



*Simulador para ensino de algoritmos de
escalonamento de discos*

Alex Sandro Rodrigues Ancioto

Dezembro / 2018

Dissertação de Mestrado em Ciência da
Computação

Simulador para ensino de algoritmos de escalonamento de discos

Esse documento corresponde ao Projeto de Pesquisa de qualificação apresentado à Banca Examinadora para qualificação no curso de Mestrado em Ciência da Computação do UNIFACCAMP – Centro Universitário Campo Limpo Paulista.

Campo Limpo Paulista, 14 de dezembro de 2018.

Alex Sandro Rodrigues Ancioto

Prof. Dr. Marcelo de Paiva Guimarães

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

**Ficha catalográfica elaborada pela
Biblioteca Central da UNIFACCAMP**

A554s

Ancioto, Alex Sandro Rodrigues

Simulador para ensino de algoritmos de escalonamento de discos /
Alex Sandro Rodrigues Acioto. Campo Limpo Paulista, SP:
UNIFACCAMP, 2018.

Orientador: Prof^o. Dr^o. Marcelo de Paiva Guimarães

Dissertação (Programa de Mestrado Profissional em Ciência da
Computação) – Centro Universitário Campo Limpo Paulista –
UNIFACCAMP.

1. Realidade virtual. 2. Algoritmo de escalonamento. 3. Imersão. 4.
Disco magnético. 5. Disco sólido I. Guimarães, Marcelo de Paiva. II.
Campo Limpo Paulista. III. Título.

CDD-005.1

Agradecimentos

Indubitavelmente as palavras aqui escritas serão insuficientes para homenagear a todas as pessoas que fizeram parte dessa importante fase da minha vida e doaram um pouco de si para o aperfeiçoamento deste trabalho.

Em primeiro lugar, gostaria de agradecer a meus pais que me apoiaram das mais diversas formas, desde sempre, e tiveram um papel fundamental nesta caminhada.

Dedico especial agradecimento ao Professor Dr. Marcelo de Paiva Guimarães pela sua incalculável dedicação, incentivo e pela orientação deste trabalho, e por meio dele, me reporto a toda comunidade da UNIFACCAMP pelo apoio incondicional.

A minha namorada Carolina Cava, pelo carinho, amor, e principalmente por me acompanhar com paciência e compreensão nessa longa jornada.

A todos os meus familiares que estiveram presentes nos momentos difíceis e sem os quais teria sido muito difícil vencer este desafio.

Aos pesquisadores e professores da banca examinadora pela atenção e contribuição dedicada a este estudo.

E por fim, mas não menos importante, desejo um carinhoso agradecimento aos amigos que nobremente colaboram direta ou indiretamente pelas preciosas sugestões e críticas, sempre construtivas.

Resumo. Realidade Virtual é uma tecnologia que proporciona ao usuário a sensação de substituir o ambiente real por um gerado por computador, por meio do qual é possível criar efeitos visuais, sonoros e até táteis em um ambiente virtual interativo. A Realidade Virtual vem se demonstrando muito eficiente no auxílio de processos definidos, atividades complexas e experiências que não são possíveis no mundo real, porém, não é possível afirmar que ela consegue auxiliar no aprendizado nos mais diversos contextos, como nos abstratos. Com o intuito de explorar o uso da tecnologia de Realidade Virtual sobre conceitos abstratos na disciplina de sistemas operacionais, foi desenvolvido um simulador imersivo e interativo. Ele foi desenvolvido para simular os algoritmos de escalonamentos de discos FCFS (*First-Come, First-Served*), NOOP e o PIQ (*Parallel Issue Queuing*). Além disso, apresenta arquitetura física dos discos magnéticos e sólidos. O uso da tecnologia pode tornar a aprendizagem mais significativa e sobretudo mais atraente, contudo, a simples presença da tecnologia computacional no ambiente de aprendizagem não é suficiente para assegurar melhorias no processo educativo. Assim, uma vez desenvolvido o software foi necessário validar se ele atende suas funções e, para isso, foi realizado um teste de usabilidade com o objetivo de detectar os pontos a serem melhorados e um teste análise de aspecto motivacional com intuito de avaliar a motivação dos alunos quanto ao uso do simulador. Os testes foram realizados tanto em um dispositivo HMD (*Head Mounted Display*) quanto em um desktop, sendo que um comparativo entre eles foi criado. Por último, foi executado um teste de conhecimento para avaliar o quanto os alunos evoluíram na aprendizagem. Para isso, foi feito um comparativo entre três métodos de ensino: método tradicional e o uso do simulador tanto em um ambiente com desktop como no ambiente com um HMD. Apesar dos testes de avaliação de conhecimento não terem demonstrado uma diferença estatística significativa na avaliação entre os métodos de ensino, pode-se afirmar a partir dos testes de aspectos motivacionais, uma maior motivação quanto ao uso do simulador no ambiente HMD (imersivo) e, por meio do teste de usabilidade, foi possível detectar melhorias a serem feitas como, por exemplo, controle e liberdade do usuário, ajustes na interface gráfica, controle de som, etc.

Palavras-chave: *Realidade Virtual, algoritmo de escalonamento, imersão, disco magnético, disco sólido.*

Abstract: *Virtual Reality is a technology that gives the user the feeling of replacing the real environment with a computer generated one, through which it is possible to induce visual, sound and even tactile effects in an interactive virtual environment. Virtual Reality is proving to be very efficient in the aid of defined processes, complex activities and experiences that are not possible in the real world, but it is not possible to affirm that it can help in learning in diverse contexts, as in abstract ones. In order to explore the use of Virtual Reality technology over abstract concepts in the operating systems discipline, an immersive and interactive simulator was developed. It was developed to simulate the scalability algorithms of First-Come, First-Served (FCFS), NOOP and Parallel Issue Queuing (PIQ) disks. In addition, it presents physical architecture of magnetic and solid disks. The use of technology can make learning more meaningful and above all more attractive, however, the mere presence of computational technology in the learning environment is not enough to ensure improvements in the educational process. Thus, once the software was developed it was necessary to validate if it fulfills its functions and, for this, a usability test was carried out with the objective of detecting the points to be improved and a test motivational aspect analysis in order to evaluate the motivation of the the use of the simulator. The tests were performed in both an HMD (Head Mounted Display) device and a desktop, and a comparison was made between them. Finally, a knowledge test was conducted to assess how much students have evolved in learning. For this, a comparison was made between three teaching methods: traditional method and the use of the simulator in both a desktop environment and the environment with an HMD. Although the knowledge assessment tests did not show a statistically significant difference in the evaluation between the teaching methods, it is possible to affirm from the tests of motivational aspects a greater motivation regarding the use of the simulator in the HMD (immersive) environment and, through of the usability test, it was possible to detect improvements to be made for example, control and freedom of the user, adjustments in the graphical interface, control of sound, and others.*

Keywords: *Virtual Reality, scheduling algorithm, immersion, magnetic disk, solid disk.*

Sumário

Glossário	v
Lista de Tabelas	vii
Lista de Figuras	viii
1. INTRODUÇÃO	1
1.1. Contexto e Motivação	1
1.2. Objetivos	3
1.3. Organização do Trabalho	4
2. ARMAZENAMENTO SECUNDÁRIO: DISCOS RÍGIDOS E SÓLIDOS	5
2.1. Introdução	5
2.2. Disco Magnético	5
2.3. Disco Sólido	8
2.4. Considerações finais do capítulo	11
3. ALGORITMOS DE ESCALONAMENTOS DE DISCO	12
3.1. Introdução	12
3.2. FCFS (First-Come, First-Served)	14
3.3. SSTF (Shortest Seek Time First)	16
3.4. SCAN	17
3.5. NOOP	18
3.6. DeadLine	19
3.7. PIQ (Parallel Issue Queuing)	19
3.8. Considerações finais do capítulo	21
4. REALIDADE VIRTUAL (RV)	23
4.1. Introdução	23
4.2. Dispositivos de Entrada	25
4.2.1. Rastreadores de Posição	25

4.2.1.1. Luvas de Dados	26
4.2.1.2. Dispositivo de Entrada Biológico	26
4.3. Dispositivos de Saída	27
4.3.1. Head-Mounted Display	27
4.3.2. Ambientes de Projeções	28
4.4. Dispositivos de Processamento	29
4.5. Ferramentas de Desenvolvimento	31
4.6. Considerações finais do capítulo	32
5. TRABALHOS RELACIONADOS	34
5.1. Introdução	34
5.2. Considerações finais do capítulo	41
6. PROJETO E IMPLEMENTAÇÃO	43
6.1. Introdução	43
6.2. Etapa Análise	44
6.3. Etapa Projeto	45
6.4. Etapa Implementação	46
6.5. Considerações finais do capítulo	56
7. METODOLOGIA DOS TESTES	57
7.1. Introdução	57
7.2. Avaliação do simulador como método de apoio educacional (Conhecimento)	59
7.2.1. Etapa 1	60
7.2.2. Etapa 2	63
7.3. Teste de usabilidade	63

7.4. Teste de aspecto motivacional	67
7.5. Considerações finais do capítulo	69
8. RESULTADOS DOS TESTES E DISCUSSÃO	70
8.1. Resultado do simulador como apoio do ensino dos algoritmos de escalonamento de disco (Conhecimento)	70
8.1.1. Resultado do teste de conhecimento aplicado na Etapa 1	70
8.1.2. Resultado do teste de conhecimento aplicado na Etapa 2	72
8.1.3. Discussão dos resultados do simulador como apoio do ensino dos algoritmos de escalonamento de disco (conhecimento)	74
8.2. Resultados dos testes de usabilidade	75
8.2.1. Resultado do teste de usabilidade aplicado na Etapa 1	75
8.2.2. Resultado do teste de usabilidade aplicado na Etapa 2	77
8.2.3. Discussão sobre o teste de usabilidade etapas 1 e 2	79
8.2.3.1. Discussão dos resultados do teste de usabilidade aplicado Etapa 1	79
8.2.3.2. Discussão dos resultados do teste de usabilidade aplicado na Etapa 2	81
8.3. Teste Motivacional	82
8.3.1. Resultado do teste de aspecto motivacional aplicado na Etapa 1	82
8.3.2. Resultado do teste de aspecto motivacional aplicado na Etapa 2	83
8.3.3. Discussão dos testes de aspectos motivacionais aplicados na Etapa 1 e Etapa 2	85
8.3.3.1. Discussão do teste de aspecto motivacional aplicado na Etapa 1	85
8.3.3.2. Discussão do teste de aspecto motivacional aplicado na Etapa 2	88
8.4. Considerações finais do capítulo	88
9. CONCLUSÕES E TRABALHOS FUTUROS	89

Glossário

2D - Duas dimensões

3D - Três dimensões

DOF - *Degrees of Freedom*

HMD - *Head Mounted Display*

IFSP - Instituto Federal de São Paulo

RAM - *Random Access Memory*

RV - Realidade Virtual

MR - *Mixed Reality*

SO - Sistema Operacional

MUVE - Multiuser Virtual Environments

FCFS - *First-Come, First-Served*

SJF - *Shortest Job First*

CFQ - *Completely Fair Queuing*

PIQ - *Parallel Issue Queuing*

PAQ - *Physically Addressed Queuing*

SSD - *Solid State Drives*

SAN - *Storage Area Networks*

HPC - *High Performance Computing*

HDD - *Hard Disk Drive*

HIL - *Host Interface Layer*

FTL - *Flash Translation Layer*

HAL - *Hardware Abstraction Layer*

SSTF - *Shortest Seek Time First*

LBN - *Logical Block Number*

RWC - *Read Write Conflict*

RRC - Read Read Conflict

WWC - Write Write Conflict

RWS - Read Write Separation

RRS - Read Read Separation

WWS - Write Write Separation

Lista de Tabelas

Tabela 1: Tabela comparativa dos simuladores analisados	41
Tabela 2: Tabela Recursos Unity	48
Tabela 3: Perfil dos usuários	58
Tabela 4: Resumo dos testes da Etapa 1 com o algoritmo FCFS	58
Tabela 5: resumo dos testes da Etapa 2 com os algoritmos NOOP e PIQ	59
Tabela 6: Questionário de conhecimento (pré-teste e pós-teste)	62
Tabela 7: Questionário de conhecimento (pré-teste e pós-teste) NOOP e PIQ	63
Tabela 8: Questionário – Teste de usabilidade	66
Tabela 9: Questões abertas teste de usabilidade	66
Tabela 10: Questões sobre aspectos motivacionais	68

Lista de Figuras

Figura 1: Visão lateral e esquemática de um disco magnético. Adaptado de (Deitel <i>et al.</i> 2005).	6
Figura 2: Arquitetura interna SSD. Adaptado de (Jung <i>et al.</i> 2012)	9
Figura 3: Camada lógica SSD. Adaptado de (Jung <i>et al.</i> 2014)	10
Figura 4: Algoritmo FCFS - Adaptado de (Deitel <i>et al.</i> 2005)	15
Figura 5: Limitações do algoritmo FCFS - Adaptado de (Deitel <i>et al.</i> 2005)	15
Figura 6: Algoritmo SSTF - Adaptado de (Deitel <i>et al.</i> 2005)	16
Figura 7: Algoritmo SSTF - Adaptado de (Deitel <i>et al.</i> 2005)	17
Figura 8: Algoritmo NOOP. Adaptado de (Gao <i>et al.</i> 2014)	18
Figura 9: Algoritmo PIQ. Adaptado de (Gao <i>et al.</i> 2014)	21
Figura 10: Grau de liberdade. Adaptado de (Siscoutto and Costa 2008)	26
Figura 11: Luva de dados (Fonte: Siscoutto and Costa 2008)	26
Figura 12: Exemplo de HMD desenvolvido pela empresa Oculus (Fonte: https://www.oculus.com)	28
Figura 13: Imagem de uma caverna e seus usuários (Fonte: “Adaptada” de Tori <i>et al.</i> 2017)	29
Figura 14: Simulador Web (Fonte: Souza)	34
Figura 15: Simulador Track Animation (Fonte: Zanduetta 2013)	35
Figura 16: Simulador de algoritmos (Fonte: Suranauwarat, 2017)	36
Figura 17: Simulador de algoritmos (Fonte: Robbins, 2004)	38
Figura 18: Simulador EagleTree (Fonte: Dayan <i>et al.</i> 2013)	39
Figura 19: Etapas do Modelo incremental	43
Figura 20: Fluxograma de atividades	44
Figura 21: Caso de Uso do simulador	46
Figura 22: Tela inicial do simulador	47
Figura 23: Algoritmo FCFS Conceitos	49
Figura 24: monitor de simulação do algoritmo FCFS Conceitos	50
Figura 25: Algoritmo FCFS Limitações	50
Figura 26: Monitor simulação algoritmo FCFS Limitações	51
Figura 27: Disco Magnético	52
Figura 28: Disco SSD	52
Figura 29: Conceitos do algoritmo NOOP	53
Figura 30: Simulação do algoritmo NOOP	54
Figura 31: Conceitos do algoritmo PIQ	54

Figura 32: Simulação do algoritmo PIQ	55
Figura 33: Imagens do algoritmo PIQ para visualização 3D (olho esquerdo e direito)	56
Figura 34: Imagens limitações do algoritmo FCFS para visualização 3D (olho esquerdo e direito)	56
Figura 35: Acertos obtidos no método tradicional de Ensino – FCFS	71
Figura 36: Acertos obtidos no método de ensino auxiliado pelo simulador executado em desktop – FCFS.....	71
Figura 37: Acertos obtidos no método de ensino auxiliado pelo simulador executado em Gear VR – FCFS	71
Figura 38: Análise estatística dos dados ANOVA	72
Figura 39: Acertos e erros obtidos no método tradicional de Ensino – NOOP e PIQ.....	73
Figura 40: Acertos obtidos no método de ensino auxiliado pelo simulador executado em desktop – NOOP e PIQ	73
Figura 41: Acertos obtidos no método de ensino auxiliado pelo simulador executado em Gear VR – NOOP e PIQ	73
Figura 42: Análise estatística dos dados ANOVA	74
Figura 43: Teste de usabilidade do simulador executando em um desktop – Etapa 1	76
Figura 44: Teste de usabilidade do simulador executando em um Gear VR - Etapa 1	76
Figura 45: Média de usabilidade por questão – Etapa 1.....	77
Figura 46: Teste de usabilidade do simulador executando em um desktop – Etapa 2	77
Figura 47: Teste de usabilidade do simulador executando em um Gear VR. – Etapa 2	78
Figura 48: Média de usabilidade por questão segunda etapa – Etapa 2	78
Figura 49: Média geral de usabilidade por etapa/grupo	78
Figura 50: Teste Motivacional Desktop e Gear VR – Etapa 1	82
Figura 51: Média motivacional por questão – Etapa 1.....	83
Figura 52: Teste Motivacional desktop – Etapa 2.....	83
Figura 53: Teste Motivacional Gear VR – Etapa 2	84
Figura 54: Média por questão – Etapa 2	84
Figura 55: Média de aspecto motivacional por fase/grupo	84
Figura 56: Resumo das discussões por ambiente	87

1. INTRODUÇÃO

1.1. Contexto e Motivação

Conforme Deitel *et al.* (2005), Sistema Operacional (SO) é uma camada de *software*¹ que separa as aplicações do *hardware*² que elas acessam e tem como funcionalidade fornecer serviços que permitem às aplicações serem executadas com segurança e efetividade. O avanço dos SOs está ligado à evolução dos computadores nos quais são executados (Tanenbaum, 2010). Tal evolução é implementada por meio de abstrações³ cada vez mais sofisticadas que tornam os SOs mais poderosos e escondem os mecanismos de *hardware* e *software* subjacentes, que de fato suportam suas funcionalidades.

As funcionalidades dos SOs são inúmeras e independentes entre si, dentre as quais cita-se o gerenciamento de processos, a gerência de memória, a gerência de arquivos e a de dispositivos de entrada e saída (Machado, 2007). Nos cursos da área de Computação a disciplina de SO tem um papel de destaque, pois é em seu conteúdo que os alunos aprendem tais funcionalidades.

Tradicionalmente a disciplina de SO é ensinada utilizando recursos convencionais, como projetores multimídia e lousa. Nesse contexto é comum os alunos apresentarem dificuldades de entendimento, principalmente porque envolve muitos conceitos abstratos. Por outro lado, nos últimos anos surgiram novas tecnologias computacionais que podem ser utilizadas para auxiliar o ensino como, por exemplo, a Realidade Virtual (RV). Essa tecnologia permite a construção de simuladores 3D (tridimensionais) interativos e imersivo.

A imersão psicológica é o estado mental de ser completamente absorvido ou envolvido com alguma coisa (Liu *et al.*, 2017). Por exemplo, um jogo em um ambiente de RV atrai os espectadores para o mundo retratado na tela, e eles se sentem presos nesse ambiente virtual. O uso de narrativa e simbolismo cria situações confiáveis e envolventes (Dawley *et al.*, 2013) neste caso, cada participante pode influenciar o que acontece por

¹ Entende-se como software em um sistema computacional, o conjunto lógico de programas.

² Entende-se como hardware os componentes eletrônicos e físicos.

³ As abstrações envolvem desde a ocultação dos detalhes de hardware até o gerenciamento do próprio sistema operacional.

meio de suas ações e pode interagir com os outros. Por meio de estímulos mais ricos, displays montados na cabeça podem criar imersão sensorial para aprofundar o efeito da imersão psicológica, além de induzir presença virtual (ilusão de lugar), a sensação de estar em um local no mundo virtual (Slater, 2017).

Portanto, a RV vem sendo amplamente usada com o intuito de aprimorar o entendimento de conceitos em várias áreas de conhecimento, por exemplo, na área da Matemática e suas Tecnologias, Kaufmann *et al.* (2000) em seus estudos, propuseram uma ferramenta para abordagem de conceitos abstratos em geometria. Roussou (2009) utilizou RV para desenvolver um “*playground virtual*” para o ensino da matemática, abordando problemas de fração, para crianças. Lin *et al.* (2013) utilizaram a RV na aprendizagem de geografia. Coulter *et al.* (2007) usaram a RV na medicina para o ensino sobre tratamento de traumatismo craniano.

Embora o uso da RV seja visto como positivo no processo educacional (Coulter *et al.*, 2007; Roussou, 2009; Liu *et al.*, 2017; Lin *et al.*, 2013), ela não torna intrinsecamente melhor toda experiência de seu uso em termos de motivação e aprendizagem. O uso indiscriminado da imersão sensorial pode até piorar a experiência educacional, como exemplo: a utilização por longo período de tempo em um ambiente imersivo pode proporcionar desconforto, a riqueza proporcionada em um ambiente de RV pode reduzir o foco do objetivo e a resolução visual pode dificultar a leitura (Dede *et al.*, 2017). Nestes casos, a simulação em 2-D (não-imersivo) pode ser tão ou mais eficiente quanto a mídia imersiva se usada para aprendizagem no processo de construção de conhecimento. Dores *et al.* (2012) em sua revisão sistemática também elencam os aspectos positivos e negativos quanto ao uso da RV. Apesar da RV ser amplamente explorada para proporcionar melhoria no processo educacional em várias áreas e estar consolidada no ensino-aprendizagem com conteúdo concreto, por exemplo, pilotagem de avião, o seu uso em contexto abstrato ainda não está consolidado. Dessa forma, diante de inúmeras vantagens e desvantagens, certezas e dúvidas, existe uma oportunidade de investigação do uso da RV em contextos abstratos, como na disciplina de SO.

1.2. Objetivos

A RV vem se demonstrando muito eficiente no auxílio de processos definidos, atividades complexas e experiências mediadas que seriam inviáveis no mundo real, conforme apresentado em Lin *et al.* (2013), porém, não é possível afirmar que RV consegue auxiliar no aprendizado em contextos abstratos.

Este trabalho tem como objetivo analisar o uso da tecnologia de RV em um contexto abstrato, construindo um simulador imersivo e interativo para apoiar o ensino de algoritmos de escalonamento de disco. Além disso, apresenta o teste de usabilidade do simulador e uma análise de aspecto motivacional. Também é mostrado um comparativo do ensino dos algoritmos FCFS (*First Come, First Served*), NOOP e PIQ (*Parallel Issue Queuing*), utilizando a sala de aula tradicional (lousa), um laboratório de computação (visualização-2D do simulador) e um ambiente imersivo 3D baseado no uso de um HMD (*Head Mounted Display*).

O simulador foi desenvolvido de modo a permitir que os alunos conheçam a estrutura interna de um disco magnético e sólido. É esperado que o simulador auxilie o processo de ensino-aprendizagem dos algoritmos, transformando o aprendizado em uma situação de exploração, descoberta, observação e construção de conhecimento na medida em que proporciona uma nova forma de entender de forma prática conceitos abstratos. O simulador apresenta as principais características:

1. Proporciona aos usuários uma experiência imersiva, interativa e em tempo real;
2. Simula os algoritmos FCFS, NOOP e PIQ;
3. Simula e apresenta a ordem de atendimento das requisições;
4. Permite a interação de usuários com o ambiente usando dispositivos de entrada como controle, teclado e mouse;
5. Apresenta e simula no ambiente a arquitetura física de um disco magnético ou sólido;
6. Permite o usuário navegar em um laboratório com bancadas específicas para a explicação dos conceitos e limitações dos algoritmos usando dispositivos não imersivos, como desktops, e imersivos, como *Google Cardboard* ou *Gear VR-Samsung*.

1.3. Organização do Trabalho

O restante do trabalho está organizado da seguinte forma: o capítulo 2 mostra o embasamento teórico sobre discos magnéticos e discos sólidos, focando no conceito de sua arquitetura física interna; o capítulo 3 apresenta os principais algoritmos de escalonamento de disco FCFS, SSTF, SCAN, NOOP, Deadline e PIQ; o capítulo 4 trata dos conceitos sobre RV; o capítulo 5 relata trabalhos relacionados; o capítulo 6 apresenta o simulador desenvolvido; o capítulo 7 descreve a metodologia utilizada nos testes de usabilidade, nas análises de aspectos motivacionais e de aprendizagem realizados; no capítulo 8 são apontados os resultados dos testes e discussão dos resultados e, por fim, o capítulo 9 apresenta as conclusões e trabalhos futuros.

2. ARMAZENAMENTO SECUNDÁRIO: DISCOS RÍGIDOS E SÓLIDOS

O armazenamento secundário trata de conservar os dados de forma segura em dispositivos não voláteis. É um assunto fundamental na disciplina de SOs e é o tema abordado no simulador desenvolvido. Este capítulo apresenta os principais conceitos da arquitetura interna dos discos magnéticos, assim como a arquitetura dos discos sólidos, levando-se também em consideração os conceitos da camada lógica.

2.1. Introdução

A memória principal (armazenamento primário) é geralmente implementada em módulos de memória RAM (*Random Access Memory* – Memória de Acesso Randômico). Apesar de possuir alta velocidade de acesso em comparação aos discos rígidos, é volátil, ou seja, os dados armazenados são perdidos caso a alimentação de energia seja interrompida e, em muitos momentos, não possui espaço suficiente para armazenar a demanda de dados e programas do sistema. Por isso, a maioria dos computadores possuem uma memória secundária com o objetivo de armazenar grandes quantidades de dados de forma não volátil, ou seja, os dados não são perdidos caso a alimentação de energia seja interrompida. Assim sendo, os dados armazenados na memória secundária são gravados de forma permanente. Os tipos de dispositivos de armazenamento secundário mais utilizados atualmente são os magnéticos e os de estado sólido.

Normalmente, os programas dos usuários são armazenados em disco até serem carregados na memória principal. Somente após carregados o processador pode acessar e executar (Silberschatz *et al.*, 2001). Dessa forma, um gerenciamento de armazenamento de disco eficiente é indispensável para um sistema de computação. A memória principal e secundária tem como objetivo o armazenamento, contudo se diferenciam em relação a velocidade, custo, tamanho e volatilidade.

2.2. Disco Magnético

Os discos magnéticos são também conhecidos como disco rígido HDD (*Hard Disk Drive*), disco fixo ou simplesmente disco. Eles são não voláteis e, diferentemente da memória principal, fornecem velocidade de acesso variável ao seu conteúdo conforme a posição relativa entre o cabeçote de leitura e do dado requisitado.

Segundo (Silberschatz *et al.*, 2001), um disco magnético é constituído de lâminas de disco que são planas e circulares semelhantes a um CD (*Compact Disc*). Suas superfícies, superior e inferior, são cobertas por um revestimento magnetizado no qual os dados são gravados. As lâminas de discos presentes nos discos magnéticos também são referenciadas como discos ou pratos (*platters*). Prato é o termo que será utilizado nesta dissertação.

Conforme Morimoto (2011), a capacidade de um disco magnético é determinada por dois fatores: a tecnologia utilizada que determina a densidade (material empregado para construir a superfície magnética do prato) e o diâmetro do prato (determinado em polegadas).

Os dados são lidos ou gravados nos discos magnéticos por meio de uma bobina condutora, denominada cabeçote ou cabeça de leitura. Para que os dados sejam lidos ou gravados, o cabeçote se movimenta pairando a uma pequena distância da superfície de cada prato - cada superfície possui o respectivo cabeçote (Deitel *et al.*, 2005). Assim, o acesso aos dados ocorre tanto acima ou abaixo de cada prato. A Figura 1 mostra a visão lateral e esquemática dos componentes básicos de um disco.

