



ISSN: 2595-1661

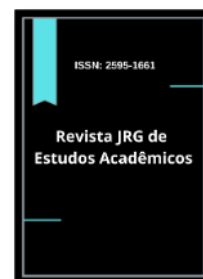
ARTIGO

Listas de conteúdos disponíveis em [Portal de Periódicos CAPES](http://portal.periodicos.capes.gov.br)

Revista JRG de Estudos Acadêmicos

Página da revista:

<https://revistajrg.com/index.php/jrg>



Segurança do Trabalho: O uso da Realidade Virtual na Educação Profissional e Tecnológica

Occupational Safety: The Use of Virtual Reality in Professional and Technological Education

DOI: 10.55892/jrg.v8i19.2557

ARK: 57118/JRG.v8i19.2557

Recebido: 14/10/2025 | Aceito: 22/10/2025 | Publicado on-line: 23/10/2025

João Senna de Andrade da Rosa¹

<http://lattes.cnpq.br/7553117357534129>
Universidade Federal de Santa Maria, RS, Brasil
E-mail: joao-senna1@hotmail.com

Simone Regina Ceolin²

<https://orcid.org/0000-0003-3750-2007>
 <http://lattes.cnpq.br/7514884760338859>
Universidade Federal de Santa Maria, RS, Brasil
E-mail: email@gmail.com

Rogério Correa Turchetti³

<https://orcid.org/0000-0002-5242-5057>
 <http://lattes.cnpq.br/1286016553991455>
Universidade Federal de Santa Maria, RS, Brasil
E-mail: email@gmail.com



Resumo

A Realidade Virtual (RV) tem o potencial de criar ambientes educativos inovadores, como a simulação de situações de trabalho que podem ser perigosas. O uso de equipamentos de RV em escolas técnicas pode enriquecer o aprendizado em Segurança do Trabalho, permitindo que os estudantes pratiquem a escolha e uso de Equipamentos de Proteção Individual (EPIs) em um ambiente virtual. Utilizando a Unreal Engine e ferramentas como Blender e GIMP, o projeto visa criar um vestíário virtual onde os alunos interagem com os EPIs, escolhendo os adequados para diferentes tarefas. O objetivo é reforçar o hábito do uso dos EPIs, integrando o conhecimento teórico com a prática virtual. Além disso, o uso de RV em ambientes educacionais pode aumentar a motivação e a retenção de conhecimentos dos alunos, simulando situações reais sem os riscos associados. Este projeto serve como um passo inicial para investigar a viabilidade do uso de realidade virtual e o desenvolvimento de um ambiente de aprendizagem virtual mais amplo e sofisticado.

¹ Possui graduação em Sistemas de Internet pela Universidade Federal de Santa Maria (2021). Atualmente é técnico administrativo em educação da Universidade Federal de Santa Maria. Tem experiência na área de Ciência da Computação.

² Possui graduação em Sistemas de Informação pelo Centro Universitário Franciscano (2001), mestrado em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul (2004) e doutorado em Ciência da Computação pela Universidade de York - Reino Unido (2011).

³ Possui graduação em Sistemas de Informação (UFN 2001), mestrado em Engenharia de Prod. ênfase em Tec de Informação (UFSM 2006) e doutorado em Informática (UFPR 2017). Atualmente é professor associado da Universidade Federal de Santa Maria, atuando nos cursos de graduação em computação (presencial e EaD) e no Mestrado Acadêmico em Educação Profissional e Tecnológica do CTISM.

Palavras-chave: Realidade Virtual, Educação Profissional e Tecnológica, Segurança do Trabalho.

Abstract

Virtual Reality (VR) has the potential to create innovative educational environments, such as the simulation of work situations that may be dangerous. The use of VR equipment in technical schools can enhance learning in Occupational Safety, allowing students to practice selecting and using Personal Protective Equipment (PPE) in a virtual environment. Utilizing Unreal Engine and tools like Blender and GIMP, the project aims to create a virtual locker room where students interact with PPE, choosing the appropriate ones for different tasks. The goal is to reinforce the habit of using PPE, integrating theoretical knowledge with virtual practice. Moreover, the use of VR in educational settings can increase student motivation and knowledge retention by simulating real situations without the associated risks. This project serves as an initial step to investigate the feasibility of using virtual reality and developing a more comprehensive and sophisticated virtual learning environment.

Keywords: *Virtual Reality; Vocational and Technological Education; Occupational Safety.*

1. Introdução

A Realidade Virtual (RV) possui o potencial de imergir o usuário em ambientes que seriam, de outro modo, inalcançáveis e inimagináveis, segundo Serin (2020) "As aplicações de realidade virtual na educação permitem que os estudantes adquiram experiências que são perigosas ou impossíveis de serem obtidas na vida real [...]". Mesmo que nenhum astronauta tenha pousado em um planeta fora da Terra, a RV permite ao usuário vivenciar visualmente cenários de outros planetas por meio de dispositivos como óculos de RV. De acordo com o dicionário, o termo "virtual" refere-se a algo que "existe como uma possibilidade, sem materialização completa", o que posiciona a Realidade Virtual como uma representação potencial e parcialmente simulada da realidade.

