

REALIDADE ESTENDIDA (XR) E IMMERSIVE LEARNING

INTRODUÇÃO

Aliadas ao poder de processamento gráfico dos computadores pessoais, estão disponíveis, no mercado de softwares, interfaces gráficas cada vez mais intuitivas. Para além dos computadores pessoais, dispositivos móveis de alta resolução, teclados ergonômicos, câmeras de captura de movimento, óculos de realidade aumentada (RA) e/ou virtual (RV) e videogames são ferramentas capazes de representar mundos e seres vivos tridimensionais, permitindo-nos interagir com eles, tão verossímeis que chegam a dar vertigem. Todo esse aparato é capaz de produzir tipos variados de emoções por meio de suas mais diversas formas de interface, capturando a atenção de jovens e adultos por horas a fio, tamanha a riqueza de detalhes que alcançam, imitando artificialmente os fenômenos físicos, químicos e biológicos observáveis na natureza.

Anteriormente, caracterizamos, em detalhes, os conceitos de Realidade Mista (RM) e Realidade Mediada (XYR), apresentando uma abordagem aprofundada dessas tecnologias, além de trazer alguns exemplos de como elas estão inseridas no nosso dia a dia. Sequencialmente, o conteúdo desta etapa tem por objetivos:

- apresentar alguns exemplos de dispositivos de entrada e saída para sistemas de RV, com enfoque especial nos *headsets*;
- abordar os requisitos básicos e fundamentais no processo de desenvolvimento de sistemas de RV;
- caracterizar os ambientes virtuais e os conceitos de modelagem e programação a eles atrelados.

Para isso, ela foi estruturada da seguinte forma:

- dispositivos de entrada e saída para sistemas de RV;
- requisitos básicos para um sistema de RV;
- modelagem e programação em ambientes virtuais;
- processo de desenvolvimento de sistemas de RV;
- conclusão e próximos passos.

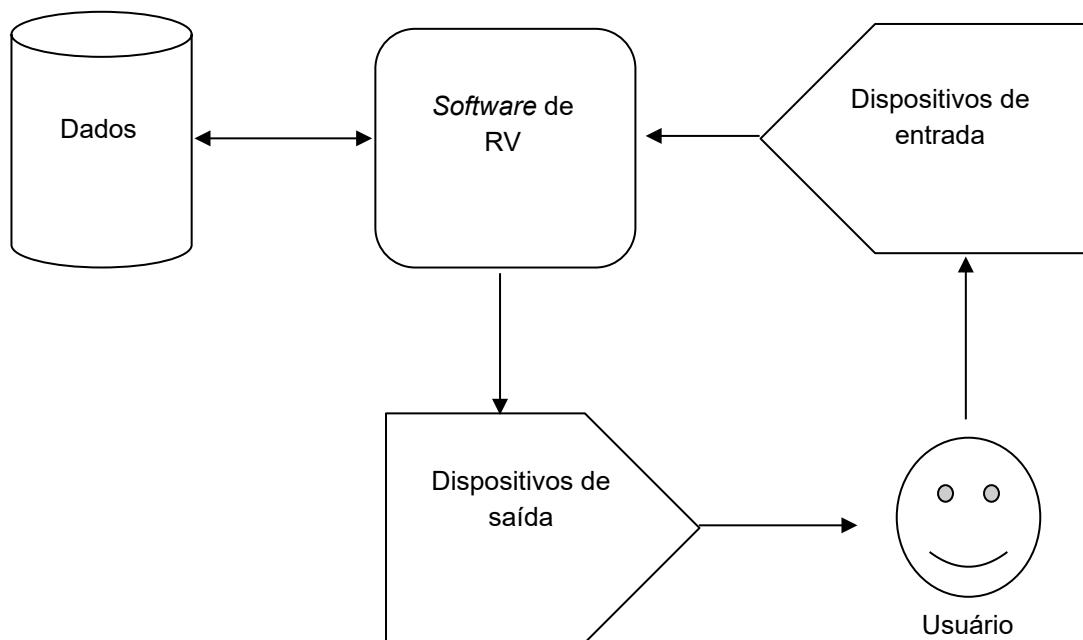
Aproveite e tenha bons estudos!

TEMA 1 – DISPOSITIVOS DE ENTRADA E SAÍDA PARA SISTEMAS DE RV

A utilização de dispositivos específicos para entrada e saída de informações em um sistema de RV visa a aumentar os níveis de imersão do usuário e a prover modos mais intuitivos de interação. Os dispositivos utilizados em um sistema de RV podem ser divididos em duas categorias: os de entrada e os de saída, responsáveis por toda a comunicação usuário-sistema (Machado; Cardoso, 2006).

Os dispositivos de entrada procuram captar movimentos e ações do usuário para alimentar o sistema de RV, que retornará o resultado do processamento dessa interação, na forma de estímulos a pelo menos um dos cinco sentidos humanos, por meio dos dispositivos de saída. Esses dispositivos são específicos para que o sistema de RV possa prover um meio intuitivo de comunicação usuário-sistema. No entanto, grande parte dos sistemas de RV integra, também, dispositivos convencionais, como *mouse* e teclado, os quais, na maioria das vezes, são utilizados para selecionar menus e objetos ou para navegar pelo ambiente. A Figura 1 apresenta um esquema com os elementos-chave de um sistema de RV; é possível notar, nela, a importância dos dispositivos de entrada e de saída de dados (Machado; Cardoso, 2006).