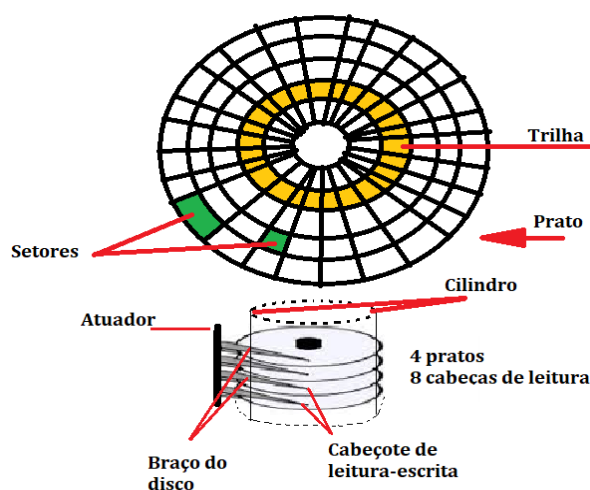


Figura 1: Visão lateral e esquemática de um disco magnético. Adaptado de (Deitel *et al.*, 2005)

No processo de leitura, o cabeçote lê os dados medindo como o meio magnetizável altera a corrente. Já no processo de gravação, o cabeçote grava os dados utilizando a corrente para alterar a magnetização do meio. Os cabeçotes são presos a um braço (*arm*), que permite acesso a todo o prato. Conforme Morimoto (2011), o braço de leitura é uma

peça triangular, normalmente feita de alumínio, que permite o acesso a todo o disco. O mecanismo que movimenta o braço de leitura é chamado de atuador (*actuator*). Os pratos são conectados a uma haste que gira em rotação constante graças ao motor de rotação, cuja velocidade é medida em rotação por minuto (RPM). Os discos magnéticos mais antigos utilizavam motores de 3.600 rotações por minuto, enquanto que, atualmente, são utilizados motores de 7.200, 10.000 ou 15.000 RPM.

A superfície de um disco é logicamente dividida em faixas circulares concêntricas, chamadas trilhas (*tracks*), numeradas em ordem crescente da parte externa do disco ao centro. As trilhas são divididas com base no número de setores de tamanho fixo que contém as informações que podem ser lidas ou gravadas. Um setor é a menor unidade que pode ser lida ou gravada; normalmente, contém 512 bytes de dados (Deitel *et al.*, 2005). O conjunto de trilhas que está em uma posição do braço forma um cilindro. Portanto, conforme Deitel *et al.* (2005), o SO realiza a leitura e gravação de dados no prato definindo: o cabeçote (determina qual a superfície do prato), o cilindro (indica a trilha que deve ser realizada a operação) e o setor no qual o dado será gravado ou lido. A disposição do cabeçote de leitura e escrita na região correta baseia-se em sinais de retorno (*feedback*) que orientam o posicionamento (Morimoto, 2011).

Nos discos antigos o número de setores em todos os cilindros eram os mesmos, tal arquitetura limitava a capacidade de armazenamento, já que a quantidade de setores nas trilhas mais externas eram as mesmas das mais internas. Uma outra abordagem, na qual os discos são divididos por zonas (*Zoned bit Recording - ZBR*), permite que as zonas mais externas tenham mais setores que as mais internas (Tanenbaum, 2010). Isso possibilita uma distribuição mais racional do espaço em disco e, conseqüentemente, uma maior densidade de gravação (Morimoto, 2011). Para isso, detalhes da arquitetura física do disco é ocultada e uma geometria virtual é apresentada ao SO, possibilitando então a visualização da estrutura física do disco de forma abstrata. Neste caso o SO passava a agir como se tivesse x cilindros, y cabeçotes e z setores por trilha. A controladora então mapeava as requisições com coordenadas (x, y, z) em um posicionamento real de cilindro, cabeçote e setor.

Com a evolução do sistema de armazenamento, o modelo por zona passou a tornar-se ineficiente, já que era necessário manter compatibilidade com as limitações do IBM PC

original⁴. Para superar esses limites, os discos passaram a suportar um sistema chamado endereçamento lógico de bloco (*Logical Block Addressing - LBA*), que enumera os setores consecutivamente a partir do setor zero, sem considerar a geometria do disco (Tanenbaum, 2010). Ou seja, os blocos de dados passaram a ser endereçados de forma linear, sendo o primeiro bloco LBA 0, o segundo LBA 1 e assim sucessivamente.

O acesso às posições do disco é exercido por intermédio das controladoras, que usam protocolos padronizados (por exemplo, ATA, IDE, SATA, SCSI) para realizar a comunicação entre o processador e o disco. Para comunicação, os comandos são convertidos conforme o protocolo, permitindo então o controle da rotação do motor, movimentação do cabeçote de leitura sobre os setores corretos, verificação dos possíveis erros de leitura e o controle do *cache* de disco (Morimoto, 2011).

2.3. Disco Sólido

Os discos rígidos por serem na sua maior parte mecânico, que talvez seja um dos maiores dos seus problemas, foram considerados por um grande período de tempo um dos principais gargalos dos sistemas computacionais devido à alta latência para acessos de dados aleatórios e baixo *throughput*⁵ para lidar com várias requisições simultâneas. Como evolução, surgiram os dispositivos baseados em memória *flash*, conhecidos como discos de estado sólido (*Solid State Drives (SSDs)*), que são construídos inteiramente de chips de semicondutores sem partes mecânicas. Esses discos vêm se tornando a tecnologia de armazenamento dominante por vários motivos, entre eles, a compatibilidade com as interfaces utilizadas pelos HDDs atuais, baixa latência de leitura e escrita, baixo consumo de energia e resistência maior a choques mecânicos, quando comparados com os magnéticos (Agrawal *et al.*, 2008). Tais vantagens vêm tornando o uso deles cada vez mais presente em computadores pessoais, como *laptops* e em aplicações empresariais, como *Storage Area Networks (SANs)*, e até mesmo computação de alta performance (HPC - *High Performance Computing*).

⁴ IBM PC (*Personal Computer*) também conhecido como "computador pessoal" foi lançado em 1981. A versão original deu origem a plataforma de hardware dos "IBM PC" no qual, recebeu a denominação IBM 5150.

⁵ *throughput* pode ser considerado como taxa de transferência ou, ou a quantidade de dados processados em um determinado espaço de tempo.

A Figura 2 mostra um exemplo da arquitetura física interna de um SSDs, na qual eles são organizados através de chips de memória *flash NAND* interconectados por meio de canais a um controlador.

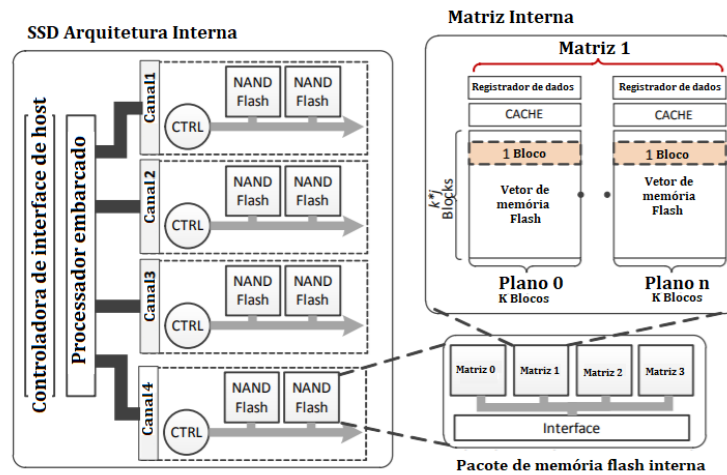


Figura 2: Arquitetura interna SSD. Adaptado de (Jung *et al.*, 2012)

Basicamente um SSD é composto de múltiplos canais, sendo que cada um é conectado a vários chips de memória *flash NAND*. Cada chip de memória *flash NAND* é composto de múltiplas matrizes composta por múltiplos planos e cada plano contém várias páginas (Gao *et al.*, 2014). As páginas são consideradas as menores unidades unidade de leitura/gravação em um disco SSD (Jung *et al.*, 2014).

As operações de leitura e gravação na memória *flash* são processadas em unidade de página. Diferente dos discos magnéticos, as páginas de memória *flash* devem ser apagadas antes de serem reescritas. Porém, uma operação de apagamento ocorre em um conjunto de páginas, chamado coletivamente de bloco (*block*), que é a menor unidade para uma operação de apagamento (Jung *et al.*, 2014).

As operações de leitura em dispositivos SSD tendem a ter melhor desempenho que as operações de gravação. Um dos pontos que podem levar a essa disparidade de desempenho é a exigência de que, se um bloco a ser escrito ainda não estiver livre, um apagamento deve preceder a gravação. Portanto, se uma gravação ocorrer em um bloco ocupado, uma latência de apagamento mais a latência de gravação será incorrida. Neste caso, poderia degradar ainda mais o desempenho se no processo de regravação for necessário gravar em um subconjunto de um bloco de modo que parte dos dados existentes nele sejam mantidos.

A Figura 3 ilustra as camadas lógicas de um SSDs (Jung *et al.*, 2012). Na parte superior estão o kernel (núcleo) do SO e o driver disco.

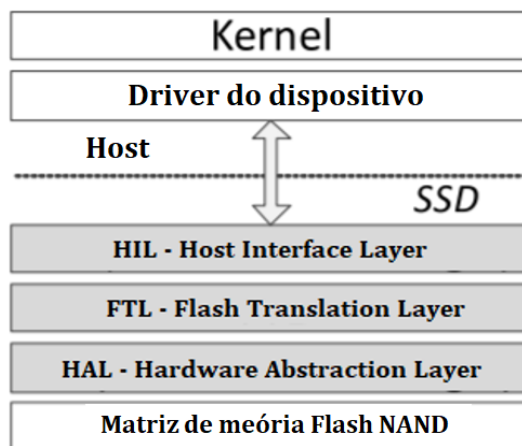


Figura 3: Camada lógica SSD. Adaptado de (Jung *et al.*, 2014)

- Camada lógica de interface do *host*-HIL (*Host Interface Layer*): responsável pela comunicação entre o sistema *host*⁶ e as camadas subjacentes no SSD. É a camada onde comandos de Entrada/Saída (E/S) são analisados e as operações com base nos protocolos de interface são executadas como: iniciar uma transferência de dados ou confirmar transações *NAND* para camadas subjacentes;
- Camada de tradução *flash*-FTL (*Flash Translation Layer*): responsável pela tradução de endereços do espaço do *host*, que contém endereços virtuais ou lógicos, em endereço do espaço físico onde os dados estão localizados no chip de memória *flash* (canais, chips, matrizes, planos e páginas);
- Camada de abstração de *hardware*-HAL (*Hardware Abstraction Layer*): gerencia fisicamente a memória *flash NAND*. É encarregada de enviar transações *NAND* para a memória *flash* subjacente e monitorar os status das transações.

Assim como nos discos magnéticos, o desempenho dos SSD também depende dos algoritmos de escalonamento. No caso dos SSD existem quatro níveis principais que podem ser explorados para aumentar o paralelismo em um SSD, sendo eles: canais, chip de memória *flash*, matrizes e planos (Gao *et al.*, 2014). Muitos trabalhos concentram esforços para melhorar o desempenho dos discos SSDs explorando em diversas perspectivas o paralelismo no disco (Jung *et al.*, 2012), (Gao *et al.*, 2014).

⁶ Host refere-se a qualquer máquina ou computador conectado a uma rede.

Gao *et al.*, (2014) propôs um algoritmo de escalonamento que visa melhorar o desempenho de leitura e gravação através da diminuição de conflitos de acesso leitura e gravação, leitura e leitura, e de gravação e gravação conforme será explicado na seção 3.6. No mesmo sentido, Jung *et al.* (2012) propôs mudanças na arquitetura física dos dispositivos SSDs com objetivo de melhorar o desempenho de leituras aleatórias por meio de detecção de conflitos utilizando endereços físicos. Ambos os trabalhos visaram melhorar o desempenho explorando o paralelismo através da eliminação de conflitos.

2.4. Considerações finais do capítulo

Para reduzir o tempo gasto na busca de registros e proporcionar melhora no desempenho do sistema, estratégias de escalonamento de disco devem ser adotadas. No próximo capítulo são apresentados os principais algoritmos de escalonamento de discos abordados na disciplina de SO.

3. ALGORITMOS DE ESCALONAMENTOS DE DISCO

Em um sistema multitarefas, várias aplicações e processos do sistema podem solicitar acessos concorrentes ao disco, para escrita e leitura de dados. Neste caso, se faz necessário criar uma fila de acessos de requisições pendentes onde cada nova requisição de acesso ao disco é inserida. Para gerenciar essas filas, escalonadores de discos, que são um dos principais assuntos desta pesquisa, devem ser utilizados. Neste capítulo são apresentados os conceitos dos algoritmos de escalonamento de discos FCFS, SSTF, SCAN, NOOP e PIQ. Os algoritmos, FCFS, SCAN e SSTF, foram escolhidos por serem propostos e discutidos na maioria dos livros didáticos de sistemas operacionais de graduação, os algoritmos, NOOP e PIQ, por serem voltados para disco de estado sólidos e este último, por explorar o paralelismo em disco SSD.

3.1. Introdução

De acordo com (Tanenbaum, 2010), o tempo gasto para ler ou escrever algum dado no disco magnético pode ser determinado com base em três fatores:

- Tempo necessário para movimentar o braço do disco até o cilindro correto. Esse intervalo é conhecido como tempo de posicionamento. Ele tem influência direta na dinâmica do sistema, de modo que sua redução pode melhorar substancialmente o desempenho do sistema computacional. Muitos processos podem gerar simultaneamente requisições de leitura ou escrita em um disco e gerenciar filas para reter tais requisições. Isso se deve ao fato de que esses processos costumam ser mais rápidos em fazer as requisições do que o disco em às atender (Deitel *et al.*, 2005);
- Atraso de rotação, que correspondente ao tempo necessário para rotacionar o disco de tal forma que seja posicionado o setor correto sob o cabeçote de leitura em que a operação será realizada. Para (Deitel *et al.*, 2005), a geometria do disco é fator importante no desempenho computacional. Neste caso, ele enfatiza que na localização de cilindros em um disco a otimização do tempo de posicionamento deve ser almejada enquanto na otimização de atrasos rotacionais, a localização de setores deve ser priorizada; e

- Tempo de transferência real dos dados.

Todo acesso a disco é realizado por meio de requisições. Elas são geradas pelos processos e são atendidas caso o dispositivo esteja ocioso e, caso contrário, são armazenadas em uma fila de espera. Para reduzir o tempo gasto na busca por requisições de acesso ao disco geralmente a fila é reordenada. Esse processo é denominado escalonamento de disco.

Então, o desempenho do atendimento das requisições envolve uma análise da fila e da reordenação. A estratégia de escalonamento depende do objetivo do SO, mas a maioria adota políticas para maximizar o rendimento e minimizar o tempo de resposta. Para atender tais objetivos, toma-se como base critérios como rendimento (número de requisições atendidas por unidade de tempo), tempo médio de resposta (tempo médio de espera para que uma requisição seja atendida) e variância dos tempos de resposta (medida da previsibilidade dos tempos de resposta) para minimizar o tempo de posicionamento do cabeçote de leitura para um conjunto de requisições em uma fila, proporcionando assim, a menor movimentação mecânica possível no disco, no caso dos magnéticos (Deitel *et al.*, 2005).

Quando os critérios rendimento e tempo médio de resposta dos algoritmos de escalonamento de disco são otimizados, o desempenho do sistema melhora, mas as requisições individuais podem sofrer atrasos (Deitel *et al.*, 2005). O algoritmo de escalonamento de disco deve, nesse sentido, implementar uma estratégia para impedir o adiamento indefinido, ou seja, cada requisição de acesso a disco deve ser atendida dentro de um período de tempo aceitável. Tal estratégia pode ser alcançada por meio do critério de variância, que permite certa medida de previsibilidade do tempo de resposta às requisições.

De acordo com (Deitel *et al.*, 2005), a variância mede como as requisições individuais são atendidas em relação ao desempenho médio do sistema. Portanto, quanto menor for a variância, maior será a probabilidade de que a maioria das requisições do disco sejam atendidas. Nesse caso, a variância pode ser vista como uma medida de justiça e de previsibilidade, já que todas as requisições têm um período de tempo de espera comum.

Conforme Rocha *et al.* (2011), duas requisições são ditas sequenciais quando a posição do final de uma requisição é adjacente ao início da próxima; caso contrário elas são denominadas não-sequenciais ou aleatórias. Caso a solicitação de uma requisição dependa do término da anterior, elas são denominadas síncronas. Caso não exista relações de dependência são denominadas requisições assíncronas.

São apresentados a seguir os principais algoritmos de escalonamento de disco magnéticos e sólidos.

3.2. FCFS (*First-Come, First-Served*)

O algoritmo FCFS atende as requisições na ordem que elas chegam, ou seja, a requisição que chegou mais cedo é atendida em primeiro lugar. É considerado um dos algoritmos mais simples por não reordenar as requisições pendentes. Então, pode ser utilizado tanto nos discos magnéticos quanto nos sólidos.

Esse algoritmo é intrinsecamente justo, pois as requisições que chegam não podem entrar na fila à frente das que estão à espera. Esse comportamento garante que nenhuma requisição seja adiada indefinidamente, mas também significa que ele pode resultar em baixo rendimento quando há várias requisições sucessivas (Deitel *et al.*, 2005).

Como os SSDs não possuem partes móveis, então o desempenho do FCFS é insensível a problemas como tempo de busca e latência rotacional. Contudo, no caso de discos magnéticos, a alta taxa de acesso ao disco alternando entre requisições de acesso do cilindro mais interno ao mais externos, pode fazer com que o algoritmo realize longa operação de busca para atender à requisição seguinte, mesmo que outra requisição esteja mais próxima ao cabeçote. Neste caso o FCFS ignora as relações posicionais entre as requisições pendentes na fila. Conforme (Tanenbaum, 2010), se o *driver* (funções integradas ao SO para controlar a comunicação com um determinado periférico) do disco recebe as requisições e atende uma após outra na ordem em que elas foram recebidas, isto é, FCFS, quase nada pode ser feito para otimizar o tempo de posicionamento.

A Figura 4 ilustra o FCFS em um disco magnético. Considere que inicialmente o cabeçote de leitura esteja posicionada sobre o bloco 60. A fila de pedidos de acesso pendentes corresponde aos seguintes blocos de requisições do disco, em sequência: 20, 10, 30, 70, 110, 80, 120, 60, 20, 50. Considerando-se que nenhuma nova requisição

chegará à fila durante a execução dos algoritmos e que o escalonador já terá associado o número do bloco de cada requisição. Como a cabeça de leitura está inicialmente no bloco 60, ela primeiro passará do bloco 60 para o bloco 20, depois para os blocos 10, 30, 70, 110, 80, 120, 60, 20 e, finalmente, para o bloco 50.

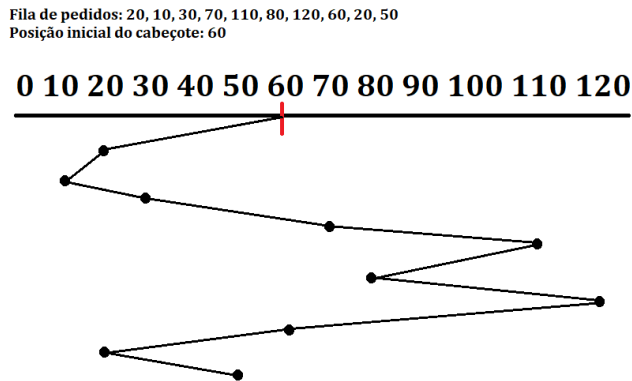


Figura 4: Algoritmo FCFS - Adaptado de (Deitel *et al.*, 2005)

Para ilustrar as limitações do algoritmo FCFS, Figura 5 demonstra que a relação posicional dos blocos de requisições influencia no desempenho do sistema. Nesse caso, a fila de pedidos de acesso pendentes relativa aos blocos do disco teve a seguinte sequência: 120, 20, 110, 20, 110, 120, 30, 120, 10, 90. Como a cabeça de leitura está inicialmente no bloco 60, ela primeiro passará do bloco 60 para o bloco 120, depois para os blocos 20, 110, 20, 110, 120, 30, 120, 10 e, finalmente, para o bloco 90.

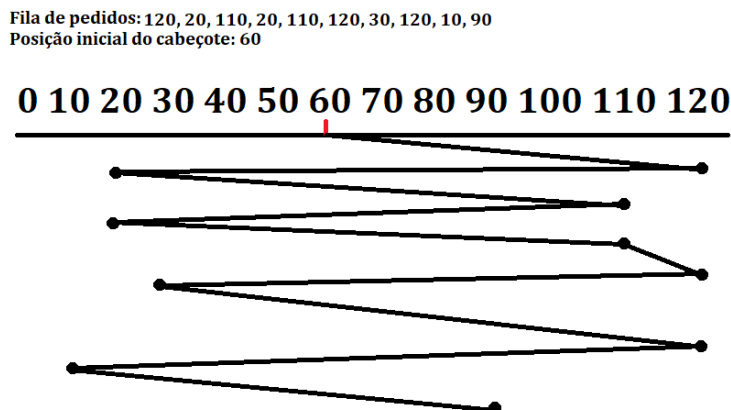


Figura 5: Limitações do algoritmo FCFS - Adaptado de (Deitel *et al.*, 2005)

3.3. SSTF (*Shortest Seek Time First*)

O SSTF ou escalonamento do tipo “tempo de busca mais curto primeiro” atende a todas as requisições mais próximas à posição atual da cabeça de leitura antes de mover o braço de leitura para atender outras requisições mais distantes (Silberschatz *et al.*, 2001), por isso é voltado para discos magnéticos. Geralmente é inviável prever tempos de busca exatos, mas o SSTF pode prever um tempo aproximado com o uso de busca pela distância de uma requisição a outra (Denning, 1967). Neste caso, por reduzir o tempo médio de busca, o SSTF consegue obter taxas de rendimento mais altas que o FCFS, e por consequência, o tempo de resposta tende a ser mais baixo para carga de trabalho moderada (Deitel *et al.*, 2005).

O SSTF não garante justiça e pode resultar em variância (medida da previsibilidade dos tempos de resposta) mais alta dos tempos de resposta para requisições individuais devido a discriminação contra essas requisições quando estão nas trilhas mais externas ou internas, podendo no limite ocasionar inanição (tempo de resposta insatisfatório para as requisições que estejam nas trilhas mais distantes (Deitel *et al.*, 2005).

Levando em consideração a mesma fila de pedidos do conceito do algoritmo FCFS, a Figura 6 ilustra o funcionamento do algoritmo SSTF. Como a cabeça de leitura está inicialmente no bloco 60, ela primeiro passará do bloco 60 para o bloco 70, depois para os blocos 80, 110, 120, 50, 30, 20, 20 e, finalmente, para o bloco 10.

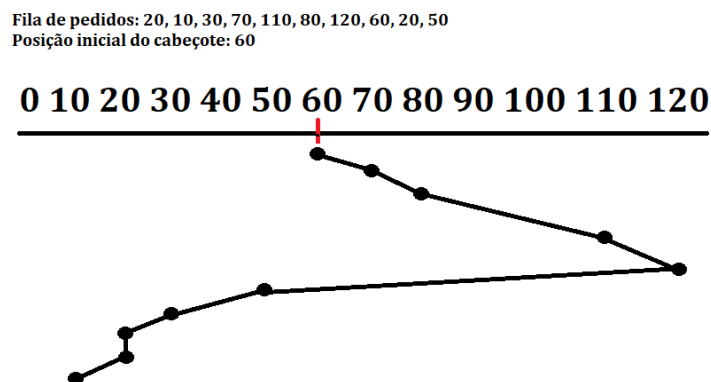


Figura 6: Algoritmo SSTF - Adaptado de (Deitel *et al.*, 2005)

3.4. SCAN

No algoritmo SCAN o braço do disco inicia em uma extremidade do disco e se move na direção a outra extremidade, atendendo as requisições que requerem a menor distância de busca na direção predefinida. Ao chegar à outra extremidade do disco a direção do braço é invertida (Silberschatz *et al.*, 2001). Como o algoritmo se baseia nos movimentos dos braços do discos, então é voltado para discos magnéticos.

O SCAN é as vezes denominado de algoritmo do elevador, uma vez que o braço do disco se comporta análogo a um elevador em um prédio, atendendo a todas as requisições em seu caminho em uma só direção e depois, inverte o movimento para atender as requisições em outro sentido. Como o cabeçote de leitura pode oscilar entre as extremidades de cada prato, as trilhas mais externas podem sofrer maior tempo para serem acessadas e as requisições que chegam serem atendidas antes das requisições que estão na fila de espera. Assim como o SSTF, o SCAN pode sofrer inanição (ou seja, tem menor variação no tempo de resposta para requisições que estão na extremidade do prato) (Deitel *et al.*, 2005), levando em consideração a mesma fila de pedidos do conceito do algoritmo FCFS.

A Figura 7 ilustra o funcionamento do algoritmo SCAN. Como a cabeça de leitura está inicialmente no bloco 60 para o bloco 70, depois para os blocos 80, 90, 100, 110, 50, 30, 20, 20 e, finalmente, para o bloco 10.

Fila de pedidos: 20, 10, 30, 70, 110, 80, 60, 20, 90, 100, 50
Posição inicial do cabeçote: 60

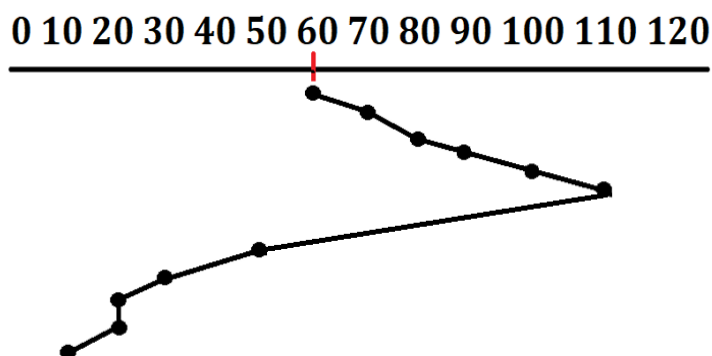


Figura 7: Algoritmo SCAN - Adaptado de (Deitel *et al.*, 2005)

Como todo prato é percorrido em ambos os sentidos, como resultado, o SCAN resiste à inanição de forma mais eficaz do que o SSTF (Denning, 1967).

3.5. NOOP

O algoritmo de escalonamento NOOP foi introduzido no sistema operacional Linux versão 2.6 (Stephen *et al.*, 2014) e atende um mínimo de operações na fila de requisições pendentes antes de despachá-lo para o dispositivo físico subjacente (Dunn 2010). O NOOP é baseado no FCFS, no qual todas as requisições são inseridas em uma mesma fila e são executadas na ordem de chegada podendo neste caso, apenas agrupar os pedidos direcionados ao mesmo bloco (Maziero, 2014). Este escalonador é voltado para discos de estado sólido (SSD) ou sistemas de armazenamento que façam seu próprio escalonamento, como sistemas RAID⁷ (Maziero, 2014).