"Realidade Virtual é uma interface avançada do usuário que permite acessar aplicações em um computador, sendo assim, podemos utilizar para criar uma nova forma de interagir com a própria realidade através do computador" (KIRNER; SISCOOTTO, 2007).

O uso de ferramentas digitais, como plataformas de aprendizado online, simuladores e laboratórios virtuais, facilita a integração entre a teoria e a prática, como evidenciado no trabalho de Alarcon e Mendoza (2024), intitulado "*Prácticas de uso de tecnologías en la Educación Profesional y Tecnológica (EPT): un potencial innovador para el desarrollo de las clases presenciales*". Essas inovações promovem a autonomia dos estudantes, preparando-os para os desafios de uma economia cada vez mais digitalizada. Assim, a incorporação de tecnologias na EPT não apenas expande o alcance do ensino, mas também contribui para a formação de profissionais mais capacitados e prontos para os desafios do futuro.

No Colégio Técnico Industrial de Santa Maria da Universidade Federal de Santa Maria (CTISM/UFSM), há infraestrutura de RV disponível para atividades didáticas. Parte significativa dos cursos exige conhecimentos de Segurança do Trabalho, visando minimizar riscos em aulas práticas com maquinário.

Entre a instrução teórica e a execução da prática há um intervalo que pode ser melhor aproveitado por meio de um ambiente virtual, onde estudantes selecionam EPIs de forma rápida e contextualizada, reforçando a consciência preventiva.

1.1 Objetivo geral

O objetivo geral é desenvolver e documentar um ambiente virtual imersivo (vestiário + estações de avaliação) para treinamento de seleção e uso de EPIs em contextos de práticas de oficinas/laboratórios no CTISM.

1.2 Objetivos específicos

- Levantar e analisar trabalhos e ferramentas correlatas (UE, *Blender*, *GIMP*) quanto a requisitos de desempenho e usabilidade;
- Projetar o fluxo de interação (exploração do vestiário → coleta de EPIs → escolha da prática → avaliação);
- Implementar protótipo funcional em *Unreal Engine* com modelos 3D (autoria própria e *assets* livres), priorizando alto FPS e a redução de *motion sickness*;
- Registrar telemetrias simples para uso didático em ciclos de prática;

1.3 Justificativa

Acidentes de trabalho são recorrentes e, muitas vezes, evitáveis com o uso correto de EPIs. Um treino repetível, seguro e escalável em RV favorece a internalização do hábito (automatização) sem expor o aprendiz a riscos reais, além de permitir maior volume de prática que o tempo de laboratório usual.

1.4 Proposta

Implementar uma aplicação de RV com foco em seleção consciente de EPIs: o estudante interage em um vestiário virtual, coleta itens, escolhe uma prática (p.ex., torno mecânico) e tem sua seleção avaliada por um painel de controle. A solução enfatiza qualidade gráfica e estabilidade de *framerate* para reduzir desconforto (JUMANI et al., 2024; CHANG; KIM; YOO, 2020).

1.5 Estrutura do artigo

O presente artigo está organizado da seguinte forma: a Seção 2 apresenta a Metodologia da pesquisa. A Seção 3 aborda os fundamentos conceituais da Realidade Virtual (RV), além de trazer trabalhos que se relacionam diretamente com o tema. Na Seção 4 apresenta os resultados e discussões, tratando dos pontos relevantes da implementação e o resultado final da aplicação. A Seção 5 apresenta a conclusão do trabalho com o aprendizado adquirido durante a sua execução.

2. Metodologia

Em um primeiro momento, será realizada uma pesquisa bibliográfica sobre o tema, visando identificar as implementações existentes e os resultados observados. Também será possível comparar as ferramentas empregadas em cada implementação, permitindo destacar os pontos fortes e fracos do motor gráfico selecionado para este estudo.

Além disso, será analisada a documentação oficial da ferramenta escolhida, a *Unreal Engine*. Serão utilizados materiais de apoio para agilizar a compreensão tanto dos conceitos fundamentais quanto dos conceitos específicos relacionados à

ferramenta. Ainda na primeira etapa, será avaliada a capacidade de cada dispositivo disponível no colégio. Ao identificar as limitações de cada hardware, será possível selecionar o mais adequado aos objetivos da aplicação. Embora a aplicação seja simples e ocupe um espaço reduzido, a escolha das texturas e modelos de alta qualidade pode exigir considerável desempenho dos equipamentos.

No segundo momento, será desenvolvido o cenário da aplicação, configuradas as regras do jogo e implementado o algoritmo responsável por definir os diferentes objetos que serão posicionados no ambiente, criando um desafio a cada nova execução.

Por fim, serão realizados testes com a aplicação para verificar se ela opera conforme o esperado e atende às necessidades propostas. Não serão feitos testes com alunos neste estágio, pois é necessário, em uma fase futura, investigar quais materiais são utilizados em cada equipamento. O aprimoramento do jogo para ser utilizado como recurso didático depende da colaboração de docentes das áreas envolvidas. Uma vez que a aplicação funcione de forma consistente, ela poderá ser facilmente adaptada para diferentes contextos.

