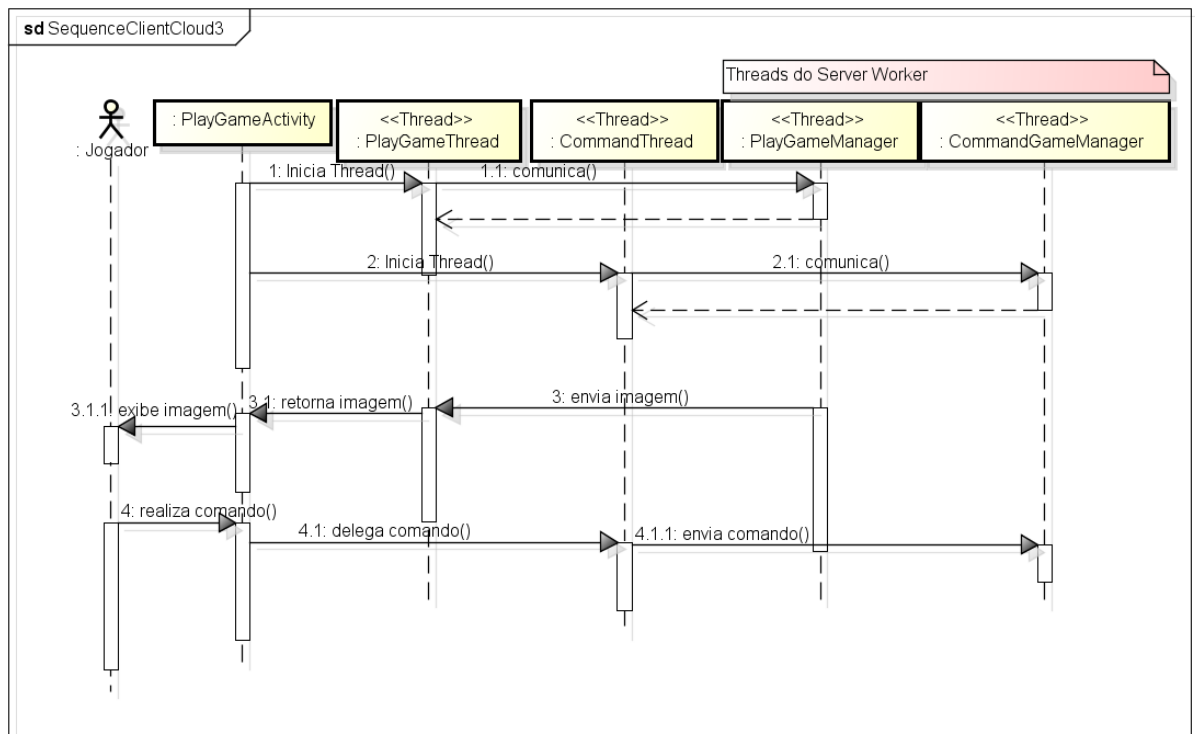
Na mesma figura, pode-se visualizar na ação 1.1.1.1 que, após o `ServerManager` ter delegado para o `ServerWorker` toda a responsabilidade com relação ao aplicativo cliente, é o `ServerWorker` que estabelece a comunicação direta e controla o envio de imagens e recebimento de comandos.



powered by astah

Figura 3.9: Diagrama que contém os passos seguintes depois que o jogador escolhe o jogo.

O diagrama contido na Figura 3.10 demonstra quais as ações e comunicações que ocorrem quando o jogador está jogando. Pode-se notar na ação 3 a responsabilidade do ServerWorker de enviar as imagens para o aplicativo cliente, e na ação 4.1.1 o ServerWorker recebendo os comandos do jogo.