A Figura 8 mostra o NOOP em operação. Para isso, foi suposto que há seis requisições na fila pendente (três requisições de leitura (R0, R1, R2) e três requisições de gravação (W0, W1, W2), cada requisição acessa a dois chips de memória *flash*. Essas seis requisições são adicionadas à fila pendente na ordem FCFS, da esquerda para a direita. A arquitetura física do SSD é organizada com quatro chips de memória *flash* sendo um chip por canal. A localização dos dados das seis requisições é marcada nos quatro chips e os vetores de localização dessas requisições são produzidos.

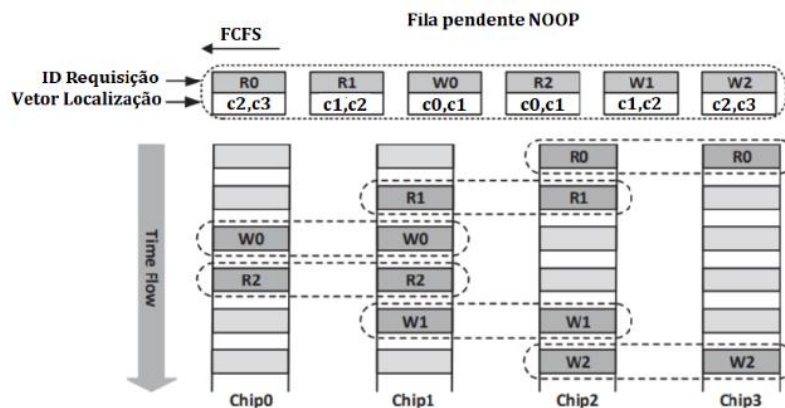


Figura 8: Algoritmo NOOP. Adaptado de (Gao *et al.*, 2014)

Nesse caso, essas seis requisições são processadas sequencialmente pois, as requisições são planejadas na ordem de FCFS, o que requer três ciclos de leitura e três de gravação para atender a essas requisições, totalizando seis acessos ao disco.

⁷ Redundant Arrays of Inexpensive Disks (RAID) – Técnica de gerenciamento de discos que otimizavam as operações de E/S e implementavam redundância e proteção de dados (Machado, 2007).

3.6. DeadLine

Devido à simplicidade do algoritmo de escalonamento NOOP, nada pode ser feito para garantir o tempo de serviço para as requisições que chegam a fila de espera. O escalonador DeadLine, também voltado para SSDs, tem como objetivo garantir que todas as requisições sejam atendidas dentro de um período de tempo especificado.

Para isso, esse algoritmo adiciona dois recursos: (1) listas ordenadas na tentativa de minimizar tempos de busca; e (2) um tempo de expiração (*deadline*) em cada requisição a ser atendida durante a fase de enfileiramento para tentar garantir um tempo de serviço inicial para as requisições (Jacobsen, 2007). A noção de um prazo de expiração por requisição também é usada para atribuir uma preferência maior para operações de leitura a escrita. Conforme Jacobsen (2007), o tempo de expiração para operações de leitura e escrita são diferentes.

O algoritmo DeadLine opera com cinco estruturas. Há uma fila para requisições de leitura e outra para as requisições de gravação (as requisições presentes nessas listas são classificadas por endereços de blocos lógicos - lista ordenada). Além disso, o escalonador mantém mais duas listas, cada uma classificada pelo fator tempo de expiração (lista FCFS). Por último, há uma lista para envio das requisições ao *driver* do dispositivo de armazenamento (Pratt and Heger 2004).

De acordo com Pratt and Heger, (2004), inicialmente o algoritmo verifica se há requisições pendentes na lista de envio. Se sim, essas requisições são atendidas. Do contrário, conjuntos de requisições são movidas de uma das outras 4 listas (ordenada ou FCFS) para fila de envio, o que ocorrer primeiro: (a) se houver requisições de escrita na lista ordenada e, esta não foi selecionada por um determinado período de tempo; (b) se na fila de leitura FCFS há requisições com prazo vencido; (c) se na fila de leitura ordenada há requisições pendentes; (d) se há, em qualquer lista, requisições de escrita pendentes.

3.7. PIQ (*Parallel Issue Queuing*)

PIQ (Gao *et al.*, 2014) é um algoritmo que visa melhorar o desempenho da leitura e gravação, sem alteração do design da arquitetura física do disco SSD. Para isso, busca minimizar os conflitos entre requisições.

O conflito de requisição em um SSD é altamente correlacionado o tempo em que as requisições são geradas e com sua localização que é baseada no endereço físico do disco. Se o intervalo de tempo entre as requisições for pequeno o suficiente e as requisições de acesso ao disco estarem destinadas ao mesmo chip de memória *flash*, há uma grande probabilidade dessas requisições de estarem em conflito.

PIQ baseia-se em uma abordagem de detecção de conflitos por meio do número de bloco lógico (LBN). LBN representa o endereço lógico dos dados requisitados pelos sistemas de *hosts*, que é usado na determinação da localização dos dados.

De acordo com os tipos de requisições, os conflitos de acesso são classificados em três tipos: conflitos de leitura e gravação (RWC), conflitos de leitura e leitura (RRC) e conflitos de gravação e gravação (WWC). Baseado nessa classificação, o esquema de separação por tipo de conflito pode se dar em três etapas:

1. *Read and Write Separation* (RWS): As requisições de leitura e gravação são separadas em pacotes diferentes para evitar o RWC.
2. *Read and Read Separation* (RRS): As requisições são separadas em pacotes diferentes para evitar o RRC entre as requisições de leitura.
3. *Write and Write Separation* (WWS): As requisições são separadas em pacotes diferentes para evitar o WWC entre as requisições de gravação.

Com base nos tipos de requisições e na classificação dos conflitos, o algoritmo PIQ é organizado em três estruturas, sendo a primeira uma fila de requisições FCFS, a segunda estrutura uma lista de separação de leitura e escrita e a terceira a criação de pacotes de requisições sem conflitos, a qual é inserida na lista pacote de requisição. Tal estrutura pode ser visualizada na Figura 9.

A Figura 9 mostra o PIQ em operação e para isso, foi suposto que há seis requisições na fila FCFS (três requisições de leitura (R0, R1, R2) e três requisições de gravação (W0, W1, W2), cada requisição acessa a dois chips de memória *flash* na qual, pode ser observado através do vetor de localização exemplo: a requisição R0 possui vetor de localização (c2 e c3) sendo esses, referências lógicas ao chip 2 e chip 3 na arquitetura física do disco sólido. A detecção dos conflitos ocorre conforme as seguintes etapas. Na primeira etapa, o RWS, é aplicado à fila de requisições FCFS onde as requisições de leitura e gravação são separadas, para evitar o RWC e, em uma segunda etapa, são aplicados o

WWS e o RRS onde são formados pacotes de requisições sem conflitos evitando assim, o RRC e o WWC. Por fim, o PIQ envia os pacotes de requisições sem conflitos a uma lista de pacote de requisição, onde todos os pacotes são processados na ordem de criação.

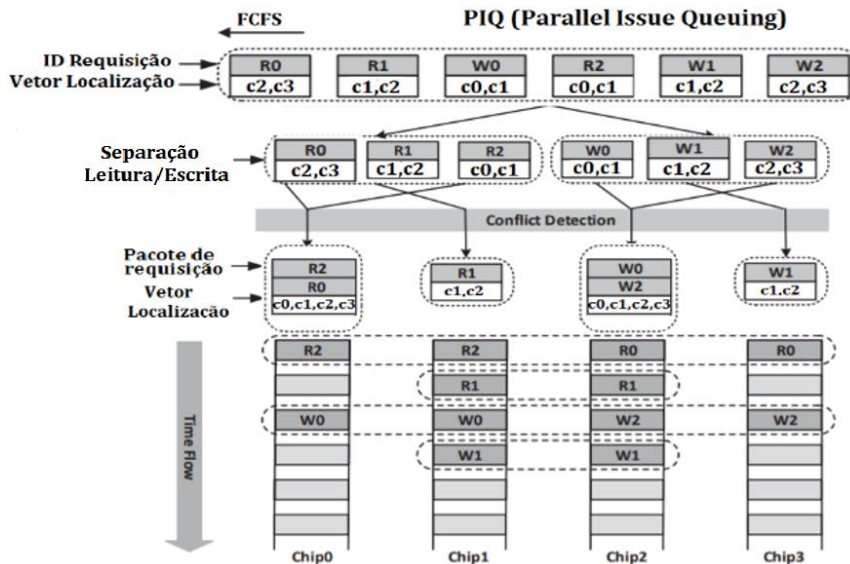


Figura 9: Algoritmo PIQ. Adaptado de (Gao *et al.*, 2014)

Neste caso, as requisições R1 e R2, conforme vetor de localização, estão em conflito no chip 1 já que, ambas acessam o mesmo chip e, as requisições R0 e R1 estão em conflito no chip 2. Portanto, entre as requisições de leitura, os pacotes de requisições sem conflito podem ser formados apenas entre as requisições R2 e R0 pois, entre elas não há conflito de acesso ao mesmo chip. Já entre as requisições de gravação há conflitos entre as requisições W0 e W1 onde ambas acessam o chip 1 assim como, entre as requisições W1 e W2 que acessam o chip 2, portanto, pacotes de requisições sem conflitos podem ser formados apenas entre as requisições W0 e W2.

Com o algoritmo PIQ pode ser observado que apenas dois ciclos de leitura e dois de gravação são necessários para atender a essas requisições.

3.8. Considerações finais do capítulo

A disciplina de SO envolve conteúdos abstratos, como os algoritmos de escalonamento de discos. Na maioria dos casos, essa disciplina é abordada de maneira teórica por meio de recursos como quadro negro e projetor multimídia, dificultando a

capacidade de entendimento dos alunos. Neste sentido, o uso de ferramentas tecnológica, para apoiar o ensino, pode facilitar a aprendizagem.

A RV é uma interface que permite a construção de situações de ensino, pois permite a simulação de cenários de exploração imersiva com interação. Em seu conjunto, RV configura-se como uma interface avançada de última geração para aplicações computacionais, a qual permite a interação do usuário em tempo real utilizando dispositivos multisensoriais a partir de um ambiente tridimensional (Tori *et al.*, 2006). Neste sentido, há várias razões para se investigar o uso da RV no contexto abstrato como o de SO, dentre elas: (i) possibilidade de transformar o abstrato em concreto; (ii) viabilidade de fazer em vez de apenas observar; (iii) capacidade do inviável ou impossível se tornar prático; (iv) oportunidade de explorar cenários simulados; (v) e capacidade de ir além da realidade (Slater, 2017).

4. REALIDADE VIRTUAL (RV)

Representação de objetos reais ou abstratos sempre fez parte da vida dos seres humanos e, por meio da RV, tais objetos passaram a ser projetados e simulados em ambientes imersivos, rompendo assim, os limites físicos existentes. Este capítulo apresenta os principais conceitos, definições e classificações das diferentes tecnologias de RV.

4.1. Introdução

O termo RV foi forjado pelo cientista da computação e artista Jaron Lanier por conseguir unir dois conceitos adversos, real e virtual, em um novo conceito que melhor retrata a essência dessa tecnologia (Tori *et al.*, 2006).

A definição desta tecnologia é bastante abrangente. Conforme Tori *et al.* (2006), a RV pode ser considerada uma “interface avançada do usuário”, tendo como características não só a visualização e movimentação em ambientes tridimensionais, como também a interação com elementos desse ambiente em tempo real. O enriquecimento da experiência de interação do usuário com o uso de RV pode ser proporcionada através da estimulação de todos os sentidos humanos.

Para Cardoso *et al.* (2007), a RV pode ser visualizada como um sistema computacional que permite ao usuário criar ambientes artificiais, nos quais este não só se sente envolvido como também possibilitado a interagir, navegar e imergir no ambiente 3D, gerado pelo computador por meio de canais multissensoriais. Outras definições podem ser encontradas em Steuer (1992) e Liu *et al.* (2017).

Um dos primeiros projetos a dar início às ideias que levaram ao desenvolvimento da RV foi o projeto SENSORAMA, proposto pelo cineasta Morton Heilig na década de 1950, o qual projetou um sistema imersivo onde o usuário era submetido a diversas sensações, movimentos, sons, ventos, odores e visão estereoscópica (Tori *et al.*, 2006). Em seguida, RV apareceu com os simuladores de voo da Força Aérea Americana nos anos 60 ou 70, os quais eram utilizados em treinamentos militares (Liu *et al.*, 2017).

A RV sempre envolveu altos custos devido ao uso de equipamentos e *software* específicos. Por isso, era exclusiva de instituições de pesquisa e/ou governamentais.

Contudo, o avanço tecnológico da última década tornou essa tecnologia acessível, principalmente devido o barateamento dos dispositivos e da evolução dos *software* (Liu *et al.*, 2017).

Como exemplo de popularização pode citar o *Pokemon GO*, que teve 100 milhões de downloads, desde o seu lançamento em 6 de julho de 2016 até dezembro de 2016, e um faturamento de mais de US\$ 10 milhões por dia no *IOS* e no *Google Play* combinados (Liu *et al.*, 2017).

A RV permite a interação do usuário diretamente com objetos em um ambiente 3D e, com o uso de dispositivos especiais, é possível no ambiente virtual ampliar em tempo e espaço a intensidade dos sentidos do usuário, como ouvir e sentir ou viajar muito além das capacidades humanas (Tori *et al.*, 2006). A interação juntamente com estímulos provenientes de dispositivos externos, como HMD (*Head mounted display*), pode criar nos participantes uma sensação de imersão sensorial e aprofundar o efeito da imersão psicológica⁸, além de induzir presença virtual (ilusão de lugar) (Liu *et al.*, 2017).

Para prover senso de presença, permitir a interação e tornar o ambiente virtual o mais real possível, um sistema RV inclui tanto dispositivos de *hardware* quanto de *software*. De modo geral, os dispositivos (*hardware*) utilizados em um sistema de RV podem ser classificados como dispositivos de entrada (*joysticks*, teclado, câmeras etc.), os quais permitem o envio da ação do usuário, e dispositivos de saída (incluindo vídeo, áudio, retorno tátil etc.) que têm como função enviar as informações aos usuários. Já os componentes de *software* têm como função identificar a informação de um dispositivo de entrada ou representar a informação na saída do sistema, de forma que possa ser reconhecida pelo usuário ou interpretada por dispositivos de saída (LaViola Jr *et al.*, 2017).

⁸ A imersão psicológica é o estado mental de ser completamente absorvido ou engajado com alguma coisa (Liu *et al.*, 2017).

4.2. Dispositivos de Entrada

Os dispositivos de entrada como teclado, mouse, rastreadores, luvas e trajes têm como papel enviar as ações do usuário em um ambiente 3D ao sistema, permitindo assim a interação com os objetos do mundo virtual. Os mais comuns são o teclado e o mouse, que tem como vantagem serem dispositivos conhecidos pelos usuários e fáceis de serem encontrados, porém, limita o usuário a uma mesa fixa e restringe a manipulação no ambiente virtual (Gnecco *et. al*, 2007). Outro dispositivo comum são os *joysticks*, que são geralmente utilizados em simuladores de voo e em jogos.

Já os rastreadores, são dispositivos utilizados para mapear o movimento que o usuário faz durante a interação com o ambiente virtual a partir de um ponto ou estado de referência (Siscoutto and Costa, 2008; Bogoni *et. al*, 2015), assim como determinar a posição do usuário.

4.2.1. Rastreadores de Posição

Os rastreadores de posição são construídos utilizando diversas tecnologias, como exemplo, os rastreadores mecânicos: que utilizam dispositivos mecânicos conectado ao corpo do usuário, os quais permitem rastrear com precisão a posição do objeto rastreado ou, eletromagnéticos: que são rastreadores que possuem bobinas que geram ondas eletromagnéticas que definem o ponto de referência de todo o sistema de coordenadas dos sensores monitorados dentro do mundo real.

A escolha do melhor rastreador a ser utilizado depende de diversos fatores, como a precisão do rastreamento que se deseja, tipo de ambiente que será monitorado como também, a conexão entre o sensor e computador (Bogoni *et. al*, 2015). Para que se possa rastrear a posição exata do objeto, a navegação no mundo virtual deve ser similar à navegação no mundo real que se baseia em um espaço tridimensional para isso, é necessário conhecer 3 coordenadas (x,y,z) relacionadas à orientação do objeto no espaço e 3 coordenadas (x,y,z) referentes a sua orientação. Tais coordenadas são conhecidas como Grau de Liberdade (*Degrees of Freedom* - DOF). Portanto, para que um objeto seja rastreado em um ambiente 3D é necessário que os rastreadores possuam 6 DOF (Rolland *et al.*, 2001). Os rastreadores permitem ao usuário navegar seis graus de liberdade, o que implica na possibilidade de três rotações e três translações conforme Figura 10.

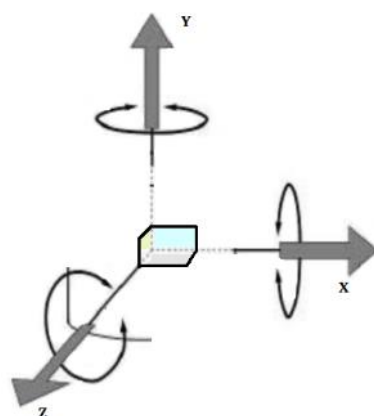


Figura 10: Grau de liberdade. Adaptado de (Siscoutto e Costa, 2008)

Rastreadores estão presentes na maioria dos dispositivos de interação com o ambiente virtual entre eles a luva de dados e os dispositivos de entrada biológicos.

4.2.1.1. Luvas de Dados

Por meio das luvas de dados, é possível identificar os movimentos dos dedos da mão do usuário como arrastar, impulsionar, manipular, selecionar e segurar objetos dentro do ambiente virtual. Para isso, sensores são utilizados na luva de dados com o intuito de rastrear a localização e orientação da mão do usuário no espaço. Tais sensores podem ser mecânicos ou de fibra óptica. A Figura 11 ilustra uma luva de dados.

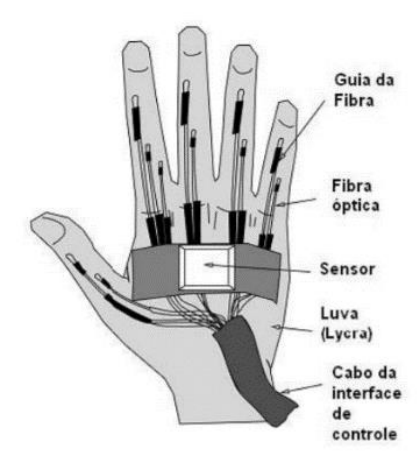


Figura 11: Luva de dados (Fonte: Siscoutto e Costa, 2008)

4.2.1.2. Dispositivo de Entrada Biológico

Os dispositivos de entrada biológicos possuem sensores capazes de reconhecer atividades indiretas, como comando de voz e sinais elétricos musculares. O comando de

voz pode ser utilizado no ambiente virtual quando o uso da mão está determinado a outras atividades ou não pode ser utilizada. Já os sinais elétricos musculares utilizam sensores colocados na pele no qual, é utilizado para detectar a atividade muscular, permitindo ao usuário movimentar-se pelo ambiente virtual (Siscoutto e Costa 2008).

Outros dispositivos encontrados são os rastreadores dos olhos do usuário, os quais permitem, por meio de câmeras posicionadas na frente do usuário, identificar a direção para a qual o usuário está olhando ou, dispositivos de rastreamento da cabeça do usuário, que determina a direção de sua cabeça. A escolha por um determinado tipo de rastreador depende diretamente do objetivo do sistema em RV utilizado.

4.3. Dispositivos de Saída

Os dispositivos de saída são responsáveis pelo envio de informações aos usuários e, por meio de dispositivos específicos, exploram seus sentidos como visão, audição e tato (Siscoutto and Costa 2008). Como dispositivos de saída citam-se os dispositivos visuais que são de extrema importância para se obter imersão em um ambiente virtual já que a imagem gerada por eles influencia na sensação de imersão do usuário em um sistema de RV. Exemplos de dispositivos visuais são os vídeo-capacetes também conhecidos como HMD, sistemas de projeção, entre outros.

4.3.1. *Head-Mounted Display*

Os HMDs são dispositivos visuais colocados na cabeça do usuário ou como parte de um capacete que possui display próximo aos seus olhos. Desta forma, os dispositivos HMDs isolam os usuários do mundo real ao projetar imagens que dão a percepção de estar totalmente imerso no ambiente virtual, já que estes não têm visão direta do mundo real (Burdea e Coiffet, 2003). A Figura 12 exhibe um dispositivo HMD desenvolvido pela empresa *Oculus*⁹.

⁹ Oculus VR é uma empresa americana de tecnologia especializada em produtos de *hardwares* e *software* de realidade Virtual, fundada por Palmer Luckey, Brendan Iribe, Michael Antonov e Nate Mitchell em 2012.



Figura 12: Exemplo de HMD desenvolvido pela empresa Oculus (Fonte: <https://www.oculus.com>)

Estes dispositivos apresentam imagens estereoscópicas e muitas vezes possuem rastreadores de posição que identificam para qual direção o usuário está olhando.

A estereoscopia pode ser considerada um processo de reconstrução em uma única representação de duas imagens simultâneas visualizadas a partir de pontos de observações diferentes, o que permite a visualização em um ambiente 3D com um mais alto grau de realismo, fornecendo ao usuário sensação de profundidade espacial (Burdea *et al.*, 2003).

Atualmente é possível encontrar adaptadores, também chamados de VR Mobile Phone 3d Glasses, que transformam *smartphones* em HMDs. Estes dispositivos possuem duas lentes próximas aos olhos do usuário e por meio de suportes específicos é possível encaixar o celular nos HMDs. Isso permite ao usuário visualizar as imagens do ambiente virtual a partir tela do celular. Como exemplo desses dispositivos citam-se o *Google Cardboard* e o Gear VR.

4.3.2. Ambientes de Projeções

Em ambientes de projeções, o uso de monitores convencionais ou projetores multimídia podem ser utilizados. Neste caso, permitem a visualização do ambiente virtual de forma monoscópica ou estereoscópica (Burdea *et al.*, 2013).

Como exemplo de sistema de multiprojeção cita-se a CAVERNAsTM Digitais¹⁰ (CAVETM – CAVE Automatic Virtual Environment), que é um sistema de multiprojeção

¹⁰ CavernaTM Digital é a denominação dada ao primeiro sistema integrado de multiprojeção estereoscópica totalmente imersivo e interativo desenvolvido na América Latina. Esse sistema foi construído

estereoscópico composto por 3 ou mais telas de projeção (Tori *et al.*, 2017), com formato de um cubo, onde imagens são projetadas em cada uma das paredes promovendo ao usuário a percepção de estar totalmente inseridos (imersos) em um ambiente virtual gerado por computador. Para isso, o usuário deve usar óculos de estereoscopia ativa.

No ambiente de uma CAVE é possível ter mais de um usuário compartilhando a mesma experiência no ambiente virtual, porém, apenas um usuário pode realizar ações dentro do ambiente, enquanto os demais apenas visualizam em tempo real (Tori *et al.*, 2017). A Figura 13 apresenta um ambiente CAVE.



Figura 13: Imagem de uma caverna e seus usuários (Fonte: “Adaptada” de Tori *et al.*, 2017)

Para que os dados enviados por meio dos dispositivos de entrada possam ser visualizados ou reconhecidos nos dispositivos de saída, dispositivos de processamento se fazem necessários.

4.4. Dispositivos de Processamento

Os computadores usados para suportar sistemas de RV variam de estações de trabalho com múltiplos processadores ou mesmo redes de computadores trabalhando como *clusters*¹¹, até dispositivos móveis e computadores pessoais, equipados com placas

pelos pesquisadores do Laboratório de Sistemas Integráveis da Escola Politécnica da Universidade de São Paulo – Butantã (Zuffo *et al.*, 2001).

¹¹ *Cluster* - Conjunto de estação de trabalho ou PCs semelhantes, conectados por meio de uma rede de alta velocidade (Tanenbaum *et al.*, 2007).

gráficas adequadas (Tori, 2017). Na prática, os dispositivos de processamento devem codificar os sinais de entrada e saída em tempo real com a menor degradação possível.

Conforme Gnecco *et al.*, (2007), utilizou-se por muito tempo o uso de computadores fortemente acoplados¹² para controlar sistemas de RV. Estes computadores são montados com *hardwares* especializados com o objetivo de lidar com computação gráfica. Um exemplo de ambiente de utilização dos computadores fortemente acoplados são os de sistemas de multiprojção, como as CAVERNAs Digitais, porém são de alto custo, o que os limita a grandes centros de pesquisas (Guimarães 2004). Em substituição aos sistemas fortemente acoplados o uso de aglomerados de computadores pessoais (PC's), de baixo custo, podem ser adotados (sistemas fracamente acoplados / *cluster*) (Guimarães, 2004).

Com a evolução do poder de processamento tanto dos dispositivos móveis, como *smartphone*, quanto dos computadores pessoais (PC) e, somado ao baixo custo, permitiu que tais dispositivos, de forma individual, pudessem ser utilizados para o processamento de boa parte das aplicações de RV. Neste caso, os processadores e as placas gráficas devem ser eficazes a ponto de tratarem as necessidades do ambiente de RV, garantindo a execução em tempo real das seguintes atividades: renderização de imagens de alta definição; saída de áudio; varredura de dispositivos de entrada com ênfase no rastreamento; incorporação de som, entre outras.

O uso de *smartphones* tem suas vantagens, como mobilidade, facilitar a implantação em situações novas e, muitos já possuem componentes necessários para uma RV imersiva: uma GPU móvel capaz de renderizar imagens de alta definição e, saída de áudio (Steed, 2013), porém ainda a fidelidade gráfica é menor do que a que pode ser produzida usando o poder computacional de um PC, e a falta de hardware de resfriamento eficiente na maioria dos dispositivos significa que eles se aquecem rapidamente com qualquer aplicativo que exija exibição contínua de gráficos (Powell *et al.*, 2016).

Para que os *smartphone* sejam capazes de exibir imagens em ambientes virtuais, o uso de dispositivos especializados deve ser adotado, como Google Cardboard e o Gear

¹² Nos sistemas fortemente acoplados, existem vários processadores, compartilhando uma única memória física e dispositivos de entrada e saída controlados por apenas um sistema operacional (Machado, 2007).

VR. Assim, é possível obter visão estereoscópica e permitir imersão no ambiente já que o dispositivo exibe separadamente as imagens de cada olho.

Além de um hardware eficiente, ferramentas de desenvolvimentos e de modelagem são de extrema importância para a construção de um ambiente de RV.

4.5. Ferramentas de Desenvolvimento

Conforme Guimarães *et al.* (2007), as ferramentas de desenvolvimento de ambiente virtual devem atender aos requisitos básicos de software como: usabilidade, suporte a novas tecnologias, desempenho, manutenibilidade, tolerância a falhas, portabilidade e reusabilidade. Assim como fornecer recursos específicos para a área de RV, como suporte a dispositivos de entrada e geração de imagem em tempo real. Além disso, a criação de um ambiente virtual requer também a modelagem da cena: ambiente virtual, os personagens, objetos, etc. Para o desenvolvimento da aplicação e modelagem, utilizam-se programas capazes de manipular a geometria, texturas e preparar animações dos objetos (Gnecco *et al.*, 2007). Para isso, ferramentas específicas devem ser utilizadas como: Panda3D, Unity, Blender, 3DMax entre outras.

