

TCC – AMBIENTE DE AULA EM REALIDADE VIRTUAL

Gabriel Garcia Salvador, Dalton Solano dos Reis – Orientador

Curso de Bacharel em Ciência da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

gabrielgarciasalvador@outlook.com, dalton@furb.br

Resumo: O resumo é uma apresentação concisa dos pontos relevantes de um texto. Informa suficientemente ao leitor, para que este possa decidir sobre a conveniência da leitura do texto inteiro. Deve conter OBRIGATORIAMENTE o OBJETIVO, METODOLOGIA, RESULTADOS e CONCLUSÕES. O resumo não deve ultrapassar 10 linhas e deve ser composto de uma sequência corrente de frases concisas e não de uma enumeração de tópicos. O resumo deve ser escrito em um único texto corrido (sem parágrafos). Deve-se usar a terceira pessoa do singular. As palavras-chave, a seguir, são separadas por ponto, com a primeira letra maiúscula. Caso uma palavra-chave seja composta por mais de uma palavra, somente a primeira deve ser escrita com letra maiúscula, sendo que as demais iniciam com letra minúscula, desde que não sejam nomes próprios.]

Palavras-chave: Ciência da computação. Sistemas de informação. Monografia. Resumo. Formato.

1 INTRODUÇÃO

Devido a ameaça posta pela pandemia do vírus COVID-19 em 2020, muitas escolas e faculdades foram forçadas a cessar as atividades letivas presenciais a fim de manter os diretores, colaboradores e estudantes seguros durante essa crise de saúde pública, instruindo seu corpo docente a optar por lecionar de maneira online como forma substituta ao ensino presencial a fim de evitar a contaminação e o avanço da doença (HODGES *et al.*, 2020).

Para manter as atividades letivas, as instituições educacionais rapidamente optaram por manter a modalidade de ensino a distância o que resultou em um avanço repentino sem precedentes na educação online, com isso, diversas empresas de plataformas online de ensino propuseram apoiar e solucionar essa nova alta demanda, muitas vezes com serviços gratuitos, mas essa rápida adoção mostrou diversos gaps e deficiências nessas ferramentas (TERÄS *et al.*, 2020).

Além dos déficits das ferramentas, também existe a falta de capacitação por parte dos docentes na utilização de certas tecnologias, é necessário tempo e custo para que os encarregados pudessem treinar os docentes a utilizar das ferramentas e dos computadores, outro ponto preocupante, em relação ao ensino online, seria a forma que o conteúdo seria disponibilizado e lecionado, para conseguir manter o foco e engajamento dos alunos durante as aulas mesmo que a distância, e a instituição ainda deveria conseguir garantir um ensino acadêmico que esteja à altura do ensino presencial (BASTRIKIN, 2020).

O uso de um computador para lecionar online, utilizando de interfaces comuns como o teclado e mouse, se mostram muito limitantes como ferramentas de ensino, as dificuldades do uso dessas podem variar muito de acordo com a disciplina. Um exemplo comum onde essas interfaces geram dificuldades é nas situações em que os usuários necessitam criar representações visuais de forma manual através de desenhos como fórmulas e equações, utilizando de um mouse para realizar a tarefa, sem ter o controle e tato que teria ao escrever em um quadro físico (LEE, 2020).

Outro ponto é que no ensino, muitas vezes uma representação bidimensional como as encontradas nos livros didáticos ou em fotografias na internet, são insuficientes na compreensão de certos conteúdos, os quais requerem mais que uma imagem estática em uma página para serem compreendidos. Utilizando de modelos tridimensionais (3D) pode-se providenciar aos estudantes um conteúdo muito mais imersivo e interativo, esses Modelos 3D são ferramentas importantes para garantir uma compreensão rica do conteúdo em sala de aula (TABRIZI, 2008).

Esse artigo então, busca promover o uso da Realidade Virtual (RV) como uma ferramenta alternativa ou complementar aos docentes no ensino online, onde será desenvolvida uma aplicação de RV utilizando do motor gráfico Unity, essa aplicação poderá ser utilizada pela maioria dos Head Mounted Displays (HMD) de RV do mercado consumidor, mas terá como foco o Oculus Quest 2, devido seu baixo custo e possibilidade de ser utilizado sem a necessidade de um computador conectado, um teste será confeccionado para validar a viabilidade do uso da aplicação por professores. Nessa aplicação, será providenciada uma sala de aula virtual semelhante a o que é esperado de sua versão real, onde o professor poderá lecionar suas aulas com as devidas ferramentas necessárias dentro dela, canetão, quadro, apagador e objetos didáticos, sendo que tais ferramentas devem ser manuseáveis de forma intuitiva e que sejam de fácil acesso ao usuário e com uma baixa curva de aprendizado.

Dentro da aplicação, seria possível implementar virtualmente qualquer objeto a fim de dar uma representação visual didática mais compreensível aos alunos como, mostrar objetos e materiais exóticos, reações químicas perigosas, ação e reação de corpos físicos, dentro de virtualmente qualquer ambiente, seja esse uma sala de aula comum, um laboratório de química, ou uma estação espacial na lua. Porém, visto que essa aplicação será um conceito mais

generalizado, as ferramentas disponibilizadas serão as de caráter essencial que sejam comuns com o maior número de disciplinas didáticas.

A aplicação será utilizada por professores e também por outras pessoas, a fim de qualificar as impressões, opiniões e críticas em relação ao mesmo, considerando-as para concluir a viabilidade do uso da tecnologia de RV na educação com a tecnologia que possuímos hoje, assim como extensões propondo possíveis melhorias que possam levar a aplicação a obter outro resultado mais favorável em relação ao seu público-alvo.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, será abordado os temas presentes no artigo de forma subdividida, na primeira parte será discutido sobre o que define um material como didático e sua importância dentro de uma sala de aula, na segunda subseção será abordado o conceito, história e crescimento da tecnologia de Realidade Virtual, em seguida serão apresentados os trabalhos correlatos.

2.1 MATERIAL DIDÁTICO

Dentro de uma sala de aula, utiliza-se de materiais didáticos a fim de conseguir explicar um conteúdo para a sala de aula, alguns desses materiais podem ser específicos para certa disciplina, porém outros podem ser comuns entre diversas disciplinas. Como especificado por Freitas (2009) “os materiais e equipamentos didáticos são todo e qualquer recurso utilizado em um procedimento de ensino, visando à estimulação do aluno e à sua aproximação do conteúdo.”. Existem diversos recursos visuais, auditivos e audiovisuais que podem ser utilizados para lecionar, sejam estes objetos criados exclusivamente para fins educativos, ou que indiretamente podem ser usados para este fim. Para definir se um material pode ser considerado de uso didático, deve-se observar alguns critérios na hora de sua seleção:

- a) adequação aos objetivos, conteúdo e grau de desenvolvimento, interesse e necessidade dos alunos;
- b) adequação as habilidades que se quer desenvolver, sejam estas cognitivas, afetivas ou psicomotoras;
- c) simplicidade, baixo custo e manipulação acessível;
- d) qualidade e atração (devem despertar curiosidade).

Um material didático, consegue estabelecer uma comunicação entre professor e aluno, transformando uma aula monótona exclusivamente verbal em algo mais cativante. Assim ampliando o campo de experiência do estudante tendo em vista que agora o mesmo possui uma representação visual do elemento em pauta, que de outro modo permaneceria abstrato (FREITAS, 2009)

Historicamente na educação do Brasil, existem alguns recursos considerados universais para o ensino. Entre eles se destacam o quadro de giz ou lousa, por possuir um baixo custo, fácil instalação, ser um ótimo recurso visual acessível para todos os alunos e é versátil para diversas disciplinas. Outro equipamento é o Retroprojetor, capaz de projetar imagens em uma superfície de forma ampliada dando maior visibilidade para toda a sala, e por fim um aparelho de vídeo e DVD como recurso audiovisual permitindo a reprodução de áudio e vídeo dos assuntos necessários para a sala (FREITAS, 2009).

2.2 REALIDADE VIRTUAL

O conceito de RV já é datado desde a década de 1800, dispositivos como o Kinetoscópio e Mutoscópio permitiam ao usuário isolar sua visão de tudo ao seu redor enxergando apenas dentro de outra realidade (42GEARS, 2019). O primeiro capacete de realidade virtual ou Head Mounted Display (HMD) como o que é utilizado em dispositivos de RV hoje em dia, foi desenvolvido somente na década de 1960 por Ivan Sutherland, cujo denominou o mesmo de “Ultimate Display” (TORI; KIRNER; SISCOOTTO, 2006).

Ao longo dos anos, a tecnologia de RV se encontrava apenas em produtos de um nicho pequeno, estes que raramente se tornavam produtos bem-sucedidos no mercado como o VirtualBoy (FLANAGAN, 2018). Apenas recentemente, essa tecnologia começou a ter um atrativo maior no mercado, com diversos fabricantes desenvolvendo vários dispositivos de HMD de RV cada vez mais acessíveis e com mais recursos tecnológicos, resultado disso podem ser observados, a previsão projeta que as vendas de dispositivos de RV e Realidade Aumentada cheguem a 26 milhões por ano até o ano de 2023 (STATISTA, 2020). Hoje a tecnologia, embora ainda em constante evolução, já consegue entregar uma experiência de uso satisfatória o suficiente para agradar os seus usuários (TCHA-TOKEY et al., 2016) e, o número de vendas de produtos de RV cada vez maior é reflexo disso. O mercado ainda é recente, e seu potencial é enorme, e é necessário a mobilização dos desenvolvedores para agregar a estas novas plataformas emergentes de RV, para então evoluir essa tecnologia e integrar cada vez mais com a sociedade como uma extensão de como consumimos e entregamos conteúdo para a educação e entretenimento.