Figura 1 – Elementos básicos dos sistemas de RV

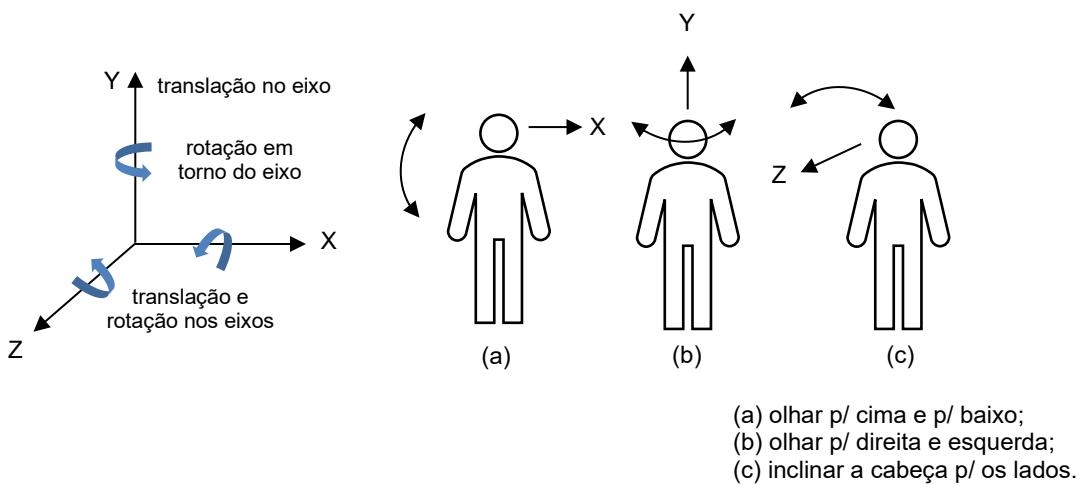


Fonte: elaborada por Guerra, 2023, com base em Machado; Cardoso, 2006, p. 40.

1.1 Dispositivos de entrada de dados

Os dispositivos de entrada de dados para sistemas de RV são utilizados para enviar informações sobre ações do usuário para o sistema. Basicamente, eles podem ser de dois tipos: de interação ou de rastreamento. Em ambos os casos, as ações do usuário são identificadas em um espaço tridimensional. É importante observar que objetos dos ambientes virtuais geralmente podem mover-se com seis graus de liberdade (os chamados *six degrees of freedom*, ou 6DOF), o que implica a possibilidade de três rotações e três translações (Figura 2) (Machado; Cardoso, 2006).

Figura 2 – Navegação com seis graus de liberdade



Fonte: elaborada por Guerra, 2023, com base em Tori; Kirner, 2006, p. 9.

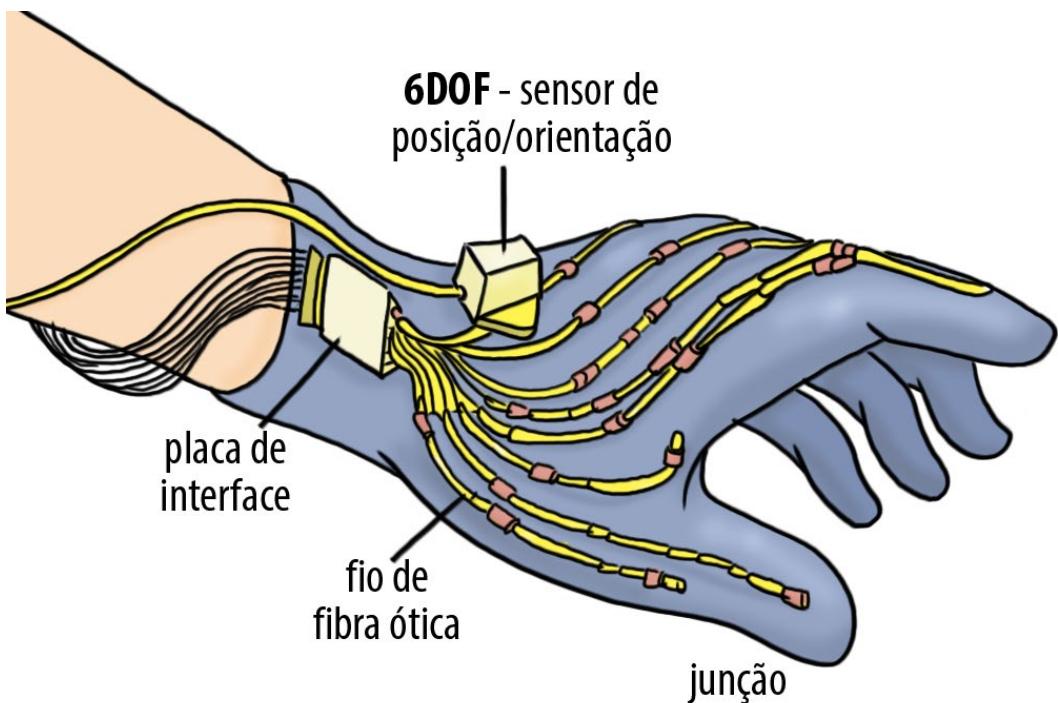
1.1.1 Dispositivos de interação

Os dispositivos de interação permitem ao usuário a movimentação e a manipulação de objetos no mundo virtual, de forma direta ou indireta. Desse modo, tais dispositivos conectam ações do usuário com elementos de cena do ambiente virtual. Existem diferentes dispositivos de interação com diferentes finalidades, sendo importante escolher o mais adequado para a aplicação da RV em questão. E a escolha de um tal dispositivo deve levar em conta não apenas a finalidade do sistema, mas também os pacotes computacionais utilizados, como linguagens e *toolkits*, pois a eficiência do sistema vai depender da capacidade de os pacotes aproveitarem as características do dispositivo (Machado; Cardoso, 2006).