- Panda3D: Panda3D (Panda, 2018) é um motor de jogo, um framework para renderização 3D e desenvolvimento de jogos. Foi desenvolvida pela Disney e pela Universidade Carnegie Mellon. Para desenvolvimento de aplicações com o Panda3D utiliza-se a linguagem de desenvolvimento Python ou C++. Com o uso da ferramenta é possível navegar em cenas 3D, monitorar o desempenho, utilizar saída estereoscópica, utilizar dispositivos de interação, sensores de posição e outros. O Panda3D é de código aberto e gratuito para qualquer finalidade, incluindo empreendimentos comerciais;
- Unity 3D: Unity 3D (Unity, 2018) é uma ferramenta de desenvolvimento usada para criar jogos para plataformas móveis, *desktop*, dispositivos móveis, aplicativos da *Web* e consoles, e que permite visualização 3D de conteúdo interativo em conjunto com a linguagem de programação C# para edição de *script*. O Unity permite a integração de pacotes adicionais como

exemplo o *Utilities for Unity 1.27*, que inclui *scripts*, *prefabs*¹³ e outros recursos. O pacote inclui uma interface para controlar o comportamento da câmera RV, uma API (*Application Programming Interface*)¹⁴ de entrada unificada para controladores, recursos avançados de renderização, captura de objeto e *scripts* hápticos para toque, ferramentas de depuração;

- Blender: é uma ferramenta de modelagem e animação 3D (Blender, 2018), que está sob a licença GNU-GPL¹⁵. Suporta uma gama de recursos e funcionalidades para criação de objetos 3D - renderização de cenas, modelagem, manipulação, animação, simulação, composição e rastreamento de movimento, edição de vídeo, entre outros. Permite aos usuários usarem uma API para o desenvolvimento de *scripts* em Python e, está disponível para várias plataformas, como *Microsoft Windows*, GNU/Linux, entre outras;
- 3D Max: é uma ferramenta de modelagem, animação e renderização 3D (3D Max, 2018). Foi desenvolvido pela Autodesk Media and Entertainment. A ferramenta permite além do desenvolvimento de jogos a aplicação de efeitos e pré-visualização de filmes. Uma linguagem de *script* que é incorporada pode ser usada para automatizar tarefas repetitivas ou para proporcionar animações aos personagens. Além disso, a ferramenta permite realizar operações nos objetos como modificar a textura, alterar o ângulo dos objetos, rotação, entre outras.

4.6. Considerações finais do capítulo

A constante evolução da área de RV vem proporcionando melhorias na qualidade e capacidade computacional (*hardware*), assim como nas ferramentas utilizadas para o desenvolvimento e modelagem do ambiente (*software*). Tal evolução somada a redução dos custos, vem transformando a tecnologia de RV em um instrumento poderoso para as múltiplas situações e contextos. Neste caso, cita-se o ambiente de pesquisa e

¹³ O *prefab* atua como um modelo a partir do qual você pode criar novas instâncias de objeto na cena (Unity, 2018)

¹⁴ API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de *software* ou plataforma baseado na *Web*.

¹⁵ A Licença Pública Geral GNU é uma licença *copyleft* gratuita para *software* e outros tipos de trabalhos.

aprendizagem, pois pode permitir ao professor explorar novas formas de ensino, com imersão, interação, senso de presença, entre outros, através de interface avançada, transformando assim a educação em um processo dinâmico e criativo.

5. TRABALHOS RELACIONADOS

Neste capítulo são apresentados simuladores de SO relacionados a esta pesquisa. Foram objetos de estudo sete simuladores que permitem a representação dos algoritmos de escalonamento de discos voltados para discos magnéticos, assim como os de estado sólido.

5.1. Simuladores de disco

O simulador descrito por Souza (2018) foi desenvolvido para suportar os algoritmos FCFS, SSTF e o SCAN. Esse simulador é 2D e executado por meio de navegadores Web. No simulador é possível determinar a posição inicial do cabeçote de leitura, tempo de *clock*, tempo de movimentação do cabeçote de leitura e o tempo de leitura das requisições. A ordem da sequência de requisições pode ser configurada pelo usuário, a qual é simulada por meio de gráficos dinâmicos. Destaca-se nesse simulador a facilidade de uso e compreensão. O simulador também apresenta estatísticas conforme os parâmetros configurados. O simulador permite ao usuário o entendimento apenas nos algoritmos de escalonamento, não dando ênfase à arquitetura interna do disco. A Figura 14 exibe a tela inicial do simulador.

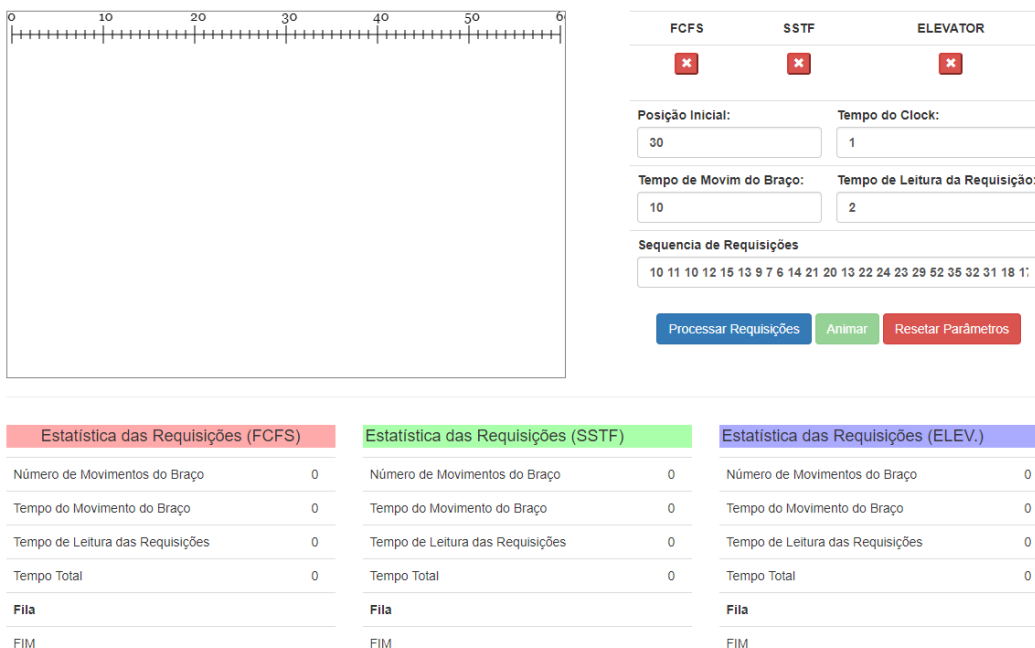


Figura 14: Simulador Web (Fonte: Souza, 2018)

Outro simulador é o Track Animation (Zanduetta, 2013), que é uma aplicação 2D desenvolvida para ser executada em um sistema operacional Microsoft Windows. Ela simula os algoritmos FCFS, SJF (*Shortest Job First*), SCAN, LOOK, CSCAN e CLOOK. A aplicação permite ao usuário visualizar o tempo total de busca, gerar uma lista aleatória de números de trilhas e reproduzir e pausar a simulação, determinando até cinco níveis de velocidade de simulação.

Através deste simulador é possível ter uma visão sobre a arquitetura do disco magnético, trilhas, atuador, cabeça de leitura, placa lógica, entre outros porém, o simulador não apresenta as características do disco de estado sólido como também, seus algoritmos. A Figura 15 mostra a interface do simulador.

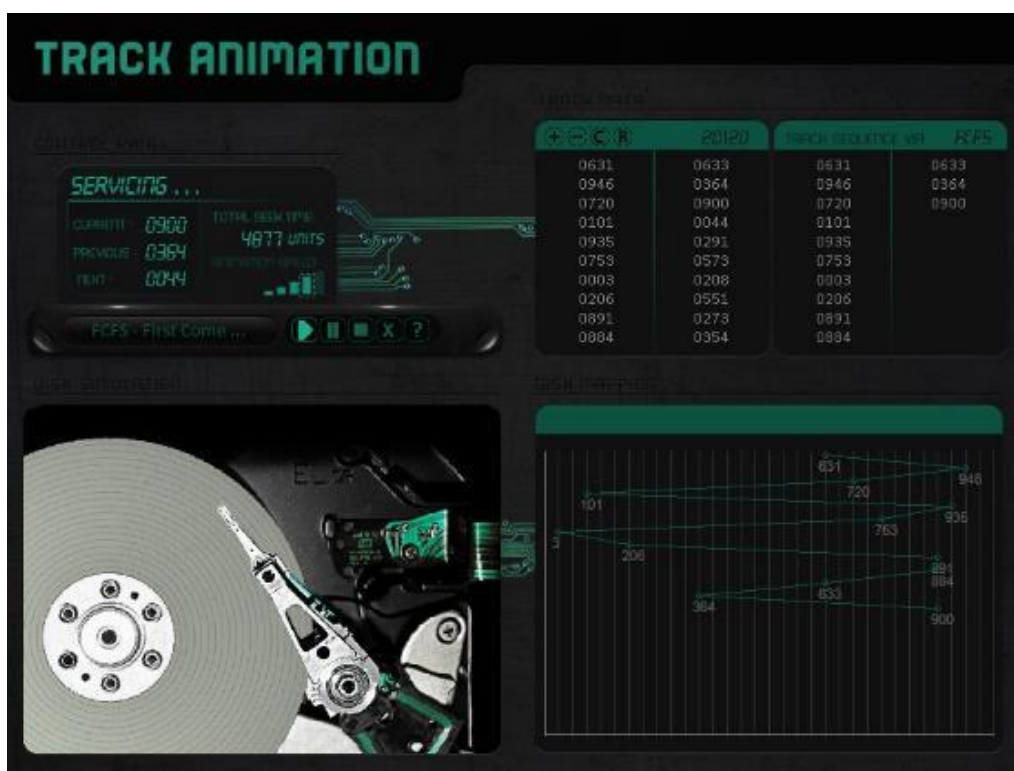


Figura 15: Simulador Track Animation (Fonte: Zanduetta 2013)

Suranauwarat (2017) apresentou um simulador 2D sobre os algoritmos de escalonamento de discos voltado para o ambiente educacional, o qual foi escrito utilizando a linguagem de desenvolvimento C#. Tal simulador permite a inserção de novas requisições de forma dinâmica, ou seja, permite que novas requisições sejam inseridas enquanto outras requisições estão sendo processadas. Além disso, permite ao usuário

comparar os resultados da simulação e as estatísticas de desempenho de diferentes algoritmos de escalonamento de disco.

O usuário pode selecionar um algoritmo de seu interesse por meio de um menu suspenso na parte superior esquerda do simulador, como mostrado na Figura 16. O simulador emula os algoritmos FCFS, SSTF, SCAN entre outros. O modo de comparação permite ao usuário comparar o desempenho dos algoritmos em termos de tempo total de busca e o tempo médio de busca das requisições. O canto superior direito mostra os parâmetros de configuração que podem ser definidos pelos usuários, entre eles: posição inicial do cabeçote de leitura, número mínimo e máximo de cilindro, milissegundos/cilindro, velocidade da animação. É permitido ao usuário especificar um conjunto de números de cilindros a serem acessados na caixa de texto “*Disk Requests*” separando cada número do cilindro por uma vírgula ou espaço. Por exemplo, “1, 36, 16, 34, 9, 12”. Como alternativa, o usuário pode usar um conjunto aleatório de requisições clicando no botão “*Random*” e, em seguida, especificar o número total de solicitações de disco que o usuário deseja que o simulador gere. Tal visualização pode ser vista na Figura 16.

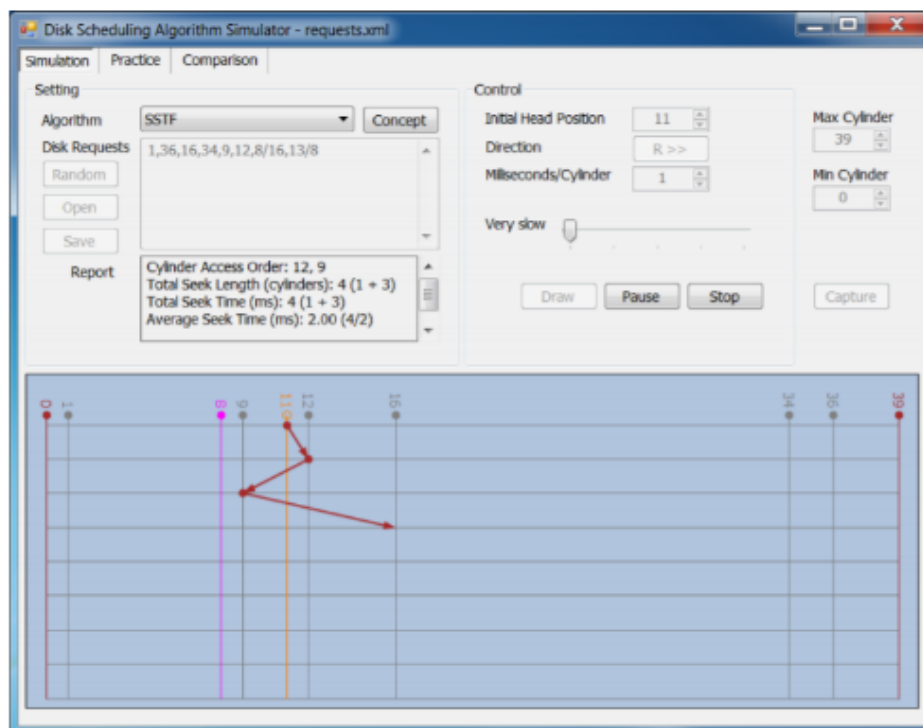


Figura 16: Simulador de algoritmos (Fonte: Suranauwarat, 2017)

Robbins (2004) apresentou um simulador 2D que pode ser executado como um *Applet Java*, seja remotamente, a partir de um navegador, ou localmente como um aplicativo *Java*. O simulador foi desenvolvido para a análise dos algoritmos FCFS, SSTF, SCAN, C-SCAN, LOOK e C-LOOK. O simulador permite que se crie um arquivo contendo a lista de requisições a serem acessadas como também a especificação do horário de chegada de cada requisição. Com o simulador é possível calcular estatísticas com base na distância percorrida do cabeçote de leitura e também calcular o tempo entre a chegada de uma requisição e o final da busca da requisição correspondente.

O ponto de destaque desse simulador é a comparação e análise de desempenho dos algoritmos onde os resultados da simulação podem ser exibidos graficamente ou em tabelas, permitindo assim, aos alunos a realização de experimentos entre os algoritmos de escalonamento de disco magnético.

A Figura 17 apresenta a comparação entre os algoritmos FCFS e SSTF. Para isso, foi assumido que a cabeçote de leitura encontrava-se no cilindro 53 e a fila de requisições pendentes contém as requisições 98, 183, 37, 122, 14, 124, 65 e 67. A Figura 17, na imagem (A) cada linha horizontal representa uma varredura do disco em uma direção, com marcas de tiques nas posições dos blocos acessados e a imagem (B) mostra as estatísticas calculadas para essas duas execuções.

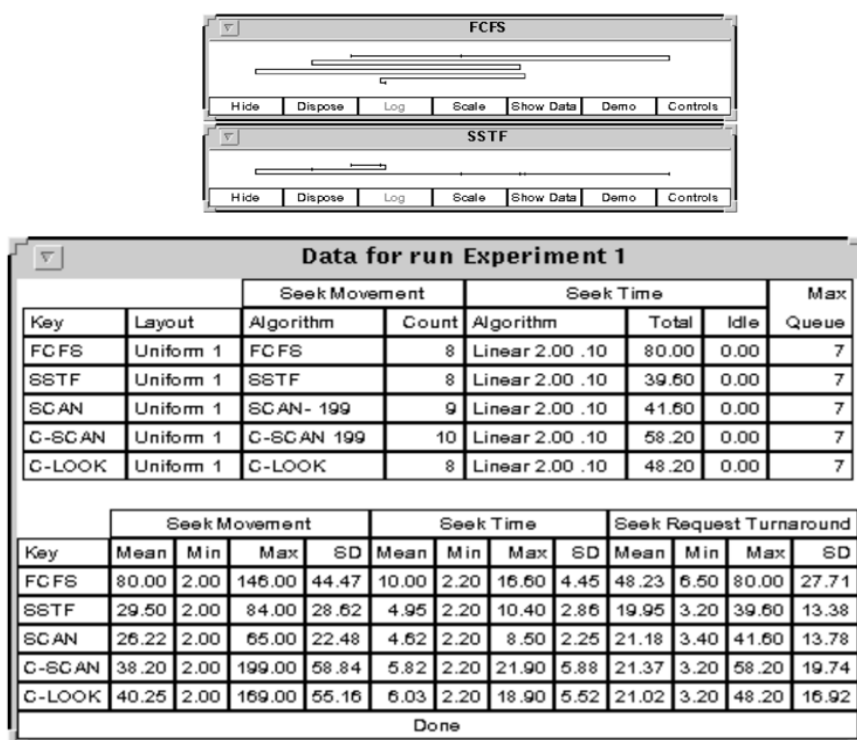


Figura 17: Simulador de algoritmos (Fonte: Robbins, 2004)

O simulador baseado em linhas de comando DiskSim (Bucy *et al.*, 2003) foi desenvolvido na linguagem C. Com o simulador é possível analisar o desempenho de subsistema de armazenamento, compreender como o desempenho do sistema de armazenamento se relaciona com o desempenho geral do sistema computacional e avaliar novas arquiteturas de armazenamento de disco. Ele inclui módulos que simulam discos, controladores intermediários, barramentos, drivers de dispositivo, escalonadores de requisições de disco, caches de blocos de disco e *array* de disco. O simulador por si só, apenas simula e relata os aspectos relacionados ao desempenho do subsistema de armazenamento.

O simulador DiskSim na sua versão original não conta com suporte à análise de disco de estado sólido, isso foi possível por meio uma extensão desenvolvida pela Microsoft Research que permitiu o uso do simulador para análise de estatística também para disco de estado sólido – SSD (Agrawal *et al.*, 2008).

O FlashSim (Kim *et al.*, 2009) foi escrito para ser interoperável com o simulador DiskSim. Ele realiza a análise de desempenho dos algoritmos e foi desenvolvido utilizando a linguagem de programação C++. Com o simulador é possível se concentrar

nas características de *software* do SSD (por exemplo, esquemas de FTL) como também nas características de *hardware* (por exemplo, pacotes, matrizes, planos, blocos). É possível usar o simulador para comparar o desempenho de dispositivos SSD que empregam diferentes esquemas de camada de tradução *flash* (FTL).

EagleTree (Dayan *et al.*, 2013) é um simulador 2D que permite não apenas a simulação das características físicas de um SSD, mas também (um subconjunto) do SO e aplicativos que o utilizam. Ele é composto de quatro camadas: a camada de *hardware*, o controlador SSD, SO e as camadas de aplicativo. Cada camada consiste de alguns parâmetros que podem ser configuráveis. Com o simulador é possível analisar o desempenho do disco simulando os algoritmos NOOP e CFQ (*Completely Fair Queuing*).

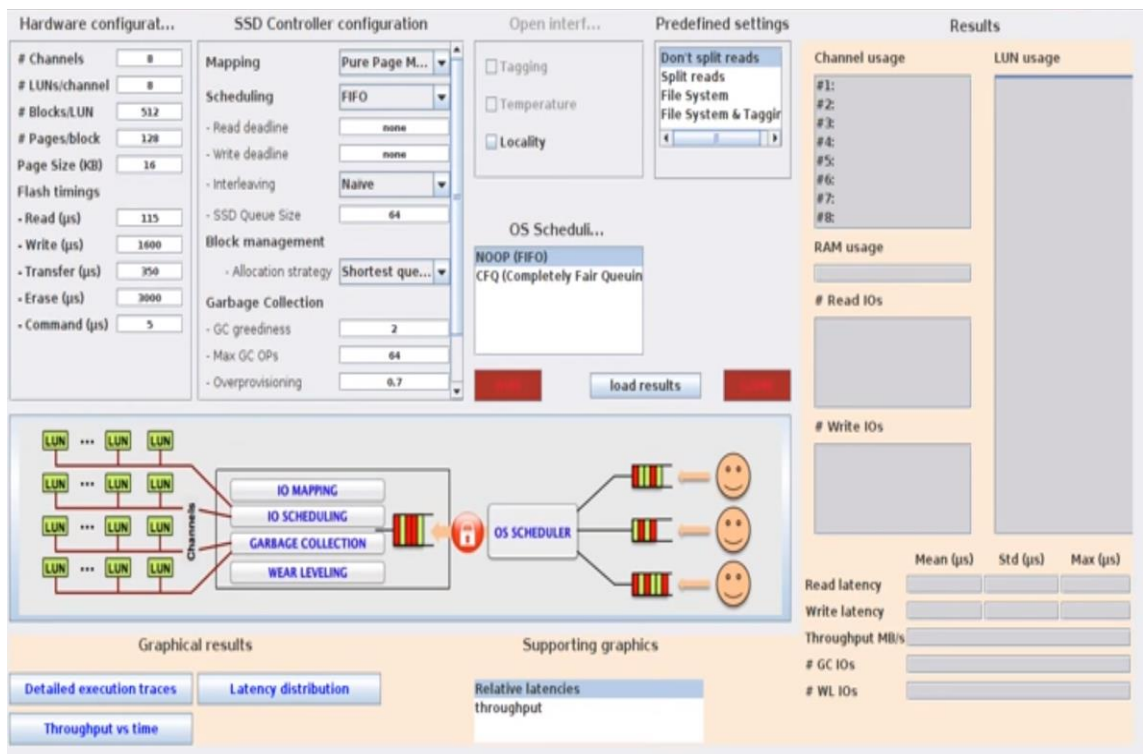


Figura 18: Simulador EagleTree (Fonte: Dayan *et al.*, 2013)

O EagleTree consiste em parâmetros configuráveis e de políticas personalizáveis levando em consideração:

1. *Hardware*: permite ao usuário configurar parâmetros referentes à arquitetura física do SSD como: números de canais, números de páginas, tempo de uma operação de leitura ou escrita e também simular transações básicas como: enviar um comando, transferir dados em um canal, ler, escrever ou apagar);

2. *SSD Controller Configuration*: o controlador SSD permite configurar o tipo de mapeamento de página, a coleta de lixo, os módulos de nivelamento de desgaste, o *deadline* das requisições de leitura e gravação, etc;
3. *OS Scheduling*: o escalonador de SO permite configurar o tipo de algoritmo de escalonamento de disco utilizado. De acordo com o tipo de algoritmo configurado é possível gerenciar as requisições simuladas por múltiplos threads concorrentes. Para isso, o simulador mantém um conjunto de requisições de cada threads e decide, baseado na política de escalonamento, quais operações devem ser enviadas ao SSD.

A Tabela 1 mostra qual(is) das técnicas listadas é(são) abordada(s) nos *software* já implementados e citados anteriormente, além de informar se a interface utilizada é 3D ou não.

Tabela 1: Tabela comparativa dos simuladores analisados

Simuladores	Interface 3D	Interface 2D	Interface Web	Linhas de comando	Algoritmos	Arquitetura física do disco sólido	Arquitetura física do disco magnético
Simulador WEB – (Souza, 2018)	Não	Sim	Sim	Não	FCFS, SSTF e SCAN	Não	Não
Track Animation- (Zanduetta, 2013)	Não	Sim	Não	Não	FCFS, SJF, SCAN, LOOK, CSCAN e CLOOK	Não	Sim
Disk Scheduling Algorithm Simulator – (Suranauwarat, 2017)	Não	Sim	Não	Não	FCFS, SSTF e SCAN	Não	Não
Simulador de Algoritmos – (Robbins, 2004)	Não	Sim	Não	Não	FCFS, SSTF, SCAN, C-SCAN, LOOK e C-LOOK	Não	Não
DiskSim - (Bucy <i>et al.</i> , 2003)	Não	Não	Não	Sim	FCFS, SSTF, SCAN, C-SCAN, LOOK e C-LOOK	Não	Não
FlashSim - (Kim <i>et al.</i> , 2009)	Não	Não	Não	Sim	FCFS, PIQ e NOOP	Não	Não
EagleTree - (Dayan <i>et al.</i> , 2013)	Não	Sim	Não	Não	NOOP e CFQ	Não	Não
Simulador desenvolvido – Anciotto	Sim	Sim	Não	Não	FCFS, NOOP e PIQ	Sim	Sim

5.2. Considerações finais do capítulo

Os simuladores de algoritmos de escalonamento, até a elaboração dessa dissertação, são baseados em linhas de comando, interfaces gráficas 2D (web ou não). Assim, eles não são capazes de proporcionar a imersão, interação e navegação dos alunos no ambiente simulado. Portanto, esses simuladores não permite a imersão sensorial.

Esses simuladores determinaram seus esforços na arquitetura física de um disco específico, magnético ou sólido e, apesar de eles permitirem o aprendizado dos algoritmos e entendimento da arquitetura física do disco, são voltados para realizar a comparação de desempenho ou fornecer diretrizes para projetos futuros de arquitetura.

O simulador apresentado nesta dissertação é uma aplicação que utiliza os recursos da RV para permitir que o aluno explore o ambiente 3D com o objetivo principal de aprender os algoritmos, inclusive as vantagens e desvantagens de cada um assim como, permite ao usuário interagir com a interface através de dispositivos de entrada e estímulos visuais e auditivos. Neste caso, o usuário pode se movimentar como se no mundo real estivesse e o cenário virtual responde para manter a ilusão de presença do corpo em um cenário simulado (Liu *et al.* 2017).

Além disso, mostra os elementos físicos da arquitetura interna de ambos os discos (magnético e sólido), ou seja, explora os elementos 3D do contexto. Nesse sentido, ele busca desenvolver as habilidades empíricas dos alunos no aprendizado do algoritmo ao mesmo tempo em que permite ao professor fazer associações com a arquitetura interna dos discos.

6. PROJETO E IMPLEMENTAÇÃO

Este capítulo apresenta os detalhes do projeto e implementação do simulador desenvolvido.

6.1. Introdução

O ambiente virtual desenvolvido simula os algoritmos de escalonamento de disco FCFS, NOOP e PIQ. Para isso, foi adotado o modelo incremental de desenvolvimento de *software* (Ramos, 2017). Esse modelo é organizado em etapas, denominadas “incrementos”, as quais produzem acréscimos no sistema até a sua versão final. Em cada incremento é realizado todo o ciclo de desenvolvimento de *software* e, mesmo não abordando todos os requisitos, produz um sistema funcional (Ramos, 2017). Após cada incremento, novos requisitos podem ser adicionados ou alterados (Wazlawick, 2004).

O modelo incremental é dividido basicamente em quatro etapas: a análise, o projeto, a implementação e os testes (Figura 19).