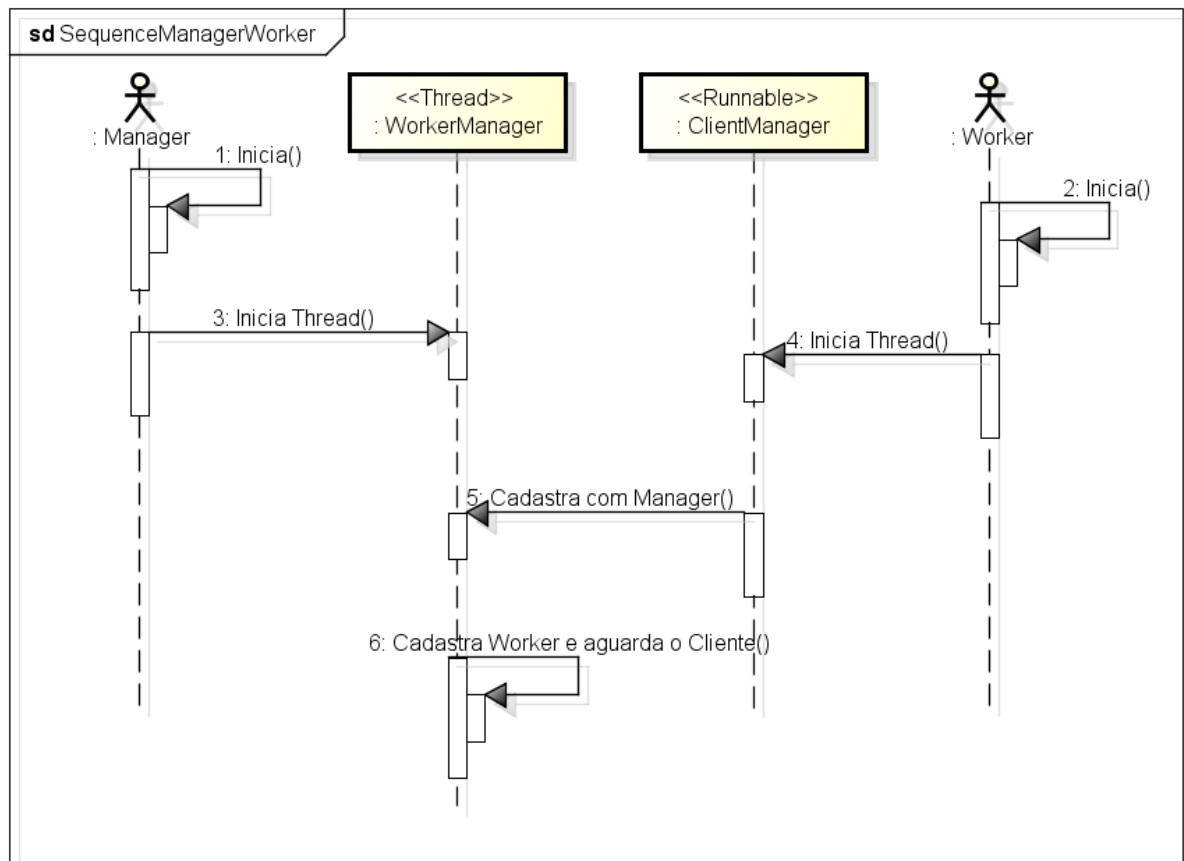


powered by astah

Figura 3.10: Diagrama que contém os passos de interação entre os sistemas durante o jogo.

É importante mencionar que, antes das interações entre nuvem e cliente, a nuvem precisa estar preparada para a comunicação com os clientes da mesma. Na Figura 3.11 pode-se analisar essa preparação.





powered by astah®

Figura 3.11: Diagrama que contém a preparação inicial da nuvem.

Na figura anterior, é possível verificar os sistemas *Manager* e *Worker* inicializando os seus trabalhos. Neste caso, é importante mencionar que o fluxo de inicialização do *Worker* pode ser executado várias vezes, iniciando vários *Workers*.

### 3.4 O Jogo

Para validar a arquitetura da nuvem proposta nas seções anteriores, foi desenvolvido um jogo que é executado na nuvem. Nesta seção está descrito como é esse jogo e quais são suas características. Aliado a isso, pode-se acrescentar que ele foi desenvolvido preenchendo os requisitos necessários para ser hospedado na nuvem. Para lembrete esses requisitos são:

- O jogo deve ser desenvolvido na linguagem Java;
- O jogo deve utilizar a biblioteca EduCloudGaming e possuir uma classe que herde

a interface `EduCloudGameCommunication`. Nela, devem constar os métodos de recebimento de comandos e envio de imagens;

- O jogo deve ser exportado para um arquivo `.jar` e adicionado ao sistema `ServerWorker`, acrescentando também as suas informações no arquivo `eduCloudGaming-config.xml`.

O *game* é um jogo de memória, em que uma série de cartas aparecem viradas para baixo e o jogador deve encontrar os pares iguais. Para isso, ele deve clicar nas cartas e "virá-las" a fim de visualizar a figura e encontrar outra carta com a figura igual. Imagens do funcionamento deste jogo, podem ser visualizadas na Figura 3.12.

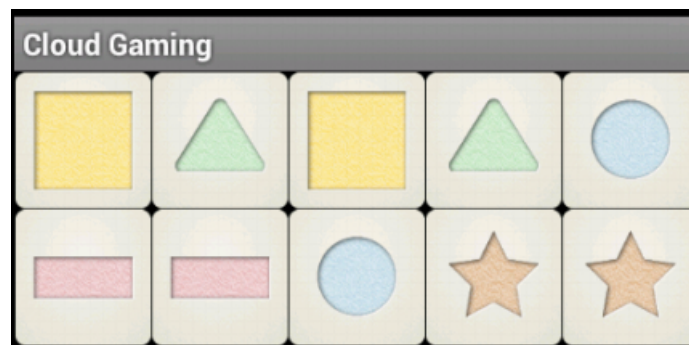
Na imagem 3.12(a) é possível visualizar o início do jogo, quando todas as cartas estão viradas para baixo. Na imagem 3.12(b), pode-se verificar dois pares de cartas que o jogador acertou: um par de círculos e um par de retângulos. Mas ainda há cartas que precisam ser viradas. Já na imagem 3.12(c) o jogador terminou o jogo, acertando todos os pares.



(a) Início do jogo.



(b) Durante o jogo.



(c) Fim do jogo.

Figura 3.12: Figuras do jogo de memória.

O Diagrama de Classes do jogo está contido na Figura 3.13. Nele pode-se verificar a classe `Communication` que herda de `EduCloudGameCommunication` e implementa os métodos da interface. Também é possível visualizar a presença de um DTO, chamado `PieceDTO`, que é uma representação de cada carta do jogo.

