

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

GUSTAVO ROCHA SPENA

Simulador de diagnósticos e tratamentos em pacientes para estudo e prática para  
estudantes na área da saúde

São Carlos - SP

2019

GUSTAVO ROCHA SPENA

Simulador de diagnósticos e tratamentos em pacientes para estudo e prática para  
estudantes na área da saúde

Monografia referente ao projeto de conclusão  
de curso dentro do escopo da disciplina  
Projeto de Formatura I (SSC0670) do  
Departamento de Sistemas de Computação do  
Instituto de Ciências Matemáticas e de  
Computação - ICMC - USP para obtenção do  
título de Engenheiro de Computação.

Área de concentração: Engenharia de  
software, computação gráfica.

Orientador: Eduardo do Valle Simões

São Carlos - SP

2019

# Dedicatória

Dedico este trabalho à minha família, principalmente aos meus pais Carlos Wanderley Spena e Salete Aparecida da Rocha Spena, por terem me dado todo o apoio necessário para chegar até aqui.

# Agradecimentos

Gostaria de agradecer a todos aqueles que conheci e que me acompanharam nesses últimos anos durante a graduação, em momentos bons e ruins, fazendo parte do meu crescimento e evolução.

Agradeço especialmente a pessoa que conheci a pouco tempo mas que se tornou muito importante para mim, e que desde então sempre esteve comigo e sempre com comidas gostosas. Mayara de Lima Prates, obrigado por me ajudar e estar sempre presente mesmo nos momentos mais difíceis que passei nesses últimos anos, sua presença foi essencial para a conclusão desta etapa em minha vida.

Agradeço também ao meu orientador Eduardo do Valle Simões pelo conhecimento e pela motivação passados para mim durante esse projeto.

*“The night is in its darkest just before dawn. But keep your eyes open. If you avert your eyes from the dark, you’ll be blinded by the rays of a new day. So keep your eyes open, no matter how dark the night ahead may be.”*

*Gintoki Sakata*

# Resumo

O projeto retratado neste Trabalho de Conclusão de Curso consiste em um simulador para o aprendizado e prática voltada a estudantes de medicina. O sistema é *open-source* e formado por um ambiente interativo para introduzir conceitos e técnicas. O ambiente tem o intuito de simular um consultório médico onde o usuário tem a liberdade de explorar e aprender utilizando as ferramentas disponíveis. Não apenas isso, o sistema também deverá ser incorporado a outros projetos com a finalidade de construir um simulador de baixo custo composto de módulos de software e hardware.

# Sumário

<b>1. Introdução</b>	<b>9</b>
<b>1.1. Motivação</b>	<b>9</b>
<b>1.2. Objetivos</b>	<b>9</b>
<b>1.3. Organização do relatório</b>	<b>10</b>
<b>2. Revisão bibliográfica</b>	<b>11</b>
<b>2.1. Tecnologias</b>	<b>11</b>
<b>2.1.2. Unity</b>	<b>11</b>
<b>2.2. Simuladores de paciente</b>	<b>12</b>
<b>2.3. Gamificação</b>	<b>15</b>
<b>3. Desenvolvimento</b>	<b>17</b>
<b>3.1. Metodologia</b>	<b>17</b>
<b>3.2. Implementação</b>	<b>17</b>
<b>3.2.1. Patient Simulator</b>	<b>17</b>
<b>3.2.2. Esquemáticos</b>	<b>18</b>
<b>3.2.3. Implementação em Java</b>	<b>19</b>
<b>3.2.4. Implementação em Unity</b>	<b>23</b>
<b>3.3. Comunicação</b>	<b>28</b>
<b>4. Conclusão</b>	<b>29</b>
<b>4.1. Resultados</b>	<b>29</b>
<b>4.2. Dificuldades e limitações</b>	<b>30</b>
<b>4.3. Trabalhos futuros</b>	<b>31</b>
<b>4.4. Relacionamento entre o curso e o projeto</b>	<b>31</b>
<b>4.5. Considerações sobre o curso de graduação</b>	<b>32</b>
<b>5. Referências</b>	<b>34</b>

# Lista de códigos

Código 1: Classe do esfigmomanômetro.	22
Código 2: Script que cria o painel de medicamentos.	26
Código 3: Script responsável pelo funcionamento do esfigmomanômetro.	28