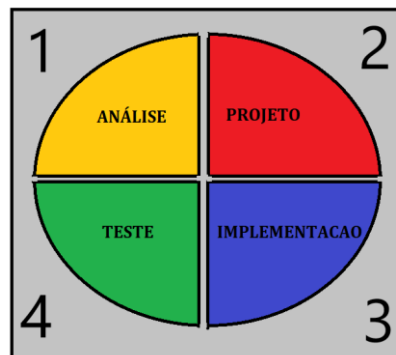


Figura 19: Etapas do Modelo incremental

As atividades desenvolvidas em cada uma destas etapas estão numeradas na sequência e representadas no fluxograma da Figura 20. Cada atividade é numerada e representada pela cor de sua etapa.

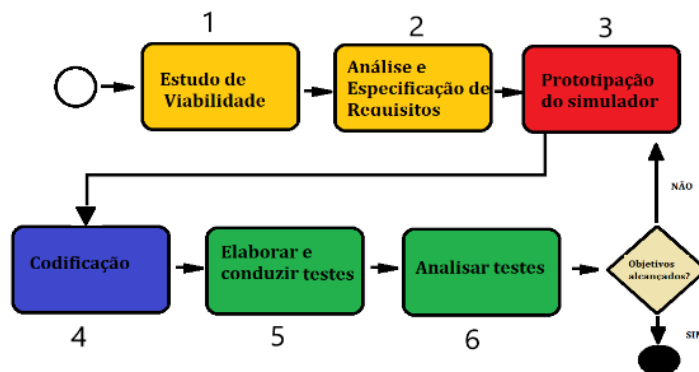


Figura 20: Fluxograma de atividades

6.2. Etapa Análise

Na etapa de análise buscou-se definir a viabilidade da concepção do simulador, bem como seu o objetivo principal. Esta etapa foi dividida em duas atividades (Figura 20). Para isso, foi realizado uma análise dos trabalhos relacionados, conforme capítulo 5, e uma revisão bibliográfica sobre os algoritmos de escalonamento de disco.

1. Atividade 1: conforme estudo comparativo, nenhum dos simuladores citados no capítulo 5 utilizou-se de RV para simular os algoritmos de escalonamento de discos e também não abordaram conceitos da arquitetura física do disco magnético e sólido.
2. Atividade 2: a fim de validar a motivação desta dissertação, nesta atividade foi realizado uma elicitação dos requisitos com base nos estudos dos trabalhos relacionados e revisão bibliográfica, permitindo assim, levantar os requisitos funcionais e não-funcionais:

Os requisitos funcionais estão relacionados à funcionalidade do sistema. Neste caso:

- a. Deve permitir ao usuário escolher o algoritmo a ser simulado. Deu-se prioridade ao algoritmo FCFS por ser um dos algoritmos presente em todos os simuladores estudados e também por ser abordado geralmente na disciplina de SO. O algoritmo NOOP e o PIQ são algoritmos presente nos simuladores quando a arquitetura física é voltada a disco sólido;

- b. Deve prover ajuda ao usuário;
- c. Deve possuir características físicas da arquitetura do disco com o intuito de permitir ao docente estimular a curiosidade do aluno;
- d. Deve proporcionar imersão ao usuário;
- e. Deve apresentar as vantagens e desvantagens dos algoritmos.

Como requisitos não funcionais:

- a. O simulador deve ser executado no SO Android;
- b. Deve possuir fácil portabilidade;
- c. Deve ser fácil de usar;
- d. O sistema deve permitir ao usuário navegar no sistema de forma segura;
- e. Cada algoritmo de escalonamento deve ser apresentado isoladamente;
- f. O sistema deverá incluir áudios ou outras mídias que facilitem o entendimento da simulação, de modo que a experiência seja multissensorial;

6.3. Etapa Projeto

Na Etapa Projeto foi estruturada a arquitetura do simulador representado pela Atividade 3 na Figura 20. O ambiente virtual desenvolvido foi projetado para possuir elementos gráficos que representem os principais componentes de um disco real, com o intuito de estimular a curiosidade do aluno durante a explicação dos conceitos de algoritmos de escalonamento de discos transformando, assim, o aprendizado em uma situação de exploração, e monitores que permitem visualizar a ordem das requisições. Além disso, mensagens de texto e áudio foram inseridas de modo a permitir ao aluno melhor compreensão dos conceitos dos algoritmos.

Para simular os conceitos dos algoritmos, bancadas específicas no interior de uma sala foram desenvolvidas para cada algoritmo simulado. O usuário ao se aproximar de cada bancada a simulação é iniciada. O simulador apresenta a seguinte arquitetura:

1. Ambiente virtual para simular algoritmos de escalonamento: a versão atual simula os algoritmos FCFS, NOOP e PIQ e permite ao aluno explorar a arquitetura interna dos discos magnético e sólido;

2. Interação: permite aos alunos manipularem o simulador usando dispositivos de entrada como controle, teclado e mouse. A interação visa a promover engajamento e imersão na simulação;
3. Navegação: possibilita aos usuários percorrerem o ambiente de uma sala na qual são simuladas bancadas específicas para a explicação dos conceitos e limitações do algoritmo FCFS, NOOP e PIQ; e
4. Visualização: permite a visualização gráfica, do conceito e da limitação do algoritmo FCFS, NOOP, PIQ e da arquitetura interna do disco magnético e sólido, usando o monitor ou dispositivos como *Google Cardboard* ou *Gear VR Samsung*.

O diagrama de Caso de Uso, Figura 21, informa as sequências de mensagens “trocadas” entre o simulador e o usuário.

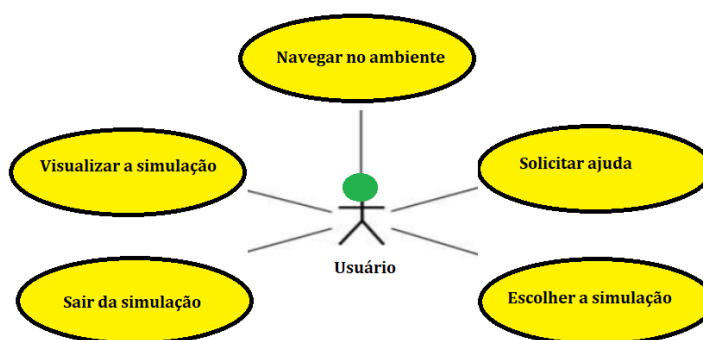


Figura 21: Caso de Uso do simulador

6.4. Etapa Implementação

Na fase de implementação o *software* é concebido, ou seja, ele é codificado de acordo com a prototipação e análises anteriores. O simulador foi implementado utilizando o motor de desenvolvimento *Unity 3D*, que é usado para criar jogos para plataformas móveis, *desktop*, dispositivos móveis, aplicativos da Web e consoles, e que permite visualização 3D de conteúdo interativo. O *Unity* em conjunto com a linguagem de programação C# permite a edição de script.

A escolha pela plataforma de desenvolvimento *Unity* foi devido a fácil portabilidade, possuir linguagem de programação, ter suporte com ferramentas de modelagem e pacote opcional como o *Utilities for Unity 1.27*, que inclui scripts, *prefabs*

e outros recursos. O pacote inclui uma interface para controlar o comportamento da câmera de RV, uma API de entrada unificada para controladores, recursos avançados de renderização, captura de objeto e ferramentas de depuração. A modelagem dos objetos foi feita no *Blender* ou *3Ds Max 2018*.

A Figura 22 mostra o ambiente inicial do simulador, no qual são exibidos ao usuário as bancadas que simulam os diferentes algoritmos de escalonamento de disco.



Figura 22: Tela inicial do simulador

Para desenvolvimento do simulador foram utilizados recursos como: *BoxCollider*, *EventTrigger*, *GameObject*, entre outros conforme Tabela 2.

Tabela 2: Tabela Recursos Unity

Recursos	Descrição
<i>BoxCollider</i>	Um componente de “colisor” que permite ser associado a um objeto do ambiente virtual. Usado nos elementos de interação, mesas, ajuda, paredes, etc.
<i>EventTrigger</i>	Recebe eventos do <i>EventSystem</i> e chama funções registradas para cada evento. Com esse recurso, foi possível criar eventos programados a partir das ações do usuário. Exemplo: ao se aproximar do objeto ajuda, o componente referente ao áudio é inicializado.
<i>GameObject</i>	Permite instanciar elementos da cena como um objeto. Os objetos instanciados possuem componente que podem ser alterados. Exemplo: posição, rotação, escala, cor, entre outros.
<i>Text</i>	Permite criar campos textuais na cena. Recurso que foi utilizado para criar a descrição dos algoritmos.
<i>OVRCameraRig</i>	É uma câmera de RV personalizada que pode ser usada para substituir a câmera comum Unity em uma cena. Tal recurso foi utilizado para substituir a câmera do avatar.

A aproximação aos elementos como disco magnético ou disco sólido sobre as bancadas faz com que as simulações iniciem. A Figura 23 mostra a bancada da qual é possível visualizar os conceitos do algoritmo FCFS.

Requisições de acesso a disco



Figura 23: Algoritmo FCFS Conceitos

Para exemplificar o conceito do algoritmo FCFS, foram predefinidos no simulador cubos vermelhos que caem ao lado do disco magnético de modo a representar a abstração das requisições de acesso a disco e também um disco hipotético com 120 blocos, cujo cabeçote de leitura se encontra inicialmente sobre o bloco 60. A fila de pedidos de acesso pendentes corresponde aos seguintes blocos do disco, em sequência: 40, 20, 10, 30, 70, 110, 80, 120, 60, 20, 50, 90. Para facilitar o entendimento, considerou-se que nenhum novo pedido de acesso chegará à fila durante a execução dos algoritmos e que o escalonador já terá associado o número do bloco de cada requisição. Como a cabeça de leitura está inicialmente no bloco 60, ela primeiro passará do bloco 60 para o bloco 40, depois para os blocos 20, 10, 30, 70, 110, 80, 120, 60, 20, 50 e, finalmente, para o bloco 90, realizando um movimento de cabeça total de 380 blocos. A simulação da ordem atendimento das requisições é apresentada em monitores que fazem parte no ambiente simulado. A Figura 24 mostra, em destaque, um exemplo dessa simulação.

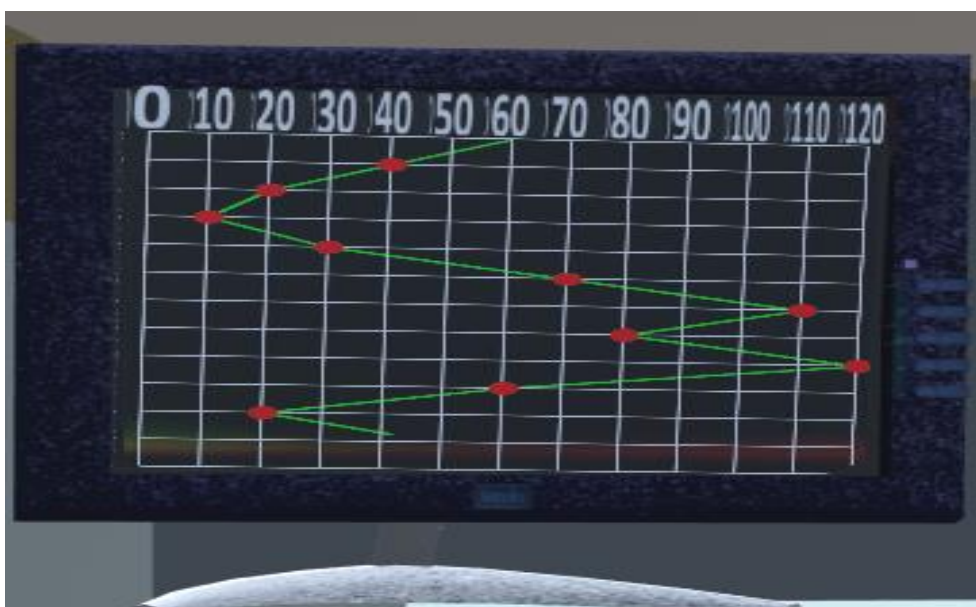


Figura 24: monitor de simulação do algoritmo FCFS Conceitos

A Figura 25 mostra a bancada de limitações do algoritmo FCFS que é apresentada para o usuário. Além de contar com elementos visuais, as explicações são realizadas com recursos como áudio e texto.



Figura 25: Algoritmo FCFS Limitações

O mesmo recurso de simulação apresentado em um monitor no ambiente foi criado para ilustrar as limitações do algoritmo FCFS, alterando a ordem das leituras dos blocos. A simulação demonstra que a relação posicional dos blocos influencia no desempenho do sistema. Nesse caso, a fila de pedidos de acesso pendentes relativa aos blocos do disco

teve a seguinte sequência: 10, 120, 20, 110, 20, 110, 120, 30, 120, 10, 90, 10. A Figura 26 mostra os deslocamentos da cabeça de leitura para atender aos pedidos de acesso da fila de pedidos pendentes do exemplo. Como a cabeça de leitura está inicialmente no bloco 60, ela primeiro passará do bloco 60 para o bloco 10, depois para os blocos 120, 20, 110, 20, 110, 120, 30, 120, 10, 90 e, finalmente, para o bloco 10, realizando um movimento de cabeça total de 980 blocos. Nesse exemplo ocorreu 275% de movimento da cabeça de leitura a mais do que no exemplo anterior.



Figura 26: Monitor simulação algoritmo FCFS Limitações

A Figura 27 mostra, em destaque, o disco magnético que é apresentado no ambiente virtual. Durante a simulação os elementos simulados se movimentam como no mundo real.

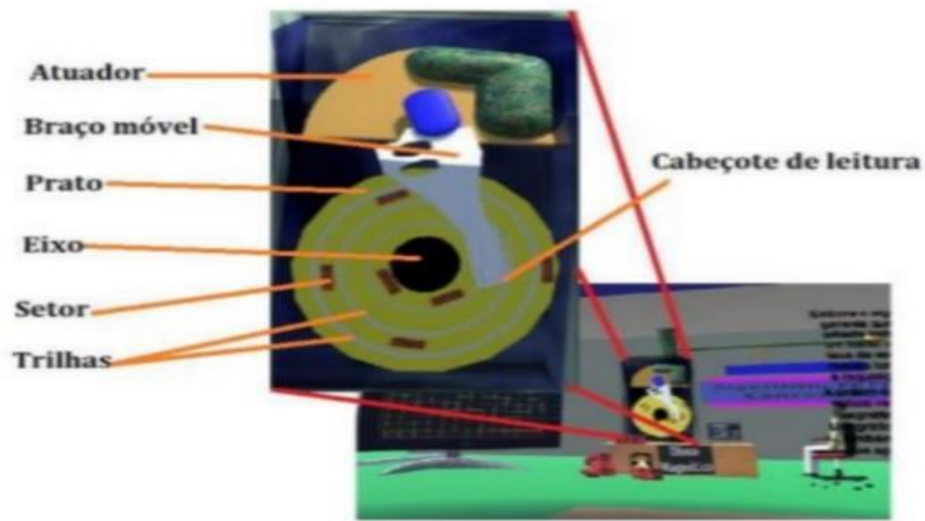


Figura 27: Disco Magnético

A Figura 28 mostra o disco sólido. Em destaque os componentes físicos e parte de sua estrutura lógica.



Figura 28: Disco SSD

Os discos são animados e possuem os principais componentes de um disco real. Isso permite que o usuário explore suas arquiteturas internas.

Para exemplos da simulação dos escalonadores NOOP e PIQ foi suposto que há seis requisições na fila pendente (*Pending Queue*) (três requisições de leitura (R0, R1, R2)

e três requisições de gravação (W0, W1, W2)), cada requisição acessa a dois chips de memória *flash*. Essas seis requisições são adicionadas à fila pendente na ordem FCFS, da esquerda para a direita. A arquitetura física do SSD no ambiente é organizada com quatro chips de memória *flash* com um chip por canal. A localização dos dados das seis requisições é marcada nos quatro chips e os vetores de localização dessas requisições são produzidos. A Figura 29 ilustra os conceitos do algoritmo NOOP.

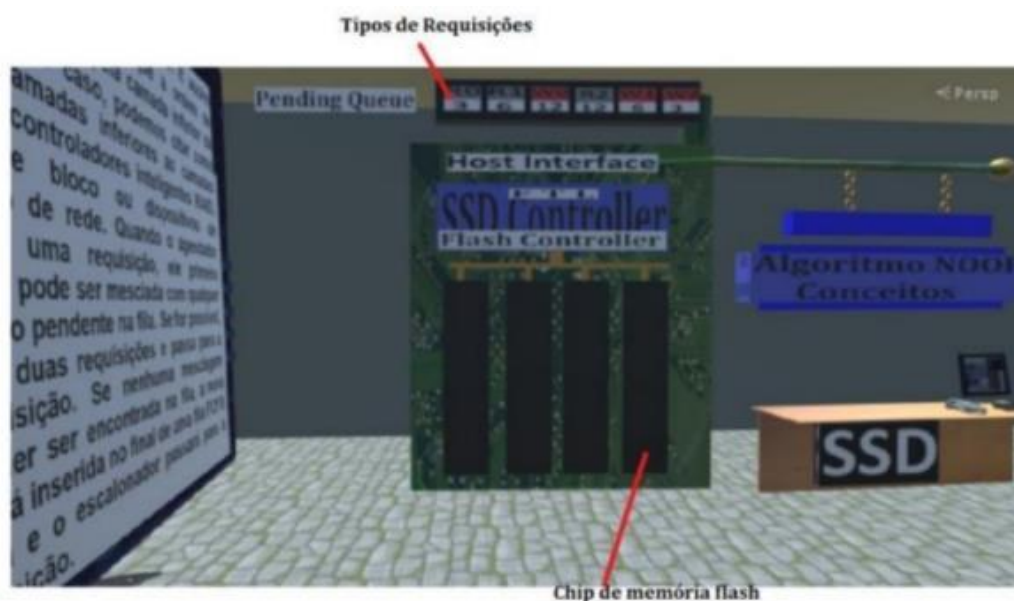


Figura 29: Conceitos do algoritmo NOOP

Nesse caso, as seis requisições são processadas sequencialmente pois, elas são escalonadas na ordem de FCFS, o que requer três ciclos de leitura e três de gravação para atender a essas requisições, conforme mostrado na Figura 30, totalizando seis acessos ao disco.



Figura 30: Simulação do algoritmo NOOP

O algoritmo PIQ (Figura 31) é organizado nas seguintes estruturas: (1) fila de requisições pendentes (*Pending Queue*), a qual é planejada conforme o algoritmo FCFS; (2) lista de separação por meio da técnica RWS; que é representada pela lista *Read Write Separation*; e (3) pacotes de requisições sem conflitos, os quais são inseridos na lista *Request Batch*.



Figura 31: Conceitos do algoritmo PIQ

A simulação do escalonador PIQ ocorre em três etapas onde o RWS, o RRS e o WWS são aplicados à fila pendente, *Pending Queue*, para separar as requisições em

diferentes pacotes. Na primeira etapa o RWS é aplicado onde as requisições de leitura e gravação são separadas, para evitar o RWC. Na segunda, o RRS e o WWS são aplicados. Assim, as requisições de leitura e gravação são separadas em pacotes diferentes e inseridas na lista *Read Write Separation*. Por fim, o PIQ detecta os conflitos e os envia às requisições em pacotes sem conflitos para uma lista (*Request Batch*), onde todos os pacotes são processados na ordem de criação. A detecção dos conflitos é feita através do número do vetor de localização presente em todas as requisições. Neste caso, com o PIQ, a simulação foi desenvolvida de modo que apenas dois ciclos de leitura e dois de gravação são necessários para atender a essas requisições. Tal simulação pode ser observada na Figura 32.

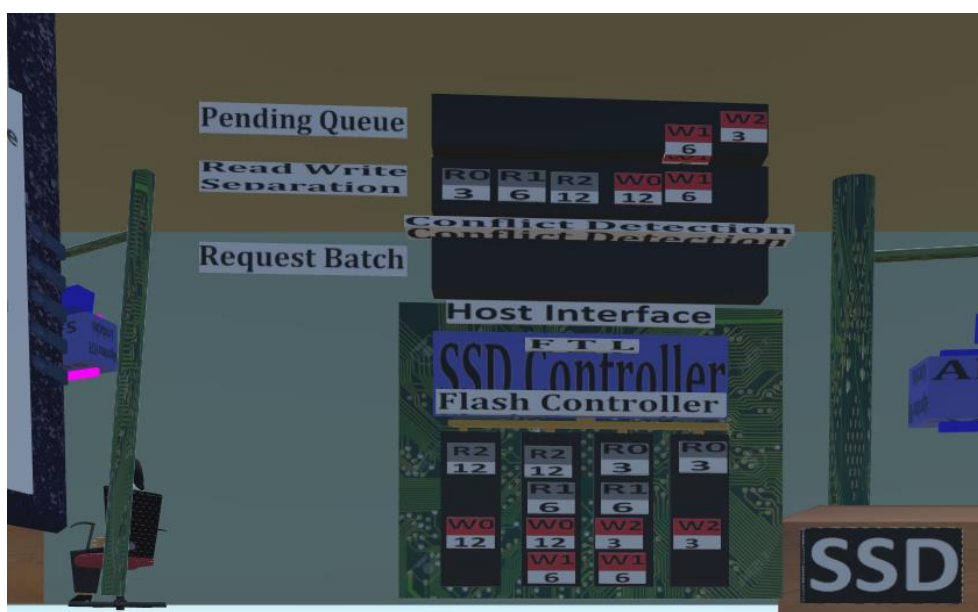


Figura 32: Simulação do algoritmo PIQ

Na comparação dos escalonadores NOOP e PIQ observa-se que com o NOOP (Figura 30) foi necessário realizar seis acessos ao disco enquanto que com o PIQ (Figura 32) as mesmas requisições foram realizadas em quatro acessos ao disco.

Nesse projeto, a imersão é fornecida pela utilização de um óculos de RV que permite ao usuário movimentar sua cabeça em 360 graus, viabilizando um olhar ao redor como o feito na vida real, o que ocasiona uma experiência quase realista. A Figura 33 e a Figura 34 mostram a visualização do simulador composta por 2 imagens voltada para

execução em um smartphone e visualizada em dispositivos como *Google Cardboard* e *Gear VR*. A interação é realizada utilizando um *joystick* com conexão *Bluetooth*.

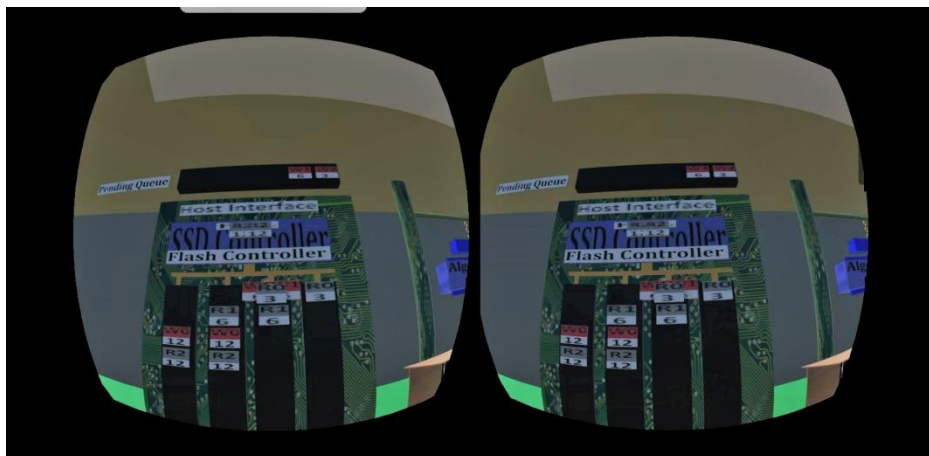


Figura 33: Imagens do algoritmo PIQ para visualização 3D (olho esquerdo e direito)

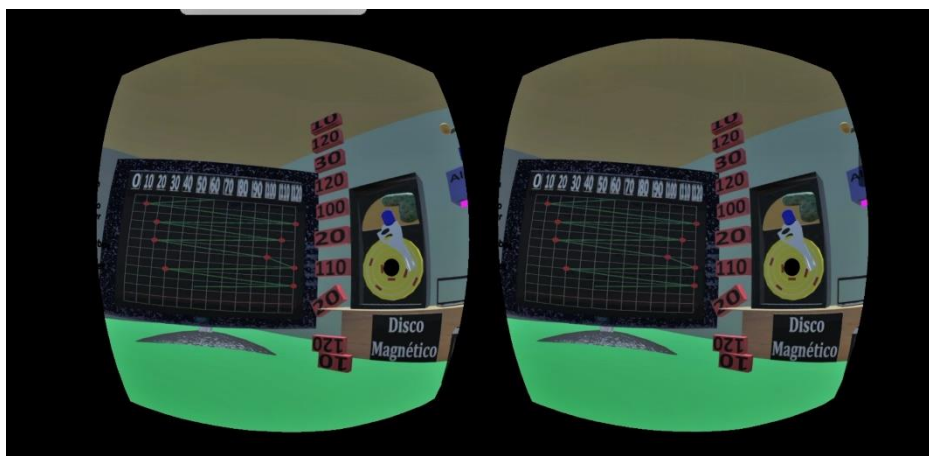


Figura 34: Imagens limitações do algoritmo FCFS para visualização 3D (olho esquerdo e direito)

6.5. Considerações finais do capítulo

O simulador de RV foi desenvolvido de modo a simular os algoritmos de escalonamento de disco FCFS (conceitos e limitações), NOOP e PIQ assim como, a visualização da arquitetura interna dos discos magnético e sólido. Tal simulador permite a interação do usuário por meio de dispositivos de entrada como também a navegação e visualização do ambiente.

7. METODOLOGIA DOS TESTES

Este capítulo apresenta a metodologia utilizada nos testes de conhecimento, usabilidade e de aspecto motivacional.

7.1. Introdução

O simulador foi desenvolvido em duas etapas. Após a implementação da primeira etapa do simulador, foram realizados testes de conhecimento, usabilidade e de aspecto motivacionais. O objetivo do teste de conhecimento foi analisar se o simulador é benéfico na introdução de conceitos sobre algoritmos de escalonamento de disco, no caso o FCFS. Mais precisamente, procurou-se investigar suas funcionalidades e se a incorporação do simulador utilizando um dispositivo como o *Gear VR* e o *desktop* podem produzir melhores resultados do que o método tradicional (isto é, aula expositiva).

A tecnologia por si só não assegura melhorias no ensino se não for observada a qualidade do *software* utilizado (Brandão 1998). Neste caso, o simulador precisa ter uma interface amigável, ser de fácil uso, entre outras características (conforme explicado na seção 7.3 e 7.4). Do contrário, o usuário pode facilmente se desmotivar e não utilizar o simulador. Portanto, uma vez desenvolvido o *software*, foi necessário validar se ele atendia às necessidades do ponto de vista do usuário final (Wazlawick, 2004). Para isso, o simulador foi submetido a um teste de usabilidade, com objetivo de detectar possíveis falhas no desenvolvimento e a um de aspectos motivacionais, para analisar em qual ambiente (*desktop* ou *Gear VR*) o simulador proporcionou melhor motivação.

Em uma Etapa 2, melhorias e sugestões foram introduzidas no simulador com base nos resultados obtidos na Etapa 1 e novos algoritmos de escalonamento de disco como o NOOP e PIQ foram implementados, assim como a arquitetura física do disco de estado sólido. Testes de usabilidade, motivacional e de conhecimento, foram reaplicados.