Vale ressaltar que, devido ao avanço tecnológico, novos dispositivos são constantemente desenvolvidos, com objetivo de oferecer modos mais intuitivos de interação. Mesmo assim, é possível citar alguns tipos mais comuns, sendo eles (Machado; Cardoso, 2006):

- **dispositivos com 2DOF:** a interação em um mundo virtual nem sempre requer o uso de um dispositivo complicado e/ou caro. Muitas tarefas podem ser executadas com simples dispositivos com 2DOF, como um *mouse* ou um *joystick*. Apesar de limitar as possibilidades de movimento, tais dispositivos reduzem o tempo de resposta do sistema (seus eventos são mais rapidamente processados) e são de fácil utilização;
- **dispositivos com 6DOF:** permitem uma movimentação bastante ampla quando utilizados em sistemas de RV, pois possibilitam movimentos em todas as direções do espaço em 3D, incluindo os de rotação. Algumas empresas procuraram modificar o projeto do *mouse* padrão para que ele pudesse funcionar com sensores de trajetória de 6DOF ou 3DOF;
- **luvas de dados:** são utilizadas para reconhecer e capturar os movimentos dos dedos da mão do usuário. Na maioria dos equipamentos disponíveis, são utilizados sensores mecânicos ou de fibra ótica – as versões mais populares empregam a fibra-ótica. O esquema básico desse tipo de luva pode ser visto na Figura 3;
- **sensores de entrada biológicos:** processam atividades ditas *indiretas*, como comandos de voz e sinais elétricos musculares. Estudos sobre reconhecimento de voz existem há mais de 20 anos, e, em sistemas de RV, o reconhecimento de comandos de voz pode facilitar a execução de tarefas no mundo virtual, principalmente quando as mãos estiverem ocupadas.

Figura 3 – Elementos de uma luva de dados



Créditos: Jefferson Schnaider.

1.1.2 Dispositivos de rastreamento

Os dispositivos de interação podem estar associados a outro dispositivo, responsável pela tarefa de detecção da trajetória, conhecido como *dispositivo de rastreamento*. Nesse caso, esses dispositivos operam baseados na diferença de posição ou orientação em relação a um ponto ou a um estado de referência. Basicamente, existem uma fonte emitindo um sinal, podendo estar localizada no dispositivo de interação, um sensor que recebe esse sinal e uma caixa controladora, a qual processará o sinal e fará a comunicação com o computador (Machado; Cardoso, 2006).

A maioria das aplicações que utiliza detecção de trajetória faz uso de pequenos sensores colocados sobre as partes do corpo ou sobre o objeto, se for o caso – técnica essa conhecida como *tracking ativo*. Nesse caso, são empregadas técnicas eletromagnéticas, ultrassônicas, mecânicas ou óticas para a medida dos movimentos. Como alternativa, o *tracking passivo* utiliza câmeras ou sensores ópticos – ou de inércia – para monitorar o objeto e determinar sua posição e orientação. Diferente dos dispositivos que utilizam *tracking ativo*, os de *tracking passivo* têm apenas um sensor para rastrear o objeto (Machado; Cardoso, 2006).

Algumas das principais características técnicas avaliadas para a escolha e a utilização de dispositivos de rastreamento são (Machado; Cardoso, 2006):

- número de medidas efetuadas por segundo;
- sensibilidade a interferências externas;
- grau de ruído;
- qualidade das medidas efetuadas (taxa de erro);
- presença ou não de fios;
- área de captura.

Como exemplos de dispositivos de rastreamento, pode-se citar três tecnologias empregadas em luvas de dados tanto para localizar a mão em relação ao espaço como para detectar a orientação de sua palma. A primeira baseia-se em câmeras para monitorá-la, colocadas a certa distância (*tracking passivo*); a segunda, em radiação de pulsos magnéticos emitidos pela luva (*tracking ativo*); e a terceira, em acústica (*tracking ativo*), com dispositivos ultrassônicos transmitindo a posição da mão (Machado; Cardoso, 2006).

1.2 Dispositivos de saída de dados

Os dispositivos de saída de dados são responsáveis pelo envio das informações ao(s) usuário(s). Uma vez que sistemas de RV buscam explorar os cinco sentidos, os dispositivos de saída os estimulam por meios específicos. Atualmente, os sentidos mais explorados são: visão, audição e tato. No entanto, pesquisas recentes já apresentam dispositivos para estímulo de olfato (Yanagida et al., 2004) e paladar (Iwata et al., 2004).

Uma grande porção do cérebro é dedicada ao processamento e à organização dos estímulos visuais. Devido a isso, os dispositivos visuais e o tipo de imagem gerada por um sistema de RV são fatores muito importantes na obtenção e na determinação do nível de imersão do sistema (Machado; Cardoso, 2006).