3 Referencial Teórico

O uso da realidade virtual (RV) para simulação de ambientes perigosos já é uma prática consolidada em diferentes áreas. Por exemplo, Nemec et al. (2017) desenvolveram um ambiente virtual que replica o reator de uma usina nuclear, com o objetivo de familiarizar os usuários com o ambiente sem expô-los aos riscos inerentes à interação direta com o reator.

Embora a realidade virtual tenha se mostrado uma alternativa promissora, suas aplicações ainda estão em estágio inicial em diversos campos. Um exemplo notável é o trabalho de Cortina (2023) que, em sua monografia intitulada "Justiçaverso em imersão: análise sobre os possíveis usos da realidade aumentada e realidade virtual no processo judicial brasileiro" demonstrou a utilização da RV em investigações criminais. Nesse contexto, cenas de crimes foram reconstruídas virtualmente para auxiliar na compreensão das condições do ambiente e na análise detalhada do cenário no momento do ocorrido.

Para a implementação da aplicação discutida neste estudo, foi empregada a *Unreal Engine*, uma ferramenta amplamente reconhecida por sua capacidade de criar ambientes realistas e sua compatibilidade com a linguagem de programação C++. A escolha também se deve à disponibilidade gratuita de recursos e soluções que otimizam o desempenho, dispensando configurações manuais complexas. O ambiente virtual criado é simples, consistindo em uma sala projetada para favorecer a imersão. Essa configuração, além de facilitar o uso, serve como base para um questionário interativo sobre a adequação de equipamentos de proteção individual (EPIs) em diferentes práticas, funcionando como uma atividade didática.

A aplicação foi desenvolvida com a premissa de que o movimento do usuário no ambiente real seria refletido no ambiente virtual. Essa abordagem visa mitigar o *motion sickness*, um desconforto comum associado ao uso de dispositivos de RV. Segundo Chang, Kim e Yoo (2020), criar um ambiente visualmente realista não é suficiente para eliminar o desconforto; contudo, integrar estímulos adicionais, como o movimento sincronizado entre os ambientes real e virtual, pode reduzir significativamente esse efeito.

Além disso, a construção do ambiente virtual desempenha um papel crucial na minimização do *motion sickness*. O ambiente foi projetado para ser estático, o que possibilita a aplicação de gráficos de alta qualidade e a otimização do desempenho

através dos recursos da *engine*. Estudos como o de Nemec et al. (2017), em *Using Virtual Reality in Education*, indicam que uma alta taxa de quadros por segundo é fundamental para reduzir o desconforto associado ao uso de RV. Assim, a combinação de um ambiente estático e realista, aliada às otimizações da ferramenta e à sincronização de movimentos, configura-se como uma estratégia eficaz para proporcionar uma experiência imersiva e confortável ao usuário.

3.1 Versionamento e Colaboração

Na área de programação, o versionamento de código é uma questão crucial. A capacidade de controlar as mudanças e compartilhar o código com a equipe é fundamental para o progresso eficiente de um projeto. Na própria documentação da ferramenta, são destacadas duas opções principais para essa tarefa: o *Perforce* e o *SVN*, evidenciando a importância desse fator para o sucesso da implementação.

Ambas as opções sugeridas pela UE na documentação apresentam limitações, pois dependem de uma infraestrutura própria para hospedar o servidor ou da contratação de um serviço de terceiros. O versionamento desempenha um papel essencial no processo de desenvolvimento, mas foi necessário adotar uma alternativa diferente das sugeridas. Seguindo o princípio da Navalha de Ockham, que sugere que a solução mais simples é a mais adequada, optou-se pelo *Git*, a ferramenta de versionamento mais popular e robusta.

Além de possibilitar o controle de versões do código, o *Git* também torna a colaboração entre os membros da equipe muito mais eficiente. Com ele, vários programadores podem trabalhar simultaneamente no mesmo projeto, sem que suas alterações interfiram negativamente no trabalho dos demais. Os programadores que seguem boas práticas criam versões separadas do código, e somente após validar que suas modificações não geram inconsistências, elas são integradas à versão principal do código, a *main*.

Algumas plataformas *web* oferecem servidores gratuitos para hospedar o código-fonte de projetos. Assim, bastaria iniciar um repositório no *Git*, e o projeto do jogo poderia aproveitar todas as vantagens dessa tecnologia. Mesmo considerando o tamanho total do projeto, é possível gerenciar suas versões em ferramentas como o *GitHub*, utilizando um arquivo *.gitignore* para garantir que arquivos grandes, como os resultados das *builds*, não sejam adicionados ao repositório.

No futuro, caso o *GitHub* venha a impor um limite de tamanho menor do que o necessário para o projeto, é possível armazenar os *assets* mais pesados no *Google Drive*. Utilizando a API fornecida pela Google, é relativamente simples criar um *script* em *Python* para sincronizar os *assets* do projeto com o código-fonte no *GitHub*.

3.2. Motor Gráfico

A *Game Engines* são formas de se encapsular comportamentos essenciais cuja implementação doméstica seria inviável no processo de criação de um jogo (GREGORY, 2018). Em aplicações com muitos elementos, como em experiências de Realidade Virtual (RV), o desenvolvimento seria excessivamente custoso caso fosse necessário implementar cada componente do zero.