Os testes foram realizados em instantes diferentes no Instituto Federal de São Paulo IFSP, campus Pirituba, com um total de 63 alunos de 14 a 17 anos matriculados no primeiro e segundo ano do Ensino Técnico Integrado ao Médio em Redes de Computadores. A Tabela 3 mostra o questionário utilizado para identificar o perfil dos usuários nas duas etapas.

Tabela 3: Perfil dos usuários

Nome
Idade: Sexo: () Masculino () Feminino
Há quanto tempo faz uso de computador?
Em qual local você utiliza o computador?
O que mais utiliza quando usa computadores?
Considerando a sua experiência prévia com sistemas de Realidade Virtual, como classifica esses sistemas?

Nesta seção é apresentada a metodologia aplicada ao teste de conhecimento, de usabilidade e motivacional organizados cada teste em Etapa 1 e Etapa 2. Vale ressaltar que as turmas do primeiro e do segundo ano são formadas com 40 alunos cada sendo, porém, divididas em 2 grupos (A e B) de 20 alunos para as aulas práticas de laboratório. A divisão dos grupos (A e B) é feita de forma aleatória através do sistema acadêmico da instituição. A Tabela 4 apresenta um resumo dos testes realizados na Etapa 1.

Tabela 4: Resumo dos testes da Etapa 1 com o algoritmo FCFS

Testes	Atividade	Total de alunos	Duração
Teste de conhecimento	Formação dos grupos	- 15 alunos do primeiro ano (grupo A) - 3 grupos de 5 alunos (divisão aleatória)	10 – 15 minutos
	Aplicação	- 3 grupos de 5 alunos cada	15 – 25 minutos
Teste de usabilidade	Formação dos grupos	- 12 alunos do primeiro ano (grupo A) - 2 grupos de 6 alunos (divisão aleatória)	10 – 15 minutos
	Aplicação	- 2 grupos de 6 alunos cada	7 – 10 minutos
Teste de aspecto motivacional	Formação dos grupos	- 24 alunos do segundo ano (grupo A e B) - 2 grupos de 12 alunos (divisão aleatória)	10 – 15 minutos
	Aplicação	2 grupos de 12 alunos cada	7 – 10 minutos

Os testes foram feitos após implementação das melhorias detectadas na Etapa 1. A Tabela 5 apresenta um resumo dos testes realizados no simulador na Etapa 2.

Tabela 5 resumo dos testes da Etapa 2 com os algoritmos NOOP e PIQ

Testes	Atividade	Total de alunos	Duração
Teste de conhecimento	Formação dos grupos	- 39 alunos do primeiro ano (grupo A e B) - 3 grupos de 13 alunos (divisão aleatória)	10 – 15 minutos
	Aplicação	- 3 grupos de 13 alunos	15 – 25 minutos
Teste de usabilidade	Formação dos grupos	- 36 alunos do primeiro ano (grupo A e B) - 2 grupos de 18 alunos (divisão aleatória)	10 – 15 minutos
	Aplicação	- 2 grupos de 18 alunos	7 – 10 minutos
Teste de aspecto motivacional	Formação dos grupos	- 38 alunos do primeiro ano (grupo A e B) - 2 grupos de 19 alunos (divisão aleatória)	10 – 15 minutos
	Aplicação	- 2 grupos de 19 alunos	7 – 10 minutos

7.2. Avaliação do simulador como método de apoio educacional (Conhecimento)

Kirkpatrick (1975) criou um modelo de avaliação de treinamento que ao longo dos anos vem sofrendo várias modificações (Kirkpatrick and Kirpatrick, 2011). Seu modelo permite não só avaliar a satisfação e a aprendizagem dos indivíduos como também identifica todos os critérios que influenciam esses dois aspectos. O modelo é baseado em quatro níveis, e cada nível tem sua respectiva importância. Conforme se passa de um nível para outro, o processo se torna mais complexo e demorado, porém, fornece resultados mais valiosos.

Em seu modelo, a informação de cada nível anterior serve como base para a avaliação do próximo nível. Assim, cada nível sucessivo representa uma medida mais precisa da eficácia do programa de treinamento, mas ao mesmo tempo, requer um uso mais rigoroso e demorado análise.

Os níveis descritos são:

1. Reação: Assim como a palavra implica, a avaliação neste nível mede como os participantes reagem ao processo de treinamento. Ele tenta responder a perguntas sobre percepções dos participantes onde se mede a satisfação e valor percebido;
2. Aprendizagem: leva em consideração o quanto os participantes podem mudar de atitude, ampliar seus conhecimentos e habilidades à medida que adquirem os conhecimentos, habilidades e atitudes pretendidos com base em sua participação no evento de aprendizagem. Ou seja, até que ponto os alunos avançaram em habilidades, conhecimento ou atitude. Os métodos de avaliação de aprendizagem variam de testes formais à informais, avaliação da equipe à autoavaliação. Para determinar a evolução da aprendizagem, os participantes podem fazer teste de avaliação antes do treinamento (pré-teste) e após o treinamento (pós-teste);
3. Comportamento: identifica o quanto os participantes mudaram seu comportamento em decorrência do que foi aprendido ou o quanto é aplicado do que aprenderam durante o treinamento em sua jornada de trabalho. Este nível tenta responder à pergunta: Os conhecimentos, habilidades ou atitudes recém-adquiridos são usados no ambiente cotidiano do aluno?
4. Resultados: identifica os ganhos obtidos no processo de aprendizagem e seus respectivos impactos.

Para elaboração do questionário de conhecimento, foi tomado como base o segundo nível de Kirkpatrick - a aprendizagem - que avalia o quanto os alunos aprenderam a respeito do algoritmo de escalonamento de disco.

7.2.1. Etapa 1

Na primeira etapa, havia disponível uma turma de 40 alunos do primeiro ano do curso de redes de computadores com idade aproximada de 14 anos. Para não realizar o teste inicial com os 40 alunos, foram realizados cálculos estatísticos de modo a determinar o tamanho mínimo da amostra n que representasse os 40 alunos. Para isso, foi realizado um cálculo de amostra com base na determinação do tamanho da amostra para população finita (Moore *et al.*, 2006).

Entende-se como população um conjunto de indivíduos que partilham de, pelo menos, uma característica comum, seja ela salário, nacionalidade, filiação a uma associação de voluntários, raça, matrícula na faculdade, etc. E o termo amostra, um subconjunto de indivíduos extraídos de uma população.

Para tanto, foi considerado os seguintes parâmetros:

n = Número de indivíduos da amostra.

N = Tamanho da população (40 alunos).

$Z_{\alpha/2}$ = Valor crítico que corresponde ao grau de confiança desejado. Estima-se um grau de confiança de 95%.

σ = Desvio padrão populacional da variável que se deseja calcular. Como o desvio padrão dos alunos é desconhecido, utilizou-se a nota final dos alunos dos bimestres anteriores para definir a amplitude e, conseqüentemente, o desvio. Assim, a amplitude foi de 5 (diferença entre a maior e menor nota) gerando um desvio padrão de 1.25 (amplitude/4).

E = Erro máximo de estimativa de 0.5 para cima ou para abaixo.

A fórmula para determinação do tamanho da amostra (n) com base na estimativa da média populacional:
$$n = \frac{N * \sigma^2 * (Z_{\alpha/2})^2}{(N-1) * E^2 + \sigma^2 * (Z_{\alpha/2})^2}$$

Neste caso, obteve-se n (amostra) de 15.244 aluno. Portanto, o teste de conhecimento na Etapa 1 foi realizado com 3 grupos compostos de 5 alunos de forma aleatória.

Como é complexo mensurar o conhecimento de um grupo de alunos sobre um determinado assunto e também avaliar se o simulador é benéfico no auxílio do aprendizado de algoritmos de escalonamento de disco, o teste de conhecimento foi estruturado na seguinte maneira. Ocorreu divisão aleatória dos alunos do ensino médio em três grupos, método tradicional (aula expositiva), *desktop* e *Gear VR*, de cinco alunos cada;

1. Pré-teste: todos os alunos responderam um questionário (Tabela 6), que versava sobre conceitos de algoritmos de escalonamento de disco FCFS. Foi

definido uma média de 15 minutos para que os alunos respondessem a todas as questões;

2. Pós-teste: no decorrer dessa etapa, os alunos tiveram que responder a perguntas relacionadas aos conceitos que foram anteriormente apresentados (Tabela 6). Foi definido uma média de 15 a 25 minutos para que os alunos respondessem a todas as questões.

O objetivo do pré-teste de conhecimento foi avaliar se os alunos tinham conhecimento prévio sobre o assunto.

Tabela 6: Questionário de conhecimento (pré-teste e pós-teste)

1.O algoritmo FCFS (<i>First-Come, First-Served</i>) atende às requisições para ler ou gravar dados em um disco. Segundo o algoritmo, primeira requisição a chegar é a primeira a ser servida. () Certo () Errado.
2.O algoritmo FCFS (<i>First-Come, First-Served</i>) seleciona a requisição com o menor tempo de busca em relação à posição atual do cabeçote de leitura do disco. () Certo () Errado.
3.O algoritmo FCFS (<i>First-Come, First-Served</i>) é justo pois a posição de uma requisição na fila não é afetada pelas requisições que chegam. () Certo () Errado.
4.Uma das vantagens do algoritmo FCFS (<i>First-Come, First-Served</i>) é que se trata de um algoritmo injusto, pois a posição de uma requisição na fila não é afetada pelas requisições que chegam. () Certo () Errado.
5.No algoritmo FCFS (<i>First-Come, First-Served</i>) não há otimização do tempo de deslocamento do braço do disco e a execução é lenta. () Certo () Errado.

A eficiência do simulador, como ferramenta de apoio ao ensino de algoritmos de escalonamento, foi avaliada por meio de questionários que versavam sobre conceitos de algoritmo de escalonamento de disco (Tabela 6). Após isso, foi comparada a pontuação dos alunos que foram introduzidos aos conceitos de algoritmos de escalonamento por meio do auxílio do simulador, utilizando um ambiente de RV e no ambiente do *desktop*, com as pontuações de alunos que aprenderam o mesmo conceito por meio do método tradicional. Vale ressaltar que o conteúdo, algoritmo de escalonamento de disco, foi lecionado nos 3 grupos pelo mesmo professor, porém, em momentos distintos.

Para análise estatística utilizou-se o método ANOVA, que é uma técnica estatística que visa, fundamentalmente, verificar se existe uma diferença significativa entre grupos.

7.2.2. Etapa 2

Na Etapa 2, o pré-teste e pós-teste de conhecimento foram aplicados com base nos algoritmos de escalonamento de disco NOOP e PIQ, implementados na segunda etapa Tabela 7. Participaram do experimento 39 alunos do primeiro ano, que foram divididos de forma aleatória em três grupos, método tradicional (aula expositiva), *desktop* e *Gear VR*. Após isso, foi comparada a pontuação dos alunos que foram introduzidos aos conceitos de algoritmos de escalonamento por meio do auxílio do simulador, utilizando um ambiente de RV, e no ambiente do *desktop*, com as pontuações de alunos que aprenderam o mesmo conceito por meio do método tradicional. Foi definido uma média de 15 a 25 minutos para que os alunos respondessem a todas as questões.

Tabela 7: Questionário de conhecimento (pré-teste e pós-teste) NOOP e PIQ

1- NOOP pode ser considerado um escalonador de E/S com sobrecarga de CPU que executa e fornece funcionalidades de reordenação de requisições. <input type="checkbox"/> Certo <input type="checkbox"/> Errado
2-O algoritmo de escalonamento NOOP pode ser desejado quando as funções do escalonador de E/S for tratada por uma camada inferior da pilha de E/S. <input type="checkbox"/> Certo <input type="checkbox"/> Errado
3-O algoritmo de escalonamento PIQ é implementado no sistema de host. Ele não introduz nenhuma mudança no design atual do disco SSD. <input type="checkbox"/> Certo <input type="checkbox"/> Errado
4-A ideia básica do PIQ é separar as requisições presentes na fila em lotes sem conflitos. <input type="checkbox"/> Certo <input type="checkbox"/> Errado
5-A implementação do PIQ ocorre em três etapas: a. Separar as requisições por tipos; b. Criar lotes de requisições com conflitos; c. Enviar as requisições ao disco. <input type="checkbox"/> Certo <input type="checkbox"/> Errado

Para análise estatística na segunda etapa, utilizou-se também o método ANOVA.

7.3. Teste de usabilidade

Se o simulador exigir conhecimento técnico para seu uso, não indicar claramente o que precisa ser feito, não informar o que os usuários podem fazer e se as informações textuais forem difíceis de ler, os alunos podem perder a motivação e, por consequência, eles não o utilizam como ferramenta de apoio educacional como também influência na motivação de seu uso. Para isso, o teste de usabilidade é uma condição fundamental.

O teste de usabilidade teve como objetivo de detectar as funcionalidades do simulador (possíveis falhas). Para isso, utilizou-se a heurística de (Nielsen, 1994). Vários trabalhos encontrados na literatura já têm apontado adaptações das heurísticas de Nielsen a fim de reinterpretar as especificidades de suas áreas (Martins e Moura 2011), (Sutcliffe e Gault 2004). Assim, com base em diversos trabalhos (Martins *et al.*, 2014), (Martins *et*

al., 2013), (Ferreira *et al.*, 2014), (Bowman e McMahan 2007), heurísticas foram definidas para avaliar o simulador. Além disso, assim como (Muller *et al.*, 1998) e outros a avaliação foi realizada por usuários finais.

A reinterpretação do conjunto de heurísticas de Nielsen (1994) no contexto foi a seguinte:

1. Visibilidade do status do sistema: o simulador informa ao usuário sobre as ações que estão ocorrendo; por exemplo, que está visualizando uma fila de requisições;
2. Compatibilidade do sistema com o mundo real: o simulador apresenta informações de tal forma como no mundo real; por exemplo, o disco rígido movimentando os braços como no mundo real, as explicações apresentam linguagens familiares aos usuários;
3. Controle do usuário e liberdade: o simulador possibilita que o aluno tenha controle total do ambiente, permitindo por exemplo, que saiam de situações indesejadas. Por exemplo, os alunos podem se movimentar para outra bancada, mesmo sem ter terminado a simulação da atual;
4. Consistência e padrões: sistema deve seguir uma convenção estipulada inicialmente para que o aluno não tenha que adivinhar que situações ou que ações idênticas significam as mesmas coisas; com base nisso, desenvolveu-se as duas bancadas para terem os mesmos padrões de animações;
5. Prevenção de erros: o simulador deve verificar e eliminar tendência a erros ou informar ao aluno essa possibilidade antes que ele se comprometa com a ação. Nesse sentido, foram adicionadas mensagens de áudio explicativas capazes de instruir o usuário sobre o que fazer nos momentos em que possíveis erros fossem gerados; por exemplo, instruções ao aluno para não sair do interior do ambiente;
6. Reconhecimento ao invés de relembração: minimizar a sobrecarga de memória do aluno, tornando visíveis objetos, ações e opções. Pensando neste aspecto, a simulação do algoritmo de escalonamento de disco e o disco são criados apenas quando o aluno se aproxima da bancada, proporcionando menor sobrecarga de informação no ambiente;

7. Flexibilidade e eficiência de uso: oferecer aos alunos mais experientes aceleradores que permitam que as tarefas sejam realizadas mais rapidamente. Por exemplo, botões no joystick que facilitam e aceleram o caminhar do aluno no ambiente virtual;
8. Estética e design minimalista: o simulador evita a utilização de informações irrelevantes ou pouco necessárias. Assim, uma quantidade mínima de objetos virtuais foi agregada ao ambiente, ou seja, foram adicionados somente aqueles objetos que são realmente indispensáveis;
9. Ajudar os usuários a reconhecer, diagnosticar e corrigir erros: o simulador deve indicar precisamente o problema e oferecer opções construtivas para a solução. O simulador oferece textos e áudios complementares à visualização para ajudar o usuário; e
10. Ajuda e documentação: devem, por exemplo, explicar o uso inicial do simulador. Nesse sentido, mensagens iniciais audíveis foram inseridas, por exemplo, “Seja bem-vindo ao simulador educacional de algoritmo de escalonamento de disco. Caminhe até as bancadas no interior da casa para visualização do funcionamento dos algoritmos”.

Também foram levados em consideração aspectos relacionados à facilidade de uso (Martins *et al.* 2013) uma reinterpretação da ISO (9241-11) e à imersão do usuário no uso do simulador de realidade virtual. Essas outras heurísticas são as seguintes:

1. Facilidade de uso: o quanto o simulador é convidativo ao uso, mesmo na primeira vez que é utilizado. Neste caso o objetivo é analisar a satisfação e a aceitação do simulador; e
2. Imersão: conforme Slater (2003), quanto mais um sistema fornece exibições (em todas as modalidades sensoriais) e rastreamento que preservam a fidelidade em relação a suas modalidades sensoriais equivalentes do mundo real mais ele é imersivo. Desse modo, foi analisado a sensação de imersão que os usuários tiveram ao utilizar o simulador através dos óculos de RV.

Isso resultou nas afirmações apresentadas na Tabela 8, que foram aplicadas no teste de usabilidade, e foram respondidas usando a escala de mensuração citada por Likert

(1932) para cada questão (Discordo fortemente; Discordo; Nem discordo, nem concordo; Concordo; Concordo fortemente).

Tabela 8: Questionário – Teste de usabilidade

1) Foi fácil entender como utilizar a aplicação.	Facilidade de uso
2) O tempo de resposta às ações dentro do ambiente é aceitável.	Flexibilidade e eficiência de uso
3) Eu me senti totalmente informado sobre o que estava acontecendo no sistema.	Visibilidade e status do sistema
4) A aplicação deu suporte para desfazer e refazer as ações indesejadas e proporcionou uma forma clara de navegar.	Controle e liberdade do sistema
5) Quando existiam mensagens na aplicação, sejam textuais ou em áudio, a linguagem era explicativa.	Compatibilidade do sistema com o mundo real
6) Quando existiam mensagens na aplicação, sejam textuais ou em áudio, a linguagem é facilmente entendida.	Prevenção de erros
7) A ajuda da aplicação, através de áudio e texto, instrui de forma clara o que deve ser realizado.	Ajuda e documentação.
8) As instruções para uso da aplicação são fáceis de serem reconhecidas.	Reconhecimento ao invés de lembranças.
9) Os monitores de vídeos que mostram os gráficos do algoritmo são de fácil compreensão e explicativos.	Compatibilidade do sistema com o mundo real.
10) Os objetos que me foram apresentados na simulação foram suficientes para o entendimento.	Estética e design minimalista.
11) Eu sempre executava as mesmas ações para acionar as mesmas funcionalidades.	Consistência e padrões.
12) A aplicação permite que o usuário se sinta imerso no ambiente (Questão presente apenas no teste de usabilidade com o Gear VR).	Imersão.

Questões abertas (Tabela 9) também foram feitas aos alunos com o intuito de que eles pudessem expressar suas considerações livremente.

Tabela 9: Questões abertas teste de usabilidade

1.Aspectos identificados como positivos.
2.Aspectos identificados como negativos.
3.Caso tenha sentido algum desconforto na experiência, descreva.
4.Sugestão para a aplicação

Os testes de usabilidade aplicados foram feitos para as duas etapas do simulador que estão descritos a seguir:

- Etapa 1: o teste de usabilidade foi realizado com 12 alunos do primeiro ano, separados de forma aleatória em dois grupos de seis alunos cada. O primeiro grupo utilizou o simulador em um *desktop*; o segundo grupo usou a versão no *Gear VR*. Cada aluno demorou de 7 a 10 minutos para testar a

aplicação. Após o uso, eles responderam os questionários (Tabela 8 e Tabela 9) que foram utilizados para melhorias do simulador.

- Etapa 2: após implementação das sugestões e aprimoramento dos problemas evidenciados na Etapa 1, na Etapa 2 o teste de usabilidade foi reaplicado com um grupo de 36 alunos do primeiro ano separados em dois grupos de 18 alunos cada. O primeiro grupo utilizou o simulador em um *desktop*; o segundo grupo usou a versão no *Gear VR* e cada aluno demorou de 7 a 10 minutos para testar a aplicação.

7.4. Teste de aspecto motivacional

Conforme Keller (2009), a motivação consiste na quantidade de esforço que uma pessoa está disposta a exercer em busca de uma meta no contexto da aprendizagem. Para suggestionar a motivação dos alunos nesse contexto foi proposto o modelo de designer motivacional ARCS (*Attention, Relevance, Confidence, Satisfaction*) de Keller (2009).

ARCS consiste em categorias conceituais relacionadas à motivação humana e pode ser considerado um acrônimo para as quatro categorias de estratégias para motivar os alunos na aprendizagem: atenção, relevância, confiança e satisfação.

1. Atenção: refere-se ao quanto o aluno está engajado com o objeto da aprendizagem. É um elemento motivacional e pré-requisito para a aprendizagem;
2. Relevância: no processo de aprendizagem, a proposta educacional deve estar atrelada aos seus objetivos, ou seja, deve ser útil aos alunos;
3. Confiança: a proposta deve gerar ao usuário uma perspectiva positiva que pode ser obtida por meio de suas próprias habilidade e esforços; e
4. Satisfação: o sentimento do aluno dever ser positivo no processo de aprendizagem, o que pode ser alcançado com gratificações. Ou seja, os alunos devem estar satisfeitos com o que conseguiram durante o processo de aprendizagem.

Para ser utilizado como método de avaliação a proposta de Keller (2009) possui um instrumento denominado *Instructional Materials Motivational Scale* - IMMS que se

materializa em um questionário para ser aplicado depois que os estudantes utilizam um material educacional.

Outro modelo utilizado na composição do questionário de aspecto motivacional foi o modelo de avaliação de treinamentos de Kirkpatrick (1975) conforme explicado na seção 7.2. Para isso, foi tomado como base apenas o primeiro nível - a reação - que avalia qual foi a percepção do aprendiz a respeito da experiência do aprendizado.

A reinterpretção dos dois modelos consiste no questionário de aspectos motivacionais apresentado na Tabela 10.

Tabela 10: Questões sobre aspectos motivacionais

1.O design da interface do simulador é atraente?	Atenção
2.A interface gráfica do simulador é agradável?	Atenção
3.Eu aprendi algumas coisas com o simulador que foram surpreendentes ou inesperadas.	Satisfação
4.O conteúdo do simulador é relevante para meus interesses.	Relevância
5.Eu gostei tanto do simulador que gostaria de aprender mais sobre o assunto abordado por ele.	Relevância
6.O simulador apresentou um conteúdo abstrato de forma prática.	Confiança
7.Senti que estava tendo progresso durante o desenrolar do simulador	Satisfação
8.O simulador me manteve motivado a continuar utilizando-o.	Atenção
9.Eu utilizaria este simulador novamente.	Confiança
10.Depois de utilizar o simulador, eu consigo lembrar de mais informações relacionadas ao tema nele apresentado.	Reação
11. Depois de utilizar o simulador, eu consigo compreender melhor os temas nele apresentados.	Reação
12. Depois de utilizar o simulador, eu consigo aplicar melhor os temas nele relacionados.	Reação

Questões abertas também foram feitas aos usuários no intuito de que os mesmos pudessem expressar suas considerações livremente tais como: aspectos positivos, aspectos negativos, desconfortos e sugestões.

A aplicação do questionário motivacional foi organizada em duas etapas distintas.

- Etapa 1: foi realizado com vinte e quatro alunos do segundo ano, os quais foram separados de forma aleatória em dois grupos de 12 alunos cada. O primeiro grupo utilizou o simulador em um desktop; o segundo grupo usou a versão no Gear VR. Cada aluno demorou de 7 a 10 minutos para testar a aplicação. Após o uso, eles responderam o questionário (Tabela 10).

- Etapa 2: foi reaplicado o teste com um grupo de 38 alunos do primeiro ano, separados de forma aleatória em dois grupos de 19 alunos cada. O primeiro grupo utilizou o simulador em um *desktop*; o segundo grupo usou a versão no *Gear VR*. Cada aluno demorou de 7 a 10 minutos para testar a aplicação. O questionário aplicado versava sobre as questões na Tabela 10.

7.5. Considerações finais do capítulo

Através da metodologia aplicada nos testes de conhecimento, usabilidade e motivacional, foi possível investigar a eficiência e aceitação do simulador nos aspectos de conhecimento e motivacional para os grupos de alunos selecionados em seus respectivos testes assim como, analisar critérios de usabilidade com o objetivo de detectar possíveis melhorias no simulador.

8. RESULTADOS DOS TESTES E DISCUSSÃO

Neste capítulo são apresentados os resultados e discussões dos testes no simulador de apoio ao ensino de algoritmos de escalonamento de disco magnético e sólido (conhecimento), bem como dos testes de usabilidade e motivacional. Os testes permitiram não só identificar alguns pontos que despertaram maior interesse nos alunos ou que não os agradaram, como também analisar se houve uma diferença estatística significativa no aprendizado entre os métodos de ensino avaliados. Essa seção foi organizada de acordo com os testes realizados e suas respectivas etapas.

8.1. Resultado do simulador como apoio do ensino dos algoritmos de escalonamento de disco (Conhecimento)

Para isso, esta seção é organizada de modo a exibir os resultados do teste de conhecimento tanto da Etapa 1 quanto da Etapa 2. Logo em seguida, é realizada a discussão dos resultados obtidos nos testes de conhecimento em ambas as etapas.

8.1.1. Resultado do teste de conhecimento aplicado na Etapa 1

Por meio do pré-teste (Tabela 6) foi possível detectar que os alunos do ensino médio não possuíam conhecimento sobre o algoritmo FCFS. As Figura 35, Figura 36 e Figura 37 apresentam o quantitativo de acertos obtidos na primeira etapa pelos estudantes no pós-teste de conhecimentos em algoritmos de escalonamento de disco FCFS, de acordo com o método de ensino utilizado: Método tradicional de ensino (aula expositiva), método de ensino auxiliado pelo simulador executado em *desktop* e método de ensino auxiliado pelo simulador executado no *Gear VR*, respectivamente.