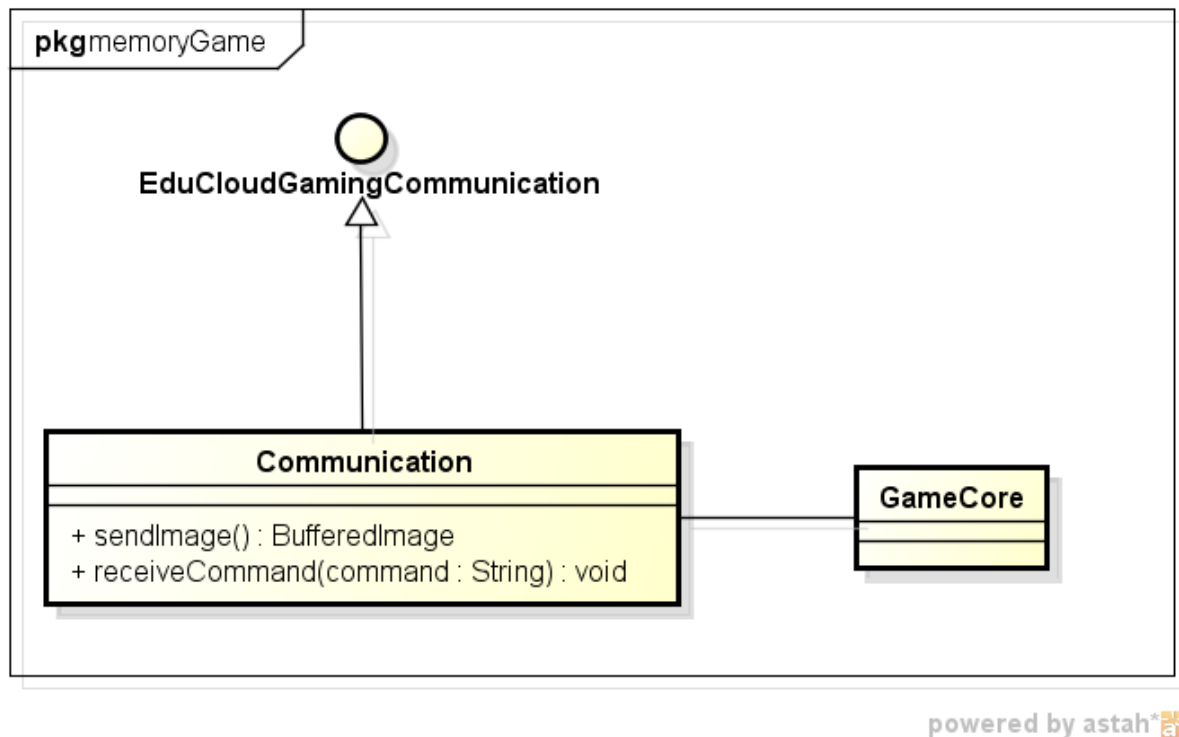
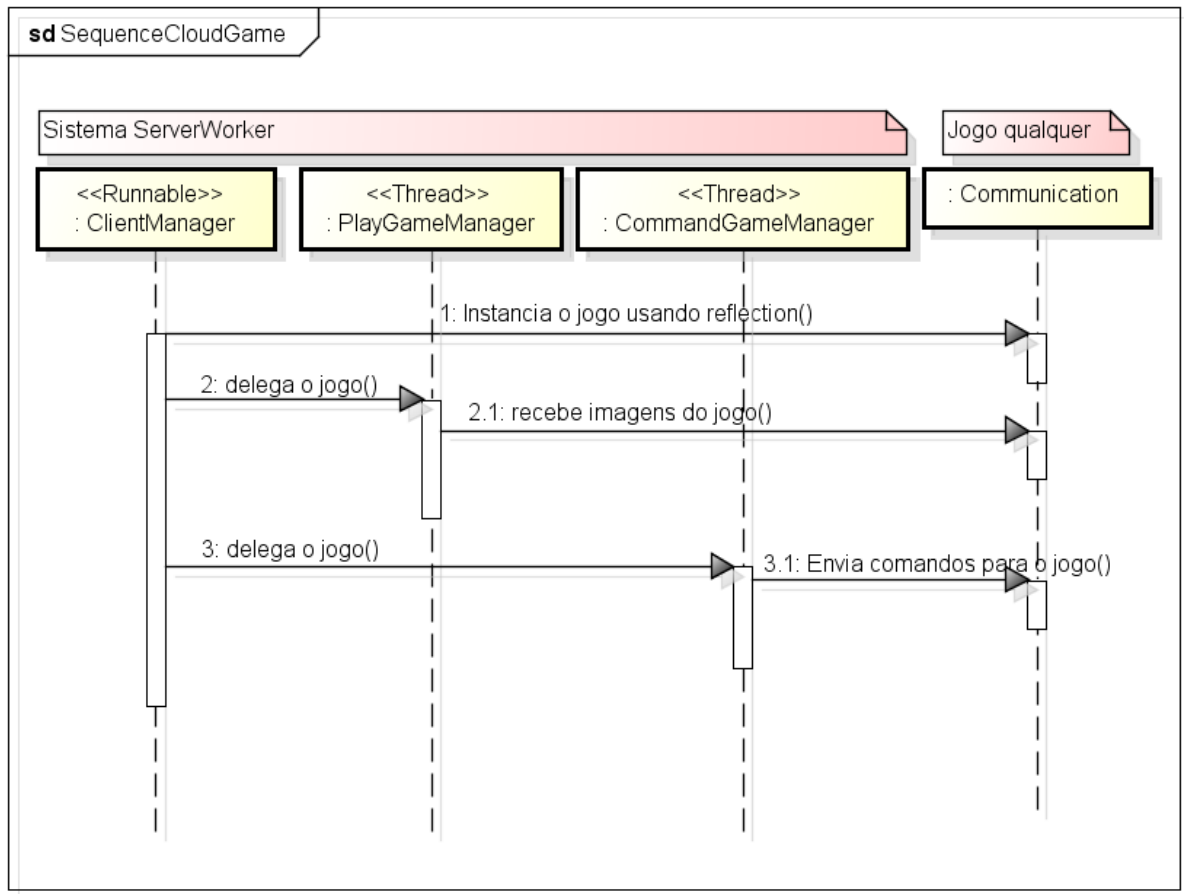


Figura 3.13: Diagrama de Classes do jogo de memória desenvolvido.

A implementação e o motor do jogo estão contidos na classe `GameCore`. Toda a lógica do jogo, as interpretações dos comandos do jogador e a criação da imagem para envio estão na mesma classe.

### 3.5 Comunicação entre o Jogo e a Nuvem

Esta seção abrange a comunicação entre o jogo e a nuvem. Os passos dessa comunicação estão presentes na Figura 3.14 que representa a comunicação da nuvem com um jogo qualquer. Observa-se que a comunicação se resume ao sistema trabalhador (`ServerWorker`) e ao jogo, considerando que o sistema gerenciador não se comunica com o jogo diretamente, sendo esta uma responsabilidade dos trabalhadores.



powered by astah

Figura 3.14: Diagrama que contém a comunicação da nuvem com o Jogo.