2.3 TRABALHOS CORRELATOS

A seguir serão apresentados os trabalhos que possuem características de alguma maneira semelhante aos objetivos principais do artigo. O primeiro, é um trabalho que relata o desenvolvimento de um sistema de Agent based Virtual Reality (AVR) que permite aos professores lecionarem, e aos alunos de estudarem em um ambiente virtual (TABRIZI, 2008). O segundo, é um artigo que conceitualiza uma sala de aula em RV que possibilite aprimorar a educação através de representações mais interativas para as diversas matérias lecionadas. E o terceiro, mostra o desenvolvimento de um ambiente virtual urbano, com ruas para os usuários aprenderem sobre educação de trânsito.

Quadro 1 – Agent and Virtual Reality-based Course Delivery System

Referência	Tabrizi (2008)
Objetivos	Maximizar a efetividade do ensino online, desenvolvendo um sistema de ensino para professores lecionarem e alunos poderem atender as aulas, realizar atividades e avaliações.
Principais funcionalidades	Uma plataforma 3D de RV com, um sistema de gerenciamento de cursos, um sistema dinâmico de avaliações, um ambiente de comunicação com base em multimídia, um modelo cliente/servidor efetivo com uma camada de segurança avançada, um quadro branco e apresentador de PowerPoint eletrônico, esse ambiente deve ter um aspecto de campus como os laboratórios, e deve poder capturar os movimentos do professor durante as aulas.
Ferramentas de desenvolvimento	O autor não especifica as ferramentas ou linguagens utilizadas para desenvolver o sistema.
Resultados e conclusões	O autor não realiza uma avaliação sobre os resultados, apenas informa que existe trabalhos de outros professores que estão estudando a efetividade de seu sistema, porém afirma que a atitude dos estudantes em relação a sistemas de jogos é amplamente acordada como algo positivo, e que o sistema AVR providenciara aos estudantes uma plataforma para aprenderem em um ambiente interativo de multimídia similar ao mundo dos jogos.

Fonte: elaborado pelo autor.

Quadro 2 – An overall solution of Virtual Reality Classroom

Referência	Dong (2016)
Objetivos	O autor (DONG, 2016) propõe uma solução para a integração geral de uma sala de aula em RV, combinando a tecnologia com as diversas disciplinas, a fim de atingir uma convergência compatível com a sala de aula presencial comum, com a sala de aula de informática, salas de multimídia entre outras.
Principais funcionalidades	Dong afirma que a sala de aula em RV deve ser constituída pela combinação de animação virtual, espaços virtuais, para criar ambientes imersivos de aprendizado 3D. então, exemplifica por matérias, de forma pontual e breve, um exemplo de como a RV pode ajudar nas mesmas: <ul style="list-style-type: none"> a) matemática: a sala RV permite visualizar conceitos matemáticos abstratos e complexos de forma sensorial, mapeamento de coordenadas, porcentagem, objetos geométricos, entre outros; b) biologia: pode-se renderizar o mundo microscópico e observar a multiplicação de células, vírus entre outros; c) física: representando forças, movimento e energia, seja física, térmica, óptica, atômica, mecânica e elétrica de forma visual ajudando a dominar as leis da física; d) química: simular reações químicas sem a necessidade de se preocupar com a obtenção, perda e perigo dos reagentes químicos; e) astronomia: pode levar os alunos a literalmente andar em algum planeta; f) engenharias: visualizar protótipos antes de produzir eles fisicamente. Também aponta que o ensino pela RV não é necessariamente uma alternativa
Ferramentas de desenvolvimento	O autor apenas conceitualiza um ambiente virtual de RV.
Resultados e conclusões	Concluiu-se pelo autor (DONG, 2016) que a proposta que a RV traz é extremamente atrativo, e vai mudar a forma que as pessoas pensam em um sentido, e até mudar o entendimento de tempo e espaço. A RV pode desenvolver novos meios de ensino e aprendizado, e terá grande importância na área educacional com o decorrer do tempo.

Fonte: elaborado pelo autor.

Quadro 3 - DEVELOPING A VIRTUAL REALITY EDUCATIONAL ENVIRONMENT FOR TRAFFIC EDUCATION

Referência	International Conference of Education, Research And Innovation (2019)
------------	---

Objetivos	Familiarizar os usuários na navegação dentro de um ambiente urbano, com a educação de trânsito, bom comportamento de motoristas, entendimento das prioridades no sistema de ruas e aprimorar a movimentação do usuário dentro do ambiente como pedestre.
Principais funcionalidades	O usuário deve poder movimentar-se livremente dentro do ambiente virtual, interagindo com objetos virtuais como, sinalização de trânsito, semáforos, outros veículos, veículos prioritários e personagens não jogáveis (NPCs)
Ferramentas de desenvolvimento	O ambiente virtual foi implementado utilizando o OpenSim, uma aplicação servidora multiplataforma, multiusuário e tridimensional. Os objetos existentes dentro do mundo virtual foram criados utilizando Blender, uma plataforma open-source para design tridimensional.
Resultados e conclusões	Educação de Trânsito é um tema crucial da educação para humanos já que o conhecimento que é adquirido com essa, é aplicado no cotidiano de muitas pessoas. Se locomover de acordo com as leis, com bom comportamento de pedestres e motoristas é o passo mais importante para ruas mais seguras, enfatizando o ensino dessa matéria dès do ensino básico, para que jovens estudantes adotem e apliquem esse conhecimento de bom comportamento no trânsito o quanto antes. Uma forma de alcançar isso, é com a aplicação da tecnologia, especialmente da RV e suas vantagens para educação. Com a aplicação desenvolvida, os usuários podem simular situações de perigo no trânsito e como agir nas mesmas sem correr nenhum perigo real, estando em um ambiente confortável.

Fonte: elaborado pelo autor.

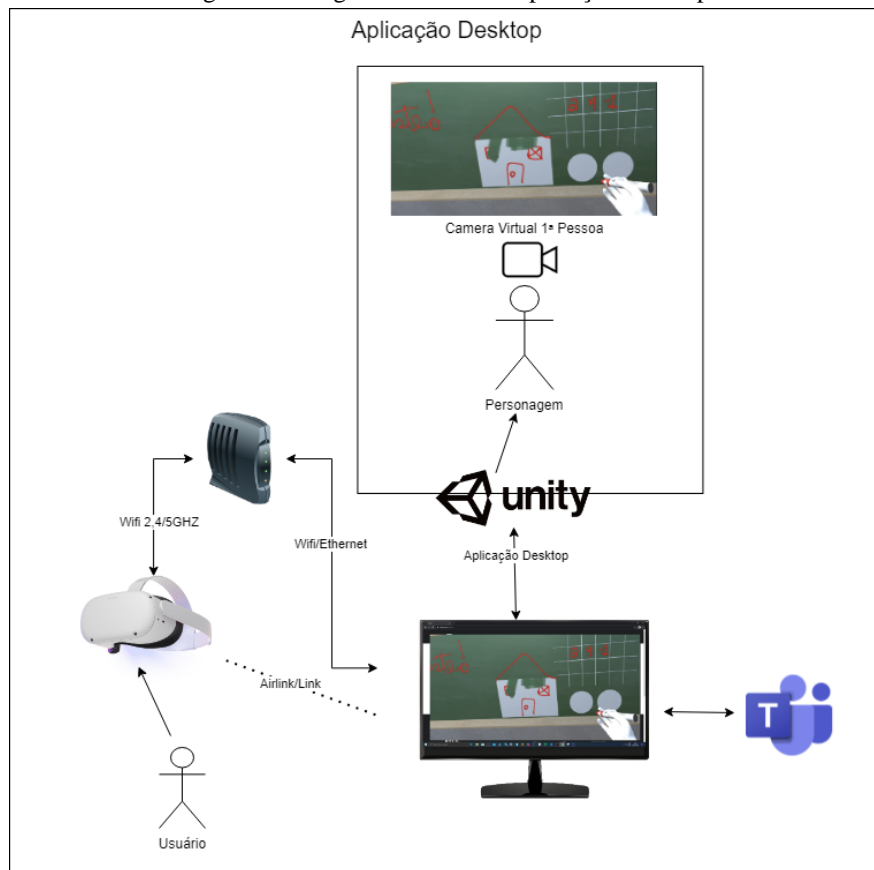
3 DESCRIÇÃO

Nesta seção devem ser descritos os **aspectos mais relevantes de especificação e implementação** para a compreensão sobre o trabalho desenvolvido. O título “DESCRIÇÃO” pode ser complementado com “DO SOFTWARE”, “DA FERRAMENTA” ou “DO PROTÓTIPO” ou aquilo que melhor representar o que foi desenvolvido. Esta seção deve estar organizada em pelo menos duas subseções: especificação e implementação

Reitera-se que, em função da limitação do número de páginas, a descrição deve contemplar o que é mais significativo para a compreensão do que foi desenvolvido.