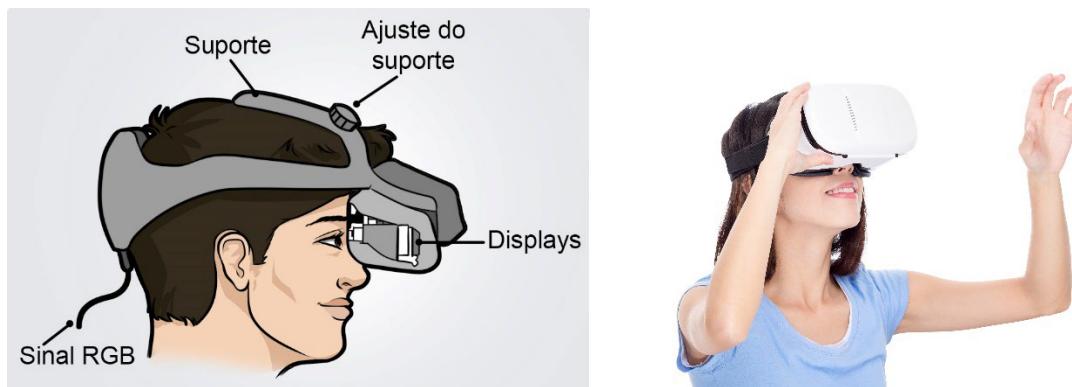
Conforme o tipo de dispositivo utilizado, podemos ter sistemas de RV monoscópicos, ou estereoscópicos. No caso de um sistema monoscópico, a mesma imagem será exibida para os dois olhos. Já no sistema estereoscópico, cada olho verá uma imagem ligeiramente diferente, sendo necessário construir um par de imagens. Nesse caso, é importante ressaltar uma característica fundamental da visão humana: em função de os olhos estarem localizados na

frente da cabeça, nosso campo visual não é de 360 graus, como o das aves; temos, pois, uma visão binocular (Machado; Cardoso, 2006).

Outro fator importante quanto à parte visual da RV é o número de quadros por segundo que aparecem no vídeo, ou seja, a velocidade da simulação. Filmes projetados para o cinema, por exemplo, apresentam, aproximadamente, 24 quadros por segundo, enquanto os projetados para TV têm, aproximadamente, 30 quadros por segundo. Em RV, busca-se uma taxa entre 15 e 22 quadros por segundo, mas isso pode variar, dependendo do tipo de interação utilizado no sistema (Machado; Cardoso, 2006).

Pode-se separar os dispositivos de visualização em duas categorias: os de visualização individual; e os de visualização coletiva. Na primeira categoria, enquadram-se os dispositivos do tipo vídeo-capacete (Head-Mounted Display – HMD) (Figura 4) e os Head-Coupled Displays – dispositivos que contam com braços mecânicos para permanecer diante do usuário. Na segunda, temos os monitores de computador e os sistemas de projeção (Machado; Cardoso, 2006).

Figura 4 – Elementos um HMD (à esquerda) e um dispositivo real (à direita)



Créditos: leungchopan/Adobe stock; Smiles ilustras.

Mais recentemente, foram desenvolvidos capacetes mais leves e fáceis de vestir, que passaram a ser chamados de Face-Mounted Displays (Figura 5). Tais dispositivos apresentam as imagens também em pequenos *displays* posicionados diante dos olhos do usuário, podendo integrar um sistema de rastreamento. Suas principais vantagens são o pequeno peso e a forma de utilização, semelhante à de óculos convencionais (Machado; Cardoso, 2006).

Figura 5 – Exemplo de Face-Mounted Display



Créditos: Ned Snowman/Shutterstock.

TEMA 2 – REQUISITOS BÁSICOS PARA UM SISTEMA DE RV

O desenvolvimento de sistemas de RV teve suas origens na criação dos softwares, com o uso das metodologias tradicionais de engenharia adaptadas à elaboração de sistemas multimídias. Além disso, a criação de produtos voltados para a indústria cinematográfica também contribuiu, e permanece contribuindo, com o desenvolvimento de sistemas de RV (Luz; Kirner, 2006). Um sistema, basicamente, é caracterizado pela integração de diversos componentes que atuam de forma inter-relacionada, visando a atingir um objetivo em comum. No caso dos sistemas de RV, os cinco requisitos detalhados no Quadro 1 são necessários.

Quadro 1 – Requisitos necessários para um sistema de RV

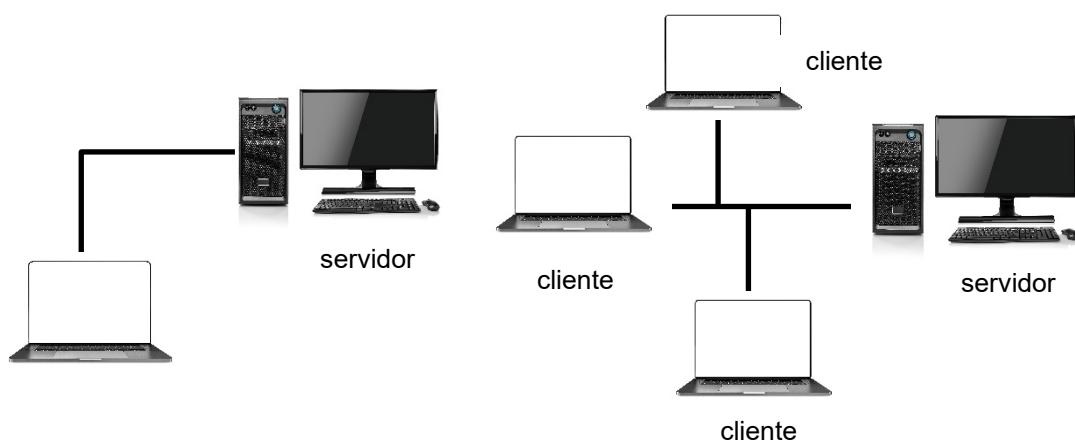
Interface de alta qualidade	A RV é utilizada como a interface de mais alto nível entre o ser humano e a máquina, pois permite que ambos interajam de maneira intuitiva para a pessoa, por imitar o que acontece em sua interação com o mundo real.
Alta interatividade	O ambiente deve reagir de maneira adequada às ações do usuário e permitir o maior número possível de ações.
Imersão	Um sistema de RV deve permitir que o usuário sinta-se dentro do mundo virtual, seja com seu corpo físico