Após o jogador escolher o jogo e algumas ações executarem, a *thread* ClientManager recebe a incumbência de iniciar o *game* escolhido. Com a ajuda do arquivo XML, exposto na seção 3.3 e a API - *Application Programming Interface* (Interface para Programação de Aplicações) de reflection do Java, o jogo é iniciado dinamicamente pela nuvem, utilizando a interface EduCloudGameCommunication presente na Figura 3.4. Esta interface deve ser herdada por uma classe em qualquer jogo que tem como objetivo ser hospedado pela nuvem, de acordo com a Figura 3.13 do jogo de memória.

No ServerWorker existem 2 *threads* responsáveis pela comunicação com o jogo: a PlayGameManager e a CommandGameManager. Depois da *thread* ClientManager iniciar o jogo, a mesma delega a responsabilidade de controle do jogo para as *threads* mencionadas.

A CommandGameManager transmite para o jogo os comandos do jogador. Uma interação genérica pode trazer algumas inconsistências, então algumas situações foram avaliadas. Neste sentido, como o sistema operacional escolhido foi o Android, e o mesmo pode

ser utilizado em *smartphones* e *tablets*, o tamanho da tela varia sem nenhuma padronização. Por isso, foi adotada uma padronização na comunicação, em que as mensagens possuem as coordenadas que o jogador pressionou e o tamanho da tela. Dessa maneira, a mensagem enviada pelo jogo possui o formato a seguir: ”<largura da tela>|<altura da tela>|<posição X do clique>|<posição Y do clique>”. Com esse padrão, qualquer jogo que tenha cliques como interação pode ser desenvolvido.

A *thread* `PlayGameManager` se comunica com o cliente Android enviando as imagens geradas pelo jogo. A imagem é criada no jogo e enviada para a nuvem no formato PNG - Portable Network Graphics, nesse momento sem nenhuma compressão.

## Testes e Resultados

---

Esta seção aborda os testes que foram realizados para validar o modelo de arquitetura da nuvem proposta. Os equipamentos usados também são descritos aqui, assim como uma abordagem comprovando que a solução possui as características de uma nuvem.

Os testes foram executados no *cluster* de desenvolvimento do SENAI-CIMATEC, chamado pelo codinome Gabi, constituído de 8 máquinas (ou *blades*) HP ProLiant DL120 G6, máquinas que possuem as seguintes configurações: processador Intel (R) Xeon X3440 2,53 GHz - *Gigahertz* com 4 núcleos, 8 GB - *Gigabyte* de memória RAM - Random Access Memory (Memória de Acesso Aleatório), sistema operacional Linux 3.2.0-23-generic X86\_64 GNU/Linux Ubuntu 12.04 LTS, interconectadas através do switch GTS Network, modelo 73.1724S de 24 portas, com 10/100 Mbps - *Megabit* por segundo.

O dispositivo móvel, que foi usado nos testes, foi um *tablet* Samsung Galaxy Tab 7.0 Plus, com processador dual core de 1.2 GHz, 1 GB de memória RAM, uma tela de 7 polegadas e o sistema operacional Android 3.2 (SAMSUNG, 2013).

Os testes não foram executados pela internet, mas na rede local presente no SENAI-CIMATEC, utilizando uma rede Wi-Fi - Wireless-Fidelity, provida pelo *access point* TP-LINK com 150 Mbps do modelo TL-WA701ND, disponibilizando uma conexão Wi-Fi para o dispositivo móvel. Todos os sistemas funcionaram corretamente e seus desempenhos foram avaliados nos testes a seguir.

A primeira validação realizada contemplou a utilização dos recursos computacionais da nuvem. Neste contexto, o principal sistema a ser analisado aqui, ou seja, de maior consumo computacional, é o sistema trabalhador (ServerWorker), pois é nele que os jogos são executados. Coletando os dados dos *blades* do *cluster*, o sistema trabalhador utiliza 10% de processamento e 1% de memória, valores verificados na Figura 4.1. Como cada máquina possui uma memória compartilhada e 4 núcleos, na teoria, seria possível iniciar 40 trabalhadores executando jogos por blade, o que contemplaria 320 trabalhadores no cluster do SENAI-CIMATEC. Vale salientar que essa estimativa está relacionada apenas aos recursos computacionais, abrangendo somente o *hardware*. Pode-se acrescentar que não foi possível testar o uso desse grande número de trabalhadores ativos, pois seria necessário o mesmo número de *smartphones/tablets/simuladores*, dispositivos estes que não estão disponíveis nesse projeto.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7397	administ	20	0	4083m	71m	14m	S	10	0.9	0:54.24	java

Figura 4.1: Utilização dos recursos computacionais.

A segunda validação está relacionada ao tamanho dos pacotes que transitam na rede durante a execução do jogo. Na Figura 4.2 é possível verificar os pacotes que entram e saem de uma máquina do *cluster*, com um trabalhador ativo, em Kbps - Quilobit por segundo.

eth0	
Kbps in	Kbps out
47.68	615.74
44.11	600.82
73.34	881.41
69.54	911.74
41.91	445.45
53.65	453.06
44.67	891.74
72.55	882.55
54.15	870.55
51.44	478.90
42.07	447.00
58.76	880.56
53.99	783.24
40.49	594.81
52.14	606.92
38.63	622.94
53.74	1178.80
33.70	22.36
70.75	1217.36
49.13	81.42
84.86	575.67

Figura 4.2: Tamanho dos pacotes entrando e saindo.