Atualmente, diversas opções estão disponíveis gratuitamente, como *Unity*, *Godot* e *Unreal*, entre outras. Entre essas opções, a *Unreal Engine*, a partir da versão 5, oferece suporte robusto à realidade virtual, integrando componentes dedicados ao *trackinging* dos controles e do *headset*, além do posicionamento da câmera.

De acordo com a documentação oficial, a *engine* disponibiliza duas abordagens para o desenvolvimento de aplicações: as *Blueprints*, uma interface *low code* em que

objetos, componentes e sistemas são representados visualmente; e uma interface de *scripts* baseada na linguagem de programação C++. Por meio do uso de C++, é possível acessar recursos de baixo nível da *engine*, possibilitando um controle mais preciso sobre suas funcionalidades.

3.3. Ferramenta de Modelagem

O *Blender* é uma ferramenta de código aberto amplamente utilizada para modelagem 3D, oferecendo recursos robustos que abrangem desde a criação de objetos básicos até formas complexas de escultura digital (*Blender*, 2024). Através de suas ferramentas de edição de malha (*mesh editing*), é possível realizar operações como extrusão, subdivisão, deformações e booleanas, permitindo ao usuário modelar objetos com alto nível de precisão e controle. Adicionalmente, o *Blender* conta com um sistema não destrutivo de modificadores, como *Mirror*, *Subdivision Surface* e *Solidify*, que facilitam a criação iterativa e a manipulação paramétrica de modelos tridimensionais.

Outro recurso do *Blender* é sua capacidade de escultura digital, que se equipara a *softwares* especializados, como ZBrush. O modo *Sculpt* do *Blender* permite que artistas trabalhem com pincéis personalizados, simulação de dinâmica de superfície, multirresolução e *dynamic topology*, permitindo a criação de modelos orgânicos altamente detalhados, como personagens e criaturas. Além disso, sua integração com ferramentas de retopologia e *baking* auxilia na conversão de modelos complexos em malhas otimizadas, adequadas para animação, jogos e renderização em tempo real.

Essas funcionalidades tornam o *Blender* uma solução viável tanto para artistas independentes quanto para produções de maior escala no mercado de entretenimento e *design*. A documentação oficial do *Blender*, constantemente atualizada pela *Blender Foundation*, é uma fonte essencial para o aprendizado e aprofundamento técnico, fornecendo instruções detalhadas e exemplos de uso para cada ferramenta e funcionalidade disponíveis no *software*.

3.4 Trabalhos Relacionados

No trabalho de Nemer et al. (2020) intitulado "Um estudo de caso sobre o uso de gamificação e da realidade virtual na educação profissional", é apresentada uma proposta de implementação de um jogo destinado ao treinamento de trabalhadores em ambientes confinados. A aplicação utilizava imagens 360° de locais reais para criar o cenário virtual, permitindo que os estudantes, a partir de um resumo da situação, selecionassem os equipamentos necessários e os procedimentos adequados a serem seguidos no contexto apresentado. Após a realização do jogo, foi aplicado um questionário para coletar o *feedback* dos estudantes sobre a experiência. Os resultados não apenas indicaram que essa tecnologia pode estimular o engajamento dos estudantes, mas também mostraram que, ao utilizar um ambiente virtual, foi possível simular situações que, no mundo real, seriam perigosas, sem os riscos associados a algumas dessas atividades.

O trabalho de Mulders, Buchner e Kerres. (2022) "*Virtual reality in vocational training: A study demonstrating the potential of a vr-based vehicle painting simulator for skills acquisition in apprenticeship training*", descreve o desenvolvimento de um simulador de pintura automotiva em realidade virtual, criado com a ferramenta *Unity*, para treinar aprendizes na área. Além da aplicação em realidade virtual, foi desenvolvida uma adaptação para o controle: uma pistola de pintura impressa em 3D, que era acoplada ao dispositivo de RV, visando aumentar o realismo da atividade.

Observou-se que o uso desse recurso aumentou a motivação dos estudantes e foi eficaz no treinamento das competências básicas necessárias para o trabalho de pintura automotiva. Além disso, é relevante destacar que essa abordagem, que transfere atividades do mundo real para o ambiente virtual, possibilita um maior volume de treinamento prático para os alunos, sem as limitações impostas pelas condições reais.

Este trabalho tem como objetivo desenvolver uma plataforma destinada a apoiar o aprendizado sobre a importância e o uso adequado de Equipamentos de Proteção Individual (EPIs). O estudante será capaz de selecionar, entre os EPIs disponíveis, aqueles necessários para a execução de práticas específicas. A ferramenta também permitirá o treinamento em cenários reais, facilitando o reconhecimento rápido e quase automático dos materiais a serem utilizados em situações práticas.

A implementação final deste projeto será parte de um trabalho de mestrado, servindo como etapa inicial para o desenvolvimento de um ambiente mais abrangente e com maiores funcionalidades. Após a criação dessa sala de aula virtual, espera-se uma significativa ampliação do conhecimento sobre as ferramentas empregadas e seus potenciais aplicações.