	ou com uma representação qualquer (avatar, vídeo, simulação de cabine etc.).
Uso da intuição/envolvimento	O sistema deve explorar a intuição do usuário envolvido pelo ambiente e, assim, proporcionar novas formas de interação. Por exemplo, o projetista de um sistema de simulação de veículo que inclua um volante, marchas e pedais tem muito mais liberdade de projetar ações do que projetista de um sistema equivalente, mas no qual o usuário tem que ficar em pé e utilizar um <i>joystick</i> . No primeiro caso, o projetista sabe que o usuário já está habituado a certas ações, por sua experiência no mundo real.
Analogia/ampliação do mundo real	O fato de criar envolvimento e utilizar a intuição faz com que o sistema de RV atue como uma transferência do mundo real, capturando tudo de proveitoso que pode ser gerado pelo fato de o usuário já ter uma noção do que deve fazer e de como fazer, mas acrescentando aspectos que não existem no mundo real. Por exemplo, um sistema de busca bibliográfica pode utilizar, de alguma forma, a habilidade de o usuário locomover-se em uma biblioteca e olhar as estantes, mas a biblioteca virtual pode destacar visualmente volumes que possam ser de seu interesse.

Fonte: elaborado por Guerra, 2023, com base em Valerio Netto; Machado; Oliveira, 2002.

Os sistemas monousuários são desenvolvidos para serem usados por um só indivíduo, de modo a não interferir em seu uso pelos demais. Já os multiusuários podem ser usados por diversos usuários e, nesse caso, a intervenção de um deles pode afetar os demais (Figura 6). Além disso, esse tipo de sistema pode ser utilizado de forma assíncrona ou síncrona: na assíncrona, os usuários podem trabalhar colaborativamente, mas não interagem, em tempo real, com os demais usuários; na síncrona, eles podem interagir entre si em tempo real (Luz; Kirner, 2006).

Figura 6 – Sistemas monousuário (à esquerda) e multiusuário (à direita)



Créditos: mahod84/Adobe stock; Denis Rozhnovsky/Adobe stock.

Basicamente, os sistemas de RV diferem, entre si, de acordo com os níveis de imersão e de interatividade proporcionados ao participante. Esses níveis são determinados pelos tipos de dispositivos de entrada e saída de dados do sistema, além de levarem em conta, também, a velocidade e a potência do computador que os hospeda (Valerio Netto; Machado; Oliveira, 2002).

Portanto, em sistemas de RV, o ponto de vista é substituído pela experiência de estar: além da atualização em tempo real e da imersão em um espaço tridimensional e dinâmico, visa-se, eminentemente, a interação multissensorial do corpo com o ambiente, o que se realiza via: dispositivos agregados ao corpo (capacetes, luvas, vestimentas, sensores, *chips* etc.); ambientes físicos específicos (salas e Cave Automatic Virtual Environments – CAVEs), capazes de detectar, reconhecer, mapear e incorporar o usuário; e métodos híbridos. Enquanto, por um lado, a incorporação de determinados dispositivos ao corpo do usuário possibilita a imersão, por outro, também se busca libertá-lo desses dispositivos, com o intuito de deixá-lo atuar de forma mais natural e próxima da comunicação interpessoal (Hanns, 2006).

TEMA 3 – MODELAGEM E PROGRAMAÇÃO EM AMBIENTES VIRTUAIS

Aplicações gráficas tridimensionais (3D) exigem um esforço computacional considerável para serem processadas. É possível distribuir esse esforço entre processadores alocados em placas gráficas ou na placa principal do computador. Conhecer os detalhes dos modelos arquiteturais disponíveis e desenvolver aplicações que explorem ao máximo a potência computacional das placas e processadores faz com que as aplicações percam portabilidade, isto é, sejam fortemente dependentes do *hardware* para o qual foram desenvolvidas. No caso de a portabilidade da aplicação ser o fator preponderante sobre o desenvolvimento, opta-se por um modelo de desenvolvimento que utiliza o conceito de camadas de abstração sobrepostas, conforme ilustra a Figura 7 (Calonego Júnior et al., 2006).

Figura 7 – Camadas de abstração



A camada denominada *hardware gráfico* corresponde a algum dispositivo de saída gráfica (uma placa gráfica usada em jogos, por exemplo). Uma vez inserida a placa no computador, a camada *sistema operacional* (SO) deve ser configurada. Os fabricantes dos dispositivos gráficos disponibilizam os programas que permitem, ao sistema operacional, ter acesso ao dispositivo gráfico. O SO, por sua vez, gerencia o acesso ao *hardware gráfico* e oferece um conjunto de primitivas que viabilizam o acesso indireto de outras aplicações ao *hardware gráfico*, e isso significa que o programador não tem necessidade de conhecer como o *hardware* opera. Além disso, há a possibilidade de troca do *hardware* sem que haja necessidade de modificar os programas que utilizam as primitivas do sistema operacional (Calonego Júnior et al., 2006).

O desenvolvimento de programas para um dado sistema operacional apresenta pouca portabilidade. O aumento dessa portabilidade é tratado na camada *biblioteca gráfica*. Ela implementa um padrão de comunicação com as primitivas do *hardware*, via sistema operacional, que aumenta essa portabilidade. Os exemplos mais comuns de camadas análogas à *biblioteca gráfica* são o OpenGL e o DirectX. Programas que usam essas camadas podem ser transportados para diferentes sistemas operacionais, desde que haja uma versão correspondente à camada usada pela aplicação que possa ser executada naquele SO. A proliferação dessas bibliotecas gráficas possibilitou a criação de pacotes de desenvolvimento de RV (Calonego Júnior et al., 2006).