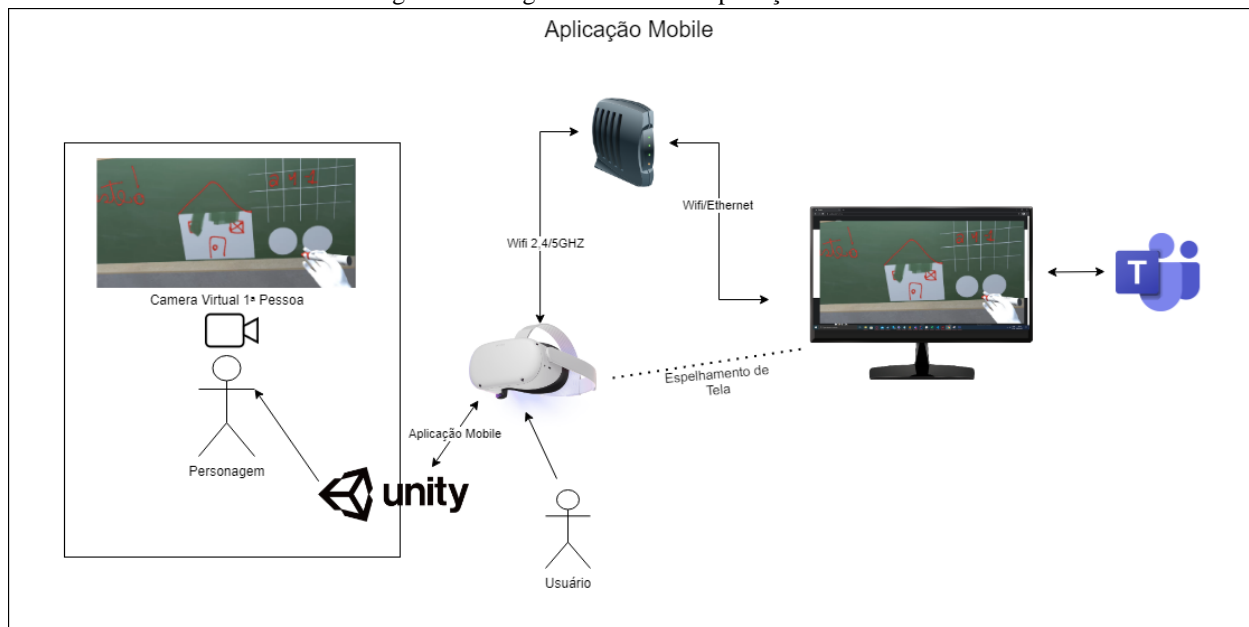
3.1 ESPECIFICAÇÃO

Figura 1 - Diagrama de uso da aplicação desktop



Fonte: Elaborado pelo autor.

Figura 2 - Diagrama de uso da aplicação mobile



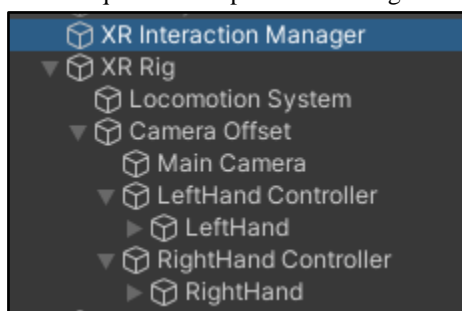
Fonte: Elaborado pelo autor.

Os diagramas inseridos devem considerar o modelo de estrutura e de comportamento do que foi desenvolvido. Se é um sistema com usuário final, inclua a lista de requisitos funcionais e não funcionais. Destaca-se que os **diagramas desenvolvidos bem como outros aspectos de especificação deverão obrigatoriamente constar nos apêndices quando não couberem nesta seção.**

3.2 IMPLEMENTAÇÃO

Para desenvolver usando Realidade Virtual dentro do Unity, foi utilizado o framework XR Interaction Toolkit, que é um pacote de componentes de alto nível para interação com dispositivos de RV desenvolvido pela Unity, com o intuito de ter um framework genérico que consiga ser compatível com o maior número possível de dispositivos de RV. Utilizando esse framework, é possível preparar uma cena em minutos com grande compatibilidade entre os diversos dispositivos de RV. Utilizando de seu componente `XR Rig`, cujo mesmo já disponibiliza de uma câmera posicionada relativa à posição do HMD, também dispõe de duas âncoras que representam o controle esquerdo e direito, em conjunto com o `XR Interaction Manager` que gerencia por de baixo dos panos os controles. Tendo esses componentes na cena, já é possível ter a posição do seu HMD e dos controles em tempo real no espaço 3d, assim como os eventos de todos os inputs dos dispositivos, sendo possível então, estruturar scripts customizados em cima desses dados para obter qualquer comportamento desejado para cada caso de uso.

Quadro 4 - Hierarquia do componente XR Rig dentro do Unity



Fonte: Elaborado pelo Autor.

Uma funcionalidade core do trabalho, é a manipulação e interação dos objetos virtuais com as mãos, o objeto quando segurado, deve manter suas propriedades físicas, deve colidir com objetos e não atravessá-los, e devem sempre se manter na posição e rotação relativa à posição virtual da mão do usuário.

Para obter esse resultado, em vez de optar por, transformar os objetos em corpos cinemáticos sem gravidade, e então, a cada frame estar atualizando a posição do objeto para a mesma posição das mãos, o que acabaria removendo a colisão desse objeto com outros corpos estáticos, foi tomada outra abordagem, de aplicar forças em cima desse objeto a fim de movê-lo até a mesma posição das mãos, assim mantendo sua colisão com todos os objetos, dando uma maior interatividade com o ambiente.

Para criar essa funcionalidade, foi criado um script, para cada mão do jogador, nomeado de `PickupHand`, esse script vai se encarregar de várias funções, escolher o objeto mais próximo a mão, pegar o objeto, movê-lo e rotacioná-lo, entre outras funções mais específicas a outras finalidades.

A grande maioria dos scripts Unity, herdam da classe `MonoBehaviour`, e possuem alguns métodos que são chamados em determinados momentos, dentre eles temos o `Awake()`, que similar ao `Start()` é um método que vai ser chamado no início do script, é comumente utilizado como inicializador de variáveis, e no script `PickupHand` utilizamos dessa função para popular a variável `hand` com o objeto que representa a mão 3D virtual que está localizado hierarquicamente abaixo do objeto que está com o script, para manipular esse objeto posteriormente.

Quadro 5 - Variáveis da classe PickupHand

```

7 public class PickupHand : MonoBehaviour
8 {
9     public float distToPickup = 0.3f;
10    public bool isHandClosed = false;
11
12    public XRController controller = null;
13    public LayerMask pickupableLayer;
14    public Rigidbody holdingTarget;
15    private GameObject hand;
16    private isTrigger isTrigger;
17    private GameObject holdingObjectHand;
18    private GameObject holdingObject;
19    private GameObject objectToHide;
20    private GameObject objectHandToHide;
21    private GameObject auxObjectToDestroy;
22    public GrabbedObjectsManager grabbedObjectsManager;
23    public PickupHand otherHand;
24    private bool firstMove = true;
25
26    public WhichHand whichHand = WhichHand.RIGHT;
27
28    4 references
29    public enum WhichHand { LEFT, RIGHT };
30
31    // Start is called before the first frame update
32    void Awake()
33    {
34        hand = this.transform.GetChild(0).gameObject;
35    }

```

Fonte: Elaborado pelo autor.

A parte principal do script, se dá dentro da função `MonoBehaviour.FixedUpdate()`, essa função é chamada de acordo com a frequência definida para o tempo do sistema de físicas do Unity, no caso do projeto, essa frequência é de cerca de 143 vezes por segundo, então esse método é chamado, 143 vezes por segundo, independente da taxa de quadros por segundo, nele é feito primeiramente uma simples verificação, utilizando do `XRController`, validamos se o botão de garra que representa o botão que vai fechar a sua mão virtual, está ou não pressionado, e então, entramos no método principal `GrabAndMoveObjects()`.

Quadro 6 - Método FixedUpdate da classe PickupHand

```

37 void FixedUpdate()
38 {
39     if (controller.inputDevice.TryGetFeatureValue(CommonUsages.gripButton, out bool gripped))
40     {
41         this.isHandClosed = gripped;
42     }
43
44     GrabAndMoveObject();
45 }
46

```

Fonte: Elaborado pelo autor

O script terá uma função diferente de acordo com o estado da variável `isHandClosed` que vai dizer se o usuário está atualmente com a mão fechada ou aberta, isso vai definir, se o usuário deveria estar segurando um objeto em suas mãos (mão fechada) ou ainda buscando algum objeto para pegar.

A primeira cláusula do `if`, vai ser executada caso a mão esteja aberta, onde será chamado o método `FindObjectToGrab()`.

Quadro 7 - Método GrabAndMoveObject da classe PickupHand

```

47 private void GrabAndMoveObject()
48 {
49     if (!isHandClosed)
50     {
51         if (auxObjectToDestroy != null)
52         {
53             Destroy(this.auxObjectToDestroy);
54             this.auxObjectToDestroy = null;
55         }
56         FindObjectToGrab();
57     }
58     else
59     {
60         //Se a mão está fechada e possui um objeto selecionado
61         if (holdingTarget && holdingTarget != otherHand.holdingTarget)
62         {
63             if (firstMove)
64             {
65                 Renderer[] childArray = holdingTarget.GetComponentsInChildren<Renderer>();
66                 foreach (Renderer render in childArray)
67                 {
68                     render.material.SetFloat("_OutlineWidth", 0.000f);
69                 }
70                 this.holdingTarget.position = transform.position;
71                 this.holdingTarget.rotation = transform.rotation;
72                 firstMove = false;
73             }
74             //Se a mão está visível, esconde ela.
75             if (hand.active)
76             {
77                 StopCoroutine(showHand());
78                 hand.SetActive(false);
79                 showHoldingObjectHand();
80             }
81             bool isObjectAgainstWall = isHoldingTargetAgainstWall();
82             MoveAndRotateHoldingObject(isObjectAgainstWall);
83         }
84     }
85 }
86
87

```

Fonte: Elaborado pelo autor.

O método `FindObjectToGrab()`, utilizando da biblioteca `Physics` do próprio Unity, calcula uma esfera de raio modificável em relação a posição da sua mão, e todo objeto que possuir um `Collider`, dentro desse raio, será então retornado dentro de um array do tipo `Collider`. Tendo isso, é feito o cálculo de distância, entre esses objetos, e então define-se que o objeto que possuir o `Collider` com a menor distância em relação a mão, será então, definido na variável `holdingTarget` como o objeto a ser “pegado”.