Outra validação constatada está relacionada ao tempo de resposta de acordo com alguma ação do usuário. Essa validação é difícil de ser executada dentro do programa, devido ao fato de as imagens não possuírem nenhuma identificação de resposta do usuário, dificultado a percepção que aquela imagem é uma resposta de algum comando do jogador. Contudo, essa validação ainda pode ser efetuada, através da interação humana. Para isso, foram coletados 10 tempos de respostas diferentes, realizando uma ação no jogo e cronometrando a sua resposta, fazendo o uso de um cronômetro comum. Sendo assim, os



resultados não são precisos, pois foram obtidos fora do ambiente computacional. Ainda assim, os mesmos estão presentes na Tabela 4.1 em que se pode verificar o melhor tempo de 1,72 segundos. Entretanto, realizando o cálculo da média dos tempos de resposta, o resultado é de 3,284, um tempo que deve ser aprimorado em trabalhos futuros.

Tabela 4.1: Tempos de resposta de uma ação do jogador.

Número	Tempo (segundos)
1	3,04
2	2,69
3	1,72
4	3,58
5	4,33
6	2,78
7	3,96
8	2,72
9	3,42
10	4,60

A validação final está relacionada à coleta da quantidade de FPS - Frames por segundo. Para realizar esta coleta, o jogo de memória foi executado 5 vezes de forma completa, e os dados foram coletados diretamente na implementação, tornando-se precisos comparados aos dados da tabela anterior.

A Tabela 4.2 contém os dados coletados nesse teste, em que se pode verificar os FPS calculados. Pode-se afirmar que a taxa de FPS está muito baixa: uma média de 1,095323131 segundos. Portanto, melhorias devem ser implementadas em trabalhos futuros para aperfeiçoá-la.

Tabela 4.2: FPS do jogo de memória executando na nuvem.

Número	Contador (Frames)	Tempo (Milisegundos)	FPS
1	109	97866	1,113767805
2	145	142510	1,017472458
3	137	111853	1,224821864
4	84	75980	1,105554093
5	63	62069	1,014999436

As características essenciais para um ambiente computacional ser considerado uma nuvem, presente na seção 2.2, estão presentes na EduCloudGaming como demonstra a Tabela 4.3:

Tabela 4.3: Elementos essenciais de uma nuvem na EduCloudGaming

<b>Elemento</b>	<b>EduCloudGaming</b>
Autoatendimento sob Demanda	O ServerManager recebe as requisições do cliente de forma automática e a comunicação entre as máquinas da nuvem, também não precisa de interação humana.
Largo acesso aos serviços de rede	No momento, o acesso a solução é local, pois o <i>cluster</i> de desenvolvimento do SENAI-CIMATEC não está disponível na internet. Entretanto, a nuvem já está preparada para o funcionamento na internet. Em relação ao acesso de vários clientes, a nuvem já está preparada, mas somente o cliente Android foi desenvolvido até o momento.
Pool de Recursos	As máquinas (físicas e virtuais) se comunicam de forma transparente para o cliente/jogador. Esta comunicação é feita através da aplicação cliente com a nuvem em diferentes máquinas, e o jogador não está ciente sobre disso.
Elasticidade Rápida	As máquinas trabalhadoras são facilmente adicionadas a nuvem e o sistema gerenciador está preparado para, a qualquer momento, receber um novo trabalhador e disponibilizá-lo para os clientes.
Serviços Mensuráveis	No atual momento, somente um cliente é alocado por trabalhador. É possível mudar essa situação, mas seria necessário implementar uma melhor gerência de memória e processamento da nuvem.

---

## Considerações finais

---

### 5.1 Conclusões

O modelo de arquitetura proposto neste trabalho, descrito no Capítulo 3, e avaliado no Capítulo 4, mostrou ser possível criar um ambiente baseado nos conceitos de *Cloud Computing* com a finalidade de disponibilizar jogos voltados para a educação. Entretanto, também avaliado no Capítulo 4, ainda existem algumas melhorias que podem ser implementadas na solução, principalmente nos tempos de respostas das ações e na taxa de FPS.

Para atender os objetivos desta pesquisa, foram desenvolvidas 4 implementações: a biblioteca EduCloudGaming; o sistema ServerManager; o sistema ServerWorker e o aplicativo para o cliente Android. Todas estas implementações formam a EduCloudGaming, que corresponde ao primeiro passo para a nuvem de jogos com fins educacionais. Atualmente, a solução pode englobar jogos gerais, mas a intenção é que ela evolua em aspectos educacionais e pedagógicos, admitindo assim um caráter de maior especificidade.

O modelo de arquitetura presente nesta dissertação está de acordo com trabalhos correlatos realizados na área, em que alguns deles estão no Capítulo 2. Aliado a isso, as características essenciais de uma nuvem estão presentes na solução, como demonstra o Capítulo 4.

A validação deste trabalho exalta, ainda mais, a limitação das redes móveis do Brasil descrita anteriormente. Na rede local, presente no SENAI-CIMATEC, em que sua latência é menor, a nuvem já apresentou pontos de melhoria. Indubitavelmente, na internet presente no Brasil, onde a internet móvel de qualidade ainda é para poucos, a utilização de soluções, como a presente neste trabalho, é quase nula.

### 5.2 Contribuições

Os objetivos do trabalho foram atingidos, ou seja, a construção de um modelo de arquitetura em que o processamento dos jogos educativos é deslocado do dispositivo móvel e direcionado para a nuvem foi desenvolvido. Portanto, esta é a principal contribuição deste trabalho, que permite a utilização de jogos digitais em ambientes desprovidos da estrutura computacional necessária.

No desenvolvimento deste trabalho foram realizadas diversas pesquisas sobre uma nuvem específica sobre jogos e educação. Foram encontrados diversos trabalhos mencionando Computação em Nuvem como um suporte tecnológico para a educação, utilizando recursos computacionais, memória, armazenamento, etc. Não foram encontrados trabalhos como o proposto aqui. Sendo assim, este trabalho é o primeiro a mencionar uma nuvem voltada para jogos específicos com fins educacionais.