# Lista de figuras

Figura 1: Utilização de manequim em aula	10
Figura 2: Simulador de baixa fidelidade	13
Figura 3: Simulador de média fidelidade	13
Figura 4: Simulador de alta fidelidade	14
Figura 5: Exemplo de simulador de paciente existente no mercado	14
Figura 6: Contextualização da gamificação	16
Figura 7: Esquemático da tela principal	18
Figura 8: Esquemático da tela com visão interna do boneco.	19
Figura 9: Tela principal do primeiro protótipo em Java.	20
Figura 10: Tela utilizando a ferramenta esfigmomanômetro.	21
Figura 11: Tela inicial do programa em Unity.	23
Figura 12: Câmera frontal do boneco paciente.	24
Figura 13: Interior ao boneco são colocados órgãos e pontos de referência.	25
Figura 14: Janela de aplicação de medicamentos.	25
Figura 15: Utilizando o esfigmomanômetro.	27
Figura 16: Raspberry pi e componentes de comunicação wifi.	29

# 1. Introdução

## 1.1. Motivação

O cenário atual do ensino na área de medicina se encontra evoluindo constantemente, consequentemente, o aparecimento de novas ferramentas, tecnologias e métodos de ensino podem facilitar o acompanhamento e garantir a qualidade do ensino nas instituições<sup>[1]</sup>. No processo de aprendizagem dos alunos é importante possibilitar o contato dos mesmos com situações reais, para que assim, seja possível prepará-los para exercer sua profissão de forma competente e correta. Além disso é necessário construir ambientes em situações controladas para que não haja riscos, tanto para possíveis pacientes quanto para os próprios estudantes, não só isso, mas a necessidade de expor os alunos a pacientes pode causar situações incômodas para estes, que procuram hospitais-escola quando há necessidade<sup>[2]</sup>. Somado a isso, o alto custo de ferramentas de última geração é um dos maiores problemas na aquisição destas ferramentas, que simulam ambientes hospitalares de maneira realista voltadas especificamente para o ensino<sup>[3]</sup>.

Tendo como principal motivação apresentar os alunos ao ambiente hospitalar e habituá-lo com os equipamentos básicos, surgiu a ideia desse projeto. Atendendo assim a demanda de ferramentas realistas e de menor custo pelas instituições de ensino da área de medicina.

## 1.2. Objetivos

Esse projeto tem como objetivo principal implementar um sistema interativo que permite ao usuário explorar e aprender sobre ferramentas de uso na medicina, porém não somente como um software de ensino mas que ofereça experiências que motivam o usuário a interagir, explorar e aprender de maneira eficiente e fluida utilizando elementos de gamificação. Com isso se espera contribuir com o atual cenário de ensino, ao fornecer ferramentas que facilitam o aprendizado.



Figura 1: Utilização de manequim em aula.

Em geral, os simuladores utilizados atualmente, como ilustrado na figura acima, consistem de manequins que são produzidos para imitar a anatomia humana e interagir com o usuário por meio de sons e sinais monitorados, podendo ainda possuir aspectos para treinamentos específicos. Por esse motivo surgiu a ideia de criar uma aplicação que possa não apenas ser usado em conjunto com um manequim mas que ao mesmo tempo possa oferecer experiências válidas de maneira independente. Um outro objetivo é o de criar um sistema que seja de baixo custo, visto que grande parte dos simuladores disponíveis no mercado tem custo muito elevado<sup>[4][5]</sup>, oferecendo assim mais acessibilidade para instituições com recursos limitados. A ideia é desenvolver uma ferramenta com funcionalidades de baixa complexidade mas acessíveis ao mercado.

O sistema ainda deverá ser capaz de se comunicar com um equipamento de hardware e de ser incorporado a outros projetos, que estão sendo e ainda serão desenvolvidos paralelamente a esse, que reproduza as condições do paciente ao mesmo tempo que é possível acompanhar e alterar suas propriedades através do sistema e vice-versa.

### **1.3. Organização do relatório**

Este relatório está estruturado da seguinte forma:

O capítulo 2 explica em detalhes os assuntos que foram necessários a serem estudados e familiarização para o desenvolvimento do projeto. O capítulo 3 descreve e exemplifica todos os módulos do sistema, desde o planejamento inicial até sua implementação e o

resultado obtido. O capítulo 4 aborda as dificuldades encontradas no desenvolvimento, os objetivos atingidos, trabalhos futuros e considerações gerais sobre todo o projeto.

## 2. Revisão bibliográfica

### 2.1. Unity

Unity 3D ou simplesmente Unity é uma IDE e um motor de jogo 3D multi-plataforma desenvolvido pela Unity Technologies. Lançado em Junho de 2005 exclusivamente para Mac OS X, e hoje, possui suporte para mais de 25 plataformas. O motor pode ser usado para criar jogos em 2D, 3D, Realidade Aumentada e Realidade Virtual, assim como simuladores. Ele também é utilizado pelas indústrias de cinema, construção, engenharia e arquitetura.