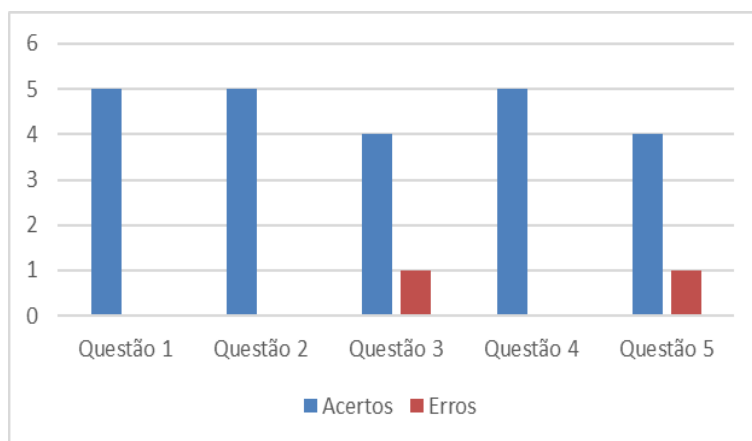


Figura 35: Acertos obtidos no método tradicional de Ensino – FCFS

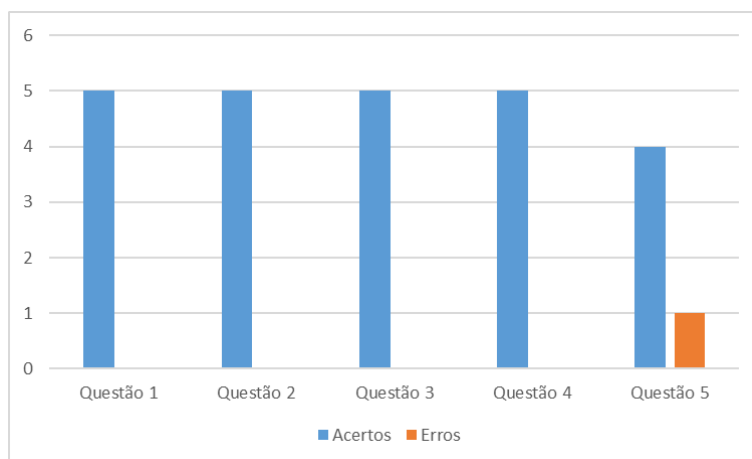


Figura 36: Acertos obtidos no método de ensino auxiliado pelo simulador executado em desktop – FCFS

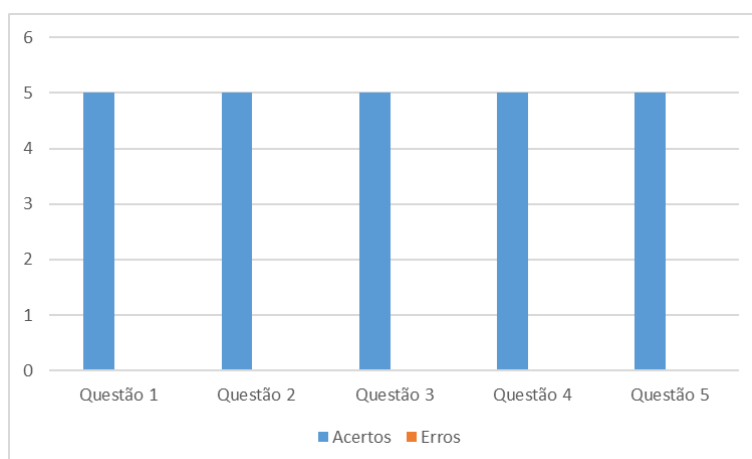


Figura 37: Acertos obtidos no método de ensino auxiliado pelo simulador executado em Gear VR – FCFS

Para o método ANOVA, se $F > F\text{-crítico}$, a hipótese nula deveria ser rejeitada, indicando que houve diferença entre as amostras. Entretanto, na análise variância, Figura

38, observou-se que $F < F\text{-crítico}$ e neste caso, a hipótese nula não deve ser rejeitada, demonstrando que não há diferença entre os métodos de ensino avaliados nos testes de conhecimento de algoritmo de escalonamento de disco. Pode-se, ainda, confirmar que não existe diferença entre os grupos ao avaliar também o valor $P = 0,334$ (maior que o nível de significância estabelecido de 5%).

Anova: factor único						
SUMÁRIO						
Grupos	Contagem	Soma	Média	Variância		
Lousa	5	23	4,6	0,3		
Desktop	5	24	4,8	0,2		
VrBox	5	25	5	0		
ANOVA						
Fonte de variação	SQ	gl	MQ	F	valor P	F crítico
Entre grupos	0,4	2	0,2	1,2	0,334897977	3,885293835
Dentro de grupos	2	12	0,16667			
Total	2,4	14				

Figura 38: Análise estatística dos dados ANOVA

8.1.2. Resultado do teste de conhecimento aplicado na Etapa 2

Através do pré-teste (Tabela 7) foi possível detectar que os alunos do ensino médio não possuíam conhecimento sobre o algoritmo NOOP e PIQ. As Figura 39, Figura 40 e Figura 41 apresentam o quantitativo de acertos obtidos na segunda etapa pelos estudantes no pós-teste de conhecimentos em algoritmos de escalonamento de disco NOOP e PIQ, de acordo com o método de ensino utilizado: método tradicional de ensino (aula expositiva), método de ensino auxiliado pelo simulador executado em *desktop* e método de ensino auxiliado pelo simulador executado no *Gear VR*, respectivamente.

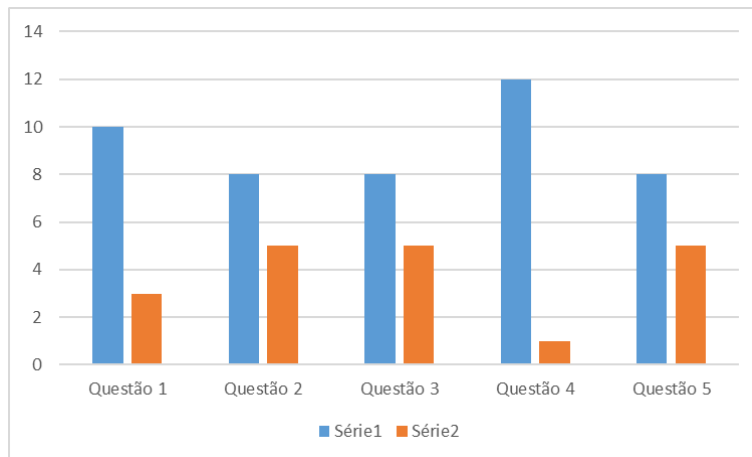


Figura 39: Acertos e erros obtidos no método tradicional de Ensino – NOOP e PIQ

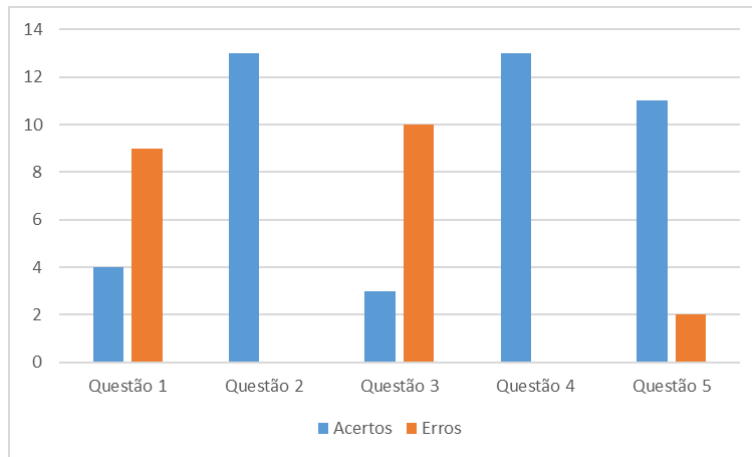


Figura 40: Acertos obtidos no método de ensino auxiliado pelo simulador executado em desktop – NOOP e PIQ

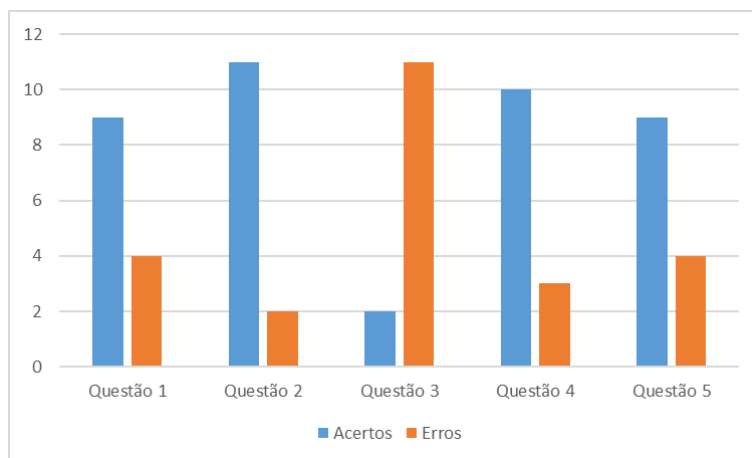


Figura 41: Acertos obtidos no método de ensino auxiliado pelo simulador executado em Gear VR – NOOP e PIQ

Nesta etapa, a análise de variância também foi realizada. Conforme Figura 42, observou-se que $F < F\text{-crítico}$ e neste caso, a hipótese nula não deve ser rejeitada, demonstrando que não há diferença entre os métodos de ensino avaliados nos testes de conhecimento de algoritmo de escalonamento de disco NOOP e PIQ. Pode-se, ainda, confirmar que não existe diferença entre os grupos ao avaliar também o valor $P = 0,910$ (maior que o nível de significância estabelecido de 5%).

Anova: factor único						
SUMÁRIO						
Grupos	Contagem	Soma	Média	Variância		
Lousa	5	46	9,2	3,2		
Desktop	5	44	8,8	24,2		
HMD	5	41	8,2	12,7		
ANOVA						
Fonte de variação	SQ	gl	MQ	F	valor P	F crítico
Entre grupos	2,53333333	2	1,26666667	0,09476309	0,91026224	3,885293835
Dentro de grupos	160,4	12	13,3666667			
Total	162,933333	14				

Figura 42: Análise estatística dos dados ANOVA

8.1.3. Discussão dos resultados do simulador de apoio do ensino dos algoritmos de escalonamento de disco (conhecimento)

A discussão dos testes de conhecimento foi organizada de modo a analisar os testes aplicados sobre algoritmos de escalonamento de disco FCFS (Etapa 1), NOOP e PIQ (Etapa 2). Portanto, essa seção abrange ambas as etapas.

O teste de conhecimento na Etapa 1 foi realizado com 15 alunos de uma população de 40 alunos do primeiro ano de redes do ensino médio conforme o cálculo de amostra considerando população finita com o intuito de aferir a viabilidade do simulador.

Na análise do teste de conhecimento sobre o algoritmo de escalonamento de disco FCFS (Figura 38) foi demonstrado não haver diferença estatística significativa no aprendizado entre os métodos de ensino avaliados. Isso pode ter ocorrido devido à pouca quantidade de alunos que participaram dos testes, o que limita os resultados.

Na Etapa 1 não se considerou a população e sim o número da amostra por considerar que correções provenientes do teste de usabilidade a julgar também o teste motivacional iriam ser feitas no simulador (seção 8.2 e 8.3). Após correções feitas no simulador, provenientes dos testes de usabilidade e motivacional, foi realizado novamente teste de conhecimento sobre os algoritmos de escalonamento de disco NOOP e PIQ com a população de 39 alunos (Etapa 2). A partir dos resultados obtidos, Figura 42, demonstrou novamente não haver diferença estatística significativa no aprendizado entre os métodos de ensino avaliados.

Mesmo não havendo diferença significativa entre os métodos utilizados para lecionar algoritmo de escalonamento de disco, durante os testes foi possível detectar maior motivação dos alunos ao utilizar o simulador tanto no desktop quanto no HMD. Tal elemento é considerado um fator importante no processo e aprendizagem. A diferença de aspectos motivacionais entre eles pode ser visualizada na Figura 56.

Deve ser destacado que durante o teste de conhecimento (Etapa 1) tais perguntas foram feitas pelos alunos durante a utilização do simulador imersivo: “*Como é formado o disco?*”, “*O que são esses quadradinhos vermelhos?*”. Nestes casos, durante a explicação do algoritmo de escalonamento de disco com o apoio do simulador, foi possível ao professor ampliar a explicação além do programado.

8.2. Resultados dos testes de usabilidade

Os resultados sobre o teste de usabilidade, ambiente *desktop* e *Gear VR*, são exibidos conforme as etapas em que foram aplicados, ou seja, etapas 1 e Etapa 2. Logo em seguida é realizada a discussão sobre os resultados obtidos em ambas as etapas.

8.2.1. Resultado do teste de usabilidade aplicado na Etapa 1

As Figura 43 e Figura 44 mostram as respostas das questões do teste de usabilidade aplicado na primeira etapa para avaliação do simulador executado no *desktop* e *Gear VR*.

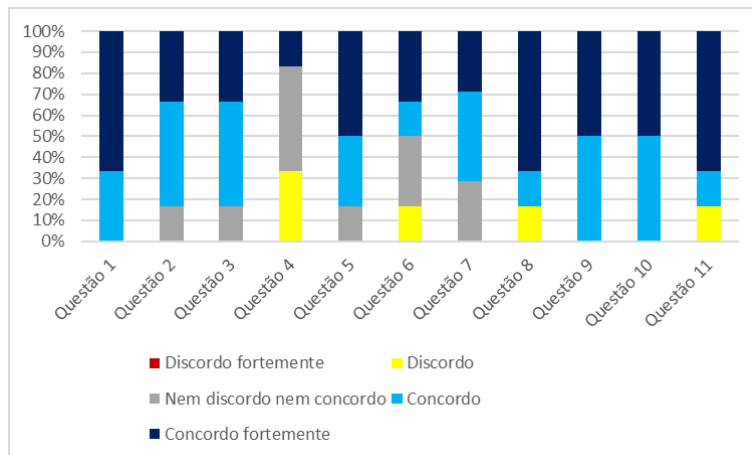


Figura 43: Teste de usabilidade do simulador executando em um *desktop* – Etapa 1

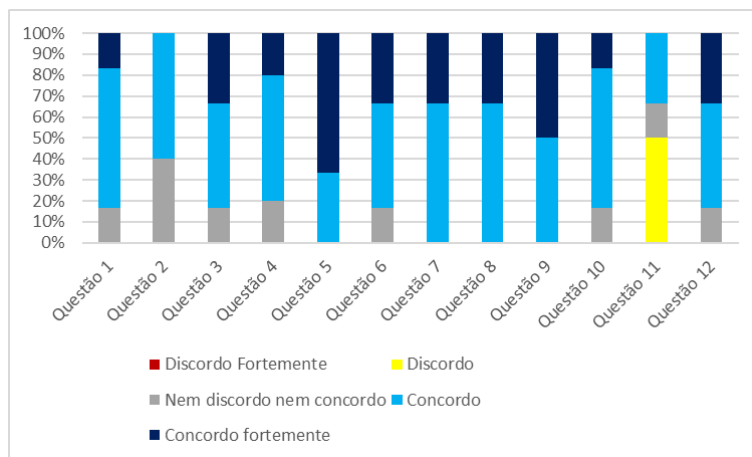


Figura 44: Teste de usabilidade do simulador executando em um *Gear VR* - Etapa 1

A Figura 45 apresenta o média do teste de usabilidade por grupo aplicado na Etapa

1.

Questão/Média	Etapa 1 - Desktop	Etapa 1 - HMD
Questão 1	4,66666667	4
Questão 2	4,16666667	3
Questão 3	4,16666667	4,16666667
Questão 4	3	3,33333333
Questão 5	4,33333333	4,66666667
Questão 6	3,66666667	4,16666667
Questão 7	4,66666667	4,33333333
Questão 8	4,33333333	4,33333333
Questão 9	4,5	4,5
Questão 10	4,5	4
Questão 11	4,33333333	1,83333333
Questão 12		4,16666667

Figura 45: Média de usabilidade por questão – Etapa 1

8.2.2. Resultado do teste de usabilidade aplicado na Etapa 2

As Figura 46 e Figura 47 mostram as respostas das questões do teste de usabilidade aplicado na segunda etapa para avaliação do simulador executado no *desktop* e no *Gear VR*.

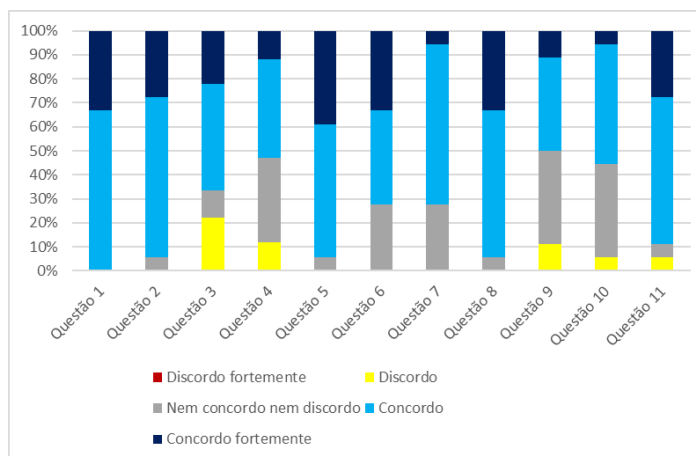


Figura 46: Teste de usabilidade do simulador executando em um *desktop* – Etapa 2

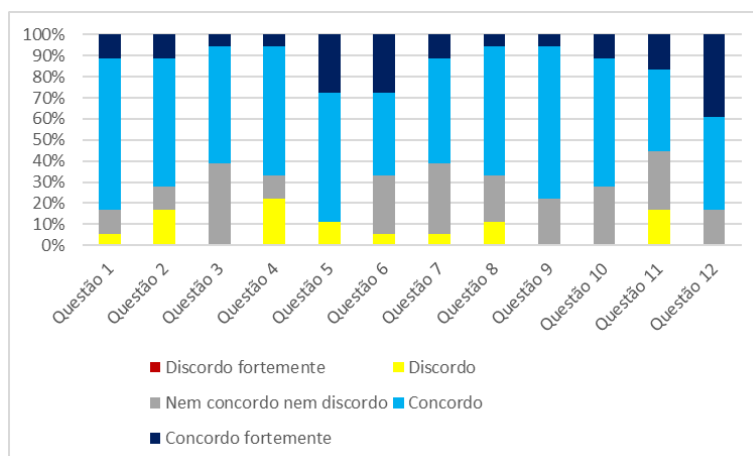


Figura 47: Teste de usabilidade do simulador executando em um Gear VR. – Etapa 2

Figura 48 apresenta a média por questões/grupos.

Questão/Média	Etapa 2 - Desktop	Etapa 2 - HMD
Questão 1	4,33333333	3,88888889
Questão 2	4,22222222	3,66666667
Questão 3	3,66666667	3,66666667
Questão 4	3,33333333	3,5
Questão 5	4,33333333	4,05555556
Questão 6	4,05555556	3,88888889
Questão 7	3,77777778	3,66666667
Questão 8	4,27777778	3,61111111
Questão 9	3,5	3,83333333
Questão 10	3,55555556	3,83333333
Questão 11	4,11111111	3,55555556
Questão 12		4,22222222

Figura 48: Média de usabilidade por questão segunda etapa – Etapa 2

A Figura 49 apresenta o resumo de média geral do teste de usabilidade, aplicado nas etapas 1 e 2, executados no ambiente *desktop* e HMD.

Etapa/Grupo	Desktop	HMD
Média Etapa 1	3,86111111	3,958333
Média Etapa 2	3,92424242	4,126263

Figura 49: Média geral de usabilidade por etapa/grupo

8.2.3. Discussão sobre o teste de usabilidade etapas 1 e 2

A discussão sobre o teste de usabilidade é organizada conforme os resultados obtidos em suas respectivas etapas.

8.2.3.1. Discussão dos resultados do teste de usabilidade aplicado Etapa 1

No que tange ao teste de usabilidade na Etapa 1, as informações coletadas permitiram identificar alguns pontos que despertaram maior interesse nos alunos ou que não os agradou. Analisando os gráficos de usabilidade, Figura 43, Figura 44 e a média de usabilidade por questões, Figura 45, é possível listar os pontos fortes do simulador:

1. Facilidade de uso: pode ser observada pela questão 1, que aponta que aproximadamente 70% dos alunos fortemente concordaram e 30% concordaram com a execução do simulador em um ambiente *desktop* totalizando uma média de 4.66. Quanto ao ambiente *Gear VR*, aproximadamente 60% dos alunos concordaram que o simulador é convidativo ao seu uso, mesmo na primeira utilização totalizando uma média 4;
2. Flexibilidade e eficiência de uso: no ambiente *Gear VR*, conforme avaliado na questão 2, totalizou aproximadamente 60% dos alunos que concordaram que o tempo de resposta às ações dentro do ambiente é aceitável, uma média de 3. Essa avaliação no ambiente de *desktop* foi de aproximadamente 80% entre fortemente concordaram e concordaram, proporcionando uma média 4.1 no *desktop* e média 3 no HMD;
3. Visibilidade e status do sistema: avaliado pela questão 3, no ambiente *Gear VR* foram 30% dos alunos que fortemente concordaram e 50% que concordaram em se sentirem totalmente informados sobre o que estava acontecendo no sistema. As mesmas análises podem ser feitas no ambiente *desktop*;
4. Compatibilidade do sistema com o mundo real: na questão 5, observa-se que as mensagens textuais e audíveis foram explicativas, já que 30% fortemente concordaram e 70% concordaram no ambiente *Gear VR*, uma média de 4.6. No ambiente *desktop* pode-se observar aproximadamente um total de 80% entre concordaram e fortemente concordaram, uma média de 4.3;

5. Prevenção de erros: os textos e áudios explicativos demonstraram atender às expectativas de modo a orientar os alunos em relação ao uso do simulador. Conforme pode ser verificado pela questão 6, 60% fortemente concordaram e 40% concordaram no ambiente *Gear VR*. No ambiente *desktop*, essa porcentagem foi de 50% entre fortemente concordaram e concordaram, proporcionando uma média de 3.6 o que se considerou como positivo;
6. Ajuda e documentação: pode-se observar que a ajuda inicial instruiu os alunos sobre como proceder ao utilizar o simulador. Na questão 7, verificou-se que 30% fortemente concordaram e 70% concordaram no ambiente *Gear VR*, uma média de 4.3, enquanto, no ambiente *desktop*, aproximadamente 70% entre concordaram e fortemente concordaram, uma média de 4.6;
7. Reconhecimento ao invés de lembranças: avaliado pela questão 8, apontou que 30% dos alunos fortemente concordaram e 70% concordaram no ambiente *Gear VR* que as instruções são fáceis de serem reconhecidas, uma média de 4.3. Já no ambiente *desktop*, aproximadamente 80% entre fortemente concordaram e concordaram, uma média 4.3;
8. Compatibilidade do sistema com o mundo real: Na questão 9, foi possível verificar que os objetos que aparecem na tela foram de fácil compreensão para o entendimento dos alunos ao ser adotada no simulador linguagem familiar a dos usuários. Nesse caso, foi boa avaliação em ambos os ambientes, pois 50% fortemente concordaram e 50% concordaram no *Gear VR* e no *desktop*, uma média de 4.5 em ambos os ambientes;
9. Estética e design minimalista: na questão 10, nessa avaliação, obteve-se que 50% dos alunos fortemente concordaram e 50% concordaram no ambiente *desktop* que os objetos apresentados foram suficientes para entendimento do algoritmo, uma média de 4.5. No ambiente *Gear VR*, aproximadamente 60% dos alunos concordaram e 20% fortemente concordaram, uma média 4;
10. Imersão: foi avaliado apenas no ambiente *Gear VR*. No caso, a avaliação pode ser observada pela questão 12 da Figura 44, que mostra que aproximadamente 50% dos alunos concordaram e 30% fortemente concordaram com ter a sensação de se sentir dentro de um ambiente imerso, uma média de 4.1. Nas considerações sobre o simulador (Tabela 9), teve-se como aspecto positivo a

seguinte frase de um dos alunos “Tem como ter uma visão dentro do computador”.

Foram enumerados possíveis correções e melhorias os itens abaixo:

1. Controle e liberdade do sistema: no momento do teste, foi verificado que o sistema não oferece suporte adequado aos alunos ao entrar em determinadas situações indesejadas. Um exemplo disso ocorreu com um aluno ao caminhar no ambiente em que não havia bancadas. Tal observação pode ser avaliada também na questão 4 do teste de usabilidade no ambiente *desktop* que tinha como objetivo avaliar se o simulador dava suporte para fazer ou refazer as ações indesejadas, Neste caso, 30% dos alunos discordaram e 60% nem concordaram, nem discordaram, tendo como média 3;
2. Consistência e padrões: esse item foi avaliado na questão 11. No ambiente de *desktop*, obteve-se uma avaliação positiva, já que 70% dos alunos fortemente concordaram. Porém, no ambiente *Gear VR*, notou-se que 50% dos alunos discordaram, uma média 1.8. Na entrevista após teste de usabilidade, foi sugerida como melhoria da movimentação do personagem, algo que atribuiu-se a essa pontuação negativa no uso do *Gear VR*. Nos aspectos negativos das considerações sobre a aplicação (Tabela 9), a seguinte frase de um dos alunos foi relatada “O movimento no ambiente virtual era um pouco travado”, tornando ainda mais evidente que melhorias no sistema da movimentação do personagem para uso no *Gear VR* precisam ser feitas.

8.2.3.2. Discussão dos resultados do teste de usabilidade aplicado na Etapa 2

Após implementadas melhorias no simulador, o teste de usabilidade foi reaplicado. Na Etapa 2 o resultado do teste de usabilidade do simulador foi similar aos resultados obtidos na Etapa 1, conforme pode ser observado na Figura 49. Como pontos positivos e negativos, enumeram-se:

1. Pontos positivos: Na segunda etapa nos testes de usabilidade, conforme Figura 48, pode-se observar uma melhora significativa na avaliação dos alunos quanto a consistência e padrões passando de uma média de 1.83 na questão 11,

Figura 45 relatado na primeira etapa para uma média de 3.5, Figura 48, na segunda etapa no ambiente HMD.

2. Pontos negativos: Com relação ao controle e liberdade do sistema: foi possível avaliar outros pontos onde o sistema não oferece suporte adequado aos alunos ao entrar em determinadas situações indesejadas. Um exemplo disso ocorreu com um aluno ao caminhar entre o disco magnético e a parede do ambiente precisando neste caso, reiniciar o simulador para prosseguir com os testes.

8.3. Teste Motivacional

Nesta seção são apresentados os resultados dos testes motivacionais, organizados em duas etapas, aplicados no ambiente *desktop* e *Gear VR*. Em seguida, é apresentada a discussão dos testes de aspectos motivacionais conforme suas etapas.

8.3.1. Resultado do teste de aspecto motivacional aplicado na Etapa 1

As Figura 50 exibe o resultado da aplicação do teste de aspectos motivacionais da primeira etapa tanto no ambiente *desktop* quanto no ambiente HMD.

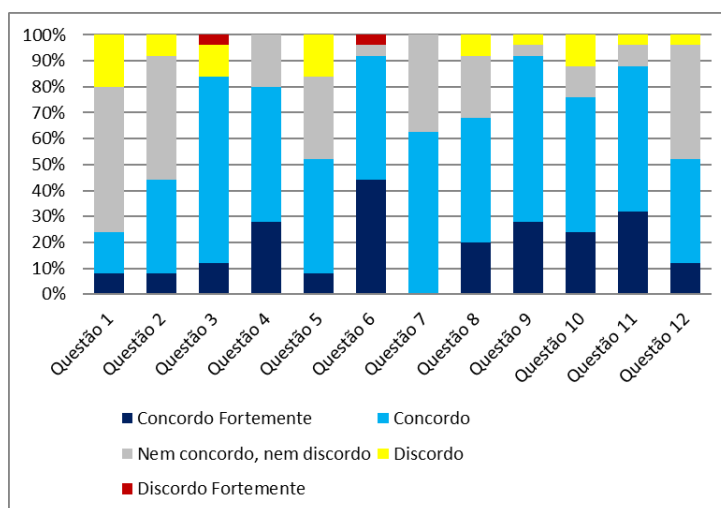


Figura 50: Teste Motivacional Desktop e Gear VR – Etapa 1

A Figura 51 apresenta a média geral do teste de motivacional aplicado na primeira etapa no ambiente *desktop* e HMD.