Quadro 8 - Método FindObjectToGrab da classe PickupHand

```

126 private void FindObjectToGrab()
127 {
128     //Mostra a mão quando aberta
129     if (!hand.active)
130         StartCoroutine(showHand());
131     if (this.holdingObjectHand != null && this.holdingObjectHand.active)
132     {
133         this.holdingObjectHand.SetActive(false);
134     }
135     //Verifica em um raio próximo a mão todos os objetos com colliders
136     Collider[] colliders = Physics.OverlapSphere(transform.position, distToPickup, pickupableLayer);
137     if (colliders.Length > 0)
138     {
139         //Se existe objeto dentro do raio, pegue o mais próximo da mão
140         float lastDistance = float.MaxValue;
141         Collider closestCollider = new Collider();
142         foreach(Collider collider in colliders)
143         {
144             float distance = Vector3.Distance(collider.transform.position, transform.position);
145             if(distance < lastDistance)
146             {
147                 lastDistance = distance;
148                 closestCollider = collider;
149             }
150         }
151         if (closestCollider != null)
152         {
153             Rigidbody aux = closestCollider.transform.root.GetComponent<Rigidbody>();
154             if (otherHand.holdingTarget != aux && aux != this.holdingTarget)
155             {
156                 Renderer[] childArray;
157                 if (holdingTarget != null)
158                 {
159                     childArray = holdingTarget.GetComponentsInChildren<Renderer>();
160                     foreach (Renderer render in childArray)
161                     {
162                         render.material.SetFloat("_OutlineWidth", 0.000f);
163                     }
164                 }
165                 holdingTarget = aux;
166                 childArray = holdingTarget.GetComponentsInChildren<Renderer>();
167                 Debug.Log(childArray.Length);
168                 foreach (Renderer render in childArray)
169                 {
170                     render.material.SetColor("_OutlineColor", new Color(0, 255, 244, 1));
171                     render.material.SetFloat("_OutlineWidth", 0.005f);
172                 }
173                 firstMove = true;
174             }
175         }
176     }

```

Fonte: Elaborado pelo autor.

Entrando na outra cláusula else do método `GrabAndMoveObject()`, que será executada caso a mão do usuário esteja fechada, vai verificar se previamente fora populado algum objeto na variável `holdingTarget`, que seria o objeto mais próximo da mão, e então, é chamado o método `MoveAndRotateHoldingObject()` que fica encarregado de aplicar forças para rotacionar e translatar o objeto até a posição da mão.

Utilizando de funções `Quaternion`, e `Vector3`, é aplicado uma vetor de velocidade ao objeto, o valor dese vetor, se dá pelo delta da distância entre a posição do objeto segurado, e a posição da mão, e também é aplicado uma velocidade angular, que no unity, se dá por um vetor tridimensional, onde cada dimensão representa um eixo de ângulo euler, a rotação de cada eixo é dada pela rotação quatérnion da mão, multiplicada pelo inverso da rotação quatérnion do objeto segurado, que então é convertida para ângulos eulers.

Quadro 9 - Método MoveAndRotateHoldingObject da classe PickupHand

```

88 private void MoveAndRotateHoldingObject(bool isObjectAgainstWall)
89 {
90     if (holdingTarget.name.Contains("ShapeParent") && this.auxObjectToDestroy == null){
91         this.auxObjectToDestroy = GameObject.Instantiate(holdingTarget.gameObject) as GameObject;
92         Destroy(auxObjectToDestroy.transform.root.GetComponent<changeColor>());
93         this.auxObjectToDestroy.GetComponent<Rigidbody>().constraints = RigidbodyConstraints.None;
94         this.auxObjectToDestroy.GetComponent<Rigidbody>().useGravity = true;
95         holdingTarget = this.auxObjectToDestroy.GetComponent<Rigidbody>();
96     }
97     //Move a posição do objeto na mão
98     this.holdingTarget.velocity = (transform.position - holdingTarget.transform.position) / Time.fixedDeltaTime;
99     if (!isObjectAgainstWall)
100     {
101         //Rotaciona o objeto na mão
102         this.holdingTarget.maxAngularVelocity = 20;
103         Quaternion deltaRot = transform.rotation * Quaternion.Inverse(this.holdingTarget.transform.rotation);
104         Vector3 eulerRot = new Vector3(Mathf.DeltaAngle(0, deltaRot.eulerAngles.x), Mathf.DeltaAngle(0, deltaRot.eulerAngles.y),
105             Mathf.DeltaAngle(0, deltaRot.eulerAngles.z));
106         eulerRot *= 0.95f;
107         eulerRot *= Mathf.Deg2Rad;
108         this.holdingTarget.angularVelocity = eulerRot / Time.fixedDeltaTime;
109     }
110 }
111

```

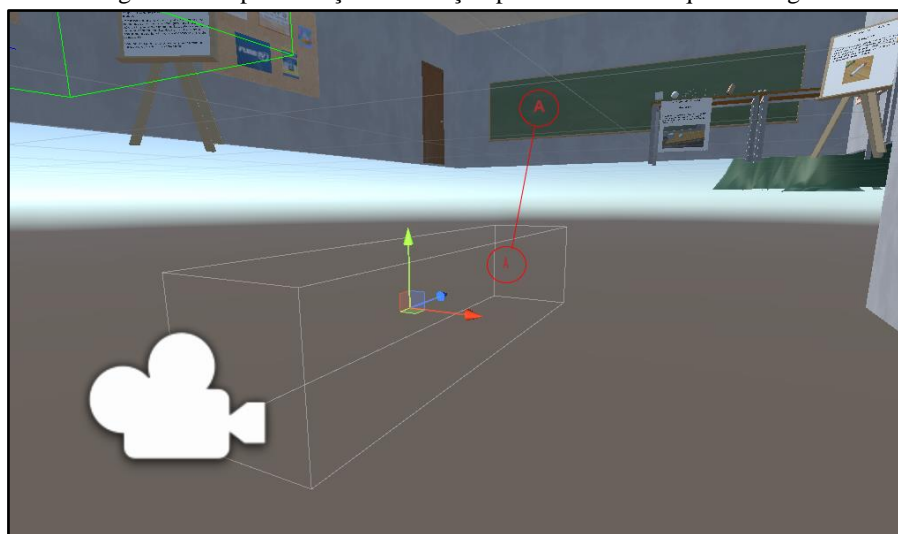
Fonte: Elaborado pelo autor.

Esses trechos de códigos, tomam conta da parte da localização e então, movimentação dos objetos na mão virtual do usuário, ainda existe mais alguns trechos de código nesse script que tomam conta de controlar algumas outras particularidades do projeto, mas que não carece de detalhes.

Tendo a parte de manipulação dos objetos codificada, também será necessário criar uma mecânica de escrita, que permita ter um quadro negro, similar ao que esperamos de um quadro negro real, onde pode-se escrever, e apagar, porém também foram implementadas outras funcionalidades, como a de poder posicionar, redimensionar formas e objetos para serem postas sobre o quadro, como quadrados, círculos, gráficos e matrizes.

Para permitir ter um objeto com uma textura que seria desenhável em tempo real, foi utilizado de uma RenderTexture, onde uma câmera virtual ortográfica, teria essa RenderTexture configurada como alvo, como resultado disso, tudo que entrasse dentro do campo de visão da câmera, seria então desenhado nessa RenderTexture, e essa RenderTexture então, seria aplicada em um material, e esse material aplicado como textura de um objeto, nesse caso, de um plano translúcido posicionado ligeiramente a frente do quadro negro.

Figura 3 - Representação da solução para desenhar no quadro negro



Fonte: Elaborado pelo autor.

Assim teríamos uma relação, entre o usuário manipulando esse objeto com o material do RenderTexture, com a câmera virtual posicionada em outro espaço. Quem controla toda a lógica por trás da funcionalidade de “desenhar” no quadro, é o script NewTexturePainter.

Dentro do `MonoBehaviour.FixedUpdate()` o script primeiramente, verifica se o usuário está segurando algum objeto que seria uma caneta ou um apagador, e então, verifica se esse objeto está pressionado contra o quadro negro, se sim, então o script vai iniciar a rotina `DoAction()`.

Quadro 10 - Parte do código dentro do `FixedUpdate` da classe `NewTexturePainter`

```
85         if (pen != null && pen.GetComponent<IsTrigger>() != null && pen.GetComponent<IsTrigger>().isTriggered)
86         {
87             if (pen.name.Contains("Shape"))
88             {
89                 PlaceOverDrawingBoard();
90             }
91             else
92             {
93                 DoAction();
94             }
95         }
```

Fonte: Elaborado pelo autor.

Dentro do método `DoAction()` é feito então, o instanciamento de uma `sprite`, que vai representar efetivamente a pintura do canetão, a textura que será visível ao escrever algo, então, é verificado se o objeto segurado é, um apagador, um canetão vermelho, azul, ou verde, e dependendo de sua cor, será alterado a coloração do material, assim como do seus componentes, também é feito um redimensionamento dessa `sprite`, para corrigir a distorção de proporção entre o material onde a textura está aplicada que no caso é um retângulo, e a resolução do `RenderTarget` que é um quadrado.