4. Resultados e Discussão

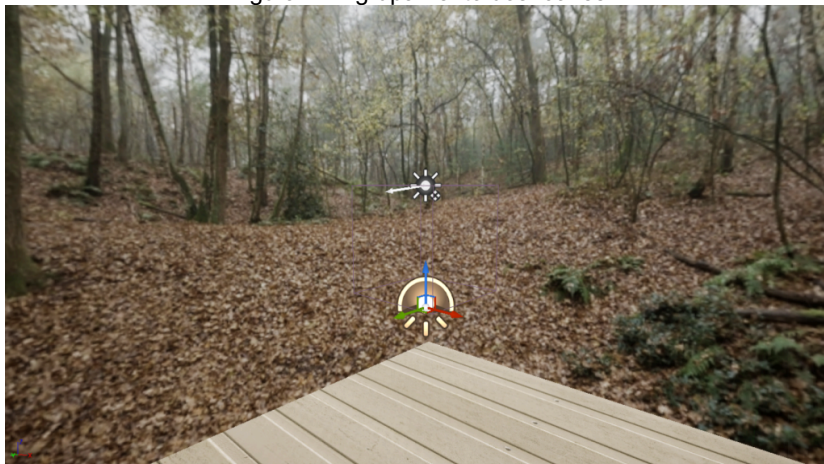
Devido à dimensão do projeto, algumas decisões foram tomadas para simplificar etapas do processo de criação na Unreal Engine (UE). Optou-se pelo *template* "First Person Shooter" (FPS) disponível na *engine*. Embora o nome sugira disparos, esse termo é apenas uma convenção da indústria para se referir a jogos em primeira pessoa, e não haverá mecânica de disparos no projeto. O *template* escolhido inclui os principais componentes e módulos necessários para o desenvolvimento. Embora exista um *template* específico para realidade virtual (RV), nas fases iniciais do projeto, o foco não foi as particularidades da RV, o que tornou o processo de desenvolvimento mais simples.

A construção do cenário, o funcionamento básico das entidades e dos sistemas foram o foco principal nas primeiras etapas do desenvolvimento do jogo. Os componentes e módulos necessários para a Realidade Virtual (RV) serão abordados no momento apropriado. No entanto, por ser um jogo simples do tipo FPS, essa abordagem facilitou o progresso nas etapas iniciais do projeto.

4.1 Criação do Mundo

Em um projeto iniciado do zero, seria necessário configurar o sistema, por exemplo: a iluminação, a atmosfera e outros elementos fundamentais para a simulação. O ambiente do jogo é composto por dois elementos principais: a *skybox* e a sala onde ocorre a interação. A *skybox* é responsável por fornecer componentes globais para a cena, incluindo o plano de fundo e a fonte de iluminação. A Figura 1 demonstra os ícones agrupados dessas configurações.

Figura 1 - Agrupamento dos ícones



Fonte: autor

O plano de fundo é uma imagem 360° de uma floresta (Figura 2), com o objetivo de permitir que, através das janelas, sejam visíveis algumas árvores, inspirado nos laboratórios do CTISM. A fonte de iluminação global na cena simula o Sol, e sua luz é ajustada para harmonizar com a imagem de fundo, promovendo uma maior imersão no ambiente.

Figura 2 - Visão externa



Fonte: autor

O *plugin Bridge* oferece acesso à biblioteca *Megascans*, que contém um vasto acervo de *assets* fotorrealistas disponíveis para uso gratuito na *Unreal Engine*. A biblioteca inclui tanto objetos 3D de alta qualidade quanto texturas, que podem ser utilizadas na *engine* para a construção do ambiente.

Assim, as três texturas diferentes foram escolhidas para os blocos formando a sala: um piso, uma textura de forro e uma textura de parede de azulejos. A ideia é parecer um ambiente de vestiário, onde o usuário vai escolher os EPIs adequados à determinada prática. Como *placeholders* para que fosse possível programar e definir o comportamento dos itens, mesmo 8 antes de eles serem modelados, foi utilizado um capacete de obras e uma mesa que estavam disponíveis na biblioteca *Megascans*, como é demonstrada na Figura 3.

Figura 3 - Vestiário



Fonte: autor

Na Figura 3 é possível observar fontes de iluminação adicionais além da proveniente da janela. Os elementos presentes na cena, incluindo as paredes, serão analisados de maneira mais detalhada no próximo tópico, no qual as entidades serão discutidas com maior profundidade.