Este trabalho também contribui para a divulgação e disseminação de jogos com fins educacionais, sendo mais um disponibilizador de *games* com este fim. Assim, facilita o conhecimento da existência desses jogos pelos alunos/jogadores.

A solução oferece um ambiente completo para desenvolvedores de jogos com fins educacionais disponibilizarem seus *games* de forma gratuita, como também o aplicativo Android para a comunicação com a nuvem. Ademais, um jogo de memória está disponível com o propósito de exemplificar como um jogo deve ser desenvolvido para que seja inserido na nuvem.

Como os códigos-fonte dessa dissertação estão disponíveis, este trabalho também auxilia a construção de outras nuvens específicas, como por exemplo, nuvens para jogos voltados à saúde.

### **5.3 Atividades Futuras de Pesquisa**

Existem várias propostas de atividades futuras para esta pesquisa. Dentre elas, a mais urgente está relacionada à melhoria dos tempos de respostas das ações e da taxa de FPS. Algumas medidas podem ser adotadas, como a compressão das imagens, diminuindo o tráfego da rede ou, ainda, outra estratégia no envio das imagens, como enviar somente partes da imagem que são diferentes da anterior.

Outra atividade futura, está na linha de segurança e tolerância a falhas. Hoje não existe nenhum processo de autenticação na nuvem, conseqüentemente qualquer pessoa pode jogar. Aliado a isso, caso o ServerManager pare de funcionar, a nuvem tem o seu funcionamento interrompido. Em síntese, um mecanismo de tolerância a falhas que ajuste esse problema seria de grande valia.

Uma linha de trabalho futuro importante que deve ser pesquisada, é a área de interface humano-computador. Hoje a nuvem somente trabalha com comandos de cliques na tela. Assim, um estudo visando a busca da interação entre o jogador e a nuvem que atenda ao usuário, seria muito importante, criando desse modo, um único "controle" para ser usado durante o jogo. Além disso, um estudo para viabilizar a utilização de sons no jogo e de

como sincronizá-los com as imagens também é importante.

Outra linha de trabalho importante, com um alto teor educacional, seria a capacidade da nuvem em executar jogos multiusuários. Atualmente, a nuvem não suporta jogos com mais de um jogador. Então, a criação de um mecanismo para permitir que a nuvem proporcione ao jogador uma maior interação com outros jogadores/alunos, isto traria benefícios relacionados a interação e motivação entre o jogador e o jogo.

Uma evolução no modelo é necessária para funcionar com outras linguagens de desenvolvimento, além da Java. Outros mecanismos de comunicação entre linguagens devem ser pesquisados para viabilizar a utilização, na EduCloudGaming, de outras linguagens de desenvolvimentos de jogos, como C, C++, lua, Javascript, etc. Outra evolução importante seria a criação de outros clientes para a nuvem, como clientes para SmartTVs, Windows (para desktop e dispositivos móveis), MAC-OS, IOS e Linux, ampliando o número de alunos/jogadores que esta solução pode atingir.

No modelo de arquitetura atual, a manipulação dos *streams* de envio das imagens foi construída utilizando código nativo Java. Então um estudo sobre bibliotecas que já manipulam *streams* e qual seria a melhor para ser utilizada nesse modelo enriqueceriam a solução.

Outra linha de trabalho futuro que deve ser mencionada é o paralelismo da execução dos jogos. Hoje o modelo de arquitetura proposto realiza o processamento do jogo somente em uma máquina. Um estudo para analisar qual a melhor tecnologia de processamento paralelo de jogos e a criação dessa paralelização seriam muito enriquecedoras para o projeto.

Por fim, um estudo importante a ser empregado nessa pesquisa é a criação de um protocolo único, para ser usado até em outras nuvens. Portanto, é importante a elaboração de um protocolo que abranja todas as comunicações da nuvem, como comunicações entre os trabalhadores e o gerente, ou comunicações entre a nuvem e o cliente.

---

## Referências Bibliográficas

---

- 1UP. *GDC 2009: OnLive Unveils "Cloud Gaming" Service*. 2009. URL: <http://www.1up.com/news/gdc-2009-onlive-unveils-cloud>. Acesso em: 16 jun. 2013. 2.6
- ANDROIDDEVELOPERS. *Android Developers*. 2013. URL: <http://developer.android.com/index.html>. Acesso em: 11 nov. 2013. 3.1
- ARMBRUST, M. et al. *Above the Clouds: A Berkeley View of Cloud Computing*. Berkeley, Estados Unidos, 2009. Technical Report No. UCB/EECS-2009-28. 2.2
- ARMBRUST, M. et al. A view of cloud computing. In: *Communications of the ACM*. [S.l.: s.n.], 2010. v. 53, n. 4, p. 50–58. 2.2, 2.5
- ASTAH. *ASTAH*. 2013. URL: <http://astah.net>. Acesso em: 31 out. 2013. 3.1
- ASTAHCOMMUNITY. *ASTAH Community*. 2013. URL: <http://astah.net>. Acesso em: 11 nov. 2013. 3.1
- BARBOZA, D. C.; LIMA, H. J.; CLUA, E. W. G.; REBELLO, V. E. F. A simple architecture for digital games on demand using low performance resources under a cloud computing paradigm. In: *Proceedings of the 2010 Brazilian Symposium on Games and Digital Entertainment*. Florianópolis, Santa Catarina, Brasil: SBGAMES, 2010. p. 33–39. (document), 1, 2.2, 2.7, 2.6
- BSNL. *BSNL Games On Demand*. 2013. URL: <http://bsnl1.indiagames.com/>. Acesso em: 24 fev. 2013. 1
- BURNS, C. *NVIDIA GRID revealed to change cloud gaming forever*. 2013. URL: <http://www.slashgear.com/nvidia-grid-revealed-to-change-cloud-gaming-forever-06263511/>. Acesso em: 25 nov. 2013. 2.6
- CERF, V. G.; KAHN, R. E. A protocol for packet network intercommunication. *IEEE Transactions on Communications Technology*, COM-22, n. 5, p. 627–641, 1974. 2.1
- CHEN, K.; CHANG, Y.; TSENG, P.; HUANG, C.; LEI, C. Measuring the latency of cloud gaming systems. In: *Proceedings of the 19th ACM international conference on Multimedia*. New York, USA: ACM, 2011. p. 1269–1272. 2.6
- CLOUDGAMINGREPORT. *Distribution and monetization strategies to increase revenues from Cloud Gaming*. 2012. URL: <http://www.cgconfusa.com/report/documents/Content-5minCloudGamingReportHighlights.pdf>. Acesso em: 12 out. 2013. 2.6
- COSTA, R. M. E. M.; CARVALHO, L. A. V. O uso de jogos digitais na reabilitação cognitiva. *XVI Simpósio Brasileiro de Informática na Educação - Workshop de Jogos Digitais na Educação*, p. 19–21, 2005. 1
- CRYTEC. *Crytec*. 2013. URL: <http://www.crytek.com/>. Acesso em: 16 jun. 2013. 2.6
- DEITEL, H. M.; DEITEL, P. J. *Java, como programar*. Porto Alegre: Bookman, 2003. 3.2.3