Quadro 11 - Trecho do código `DoAction` da classe `NewTexturePainter`

```
224     currentPaintingBrush = (GameObject)Instantiate(Resources.Load("TexturePainter-Instances/BrushEntity"));
225     currentPaintingBrush.AddComponent<TrailRenderer>();
226     if (pen.name.Contains("RED"))
227     {
228         currentPaintingBrush.GetComponent<SpriteRenderer>().color = Color.red;
229         currentPaintingBrush.transform.localScale = new Vector3(1, 1, 1) * 0.0008f;
230         trailMaterial.SetColor("_EmissionColor", Color.red);
231         currentPaintingBrush.GetComponent<TrailRenderer>().material = trailMaterial;
232         currentPaintingBrush.GetComponent<TrailRenderer>().time = 0.25f;
233         currentPaintingBrush.GetComponent<TrailRenderer>().startWidth = 1 * 0.002f;
234         currentPaintingBrush.GetComponent<TrailRenderer>().endWidth = 1 * 0.002f;
235         currentPaintingBrush.GetComponent<TrailRenderer>().minVertexDistance = 0.005f;
236         currentPaintingBrush.GetComponent<TrailRenderer>().startColor = Color.red;
237         currentPaintingBrush.GetComponent<TrailRenderer>().endColor = Color.red;
238         currentPaintingBrush.GetComponent<TrailRenderer>().shadowCastingMode = UnityEngine.Rendering.ShadowCastingMode.Off;
239         currentPaintingBrush.GetComponent<TrailRenderer>().textureMode = LineTextureMode.DistributePerSegment;
240         brushColor = Color.red;
241         brushColor.a = 1.0f;
242     }
243 }
```

Fonte: Elaborado pelo autor.

Após instanciar o objeto que será a `sprite` do desenho, é necessário buscar onde esse objeto deve ser posicionado em relação a posição do canetão, para isso, é chamado o método `HitTestUVPosition()` onde é feito um `raycast` partindo do objeto, em direção ao eixo Z frontal do mesmo, e caso o hit desse `raycast` colida com um objeto que esteja na camada `DrawingGlass` que seria o objeto desenhável, então calcula-se a partir desse acerto, a posição correspondente na câmera virtual ortográfica que está apontando para a `RenderTarget`, tendo essa posição, é então movida a `sprite` do desenho até essa localização, resultando em uma projeção dessa `sprite` no quadro negro, e essa posição é atualizada sempre que o canetão ou outro objeto especificado, esteja encostando no quadro negro. O fato de a câmera virtual, estar posicionada em um local do mundo, onde não existe nenhum fundo que deve ser desenhado pelo motor gráfico, nunca será feita a limpeza da tela a cada frame, resultando em uma trilha por onde a `sprite` foi movida.

Quadro 12 - Método HitTestUVPosition da classe NewTexturePainter

```

305 bool HitTestUVPosition(ref Vector3 uvWorldPosition)
306 {
307     RaycastHit hit;
308     if (pen.GetComponent<isTrigger>().isTriggered && Physics.Raycast(
309         pen.transform.GetChild(0).position, pen.transform.GetChild(0).forward, out hit, 10f, 3 << LayerMask.NameToLayer("DrawingGlass")))
310     {
311         MeshCollider meshCollider = hit.collider as MeshCollider;
312         if (meshCollider == null || meshCollider.sharedMesh == null)
313             return false;
314         Vector2 pixelUV = new Vector2(hit.textureCoord.x, hit.textureCoord.y);
315         uvWorldPosition.x = pixelUV.x - canvasCam.orthographicSize; //To center the UV on X
316         uvWorldPosition.y = (pixelUV.y - canvasCam.orthographicSize) / 5; //To center the UV on Y
317         uvWorldPosition.z = 0.0f;
318         return true;
319     }
320     else
321     {
322         return false;
323     }
324 }
325

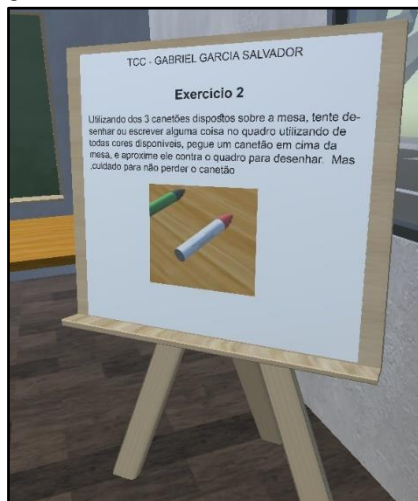
```

Fonte: Elaborado pelo autor.

A sala de aula virtual foi modelada em sua maior parte utilizando de primitivas do Unity, como planos, blocos, cilindros e esferas, em conjunto de materials e textures, outras partes da sala e alguns objetos foram adquiridos através de assets gratuitos disponibilizados no website da loja Unity, assim como de assets e texturas gratuitas disponibilizadas na internet formando assim um espaço virtual com características próximas a de uma sala de aula real, a fim de trazer uma familiaridade ao usuário.

Dentro do ambiente, foram desenvolvidos 3 exercícios básicos para que o usuário possa obter um primeiro contato com a manipulação dos objetos virtuais com os atadores do dispositivo de realidade virtual, assim como ensinar a utilizar dos recursos disponibilizados, como o canetão, apagador, e as formas desenháveis, esses exercícios foram postos virtualmente em um cavalette para que o usuário os localize e leia as instruções.

Figura 4 - Cavalette contendo o Exercício 2



Fonte: Elaborado pelo autor.

Ao desenvolver uma aplicação de RV é necessário tomar certos cuidados, pois deve se considerar que a renderização será feita duas vezes, uma para cada olho, assim possibilitando a estereoscopia, e essa imagem será de alta resolução, ainda sendo necessário manter uma taxa de quadros estável mínima de 72 quadros por segundo, o que é imprescindível para evitar desconforto ao usuário ao se movimentar, e atingir essa taxa de quadros por segundo em um dispositivo móvel como o Oculus Quest 2 se torna uma tarefa ainda mais complexa, considerando o poder computacional limitado do mesmo.

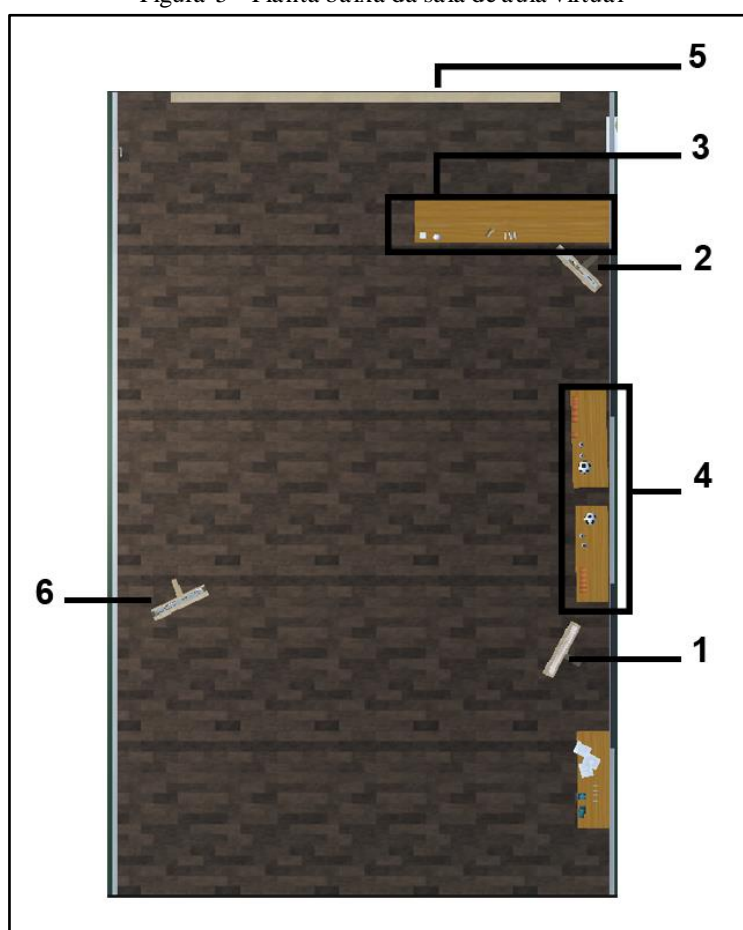
Para conseguir atingir uma taxa de quadros aceitável no deploy da aplicação, foi necessário optar por reduzir ao máximo o número de polígonos dos objetos na cena, com isso em mente, foi utilizado apenas de assets "LowPoly" para modelar o ambiente, principalmente a área externa, texturas entretanto são importantes que tenham alta resolução, pois é algo que se destaca ao usuário em uma aplicação de RV, a possibilidade de ler as coisas e enxergar os detalhes. Além disso, deve se considerar o custo computacional dos scripts que se mantém rodando por de baixo dos panos, calculando

a física, posição e comportamento dos eventos na cena até centenas de vezes por quadro. Então para mitigar esse custo, a qualidade gráfica do Unity foi definida para a menor fidelidade possível no deploy do APK, assim atingindo o objetivo de uma taxa de quadros estável de no mínimo 72 quadros com pouca flutuação.

4 RESULTADOS

Nesse capítulo serão apresentados os testes de uso realizados, assim como a forma que os mesmos foram realizados, as considerações e seus resultados. O teste foi ministrado para 3 grupos diferentes de usuários, o grupo A seria composto pelos usuários finais, subdividido entre professores da área da computação e, uma professora de química, o grupo B seria de usuários desenvolvedores com maior afinidade a tecnologia, e o grupo C de usuários sem relação com a área de educação e pouca afinidade com a tecnologia. O teste foi conduzido utilizando do HMD Oculus Quest 2 com uma build mobile da aplicação rodando no próprio *System-on-a-Chip* (SOC) do dispositivo sem a necessidade de fios.

Figura 5 - Planta baixa da sala de aula virtual



Fonte: Elaborado pelo autor.