#	Questão 1	Questão 2	Questão 3	Questão 4	Questão 5	Questão 6	Questão 7	Questão 8	Questão 9	Questão 10	Questão 11	Questão 12
DESKTOP												
Média	2,92	3,33	3,67	3,92	3,25	4,50	3,45	3,83	4,08	4,00	4,00	3,33
Moda	3,00	3,00	4,00	4,00	4,00	5,00	3,00	4,00	4,00	4,00	4,00	3,00
HMD												
Média	3,38	3,54	3,69	4,15	3,69	4,08	3,69	3,77	4,23	3,92	4,38	3,85
Moda	3,00	3,00	4,00	4,00	3,00	4,00	4,00	4,00	4,00	4,00	4,00	4,00

Figura 51: Média motivacional por questão – Etapa 1

8.3.2. Resultado do teste de aspecto motivacional aplicado na Etapa 2

A Figura 52 exibe os resultados da aplicação do teste motivacional aplicado no ambiente *desktop*.

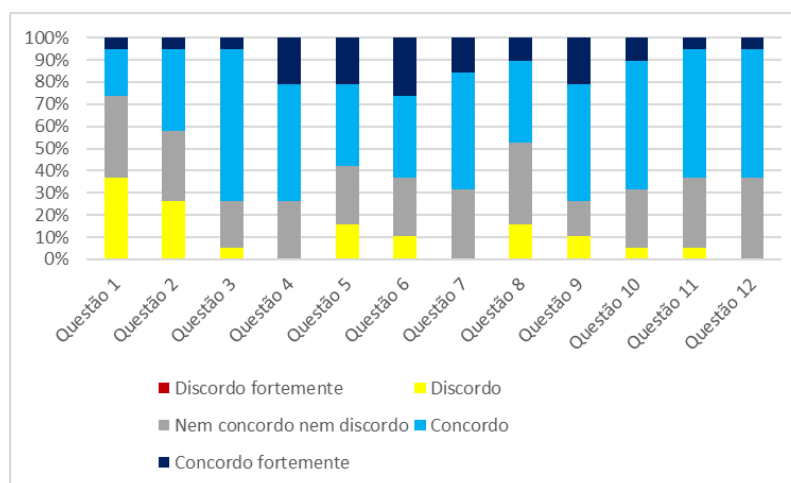


Figura 52: Teste Motivacional *desktop* – Etapa 2

A Figura 53 exibe os resultados da aplicação do teste de aspectos motivacionais da segunda etapa no ambiente *Gear VR*.

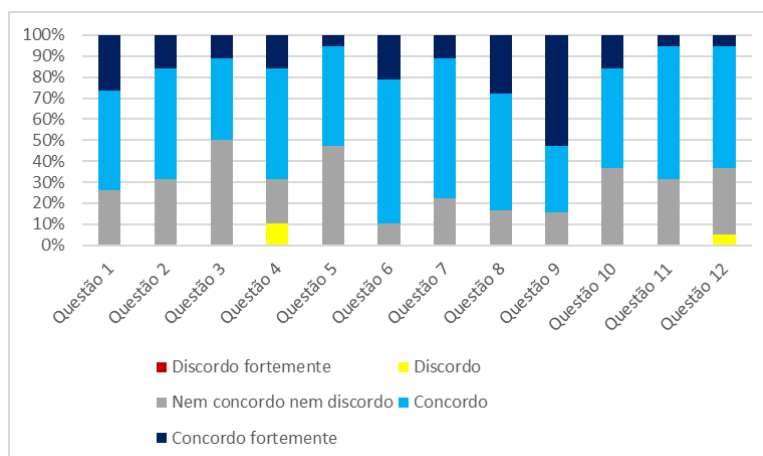


Figura 53: Teste Motivacional Gear VR – Etapa 2

A Figura 54 apresenta o resumo de média por questão da segunda etapa do teste de aspectos motivacionais por grupo.

Questão/Média	Etapa 2 - Desktop	Etapa 2 - HMD
Questão 1	2,947368421	4
Questão 2	3,210526316	3,842105263
Questão 3	3,736842105	3,421052632
Questão 4	3,947368421	3,736842105
Questão 5	3,631578947	3,578947368
Questão 6	3,789473684	4,105263158
Questão 7	3,842105263	3,684210526
Questão 8	3,421052632	3,894736842
Questão 9	3,842105263	4,368421053
Questão 10	3,736842105	3,789473684
Questão 11	3,631578947	3,736842105
Questão 12	3,684210526	3,631578947

Figura 54: Média por questão – Etapa 2

A Figura 55 apresenta o resumo de média do teste de aspectos motivacionais por grupo, tanto da primeira etapa quanto da segunda etapa.

Etapa/Grupo	Desktop	HMD
Média Etapa 1	3,666667	3,895833
Média Etapa 2	3,618421	3,815789

Figura 55: Média de aspecto motivacional por fase/grupo

8.3.3. Discussão dos testes de aspectos motivacionais aplicados na Etapa 1 e Etapa 2

A discussão dos testes de aspectos motivacionais é organizada conforme suas etapas, ou seja, Etapa 1 e 2.

8.3.3.1. Discussão do teste de aspecto motivacional aplicado na Etapa 1

No âmbito do teste motivacional, Tabela 10, pode enumerar as seguintes contribuições na primeira etapa elencando pontos positivos e negativos em cada ambiente conforme Figura 50 e Figura 51.

1. Atenção: O assunto é abordado nas questões 1, 2 e 8. Como resultado na questão 1, (Figura 51), o valor médio das respostas sobre a interface ser atraente para o ambiente *desktop* foi de 2,92 enquanto para o HMD a média foi de 3,38 mostrando que a percepção do usuário da interface é melhor no HMD. Na questão 2, a agradabilidade da interface foi melhor no ambiente HMD sendo 3,54 no ambiente HMD e 3,33 no ambiente *desktop*. Na questão 8 onde se avaliou se o simulador mantém o aluno motivado a continuar a utilizar o simulador observa-se 3,83 no ambiente *desktop* e 3,77 no ambiente HMD algo considerado positivo na avaliação. Porém, pode-se afirmar que o design da interface, baseado na questão 1, é um item que requer melhorias. Ou seja, tornar a interface mais agradável e atraente deve ser um ponto de melhoria para o projeto.
2. Satisfação: O tema é abordado nas questões 3 e 7. Os dois grupos também obtiveram respostas similares sobre aprender coisas surpreendentes e/ou inesperadas com o simulador, a média das respostas é de 3,67 no *desktop* e 3,69 no ambiente HMD para a questão 3. O segundo aspecto sobre a satisfação era se o aluno estava tendo progresso ao utilizar o simulador, o que foi aferido na questão 7. Neste caso, a média foi de 3,45 no ambiente *desktop* e 3,69 no ambiente HMD. No tangente à satisfação, os alunos se sentiram mais satisfeitos em utilizar o simulador no ambiente HMD conforme análise das médias. Pelo gráfico na Figura 50 a satisfação dos alunos ao usarem o simulador nos dois ambientes foram positivo onde, teve-se entre concordo e

fortemente concordo aproximadamente 80% na questão 3 e na questão 7 teve-se 70% dos alunos concordaram.

3. Relevância: apresentado aos usuários nas questões 4 e 5. Na questão 4 foi avaliado se o conteúdo era relevante aos alunos do curso de redes de computadores, neste caso, obteve-se uma média de 3.92 para o ambiente *desktop* e 4.15 para o ambiente HMD. Conforme Figura 50, aproximadamente 80% dos alunos marcaram entre concordam ou fortemente concordam sobre a relevância do conteúdo abordado. Na questão 5, obteve-se uma média de 3.25 no ambiente *desktop* e 3.69 no ambiente HMD. Pode-se concluir que a relevância do conteúdo abordado pelo simulador demonstrou atender aos requisitos.
4. Confiança: avaliado pela questão 6 e 9. Na questão 6 no ambiente *desktop* demonstrou-se ser mais confiante que o ambiente HMD tendo como média 4,5 no ambiente *desktop* e 4,08 no ambiente HMD. Entretanto, a média na questão 9 que avaliava se o aluno voltaria a utilizar o simulador novamente teve-se uma média de 4,23 no HMD e 4,08 no ambiente *desktop*.
5. Reação: O tema foi abordado no questionário de aspectos motivacionais nas questões 10, 11 e 12. As questões 10 e 11 abordaram a compreensão e retenção do conhecimento sobre tema apresentado pelo simulador e, em ambas as questões, obtiveram-se resultados similares no ambiente *desktop* tendo como média 4. Entretanto, no ambiente HMD a média de respostas para a aplicação do conhecimento foi maior sendo, 3,85 para HMD e 3,33 para *desktop* no qual, pode ser visualizado na questão 12.

A Figura 56 resume a discussão dos aspectos motivacionais por grupo enfatizando pela análise o ambiente de melhor desempenho.

Aspecto	Ambiente com melhor desempenho
Atenção	HMD
Satisfação	HMD
Relevância	Similar
Confiança	Desktop
Reação (Conhecimento)	HMD

Figura 56: Resumo das discussões por ambiente

Outros aspectos positivos foram percebidos dos alunos nas questões abertas. As respostas mais recorrentes dos alunos foram: *“facilidade do uso do simulador”*, *“as explicações textuais e áudios auxiliam no uso do simulador”*, *“o uso da tecnologia para apresentar um tema de difícil entendimento”*. Alguns aspectos para a melhoria do projeto também foram apontados como: *“a melhora da interface gráfica”* e *“controle de volume para os áudios”*. Um único fator de desconforto foi relatado na utilização do simulador no HMD em que o aluno se sentiu constrangido em fazer os movimentos na sala junto ao grupo.

A interação, juntamente com estímulos provenientes de dispositivos como HMD, provocou nos alunos uma sensação de imersão permitindo que eles transferissem o senso de presença do ambiente real para o virtual ao utilizar o simulador no ambiente de RV. Esse fator foi relatado nas questões abertas do teste de usabilidade *“Tem como ter uma visão dentro do computador”*. Devido à transferência do senso de presença do ambiente real para o virtual, ao usar o simulador imersivo os alunos tiveram menos interferência do ambiente real (exemplo, redes sociais, uso dos dispositivos como celulares, etc) o que permitiu a eles focarem mais no conteúdo apresentado pelo simulador em sala de aula fator este, importante para o resultado dos aspectos motivacionais. Além disso, foi possível durante o uso do simulador detectar maior curiosidade dos alunos quanto ao conteúdo apresentado neste caso, perguntas foram feitas além do conteúdo apresentado (explicado na seção 8.1.3) transformando assim, o processo de construção de conhecimento de forma dinâmica, criativa e colocando o aluno no centro dos processos de aprendizagem.

Vale destacar que nas questões abertas, tabela 3, foi possível verificar que os alunos não tinham experiência de uso com simulador imersivo fator este, que pode ter

também influenciado o melhor desempenho do simulador nos testes de aspecto motivacionais em comparação com o mesmo simulador no ambiente *desktop*.

8.3.3.2. Discussão do teste de aspecto motivacional aplicado na Etapa 2

Antes de realizar o teste de aspectos motivacionais na Etapa 2, o simulador passou por mudanças no sistema de áudio. Os textos explicativos foram digitados em sites web e por meio deles gerou-se uma narração quase realista. As cores das interfaces gráficas do simulador foram alteradas como exemplo, o piso que passou a ter textura imitando pedras. Após melhorias o teste motivacional foi reaplicado.

Apesar de se ter obtido resultados similares na análise dos testes de aspectos motivacionais das duas etapas, na segunda etapa dos testes de aspectos motivacionais, Figura 54, pode-se observar uma pequena melhora na avaliação dos alunos quanto à interface gráfica (questão 1). Também não se obteve relatos dos alunos para melhoria da interface gráfica nas questões abertas, assim como sugestões para melhoria do som.

8.4. Considerações finais do capítulo

A motivação consiste na quantidade de esforço que uma pessoa está disposta a exercer em busca de uma meta no contexto da aprendizagem e é considerada um fator importante no processo educacional. Por meio do uso do simulador imersivo e interativo foi possível aferir a motivação do aluno nos ambientes HMD e *desktop*. Assim como a motivação, a usabilidade também tem seu papel de destaque na melhoria do uso do simulador. Por meio dos testes feitos, foi possível detectar pontos a serem melhorados no uso do simulador como controle e liberdade do usuário, uma vez que o sistema não oferece suporte adequado aos alunos ao entrar em determinadas situações indesejadas, e também, melhoria na interface gráfica item de constante melhoria.

9. CONCLUSÕES E TRABALHOS FUTUROS

Embora o uso da RV seja visto como positivo no processo educacional, nos testes de conhecimento aplicados aos alunos do curso de redes de computadores integrada ao ensino médio, o uso do simulador imersivo não foi mais eficiente que o uso do simulador em um ambiente 2-D (não-imersivo) ou que o método tradicional quando usado para aprendizagem no processo de construção de conhecimento de algoritmos de escalonamento de discos FCFS, NOOP e PIQ. Neste caso, não houve diferença significativa na avaliação entre os métodos de ensino avaliados.

Vale destacar que o teste de conhecimento foi realizado com uma amostra pequena de alunos, o que limita os resultados. Portanto, há necessidade de realizar testes de conhecimento com um número maior de alunos e, além disso testes para avaliar a retenção a longo prazo do conhecimento adquirido entre os métodos lousa, *desktop* e HMD.

Apesar dos testes de avaliação de conhecimento não demonstrarem uma diferença significativa, pode-se afirmar a partir dos testes motivacionais na primeira e segunda etapa, que o simulador no ambiente de RV (imersivo) motivou os alunos durante o aprendizado de um conteúdo abstrato do contexto de SOs (algoritmo de escalonamento de disco) e estimulou a curiosidade destes quanto à estrutura interna do disco, ampliando a explicação e a construção do conhecimento em sala de aula. A motivação é um dos fatores considerados fundamentais no processo de aprendizagem. Neste sentido, pode-se observar que com o uso da RV por meio de um simulador imersivo e interativo juntamente com recursos textuais, mensagens auditivas e imagens gráficas 3D como: cubos vermelhos que caem ao lado do disco magnético de modo a representar a abstração das requisições de acesso a disco, componentes dos discos o mais próximo possível do real e monitores nos quais são possíveis visualizar a ordem das requisições, foi possível estimular essa motivação.

Vale destacar que ao usar o simulador imersivo, juntamente com estímulos provenientes de dispositivos como HMD, os alunos tiveram menos interferência do ambiente real, o que permitiu a eles focarem mais no conteúdo apresentado pelo simulador em sala de aula, fator este importante para o resultado dos aspectos motivacionais e para se obter maior curiosidade dos alunos quanto ao conteúdo apresentado.

Os testes usabilidade e motivacional também apontaram que melhorias devem ser realizadas no simulador como, por exemplo, controle e liberdade do usuário. Porém, antes de novas implementações no simulador e novos testes com os alunos se faz necessário realizar o teste de usabilidade com especialistas.

Assim, conclui-se que apesar dos testes de avaliação de conhecimento não demonstrarem uma diferença significativa, por meio dos demais testes, pode-se afirmar que a RV em seu conjunto reúne especificidades e atributos tais como: transformar o abstrato em concreto, permitir a interação em tempo real utilizando dispositivos multissensoriais a partir de um ambiente tridimensional, permitir simulações da realidade ou além dela, transferir o senso de “presença” do mundo real para o virtual, entre outros. Tais características a tornam um instrumento poderoso a serviço de todos que buscam a evolução da educação.

No futuro, planeja-se melhorar o simulador, adicionando novos algoritmos de escalonamento de discos magnéticos e também dos discos sólidos, realizar teste de usabilidade com especialista assim como, novos testes (conhecimento e aspecto motivacional) com os alunos. Além disso, planeja-se possibilitar que os usuários forneçam parâmetros durante as simulações.

10. Referências

- Agrawal, N., Vijayan P., Wobber, T., Davis, J. D., Manasse, M. S, and Panigrahy, R., Design tradeoffs for ssd performance. In USENIX Annual Technical Conference, volume 57, 2008.
- Blender, URL <https://www.blender.org/> , último acesso: maio 2018.
- Bogoni, T. N.; Gomes, O. B., Pinho, M. S., Sistemas de Realidade Virtual: Que Hardware Utilizar?, Tendências e Técnicas em Realidade Virtual e Aumentada, v. 5, p. 43-64, maio/2015.
- Bowman D. A., McMahan R. P., Virtual reality: how much immersion is enough? Computer, 40(7), 2007.
- Brandão, E. J. R., Repensando modelos de avaliação de software educacional. 3º Simpósio de Investigação e desenvolvimento de software educativo, 1998.
- Bucy, J. S., Ganger, G. R., The DiskSim simulation environment version 3.0 reference manual. School of Computer Science, Carnegie Mellon University, 2003.
- Burdea, G., Coiffet, P., Virtual Reality Technology. John Wiley & Sons, 2003
- Cardoso, A., Kirner, C., Júnior, E. L., and Kelner, J. Tecnologias e ferramentas para o desenvolvimento de sistemas de realidade virtual e aumentada. Editora Universitária UFPE, 1-19, 2007.
- Coulter, R., Saland, L., Caudell ,T., Goldsmith, T. E., Alverson, D. The effect of degree of immersion upon learning performance in virtual reality simulations for medical education. In Medicine Meets Virtual Reality, 15:155, 2007.
- Dede, C., Grotzer, T. A., Kamarainen, A., and Metcalf, S. J. (2017). Virtual reality as an immersive medium for authentic simulations. In Virtual, Augmented, and Mixed Realities in Education (pp. 133-156). Springer, Singapore.
- Deitel HM, PJ Deitel, and DR Choffnes. Sistemas Operacionais, volume 3. Prentice São Paulo, 2005.
- Denning, P. J. Effects of scheduling on file memory operations. In Proceedings of the April 18-20, 1967, spring joint computer conference, pages 9–21. ACM, 1967.

- Dores, A. R., Barbosa, F., Marques, A., Carvalho, I. P., de Sousa, L., e Castro-caldas, A. Realidade Virtual na Reabilitação: Por Que Sim e Por Que Não? Uma Revisão Sistemática. *Acta Médica Portuguesa*, 25(6), 2012.
- Dunn, M. P., A new I/O scheduler for solid state devices. PhD thesis, Texas A & M University, 2010.
- Dayan, N., Martin Kjær Svendsen, Matias Bjorling, Philippe Bonnet, and Luc Bouga nim. Eagletree: exploring the design space of ssd-based algorithms. *Proceedings of the VLDB Endowment*, 6(12):1290–1293, 2013.
- Dawley, L., Dede, C., Situated learning in virtual worlds and immersive simulations . 2013.
- Ferreira, B. M., Rivero, L., Lopes, A., Marques, A. B., and Conte, T. UsabiliCity: um jogo de apoio ao ensino de propriedades de usabilidade de software através de analogias. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 25, No. 1, p. 1273), 2014.
- Gao, C., Shi, L., Zhao, M., Xue, C. J., Wu, K., and Sha, E. H. M. Exploiting parallelism in I/O scheduling for access conflict minimization in flash-based solid state drives. In *Mass Storage Systems and Technologies (MSST), 2014 30th Symposium on* (pp. 1-11). IEEE, 2014.
- Gnecco, B. B., Guimarães, M. P. Ambientes de Hardware e Software para Aplicações de Realidade Virtual e Aumentada. Capítulo do livro *Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações*, 73, 2007.
- Guimarães, M. P., Gnecco, B. B., Damazio, R. Ferramentas para Desenvolvimento de Aplicações de Realidade Virtual e Aumentada. Capítulo do livro *Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações* (2007): 109.
- Guimarães, M. P. "Um ambiente para o desenvolvimento de aplicações de Realidade Virtual baseadas em aglomerados gráficos". Escola Politécnica da Universidade de São Paulo, São Paulo (2004).
- Jacobsen, A. Implementing and testing the APEX I/O scheduler in Linux, Master's thesis, 2007.

- Jung, Myoungsoo, Ellis, H. W. III, and Mahmut K., Physically addressed queueing (paq): improving parallelism in solid state disks. In *ACM SIGARCH Computer Architecture News*, volume 40, pages 404–415. IEEE Computer Society, 2012.
- Jung, Myoungsoo, Wonil Choi, Shekhar Srikantaiah, Joonhyuk Yoo, and Mahmut T Kandemir. Hios: A host interface i/o scheduler for solid state disks. In *ACM SIGARCH Computer Architecture News*, volume 42, pages 289–300. IEEE Press, 2014.
- Kaufmann, H., Schmalstieg, D., Wagner, M. Construct3D: a virtual reality application for mathematics and geometry education. *Education and information technologies*, 5(4), 263-276, 2000.
- Keller, J. M. *Motivational design for learning and performance: The ARCS model approach*. Springer Science & Business Media, 2009.
- Kim, Y., Brendan Tauras, Aayush Gupta, and Bhuvan Urgaonkar. Flashsim: A simulator for nand flash-based solid-state drives. In *Advances in System Simulation*, 2009. SIMUL'09. First International Conference on, pages 125–131. IEEE, 2009.
- Kirkpatrick, D., Kirkpatrick, J. D. *The Kirkpatrick Four Levels*. Kirkpatrick Partners, 2011.
- Kirkpatrick, D. L. *Evaluating training programs*. Tata McGraw-Hill Education, 1975.
- LaViola Jr, J. J., Kruijff, E., McMahan, R. P., Bowman, D., Poupyrev, I. P., *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.
- Likert, R. A technique for the measurement of attitudes. *Archives of psychology*, 1932.
- Lin, H., Chen, M., Lu, G., Zhu, Q., Gong, J., You, X., Hu, M. Virtual geographic environments (VGEs): a new generation of geographic analysis tool. *Earth-Science Reviews*, 126, 74-84, 2013.
- Liu, D., Dede, C., Huang, R., Richards, J. (Eds.), *Virtual, Augmented, and Mixed Realities in Education*. Springer, 2017.
- Machado, F. B., MAIA, L. P., *Arquitetura de sistemas operacionais*. LTC, 2007.
- Martins V. F., Moura, L. A. Evaluation methodology criteria for automatic transcription system of radiology reports. *Communications in Computer and Information Science*, 3:194–203, 2011.

- Martins, V. F., Guimarães M. P., Correa, A. G., Usability test for augmented reality applications. In Computing Conference (CLEI), 2013 XXXIX Latin American, pages 1–10. IEEE, 2013.
- Martins, V. F., Corrêa, A. G. D., Guimarães, M. P., Usability inspection of a smart tv. In Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on, pages 1–7. IEEE, 2014.
- Maziero, C. A. "Sistemas Operacionais: Conceitos e Mecanismos." Livro aberto. Acessível em: <http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php> (2014).
- Morimoto, C. E., Hardware, V. O Guia Definitivo. Editora Sulina, 2011.
- Muller, M. J., Matheson, L., Page, C., and Gallup, R. Methods & tools: participatory heuristic evaluation. *interactions*, 5(5), 13-18, 1998
- Moore, D. S., McCabe, G. P., DUCKWORTH, W. M., & Sclove, S. L. A prática da estatística empresarial: como usar dados para tomar decisões. Rio de Janeiro: LTC, 2006.
- Nielsen, J. Usability inspection methods. In Conference companion on Human factors in computing systems, pages 413–414. ACM, 1994.
- Nielsen, J. Why you only need to test with 5 users, 2000. URL <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- Panda 3D. URL <https://www.panda3d.org/> último acesso: Setembro 2018
- Powell, W., Powell, V., Brown, P., Cook, M., Uddin, J. Getting around in google cardboard—exploring navigation preferences with low-cost mobile VR. In Everyday Virtual Reality (WEVR), 2016 IEEE 2nd Workshop on (pp. 5-8). IEEE.
- Pratt, S., Heger, D. A., Workload dependent performance evaluation of the linux 2.6 i/o schedulers. In Proceedings of the Linux Symposium, Vol. 2, pp. 425-448, 2004.
- Ramos, A. L. B. M., Metodologias de desenvolvimento de sistemas. 1. ed. Rio de Janeiro: Sesus, 2017.
- Robbins, S. A disk head scheduling simulator. *ACM SIGCSE Bulletin*, 36(1): 325–329, 2004.

- Rocha, P. E., Bona, L. C. E. Sistema de armazenamento compartilhado com qualidade de serviços e alto-desempenho. Dissertação de mestrado, Universidade Federal do Paraná (UFPR), 2011.
- Rolland, Jannick P.; Davis, Larry; Baillet, Yohan. A survey of tracking technology for virtual environments. *Fundamentals of wearable computers and augmented reality*, v. 1, p. 67-112, 2001.
- Roussou, M. A VR playground for learning abstract mathematics concepts. *IEEE computer graphics and applications*, 29(1), 2009.
- Siscoutto, R., Costa R. M., Realidade virtual e aumentada: uma abordagem tecnológica. Porto Alegre: SBC, 2008.
- Silberschatz, A., Galvin, P. B., and Gagne, G. *Sistemas operacionais: conceitos e aplicações*. Campus, 2001.
- Slater, M., A note on presence terminology. *Presence connect*, 3(3):1–5, 2003.
- Slater, M., *Implicit Learning Through Embodiment in Immersive Virtual Reality*. In *Virtual, Augmented, and Mixed Realities in Education* (pp. 19-33). Springer, Singapore, 2017.
- Souza., P. S. L. Simulador de algoritmos escalonador de disco. URL http://lasdpc.icmc.usp.br/~ssc640/grad/ec2015/scheduling_simulator/. Acessado em Junho, 2018.
- Steed, A., & Julier, S. (2013, March). Design and implementation of an immersive virtual reality system based on a smartphone platform. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on* (pp. 43-46). IEEE.
- Steuer J., Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4):73–93, 1992.
- Sutcliffe, A., and Gault, B. Heuristic evaluation of virtual reality applications. *Interacting with computers*, 16(4), 831-849, 2004.
- Tanenbaum, A. S. and Van S, M. "Sistemas Distribuídos: princípios e paradigmas.", 2 ed, São Paulo, Pearson, 2007 .
- Tanenbaum, A. S. *Sistemas operacionais modernos*. São Paulo, 2010.

- Tori R., Hounsell, M. S., *Introdução a Realidade Virtual e Aumentada*. Editora SBC, 2006.
- Tori R., Cláudio K., Robson A. S., *Fundamentos e tecnologia de realidade virtual e aumentada*. *Symposium on Virtual and Augmented Reality*, 2017.
- Suranauwarat, S. A disk scheduling algorithm simulator. *The ASEE Computers in Education (CoED) Journal*, 8(3), 2017.
- Unity, “Unity Soluções,” <https://unity.com/pt/solutions>, último acesso: Agosto 2018.
- Unity, “Unity Documentation,” <https://docs.unity3d.com>, último acesso: Novembro 2018.
- Wazlawick, R. S. *Análise e Projeto de Sistemas de Informação Orientados a Objetos*. 1. ed. Rio de Janeiro: Elsevier Editora, 2004.
- Zanduetta J. Track Animation, 2013. URL <https://sourceforge.net/projects/trackanimation>.
- Zuffo, J. A., Soares, L. P., Zuffo, M. K., & Lopes, R. D. Caverna digital-sistema de multiprojeção estereoscópico baseado em aglomerados de pcs para aplicações imersivas em realidade virtual. In *IV Symposium on Virtual and Augmented Reality*, 2001.
- 3D Max, URL <https://www.autodesk.com.br/> último acesso: Maio 2018.