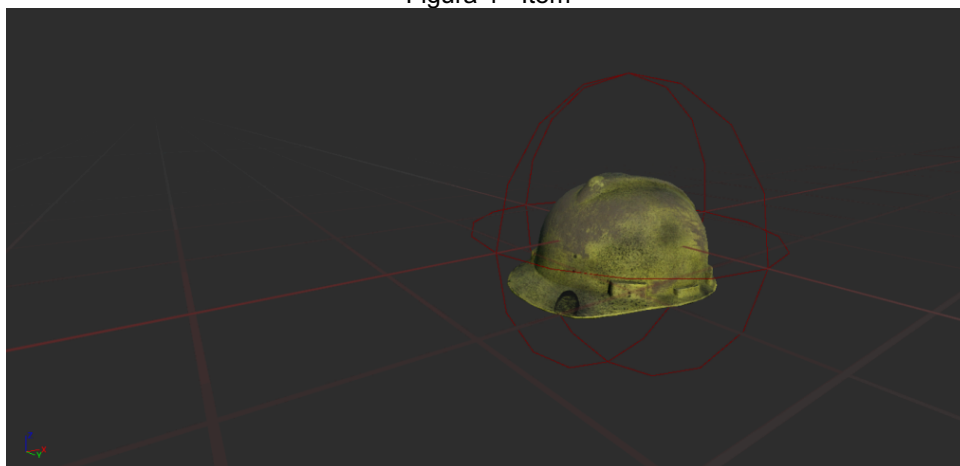
4.2 Entidades

A partir deste ponto, torna-se imprescindível abordar o padrão *Entity Component System* (ECS), amplamente empregado no desenvolvimento de jogos e aplicações de RV. Neste trabalho, os termos "entidades", "componentes" e "sistemas" já foram mencionados; portanto, é necessário esclarecer o significado dessas palavras dentro do contexto do ECS. Segundo Wiebusch e Latoschik (2015) no trabalho "*Decoupling the Entity-Component-System Pattern using Semantic Traits for Reusable Realtime Interactive Systems*", os objetos simulados na aplicação são classificados como entidades, as quais são compostas por múltiplos componentes que, por sua vez, representam diferentes propriedades da entidade. Os módulos, por outro lado, correspondem aos sistemas, sendo responsáveis por aspectos específicos da simulação; um exemplo disso é o módulo de física da simulação.

4.3 Os Itens

As entidades principais são os Equipamentos de Proteção Individual (EPIs), que são implementações da classe *Actor* da UE e são denominadas como Item demonstrado na Figura 4. O fato da classe Item ser uma subclasse de *Actor* possibilita, entre outras funcionalidades, a adição de componentes, como um componente de texto e um componente esférico de colisão. O componente de colisão responde a dois eventos: o evento de *overlap* e o evento de *endOverlap*. Quando a entidade Item é sobreposta pelo jogador, um texto com o nome do item é exibido; ao ser disparado o evento de fim de sobreposição, o texto é removido.

Figura 4 - Item



Fonte: autor

Ao contrário da entidade que representa o jogador, a qual utilizou em grande parte os recursos pré-implementados no *template*, a classe que representa os EPIs será desenvolvida em um nível mais baixo, com seus detalhes específicos definidos em C++. Embora os comportamentos e a estrutura sejam estabelecidos por meio de código, na *Unreal Engine*, as classes que precisam ser representadas diretamente na cena devem ser implementadas como *blueprints*. Assim, após a definição da classe Item em C++, a classe presente na cena é uma *blueprint* que herda de AItem.

Cada Item possui um componente de colisão, que neste caso é uma esfera com diâmetro ligeiramente superior ao do modelo 3D do item. Os dois *callbacks* de *overlap*, além de gerenciarem a exibição e ocultação do texto com o nome do item, também são responsáveis por possibilitar a interação do jogador com o item. Quando o item está a uma distância determinada, ele se torna interativo, o que exige que o jogador se desloque até o item para realizar a interação.

A *blueprint* que representa o item na cena herda todos os comportamentos definidos pela classe, e, em seguida, é associado o modelo 3D correspondente, que fica visível para o jogador. Com os comportamentos dos itens estabelecidos, o *placeholder* foi substituído pelos modelos 3D. Os EPIs utilizados também foram modelados pelos bolsistas e representam fielmente os equipamentos reais empregados nas práticas do colégio.

4.4 O Player

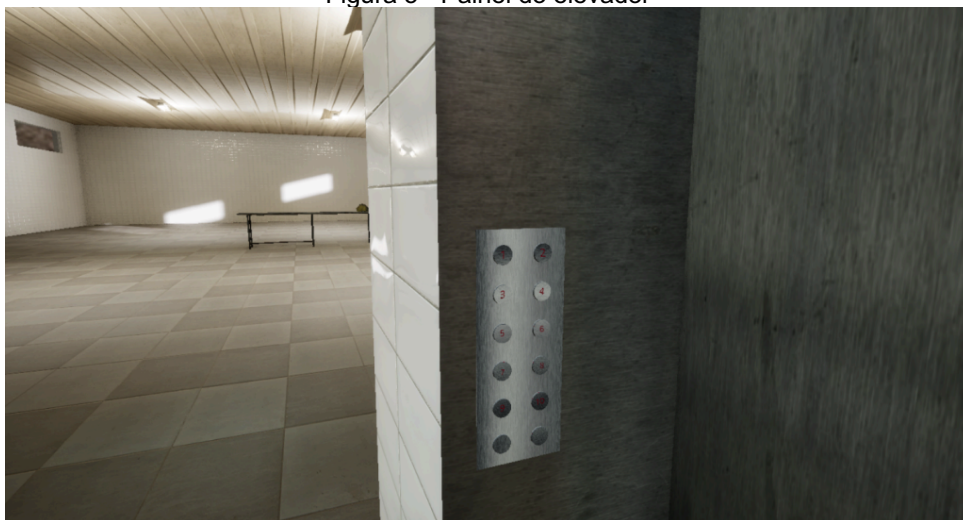
A entidade central da aplicação é aquela que representa o estudante no mundo virtual. O *Player* é uma implementação da classe *Character*, e, além dos componentes de movimento, leitura de inputs e colisão, foi criada na classe uma estrutura denominada Inventário, responsável por armazenar os EPIs coletados pelo estudante. Embora a *Unreal Engine* forneça algumas estruturas de dados prontas que podem interagir com *blueprints*, optou-se pelo uso de um Map, uma estrutura de chave-valor capaz de armazenar diferentes tipos de dados.