Quadro 13 - Legenda da planta baixa da sala de aula virtual

Identificador	Descrição
1	Cavalete contendo o exercício 1
2	Cavalete contendo o exercício 2
3	Mesa com os canetões, apagador, formas geométricas e o exercício 3
4	Mesa com objetos manuseáveis
5	Quadro Negro
6	Cavalete contendo as instruções de uso da aplicação

Fonte: Elaborado pelo autor.

4.1 GRUPO A

O teste com o grupo A de professores, foi realizado dentro da instituição da FURB - Universidade Regional de Blumenau, onde uma sala foi reservada para a agenda que teria duração de 1 hora e 30 minutos, dentro desse intervalo, cada usuário teve apenas cerca de 15 minutos para testar a aplicação. Nessa reunião, um professor da área de computação junto de uma professora da área de química testou a utilização da aplicação, compareceram também, outros professores que acompanharam os testes.

Dentro do dispositivo, foi delimitado uma área de 3x2 metros como a área segura do sistema de guardião do dispositivo, onde os usuários poderiam se movimentar fisicamente dentro desse espaço, e teriam um alerta caso ultrapassassem esse limite evitando que os mesmos colidam com pessoas e objetos, também foi configurado o espelhamento do dispositivo, para que tudo que estivesse sendo visualizado pelo usuário fosse também projetado para um notebook conectado na mesma rede local, que então projetava a imagem em um retroprojetor para melhor visualização de todos presentes.

O primeiro usuário a realizar o teste do grupo A foi o professor da área de computação, ao entrar no ambiente virtual, o mesmo se mostrou surpreso com a fidelidade da imersão propiciada pelo equipamento e a aplicação, o professor então foi instruído a manipular os controles para conseguir se locomover dentro do ambiente, e então foi dito para o mesmo se locomover até os cavaletes dentro do ambiente virtual que contam as informações de como realizar os exercícios propostos para obter uma noção do uso da aplicação, então se aproximou do quadro de exercício 1 (1), a onde foi instruído a tentar pegar blocos com as mãos virtuais e então posiciona-los de maneira a montar uma pirâmide de blocos (4), a tarefa foi realizada sem muita dificuldade. Após isso, o usuário foi até o segundo caveleto (2), onde continha o exercício 2, que requirava para o usuário, que desenhasse ou escrevesse algo utilizando dos canetões (3) no quadro negro (5), o usuário não teve muita dificuldade nesse exercício que logo foi concluído. Para o último exercício 3 (3), o usuário teria que utilizar das formas dispostas sobre a mesa (3), para conseguir desenhar as respectivas formas no quadro negro (5), nessa tarefa foi observada uma maior dificuldade, devido ao nível maior de interação necessária com os botões do controle do dispositivo, após realizar os exercícios o usuário elogiou a imersão, comentou sobre a dificuldade de utilizar os controles mas que isso seria uma questão de se acostumar, também sugeriu que os objetos poderiam ficar na mão sem ter que manter um botão pressionado para que o mesmo se mantenha na mão, e também comentou sobre uma leve vertigem devido a locomoção virtual.

O segundo usuário do grupo A, foi uma professora da área de química, que já estava observando o primeiro usuário realizar as tarefas através do espelhamento de tela do dispositivo. Ao colocar o dispositivo e entrar no ambiente a mesma mostrou uma maior dificuldade em relacionar os botões do controle com as ações no mundo virtual, assim como para se locomover, fazendo com que a mesma muitas vezes optasse por andar fisicamente até os limites do guardião em vez de utilizar da locomoção virtual, a usuária acabou pulando o exercício 1 (1) e foi diretamente para o exercício 2 (2), onde já teve uma dificuldade para saber a distância de sua mão virtual em relação aos objetos para pega-los, possivelmente devido a algum ajuste na distância intra-pupilar dos olhos que não havia sido calibrado antes da usuária utilizar o dispositivo, após conseguir pegar um canetão (3) e se locomover até o quadro negro (5), questionou como faria para desenhar, o que deveria ser realizado de maneira intuitiva, apenas aproximar o canetão no quadro, mas talvez devido a confusão no uso dos controles acabou pressupondo que seria necessário apertar algo a mais para realizar essa ação, porém após ser instruída sobre como realizar o desenho, a mesma conseguiu desenhar no quadro negro (5), e então questionou como faria para parar de desenhar, onde então foi dito para apenas afastar o canetão do quadro negro, ela desenhou uma equação de química básica, porém acabou largando o botão que mantém o objeto na mão fazendo com que ele caísse algumas vezes involuntariamente, após completar o desenho, ela se locomoveu até a mesa do exercício 3 (3), pegou a forma do cubo (3) para desenhá-lo no quadro, e então aproximou a forma no quadro (5) e apertou o botão de gatilho do controle para desenhar, mas com dificuldade e sem utilizar dos recursos que a ferramenta dispõe, como redimensionamento e troca de cores da forma. A usuária não comentou especificamente sobre a sua experiência, porém foi perceptível que seria necessário um maior tempo de uso com o dispositivo para conseguir utilizar do equipamento como um todo de forma mais natural.

Em ambos os usuários é possível verificar algumas dificuldades em comuns, a mais notável é a de locomoção dentro do ambiente virtual, que poderia ser resolvido utilizando apenas da locomoção física, dentro de um espaço físico maior ou de um espaço virtual menor de forma que os limites virtuais e físicos coincidam, tornando nula a possibilidade de *motion sickness*, outra maneira mais prática de evitar o sintoma, seria implementando a locomoção virtual com um sistema de teletransporte, onde o usuário poderia designar no chão onde que ele gostaria de se posicionar virtualmente, e então após uma transição que bloqueia o campo visual do usuário, o mesmo é movimentado até o local desejado, evitando que tenha qualquer tipo de enjoo. Outra dificuldade foi na utilização dos demais controles do dispositivo, mesmo que a curva de aprendizado não seja grande, para quem nunca utilizou do dispositivo, 15 minutos não é o suficiente para se adequar a todos os botões dos controles, mas a opção de poder apenas apertar uma vez o botão para pegar e manter na mão o objeto poderia ser uma opção customizável de acordo com a preferência do usuário. E por fim, a forma que foi implementada a ferramenta de desenho das formas no quadro também gerou dificuldade, mas que em sua grande maioria se deu pelo motivo anterior, da falta de adequação ao uso dos controles, porém uma maneira mais intuitiva de utilizar essa ferramenta pode ser uma alternativa.

4.2 GRUPO B

O segundo grupo foi composto de um usuário que é um jovem estudante desenvolvedor de softwares, que possui conhecimento de computação e já tinha afinidade com o dispositivo Oculus Quest 2. O usuário testou o software com seu próprio dispositivo sem supervisão ou observação, ele teve um pouco de dificuldade em utilizar as formas para desenhar no quadro negro, mas teve grande facilidade para executar todos os 3 exercícios propostos. Seu parecer foi positivo, ressaltando a imersão e o potencial de utilizar essa tecnologia como uma alternativa ao ensino comum no futuro próximo.

4.3 GRUPO C

O terceiro grupo foi composto de uma usuária jovem que não possui nenhuma relação com a área de computação ou educação, e que possuía um pouco de afinidade com o Oculus Quest 2, foi preparado um guardião em seu ambiente físico de 2x2 metros, e a usuária foi instruída a se locomover até o cavalete do exercício 1 (1) e completar o mesmo, apresentou pouca dificuldade para manusear os objetos virtuais, mas quando tentava posicionar objetos e quando largava os, acabava mantendo a mão no mesmo local involuntariamente, isso fazia com que a mão virtual reaparecesse e colidisse dentro do objeto fazendo com que o mesmo se deslocasse, porém conseguiu completar o exercício, então se locomoveu até o segundo exercício (2), onde conseguiu de maneira intuitiva pegar um canetão (3) e fazer um desenho no quadro (5) sem qualquer dificuldade, completando assim o exercício, por fim, no último exercício 3 (3), ao utilizar das formas para desenhar no quadro (5), apresentou certa dificuldade na compreensão dos controles para conseguir manusear e desenhar a forma no quadro, porém após alguns minutos de tentativa e erro conseguiu começar a manusear a ferramenta sem dificuldades e assim completando o exercício. A usuária elogiou a forma intuitiva que podia desenhar no quadro e manipular e montar os blocos assim como os outros objetos disponíveis, porém ficou com um pouco de enjoo devido ao desconforto do dispositivo na cabeça e com a movimentação que julgou como rápida demais.

Os resultados corroboram com as impressões

De modo a ampliar o seu caráter científico, todos os TCCs devem apresentar e discutir resultados não limitados à comparação com os trabalhos correlatos. Devem ser apresentados os casos de testes do software, destacando objetivo do teste, como foi realizada a coleta de dados e a apresentação dos resultados obtidos, preferencialmente em forma de gráficos ou tabelas, fazendo comentários sobre os mesmos. Também é sugerida a comparação com os trabalhos correlatos apresentados na fundamentação teórica.

5 CONCLUSÕES

Os resultados obtidos pelos testes foram favoráveis ao uso da tecnologia na área da educação, assim como também foi destacado pelos diversos autores ao longo do artigo, a tecnologia permite que mesmo dentro de suas próprias casas, ainda seja possível estar em um ambiente completamente diferente exercendo sua profissão sem as restrições visuais do mundo físico. Foi perceptível que os usuários submetidos ao teste, se sentiram completamente imersos durante o uso da aplicação, e conseguiram realizar os exercícios propostos dentro da aplicação, mostrando que seria possível utilizar da mesma para ministrar uma aula online. A tecnologia que encontramos nos dispositivos HMD de hoje ainda não são perfeitas, questões como ergonomia do dispositivo, densidade de pixels, campo de visão, e interfaces para interação com maior imersão e feedback háptico o para as mãos assim como novas maneiras de se locomover dentro do mundo virtual, são possíveis melhorias para as gerações futuras da tecnologia no mercado consumidor, agregando ainda mais para a experiência e viabilidade do seu uso na educação.