- DOBRA, A. *Crytek Attempted Cloud Gaming Way Before OnLive*. 2013. URL: <http://news.softpedia.com/news/Crytek-Attempted-Cloud-Gaming-Way-Before-OnLive-110232.shtml>. Acesso em: 16 jun. 2013. 2.6
- ECLIPSE. *Eclipse*. 2013. URL: <http://eclipse.org>. Acesso em: 31 out. 2013. 3.1
- ESPM; SIOUX; BLEND. *Pesquisa Game Mobile Brasil*. 2013. 1
- FOSTER, I.; ZHAO, Y.; RAICU, I.; LU, S. Cloud computing and grid computing 360-degree compared. *IEEE Grid Computing Environments Workshop*, p. 1–10, 2008. (document), 2.5, 2.3
- FOWLER, M. *Padrões de arquitetura de aplicações corporativas*. Porto Alegre: Bookman, 1994. 3.2.3
- G1. *Só 12% das escolas têm computador instalado na sala de aula, diz estudo*. 2013. URL: <http://g1.globo.com/educacao/noticia/2013/05/so-12-das-escolas-tem-computador-instalado-na-sala-de-aula-diz-estudo.html>. Acesso em: 24 set. 2013. 1
- GAIKAI. *Gaikai*. 2013. URL: <http://www.gaikai.com/>. Acesso em: 25 nov. 2013. 2.6
- GAMINGANYWHERE. *GamingAnywhere*. 2013. URL: <http://gaminganywhere.org/>. Acesso em: 17 jun. 2013. (document), 2.6, 2.6
- GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Baam: Addison Wesley, 1994. 3.2.3
- GCLUSTERA. *G-cluster Global*. 2013. URL: <http://www.g-cluster.com/>. Acesso em: 16 jun. 2013. 2.6
- GCLUSTERB. *About G-cluster Global*. 2013. URL: <http://www.g-cluster.com/eng/#about/>. Acesso em: 16 jun. 2013. 2.6
- GOOGLE. *Google Cloud Plataform*. 2013. URL: <https://cloud.google.com/>. Acesso em: 25 set. 2013. 2.5
- HUANG, C.; HSU, C.; CHANG, Y.; CHEN, K. Gaminganywhere: An open cloud gaming system. In: *Proceedings of ACM Multimedia Systems*. Oslo, Norway: [s.n.], 2013. p. 36–47. 2.6, 2.6
- IBGE. *Instituto Brasileiro de Geografia e Estatística*. 2013. URL: [http://www.ibge.gov.br/home/presidencia/noticias/noticia\\_visualiza.php?id\\_noticia=1766](http://www.ibge.gov.br/home/presidencia/noticias/noticia_visualiza.php?id_noticia=1766). Acesso em: 24 fev. 2013. 1
- ITU. *International Telecommunication Union*. 2012. URL: <http://www.itu.int/ITU-D/ict/newslog/CategoryView,category,Mobile%2Bsubscribers.aspx>. Acesso em: 10 jan. 2012. 1
- JARSCHER, M.; SCHLOSSER, D.; SCHEURING, S.; HOBFELD, T. An evaluation of qoe in cloud gaming based on subjective tests. In: *Proceedings of IEEE/ACM Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. Seoul, South Korea: [s.n.], 2011. p. 330–335. 2.6
- JAVA. *Java*. 2013. URL: [http://www.java.com/pté\\_BR/](http://www.java.com/pté_BR/). Acesso em: 13 out. 2013. 3.1

KUROSE, J. F.; ROSS, K. W. *Rede de Computadores e a Internet: uma nova abordagem*. São Paulo: Addison Wesley, 2003. 2.1

LECHETA, R. R. *Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK*. São Paulo: Novatec, 2010. 3.2.3

LECHETA, R. R. *Google Android Para Tablets: aprenda a desenvolver aplicações para o Android: de smartphones a tablets*. São Paulo: Novatec, 2011. 3.1

LEE, Y.; CHEN, K.; SU, H.; LEI, C. Are all games equally cloud-gaming-friendly? an electromyographic approach. In: *Proceedings of IEEE/ACM Network and Systems Support for Games (NetGames)*. Venice, Italy: [s.n.], 2012. p. 1–6. 2.6

MACHADO, D. *Indústria de games bate Hollywood e deve arrecadar US\$ 74 bi até 2017*. 2013. URL: <http://www.correioeuberlandia.com.br/entretenimento/industria-de-games-bate-hollywood-e-deve-arrecadar-us-74-bi-ate-2017/>. Acesso em: 24 set. 2013. 1