Essa estrutura armazena um Map, onde a chave é uma *string* que representa o nome do item e o valor é a referência na memória para o Item. Dessa forma, a gestão dos itens coletados torna-se mais simples, sem a necessidade de interações diretas com *blueprints*, o que fortalece o encapsulamento e reduz o acoplamento. Isso ocorre porque os detalhes internos, que são relevantes apenas para o contexto do código-fonte, não são expostos à camada das *blueprints*.

Também foram adicionados ao componente de leitura de *inputs* dois *callbacks*: um responsável por permitir a interação com os itens quando o jogador se aproxima o suficiente e outro que possibilita ao jogador devolver um item de seu inventário. O estudante pode, a qualquer momento, selecionar qual item deseja devolver, e esse item retornará para sua posição inicial. Como o jogador mantém uma referência na memória do item que pegou da sala, o método *spawn* da cena pode recriar o item na sala a qualquer momento. Nesse processo, a referência ao item será removida do inventário do estudante.

O jogador precisa interagir com o elevador para que suas escolhas sejam avaliadas, uma vez que simplesmente entrar no elevador e receber a resposta de forma automática seria uma transição abrupta. Dentro do elevador, encontra-se o painel de comandos, no qual cada andar corresponde a uma prática envolvendo uma máquina específica, conforme ilustrado na Figura 5.

Figura 5 - Painel do elevador



Fonte: autor

Assim, dependendo do andar que o jogador escolher, seus EPs serão avaliados como corretos ou não. Embora o elevador em si não seja uma entidade na cena, seu painel de comandos é uma entidade ativa. O painel, como qualquer outro objeto na *Unreal Engine*, é composto por diversos elementos:

- A classe definida em C++, que contém toda a lógica para ler os *inputs* e verificar os itens no inventário do estudante.
- As *Blueprints*, que herdam o comportamento da classe C++.
- As *Actions*, componentes da UE que conectam os métodos da classe aos *inputs* do jogador.
-

Quando o estudante escolhe um botão no painel, representando um andar com uma máquina específica, os itens em seu inventário são verificados quanto à sua compatibilidade com a prática daquele andar. Por exemplo, se o aluno selecionar o andar de um torno mecânico, o painel realiza uma comparação entre os itens necessários para aquela prática e os itens presentes no inventário do estudante. A estrutura utilizada para definir esses dados é um *Map*, onde a chave é uma *string* representando o andar e o valor é uma lista de *strings* contendo os nomes dos itens necessários, conforme ilustrado na Figura 4.

Quando o jogador escolhe um andar no painel, é realizada uma busca no *Map* que contém os andares do painel. A lista de itens necessários para aquele andar,

retornada pelo *Map*, será comparada com a lista de itens presentes no inventário do estudante. Caso o inventário contenha menos itens obrigatórios do que o necessário, o aluno receberá um alerta solicitando que busque mais EPIs. Por outro lado, se o inventário contiver itens em excesso, o jogador será alertado para devolver os itens não necessários.

O painel também guarda o número de tentativas necessárias para o estudante acertar os EPIs necessários, assim será possível utilizar a ferramenta para práticas desenvolvidas ao decorrer das aulas, e se observar a evolução do desempenho do estudante conforme ele avança nos estudos do conteúdo.

Ao final da implementação, os componentes de FPS da entidade Player foram substituídos pelos componentes destinados à Realidade Virtual. Assim como no caso do FPS, onde foram aproveitadas as pré-definições, o mesmo ocorreu para a RV. A implementação padrão da *Unreal Engine* foi utilizada, uma vez que o jogo não exige comportamentos complexos. Basicamente, a movimentação da câmera é controlada pela leitura dos movimentos do acelerômetro e giroscópio, enquanto um botão é usado para a interação com os objetos na cena

5. Conclusão

Embora a aplicação desenvolvida seja simples, o objetivo deste trabalho era investigar a viabilidade de criar um ambiente virtual para a aprendizagem de práticas relacionadas à Segurança do Trabalho. Mesmo considerando a curva de aprendizado da realidade virtual, ficou evidente que não só é perfeitamente viável criar um ambiente imersivo usando a ferramenta, como também é possível, com esforço mínimo, expandir a implementação para torná-la aplicável a qualquer disciplina que exija o reforço da prática no uso de EPIs antes do contato real com os equipamentos. Como parte do desenvolvimento de uma dissertação de mestrado, este trabalho proporcionou uma ampliação dos conhecimentos sobre a *Unreal Engine*, permitindo a criação de um ambiente realista e imersivo com relativa facilidade.