Uma possível extensão desse trabalho seria começar a abranger dentro do ambiente virtual, outros ambientes e objetos mais específicos para cada tipo de disciplina como colocado pelo autor Dong (2016), maximizando assim o potencial do uso da tecnologia em colocar o usuário em virtualmente qualquer local com qualquer objeto implementado. Outra possibilidade, seria implementar diferentes tipos de locomoção e acessibilidade, em específico, implementar o tracking das mãos disponível de forma nativa com o dispositivo Oculus Quest 2, de forma a substituir ou como alternativa aos controles comuns, o uso apenas do tracking reduziria o número de formas de interagir com o mundo virtual mas aumentaria a intuitividade reduzindo a curva de aprendizado. Com esses devidos ajustes, seria ótimo um novo teste ser aplicado aos usuários finais para qualificar a experiência deles com essas novas funcionalidades, e então comparar as diferenças encontradas nas impressões com as encontradas nesse artigo, assim como validar o uso da aplicação em um caso de uso de uma aula online, avaliando a opinião dos alunos.

REFERÊNCIAS

- HODGES, Charles *et al.* **The Difference Between Emergency Remote Teaching and Online Learning**. 2020. Disponível em: <https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning>. Acesso em: 18 jun. 2021.
- TERÄS, Marko; SUORANTA, Juha; TERÄS, Hanna; CURCHER, Mark. Post-Covid-19 Education and Education Technology ‘Solutionism’: a seller’s market. **Postdigital Science And Education**, [S.L.], v. 2, n. 3, p. 863-878, 13 jul. 2020. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s42438-020-00164-x>. Disponível em: <https://link.springer.com/article/10.1007%2Fs42438-020-00164-x>. Acesso em: 18 jun. 2021.
- LEE, Chris. Real Learning in a virtual classroom is difficult. **Arstechnica**, 2020. Disponível em <https://arstechnica.com/staff/2020/03/a-crash-course-in-virtual-teaching-real-learning-achieved>. Acesso em: 18 jun. 2021
- TABRIZI, M.H.N. Agent and Virtual Reality-based Course Delivery System. In: **APPLIED COMPUTING INTERNATIONAL CONFERENCE**, 08., 2008, Algarve. Applied Computing 2008. Algarve: Iadis, 2008. v. 1, p. 27-30
- FREITAS, Olga. **Equipamentos e materiais didáticos**. Brasília: Universidade de Brasília, 2009. 132 p. Disponível em: http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=614-equipamentos-e-materiais-didaticos&Itemid=30192. Acesso em: 19 jun. 2021.
- FLANAGAN, Graham. **The incredible story of the 'Virtual Boy' Nintendo's VR headset from 1995 that failed spectacularly**. 2018. Disponível em: <https://www.businessinsider.com/nintendo-virtual-boy-reality-3d-video-games-super-mario-2018-3>. Acesso em: 19 jun. 2021.
- The History of VR: 5 Eras Of Evolving A New Reality. **42GEARS**, 2019. Disponível em: <https://www.42gears.com/blog/the-history-of-vr-5-eras-of-evolving-a-new-reality/>. Acesso em: 19 jun 2021
- TORI, Romero; KIRNER, Claudio; SISCOOTTO, Robson. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. 2006. Disponível em: https://www.academia.edu/22716950/Fundamentos_e_Tecnologia_de_Realidade_Virtual_e_Aumentada_Apostila_do_Pré_Simpósio_VIII_Symposium_on_Virtual_Reality_Belém_PA_02_de_Maio_de_2006. Acesso em: 19 jun 2021.
- STATISTA. **Augmented reality (AR) and virtual reality (VR) headset shipments worldwide from 2020 to 2025**. Disponível em: www.statista.com/statistics/653390/worldwide-virtual-and-augmented-reality-headset-shipments/. 2020. Acesso em: 19 jun. 2021.
- DONG, Xisong. An overall solution of Virtual Reality Classroom. In: IEEE INTERNATIONAL CONFERENCE ON SERVICE OPERATIONS AND LOGISTICS, AND INFORMATICS, 11., 2016, Beijing. **International Conference on Service Operations and Logistics, and Informatics (SOLI)**. Beijing: Ieee, 2016. p. 120-124. Disponível em: <https://ieeexplore.ieee.org/document/7551672>. Acesso em: 20 jun. 2021.
- INTERNATIONAL CONFERENCE OF EDUCATION, RESEARCH AND INNOVATION, 12., 2019, Seville. **DEVELOPING A VIRTUAL REALITY EDUCATIONAL ENVIRONMENT FOR TRAFFIC EDUCATION**. Seville: Iceri, 2019. Disponível em: https://www.researchgate.net/publication/338106916_DEVELOPING_A_VIRTUAL_REALITY_EDUCATIONAL_ENVIRONMENT_FOR_TRAFFIC_EDUCATION. Acesso em: 20 jun. 2021.

APÊNDICE A – DIAGRAMAS DE ESPECIFICAÇÃO

É fundamental que todo projeto apresente alguma forma de especificação do que foi desenvolvido. A descrição é opcional. Assim, **este apêndice deve conter os diagramas de especificação que não couberam ao longo do texto**. Os diagramas devem conter legendas numeradas na sequência do artigo.

Cada apêndice deve iniciar em uma nova página.

APÊNDICE B – XXX

Podem ser inseridos outros apêndices no artigo tais como códigos de implementação, telas de interface, instrumentos de coleta de dados, entre outros. **Apêndices são textos elaborados pelo autor** a fim de complementar sua argumentação. Os apêndices são identificados por letras maiúsculas consecutivas, seguidas de um travessão e pelos respectivos títulos. Deve haver no mínimo uma referência no texto anterior para cada apêndice. Colocar sempre um preâmbulo no apêndice. Caso existam tabelas ou ilustrações, identifique-as através da legenda, seguindo a numeração normal das legendas do artigo.

ANEXO A – DESCRIÇÃO

Elemento opcional, **anexos são documentos não elaborados pelo autor**, que servem de fundamentação, comprovação ou ilustração, como mapas, leis, estatutos, entre outros. Os anexos são identificados por letras maiúsculas consecutivas, seguidas de um travessão e pelos respectivos títulos. Deve haver no mínimo uma referência no texto anterior para cada anexo. Colocar sempre um preâmbulo no anexo. Caso existam tabelas ou ilustrações, identifique-as através da legenda, seguindo a numeração normal das legendas do artigo.

6 DESCRIÇÃO DA FORMATAÇÃO

A seguir são apresentadas observações gerais sobre o texto do artigo do Trabalho de Conclusão de Curso (TCC).
Observa-se que esta descrição deve ser retirada do texto final.

Na confecção do texto deve-se:

- a) usar frases curtas. Segundo Teodorowitsch (2003, p. 3), “Frases com mais de duas linhas aumentam o risco de o leitor não compreender a ideia ou de entendê-la de forma equivocada.”;
- b) usar linguagem impessoal (usar a terceira pessoa do singular) e verbo na voz ativa (a ação é praticada pelo sujeito), com conexão entre os parágrafos;
- c) não usar palavras coloquiais;
- d) não usar palavras repetidas em demasia;
- e) usar verbos no presente quando for referir-se a partes do trabalho que já se encontram disponíveis no texto;
- f) destacar palavras em língua estrangeira em itálico, conforme descrito abaixo:
 - nome de software, ferramenta, aplicativo, linguagem de programação, plataforma, empresa: não deve ser escrito em itálico (exemplos: Delphi 7, Pascal, Object Pascal, Java, JavaScript, Java 2 Micro Edition, Basic, Microsoft Visual C++, C, Windows, Linux, MySQL, Oracle, Eclipse 3.0, Enterprise Architect, Rational Rose, Microsoft, Sun Microsystems);
 - citações: o sobrenome do autor ou o nome da instituição responsável pela autoria do documento citado não deve ser escrito em itálico (exemplo: Segundo Sun Microsystems (2004), ...);
 - palavras em língua estrangeira encontradas nos dicionários nacionais: não devem ser grafadas em itálico (exemplos: software, hardware, web, Internet);
 - demais palavras em língua estrangeira: devem ser escritas em itálico (exemplos: *palmtop*, *classpath*, *play*, etc.). No entanto, Teodorowitsch (2003, p. 7), sugere que alguns termos em língua inglesa devem ser substituídos por termos em português (exemplos: núcleo em vez de *kernel*, aprendizagem de máquina em vez de *machine learning*, etc.);
- g) observar as seguintes regras quanto ao uso de siglas:
 - colocar as siglas entre parênteses precedidas pela forma completa do nome, quando aparecem pela primeira vez no texto (exemplos: Associação Brasileira de Normas Técnicas (ABNT), Trabalho de Conclusão de Curso (TCC)). Caso exista uma lista de siglas na parte pré-textual do volume final, pode-se usar somente a sigla, quando aparecer pela primeira vez no texto;
 - usar apenas a sigla nas demais ocorrências no texto;
 - escrever as siglas em letras maiúsculas e não usar itálico;
 - escrever o plural das siglas sem apóstrofo (exemplos: PCs, APIs, PDAs) e determinar o gênero da sigla conforme o gênero do primeiro substantivo do seu nome (exemplo: o TCC – o Trabalho de Conclusão de Curso).