A biblioteca OpenGL permitiu a especificação e o desenvolvimento de uma camada para a descrição de objetos, eliminando a preocupação com a implementação das gráficas primitivas ou com o *hardware* a ser utilizado pela aplicação. Esse modelo foi utilizado por Gavin Bell para escrever a primeira especificação de uma linguagem de modelagem de mundos virtuais, denominada Virtual Reality Modeling Language (VRML) (Carey; Bell, 2000).

Diversos pacotes de RV utilizam programas escritos na linguagem VRML ou oferecem suporte para a conversão de códigos produzidos em outros formatos para VRML. Por exemplo, programas de desenho 3D, tais como o 3D Studio MAX e o Spazz 3D, exportam código VRML. Esses pacotes são a base para a implementação de aplicações de RV, que correspondem ao maior grau de abstração. Em geral, as aplicações são desenvolvidas utilizando-se um pacote de desenho 3D, um pacote de RV e linguagens de programação, como C, C++, Java, Delphi, VB e ECMAScript, entre outras, as quais permitem, ao programador, ter acesso ao pacote de RV para desenvolver os controles de interação da aplicação com o usuário (Calonego Júnior et al., 2006).

A programação em VRML, ou a construção de cenários virtuais, é uma tarefa complexa e exige, dos gerenciadores, o uso de estruturas de dados específicas para a otimização da visualização durante a navegação no ambiente virtual. Um *gerenciador de desenho* simplifica a tarefa de controle, fornecendo ao programador uma interface de desenvolvimento de programa, também denominada Application Programming Interface (API). O uso de uma API permite ao programador elaborar um cenário virtual para navegação e interação de forma direta. A criação de um mundo virtual é, portanto, a expressão de componentes do mundo real por meio de um modelo matemático que permita representar os elementos, bem como suas interações. Esse modelo é denominado *grafo de cena* e é usado por gerenciadores para otimizar o desenho das cenas em 3D. Atualmente, há programas tradutores de código VRML para Extensible 3D (X3D), a fim de garantir que as aplicações desenvolvidas em VRML possam ser facilmente transportadas (Calonego Júnior et al., 2006).

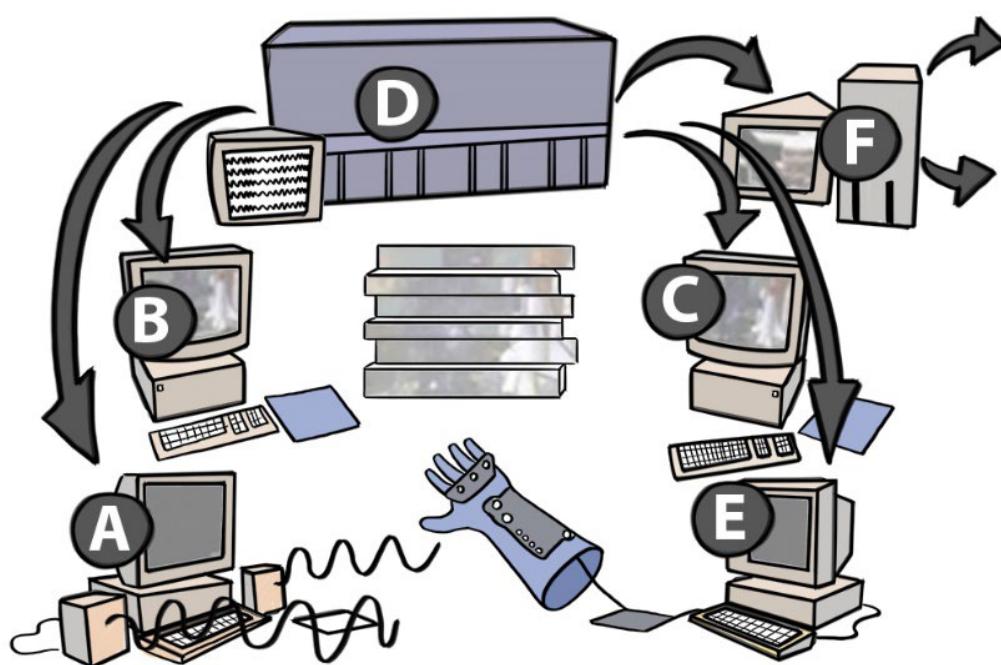
TEMA 4 – PROCESSO DE DESENVOLVIMENTO DE SISTEMAS DE RV

Em linhas gerais, um sistema de RV é composto por dois conjuntos de componentes (Luz; Kirner, 2006):

1. **Interfaces físicas e lógicas**, que incluem as entradas e saídas do sistema, representadas, respectivamente, por sensores e atuadores, as quais permitem a integração do ser humano com o sistema;
2. **Processador lógico do mundo virtual**, o componente responsável pelo controle do sistema.

A Figura 8 apresenta um exemplo de arquitetura de sistema de RV que envolve processamento distribuído. Nesse tipo de arquitetura, diversos aspectos do mundo virtual são processados por diferentes computadores. Na Figura, o computador A processa os dados referentes à geração do som, enquanto os computadores B e C formam as imagens que, em conjunto, geram a visão estereoscópica que alimenta o capacete de imersão, causando, no usuário, a sensação de profundidade. O computador D é responsável pelo processamento computacional das tarefas em tempo real e pela integração do sistema, enviando e recebendo pacotes de tarefas que compõem a interface externa do sistema (tanto com o usuário como com outros sistemas). Já o computador E faz o controle do dispositivo háptico de retorno da força enviada pelo usuário. O computador F, por fim, é responsável pela base de dados do sistema e pela atualização de dados referentes a agentes externos, utilizando, para isso, um meio de comunicação com outras bases de dados distribuídas (Luz; Kirner, 2006).