MARILL, T.; ROBERTS, L. G. Toward a cooperative network of time-shared computers. In: *AFIPS - Fall Joint Computer Conference (ACM)*. Nova York, Estados Unidos: [s.n.], 1966. p. 425–431. 2.1

MARTINS, L. F. F. *IPBrick Cloud Environment*. Dissertação (Mestrado Integrado em Engenharia Eletrotécnica e de Computadores) — Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2010. 2.2, 2.3

MATULEF, J. *Bushnell: Cloud computing will improve education*. 2011. URL: <http://www.gamesindustry.biz/articles/2011-09-08-bushnell-cloud-computing-will-improve-education>. Acesso em: 12 nov. 2013. 2.6

MEC. *Ministro entrega tablets para iniciar formação de professor do ensino médio*. 2012. URL: <http://portal.mec.gov.br>. Acesso em: 08 nov. 2013. 1

MELL, P.; GRANCE, T. The nist definition of cloud computing. In: *National Institute of Standards and Technology Special Publication*. Gaithersburg, Estados Unidos: [s.n.], 2011. 2.2, 2.3, 2.4

MOHAMED, A. *A History of Cloud Computing*. 2009. URL: <http://www.computerweekly.com/feature/A-history-of-cloud-computing>. Acesso em: 02 jun. 2013. 2.1

MURNO, G.; DRSKA, M. *Indústria de jogos deslança no Brasil*. 2013. URL: <http://economia.ig.com.br/empresas/2013-07-08/industria-de-jogos-deslanca-no-brasil.html>. Acesso em: 25 nov. 2013. 1

NVIDIAGRID. *Aceleração por GPU Fornecida pela Rede para Jogos, Aplicativos de Design e Desktops Virtuais*. 2013. URL: <http://www.nvidia.com.br/object/nvidia-grid-br.html>. Acesso em: 25 nov. 2013. 2.6

NVIDIASHIELD. *NVIDIA GRID CLOUD GAMING BETA*. 2013. URL: <http://shield.nvidia.com/grid/>. Acesso em: 06 dez. 2013. 2.6

NVIDIASHIELD. *THE BEST ANDROID GAMING DEVICE*. 2013. URL: <http://shield.nvidia.com/play-android-games/>. Acesso em: 06 dez. 2013. (document), 2.5



ONLIVE. *OnLive*. 2013. URL: <http://www.onlive.com/>. Acesso em: 24 fev. 2013. (document), 1, 2.6, 2.4

REZENDE, D. C. F. *Proposta de especificação formal em SDL de uma rede de comunicação automotiva baseada no protocolo FlexRay com geração automática de código java*. Dissertação (Programa de Pós-Graduação em Engenharia Elétrica) — Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação, Campinas, 2009. 3.3

RUIC, G. *3 em cada 10 brasileiros são donos de smartphones*. 2013. URL: <http://exame.abril.com.br/tecnologia/noticias/3-em-cada-10-brasileiros-sao-donos-de-smartphones>. Acesso em: 24 set. 2013. 1

SAMSUNG. *Galaxy Tab 7.0 Plus*. 2013. URL: <http://www.samsung.com/br/consumer/cellular-phone/cellular-phone-tablets/tablet/GT-P6200MALZTO>. Acesso em: 20 nov. 2013. 4

SANTOS, R. *Introdução à programação orientada a objetos usando Java*. Rio de Janeiro: Campus, 2003. 3.1

SETI@HOME. *SETI@home*. 2013. URL: <http://setiathome.berkeley.edu/>. Acesso em: 25 set. 2013. 2.5

SOBRAGI, C. G. *Adoção de Computação em Nuvem: Estudos de Casos Múltiplos*. Dissertação (Mestrado Acadêmico em Administração) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012. 2.3, 2.4

SULLIVAN, G. T. Aspect-oriented programming using reflection and metaobject protocols. *Magazine Communications of the ACM*, v. 44, n. 10, p. 95–97, 2001. 3.2.3

SUSELBECK, R.; SCHIELE, G.; BECKER, C. Peer-to-peer support for low-latency massively multiplayer online games in the cloud. In: *Proceedings of IEEE/ACM Network and Systems Support for Games (NetGames)*. Paris, France: [s.n.], 2009. p. 1–2. 2.6

TANENBAUM, A. S.; STEEN, M. V. *Sistemas Distribuídos: Princípios e Paradigmas*. São Paulo: Pearson Prentice Hall, 2007. (document), 2.2, 2.1

TAURION, C. *Cloud Computing - Computação em Nuvem - Transformando o Mundo da Tecnologia da Informação*. Rio de Janeiro: Brasport, 2009. 2.2, 2.2, 2.5

VELTE, A. T.; VELTE, T. J.; ELSENPETER, R. *Cloud Computing: Computação em Nuvem - Uma Abordagem Prática*. Rio de Janeiro: Altabooks, 2011. 2.2, 2.2, 2.3, 2.5

VERAS, M. *Cloud Computing - Nova arquitetura de TI*. Rio de Janeiro: Brasport, 2012. (document), 2.1, 2.3, 2.4, 2.2

WHITE, O. *Developer Productivity Report 2012: Java Tools, Tech, Devs & Data*. 2012. URL: <http://zeroturnaround.com/rebellabs/developer-productivity-report-2012-java-tools-tech-devs-and-data/4/ó/>. Acesso em: 14 out. 2013. 3.1

WLCG. *WLCG*. 2013. URL: <http://wlcg.web.cern.ch/>. Acesso em: 25 set. 2013. 2.5

*Um Modelo de Cloud Gaming para Jogos Digitais*

Alberto Vianna Dias da Silva

Salvador, Março de 2014.