6.1 FORMATAÇÃO

A formatação geral para apresentação do documento, descrita na NBR 14724 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2011), é a seguinte:

- a) o texto divide-se em capítulos, seções e subseções (até cinco divisões);
- b) a apresentação de citações em documentos deve seguir a NBR 10520 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b);
- c) a descrição das referências bibliográficas deve estar de acordo com a NBR 6023 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002a).

Observa-se ainda que todo capítulo, seção ou subseção deve ter no mínimo um texto relacionado.

O artigo deve ser digitado usando as fontes e formatação de parágrafos deste modelo, indicadas no Quadro 14.

Quadro 14– Estilos do modelo

USO	FORMATO
título de capítulo ou seção primária (1)	TF-TÍTULO 1 (Times New Roman, 10pt, negrito, maiúsculas)
título de seção secundária (1.1)	TF-TÍTULO 2 (Times New Roman, 10pt, maiúsculas)
título de seção terciária (1.1.1)	TF-Título 3 (Times New Roman, 10pt, minúsculas, exceto a 1ª letra da 1ª palavra do título e de nomes próprios)
título de seção quaternária (1.1.1.1)	TF-Título 4 (mesma formatação seção terciária)
título de seção quinária (1.1.1.1.1)	TF-Título 5 (mesma formatação seção terciária)
texto	TF-TEXTO (Times New Roman, 10pt)
citação direta com mais de três linhas	TF-CITAÇÃO (Times New Roman, 9pt, com recuo de 4 cm)
itens (alíneas)	ver descrição abaixo (Times New Roman, 10pt)

referência bibliográfica	TF-REFERÊNCIA ITEM (Times New Roman, 10pt, alinhada à margem esquerda)
fonte, legenda, texto de quadro/tabela e figura	TF-FONTE (Times New Roman, 9pt, centralizada) TF-LEGENDA, (Times New Roman, 10pt, centralizada) TF-TEXTO- QUADRO (Times New Roman, 10pt) TF-FIGURA (Times New Roman, 10pt, centralizada)

Fonte: elaborado pelo autor.

O espaçamento, também definido no modelo, deve ser conforme indicado no Quadro 15.

Quadro 15 - Espaçamento

USO	ESPAÇAMENTO
título de capítulo ou seção primária (1) título de seção secundária (1.1) título de seção terciária (1.1.1) título de seção quaternária (1.1.1.1) título da seção quinária (1.1.1.1.1)	espaço simples, com 12pt antes do parágrafo
texto	espaço simples, com 6 pt antes do parágrafo
citação direta com mais de três linhas	espaço simples com 6pt antes e depois do parágrafo
itens (alíneas)	espaço simples, com 6 pt antes do parágrafo
referência bibliográfica	espaço simples, com 6 pt antes do parágrafo
legenda e texto de ilustração/tabela	espaço simples, com 6 pt antes do parágrafo
fonte	espaço simples, com 0pt antes do parágrafo

Fonte: elaborado pelo autor.

Na disposição gráfica de itens (alíneas) devem ser observados os seguintes quesitos:

- o texto que antecede os itens termina com dois pontos;
- cada item deve iniciar com uma letra minúscula seguida de fecho parênteses e terminar com um ponto e vírgula, sendo que o último item termina com ponto (FORMATO: TF-ALÍNEA);
- o texto de cada item inicia com letra minúscula, exceto nomes próprios;
- quando contiver subitens, os mesmos devem iniciar com hífen colocado sob a primeira letra do texto do item correspondente (FORMATO: TF-SUBALÍNEA nível 1 ou TF-SUBALÍNEA nível 2, conforme o caso). Nesse caso, cada subitem deve terminar com uma vírgula, exceto o último que termina com ponto ou com ponto e vírgula.

Segue um exemplo:

- cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA);
- cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA):
 - cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 1):
 - cada subitem (nível2) inicia com letra minúscula, cada subitem (nível2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
 - cada subitem (nível2) inicia com letra minúscula, cada subitem (nível2) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 2),
 - cada subitem (nível 1) inicia com letra minúscula, cada subitem (nível 1) inicia com letra minúscula (FORMATO: TF-SUBALÍNEA nível 1);
- cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula, cada item inicia com letra minúscula (FORMATO: TF-ALÍNEA).

6.1.1.1 Exemplo de título de seção quaternária [FORMATO: TF-TÍTULO 4]

Formato: TF-TEXTO.

6.1.1.1.1 Exemplo de título de seção quinária [FORMATO: TF-TÍTULO 5]

Formato: TF-TEXTO.

6.1.2 Formatação de quadros, figuras e tabelas

Um quadro contém apenas informações textuais, que podem ser agrupadas em colunas. Uma figura contém, além das informações textuais, pelo menos um elemento gráfico. Uma tabela é uma apresentação tabular de informações **numéricas** relacionadas.

Os quadros, figuras e tabelas são identificados na parte superior por uma legenda (a qual deve estar centralizada) composta pela palavra designativa (Figura, Quadro ou Tabela, conforme o caso), seguida de seu número em algarismo arábico (usar numeração progressiva, uma sequência para os quadros, outra para as figuras e outra para as tabelas), de hífen e do título. As ilustrações devem:

- aparecer centralizadas no texto;
- estar delimitadas por uma moldura simples (com exceção das tabelas não quais não devem ser usadas as bordas (linhas) verticais em suas extremidades);
- aparecer numa única página (quando o tamanho não exceder o da página), inclusive a legenda;
- serem inseridas o mais próximo possível do trecho a que se referem pela primeira vez.

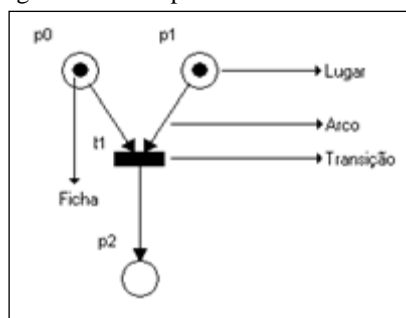
Toda ilustração deve ter fonte, centralizada. Quando foi o próprio autor que fez a ilustração, deve inserir o texto: “Fonte: elaborado pelo autor”.

Observa-se que quando um código fonte for descrito dentro de um quadro, deve-se utilizar letra do tipo `courier new 9pt.` (TF-CÓDIGO-FONTE)

Exemplos de como se deve referenciar uma figura, um quadro e uma tabela bem como descrevê-los são mostrados a seguir.

Um exemplo de uma rede de Petri pode ser visto na Figura 6.

Figura 6– Exemplo de uma rede de Petri



Fonte: Schubert (2003, p. 18).

Um exemplo de código fonte gerado a partir de uma especificação pode ser visto no Quadro 16.

Quadro 16 – Funções que verificam se as transições estão sensibilizadas

```
Function TEstruturaMalha.T1Sensibilizada: boolean;
begin
    result := (Fp2 and Fp4);
end;

function TEstruturaMalha.T2Sensibilizada: boolean;
begin
    result := (Fp1 and Fp3);
end;

function TEstruturaMalha.T3Sensibilizada: boolean;
begin
    result := (Fp2 and Fp4);
end;
```

Fonte: Schubert (2003, p. 63).

A quantidade de trabalhos finais realizados no Curso de Ciência da Computação (de 2010 até 2014) é apresentada na Tabela 1.

Tabela 1 – Trabalhos finais realizados no Curso de Ciência da Computação

Ano	Estágios	TCC's	Totais
2010/1	0	16	16
2010/2	0	21	21
2011/1	0	25	25
2011/2	0	23	23
2012/1	0	23	23
2012/2	0	22	22
2013/1	0	25	25
2013/2	0	16	16
2014/1	0	18	18
2014/2	0	13	13
	0	202	202

Fonte: elaborado pelo autor.

6.1.3 Exemplos de citações retiradas de documentos ou de nomes constituintes de uma entidade

A apresentação de citações em documentos deve seguir a NBR 10520 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b). O sistema a ser usado é o alfabético. Exemplos de citações são: “Numa publicação recente (SEBESTA, 2000) é exposto ...” e “Segundo Silva et al. (1987), execução controlada de programas é ...”.

Quando a citação referir-se a uma parte específica do documento consultado, especificar no texto do artigo a(s) página(s). Esta(s) deverá(ão) seguir a data, separada(s) por vírgula(s) e precedida(s) pelo designativo que a(s) caracteriza(m). Como exemplo, mostra-se: “(SCHIMT, 1999, p. 50)” ou “... visto que Schimt (1999, p. 50) implementou ...”.

As citações diretas (transcrição textual de parte da obra do autor consultado), no texto, com mais de três linhas, devem ser destacadas com recuo de 4 cm da margem esquerda, com letra menor que a do texto utilizado e sem aspas (FORMATO: TF-CITAÇÃO), conforme o exemplo a seguir.

A Associação Brasileira de Normas Técnicas (ABNT) é o Fórum Nacional de Normalização. As Normas Brasileiras, cujo conteúdo é de responsabilidade dos Comitês Brasileiros (ABNT/CB) e dos Organismos de Normalização Setorial (ABNT/ONS), são elaboradas por Comissões de Estudo (CE), formadas por representantes dos setores envolvidos, delas fazendo parte: produtores, consumidores e neutros (universidades, laboratórios e outros). (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2002b, p. 1).

Quando da citação de um nome (identificador) constituinte de uma entidade ou de um elemento de interface em um texto, deve-se utilizar o tipo de letra *courier new*, com tamanho nove (9). Para facilitar a formatação, existe o estilo de palavra denominado TF-COURIER9. Como exemplo cita-se nome de classe, atributo ou método. A seguir são apresentados exemplos.

As classes `TTabelaTransicao` e `TExpressaoRegular` são classes de interface, porém estão sendo consideradas como classes de domínio da aplicação.

Ao clicar no botão `Confirmar`, o software abre uma nova tela.