Figura 8 – Esquema de um sistema de RV de processamento distribuído



Crédito: Jefferson Schnaider.

Como qualquer sistema de *software*, o desenvolvimento de sistemas de RV pode basear-se nos modelos e métodos tradicionalmente indicados pela Engenharia de *Software*. Os modelos existentes, desde o tradicional Cascata (Sommerville; Sawer, 1997) até o atual Programação Extrema (Beck, 1999), podem ser adotados. Mesmo com a existência de diversos métodos, cada empresa deve utilizar o que melhor lhe convier e, se necessário, adaptá-los ou até mesmo criar seu próprio processo. Um resumo dos principais modelos de desenvolvimento de *software* está representado no Quadro 2.

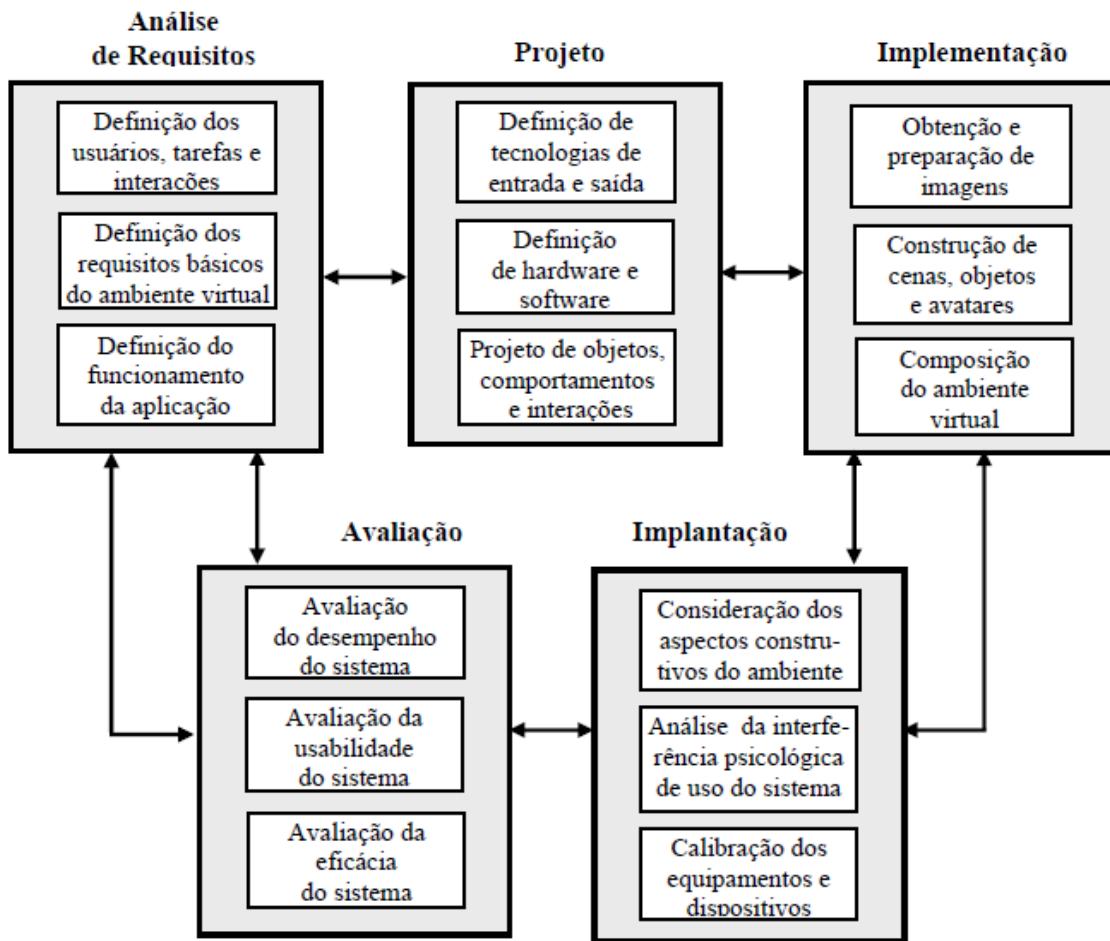
Quadro 2 – Principais modelos de *software* para sistemas de RV