Também foi possível explorar algumas potencialidades da ferramenta, como a utilização de imagens em 360°, que, conforme mencionado em um dos trabalhos, se mostrou uma abordagem interessante para a criação de um ambiente imersivo. Contudo, é importante considerar o impacto no desempenho de qualquer elemento visual que busque aumentar o realismo. Embora uma imagem usada na *skybox* não afete significativamente o desempenho dos óculos de realidade virtual, a implementação de texturas e modelos realistas deve ser cuidadosamente avaliada para garantir que não comprometa a experiência do usuário.

O principal limitante em aplicações para óculos de realidade virtual *standalone* é o desempenho do dispositivo. Atualmente, ainda não existem hardwares com alto poder de processamento, o que exige que a construção da cena seja feita com cuidado, levando em consideração que cada novo elemento pode impactar o desempenho do dispositivo. Embora a *Unreal Engine* ofereça uma ampla gama de recursos gráficos para aprimorar o realismo, muitos desses recursos não podem ser utilizados, pois reduzem significativamente a taxa de quadros por segundo, um fator crítico para evitar o *motion sickness*.

Assim, a experiência adquirida neste trabalho contribuiu de forma significativa para o aprofundamento do conhecimento técnico em *Unreal Engine* e para a compreensão das melhores práticas na criação de ambientes virtuais educacionais. Com os ajustes e otimizações apropriadas, é possível desenvolver aplicações eficientes e imersivas, tanto para a educação em Segurança do Trabalho quanto para outras áreas que se beneficiam do uso da realidade virtual.

Referências

BLENDER. Disponível em: <https://www.blender.org/>. Acesso em: 01/12/2024.

CHANG, E.; KIM, H. T.; YOO, B. Virtual reality sickness: a review of causes and measurements. *International Journal of Human–Computer Interaction*, Taylor & Francis, v. 36, n. 17, p. 1658–1682, 2020.

CHAVEZ, J. A.; MATENCIO M. Eleazar Moises. Uso del Simulador Cade SIMU en el logro de competencias en estudiantes de EPT-IE San Mateo de Huanchor-Huarochiri. 2024.

CORTINA, H. Justiça Verso em imersão: análise sobre os possíveis usos da realidade aumentada e realidade virtual no processo judicial brasileiro. In: O MANANCIAL - REPOSITÓRIO DIGITAL DA UFSM. [S.l.], 2023.

EKPENYONG, C. E.; ASUQUO, A. E. Recent advances in occupational and environmental health hazards of workers exposed to gasoline compounds. *International journal of occupational medicine and environmental health*, Instytut Medycyny Pracy im. prof. dra Jerzego Nofera w Łodzi, v. 30, n. 1, p. 1–26, 2017.

GIMP. Disponível em: <https://www.gimp.org/>. Acesso em: 01/12/2024.

GREGORY, J. Game engine architecture. [S.l.]: AK Peters/CRC Press, 2018.

JUMANI, A. K. et al. Quality of Experience That Matters in Gaming Graphics: How to Blend Image Processing and Virtual Reality. *Electronics*, v. 13, n. 15, p. 2998, 2024.

KIRNER, C.; SISCOOTTO, R. Realidade virtual e aumentada: conceitos, projetos e aplicações. In: SN. Livro do IX Symposium on Virtual and Augmented Reality, Petrópolis (RJ), Porto Alegre: SBC. [S.l.], 2007. v. 28.

MULDERS, M.; BUCHNER, J.; KERRES, M. Virtual reality in vocational training: A study demonstrating the potential of a vr-based vehicle painting simulator for skills acquisition in apprenticeship training. *Technology, Knowledge and Learning*, Springer, p. 1–16, 2022.

NEMEC, M. et al. Using virtual reality in education. In: IEEE2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA). [S.l.], 2017. p. 1–6.

NEMER, E. G. et al. Um estudo de caso sobre o uso de gamificação e da realidade virtual na educação profissional. *Revista Fatec Zona Sul (REFAS)*, v. 6, n. 5, 2020.

OBSERVATÓRIO DIGITAL DE SAÚDE E SEGURANÇA NO TRABALHO. 2024, on-line. Disponível em: <https://smartlabbr.org/sst>. Acesso em: 24 novembro 2024.

RODRIGUES, A. P.; MONTEIRO, A.; MOREIRA, J. A. Dimensões pedagógicas da sala de aula virtual: teoria e prática. *Cadernos de Pedagogia no Ensino Superior*, n. 26, p. 3–25, 2013.

SERIN, H. Virtual reality in education from the perspective of teachers. *Amazonia investiga*, v. 9, n. 26, p. 291-303, 2020.

Unreal Engine. Disponível em: <https://www.unrealengine.com/en-US/xr>. Acesso em: 20/04/2025.

WIEBUSCH, D.; LATOSCHIK, M. E. Decoupling the entity-component-system pattern using semantic traits for reusable realtime interactive systems. In: IEEE. 2015 IEEE 8th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS). [S.l.], 2015. p. 25–32.