Modelo Cascata (waterfall)	É um processo tradicional de desenvolvimento de <i>software</i> , o qual envolve a consecução das etapas de levantamento dos requisitos, análise dos requisitos, projeto, projeto detalhado, implementação, testes, implantação e manutenção. Apesar de ser largamente utilizado, possui desvantagens, pois projetos dessa natureza possuem um ciclo de desenvolvimento longo e, nesse caso, muitas das tecnologias, soluções e até mesmo metáforas podem ter que ser modificadas antes da finalização do ciclo de desenvolvimento.
Prototipagem	Esse modelo mostra-se adequado por permitir a criação de um protótipo ou produto final do sistema e colocá-lo à prova, junto aos usuários finais, em um tempo relativamente curto. Entretanto, não prevê a reformulação do sistema ao longo do tempo, pois o protótipo é criado apenas uma vez, e, depois, o ciclo de desenvolvimento transcorre linearmente.
Desenvolvimento iterativo	É um modelo importante, embora exija que as principais funcionalidades do sistema sejam cobertas já no primeiro estágio de desenvolvimento. Em muitos casos, esse modelo não leva a uma solução adequada, pois as funcionalidades podem exigir tempo e custos de desenvolvimento elevados, além de contribuir para distanciar o cliente do produto final, aumentando os riscos.
Modelo evolucionário	Apresenta características importantes, pois o ciclo de desenvolvimento de cada versão do sistema é reduzido, em relação aos modelos anteriores. Nesse modelo, o sistema é disponibilizado em versões que cumprem alguns dos requisitos totais do sistema.
Programação extrema	A Extreme Programming, com sua abordagem focada no problema e contato constante com o cliente e usuário final, procura aumentar as chances de o sistema ser desenvolvido conforme as reais necessidades do usuário. O modelo estimula a criação rápida de versões, para que o usuário final possa avaliar e interferir no próximo ciclo de desenvolvimento. Esse modelo é adequado quando os requisitos não forem totalmente esclarecidos e o contato com o cliente for possível. Para um sistema de RV completo, esse modelo pode não ser adequado, principalmente quando for exigida a integração de equipamentos especiais de alto custo. A criação de uma réplica do sistema junto ao cliente e/ou o constante deslocamento do cliente ou dos desenvolvedores para a averiguação de cada nova versão também pode elevar muito os custos.

Fonte: elaborada por Guerra, 2023, com base em Luz; Kirner, 2006, p. 116.

O processo de desenvolvimento de um sistema de RV compõe-se de etapas (Figura 9), realizadas iterativamente – a saber: análise de requisitos; projeto; implementação; avaliação; implantação (Luz; Kirner, 2006).

Figura 9 – Etapas do processo de desenvolvimento de um sistema de RV



Fonte: Luz; Kirner, 2006, p. 118.

TEMA 5 – CONCLUSÃO E PRÓXIMOS PASSOS

Com a evolução da tecnologia relacionada à visualização e aos dispositivos especiais, a RV também avança significativamente. Cada vez mais, os sistemas de RV fazem parte do cotidiano das pessoas, nas mais diferentes áreas de aplicação. Consequentemente, o domínio de um processo sistemático de desenvolvimento, adaptado às peculiaridades dos sistemas RV, tornou-se um fator altamente relevante para as empresas de *software*.

Acompanhando a tendência do desenvolvimento de jogos para dispositivos móveis, em especial para celulares, aliada ao aumento da capacidade de processamento desses dispositivos (*smartphones*), as tecnologias da realidade tendem a acompanhar uma tal evolução, tornando-se mais e mais populares. No

entanto, tudo indica que esses recursos tecnológicos não serão somente destinados ao entretenimento, mas também a aplicações mais sérias, como nas áreas de Saúde, Educação e Comércio, por exemplo.

Posteriormente, abordaremos mais profundamente os dispositivos utilizados em sistemas de RA, passando pela caracterização de sua arquitetura típica, bem como pelas vantagens e desvantagens verificadas no uso de tais recursos tecnológicos.

REFERÊNCIAS

- BECK, K. **Extreme programming explained: embrace change.** 1. ed. Boston: Addison Wesley Professional, 1999.
- CALONEGO JÚNIOR, N. et al. Modelagem e programação de ambientes virtuais interativos. In: TORI, R.; KIRNER, C.; SISCOUTTO, R. (eds.). **Fundamentos e tecnologia de realidade virtual e aumentada.** Porto Alegre: SBC, 2006. p. 98-108.
- CAREY, R.; BELL, G. **The annotated VRML 2.0 reference manual.** 3. ed. Boston: Addison-Wesley. 2000.
- HANNS, D. K. Estratégias de imersão: o corpo como interface. In: TORI, R.; KIRNER, C.; SISCOUTTO, R. (eds.). **Fundamentos e tecnologia de realidade virtual e aumentada.** Porto Alegre: SBC, 2006. p. 284-287.
- IWATA, H. et al. Food simulator. **IEEE CG&A**, v. 24, n. 1, 2004.
- LUZ; R. P.; KIRNER, T. G. Processo de desenvolvimento de sistemas de realidade virtual. In: TORI, R.; KIRNER, C.; SISCOUTTO, R. (eds.). **Fundamentos e tecnologia de realidade virtual e aumentada.** Porto Alegre: SBC, 2006. p. 109-127.
- MACHADO, L. S.; CARDOSO, A. Dispositivos de entrada e saída para sistemas de realidade virtual. In: TORI, R.; KIRNER, C.; SISCOUTTO, R. (eds.). **Fundamentos e tecnologia de realidade virtual e aumentada.** Porto Alegre: SBC, 2006. p. 39-50.
- SOMMERVILLE, I.; SAWER, P. **Requirements engineering: a good practice guide.** New York: Wiley, 1997.
- TORI, R.; KIRNER, C. Fundamentos de realidade virtual. In: TORI, R.; KIRNER, C.; SISCOUTTO, R. (eds.). **Fundamentos e tecnologia de realidade virtual e aumentada.** Porto Alegre: SBC, 2006. p. 2-21.
- VALERIO NETTO, A.; MACHADO, L. S.; OLIVEIRA, M. C. F. Realidade virtual: definições, dispositivos e aplicações. **Revista Eletrônica de Iniciação Científica**, v. 2, n. 1, p. 1-29, 2002.
- YANAGIDA, Y. et al. Projection-based olfactory display with nose tracking. In: IEEE VIRTUAL REALITY, 1., 2004. Chicago. **Anais...** Chicago: IEE Computer Society, 2004. p. 43-50.