A Unity oferece aos usuários uma API de script em C# assim como a funcionalidade de “drag and drop”, ação de clicar em um objeto virtual e "arrastá-lo" a uma posição diferente ou sobre um outro objeto virtual. Apesar de atualmente C# ser a principal linguagem de programação em uso, originalmente também eram suportadas as linguagens Boo, até 2015, e UnityScript, uma linguagem inspirada em JavaScript, até Agosto de 2017<sup>[7]</sup>.

Algumas de suas principais características são:

- Suporte para o uso de shaders;
- Programação em C# ou JavaScript;
- Suporte ao PhysX, incluindo detector de colisão, soft body e ragdoll;
- Compatibilidade com os navegadores (via o plugin Unity Web Player): Internet Explorer, Firefox, Safari, Opera, Google Chrome (versões anteriores a 45) e Camino;
- Compatibilidade com Blender, 3ds Max, Maya, Cinema 4D, Cheetah 3D, Softimage, modo, ZBrush, Lightwave, Photoshop, Fireworks, e "Substance".

Criadores de conteúdo podem desenvolver e vender recursos para outros criadores através da Unity Asset Store. Incluindo assim modelos 3D e ambientes que os

desenvolvedores podem comprar e vender, em 2018 houveram aproximadamente 40 milhões de downloads na loja virtual.

A Unity também oferece serviços para desenvolvedores, como: Unity Ads, Unity Analytics, Unity, Unity Cloud, Unity Everyplay, Unity IAP, Unity Multiplayer, Unity Performance Reporting, Unity Collaborate e Unity Hub.

A decisão de utilizar a Unity para desenvolver o projeto veio da sua grande facilidade para criar aplicações gráficas, sendo uma plataforma intuitiva e de fácil aprendizagem, além disso possui um bom suporte e sua vasta comunidade compartilha recursos informações que auxiliam na aprendizagem e oferece mais materiais para construir o projeto da maneira ideal. Outro fator importante é a compatibilidade com diversas plataformas o que permitiria portar o programa para diferentes maneiras de uso, como tablets, computadores pessoais e realidade virtual.

## **2.2. Simuladores de Paciente**

A expansão acelerada da tecnologia nos permite pensar em novos meios de aprimorar ou criar técnicas que ajudem a solucionar problemas da sociedade, inevitavelmente novas técnicas para o ensino são criadas na tentativa de tornar-lo mais completo e eficaz. Ademais na área da saúde a evolução também ocorre de forma acelerada, o que causa grande dificuldade para formar e manter os profissionais constantemente atualizados. Por esse motivo há a necessidade de criar ambientes que estimulem a aprendizagem de forma fluida e eficiente. À vista disso, tenta-se criar novas metodologias e tecnologias educacionais<sup>[9]</sup>, ainda junto a isso existe uma grande preocupação dos educadores para que os fundamentos teóricos sejam próximos das experiências reais vividas em exercício da função<sup>[10]</sup>.

Mesmo com extensos períodos de estudo e práticas, existe grande dificuldade para fazer com que os alunos apliquem todo o seu conhecimento no exercício médico<sup>[11]</sup>. Nesta situação a utilização de aplicações multimídia pode ser uma boa opção para resolver a falta de experiência com pacientes reais através da simulação. E ainda existe vários indícios mostrando que alunos apreciam as possibilidades criadas pela simulação, oferecendo a oportunidade de praticar os conhecimentos em um ambiente seguro<sup>[12]</sup>.

Simulações são classificadas quanto a sua fidelidade e realismo, onde então podem ser determinadas as atividades de cada uma. As figuras a seguir(figura 3, figura 4, figura 5) foram tiradas do curso da faculdade de enfermagem da Universidade de São Paulo em Ribeirão Preto e exemplificam os tipos de complexidade.



Figura 2: Simulador de baixa fidelidade.



Figura 3: Simulador de média fidelidade.



Figura 4: Simulador de alta fidelidade.

- Simulação de Baixa Fidelidade/Complexidade: indicado para resolução de cenários e/ou ao desenvolvimento de competências específicas<sup>[4]</sup>;
- Simulação de Média Fidelidade/Complexidade: indicado para cenários pouco complexos, que envolvem algum tipo de interação entre o aluno e o simulado<sup>[4]</sup>r, e;
- Simulação de Alta Fidelidade/Complexidade: orientado a resolução de cenários completos e complexos, onde geralmente são necessários recursos computacionais, sistemas de som e imagem, com maior grau de fidelidade ao cenário e dos simuladores<sup>[4]</sup>.

A ilustração abaixo mostra como é composto geralmente os simuladores de pacientes do mercado, com o boneco de hardware e um software em conjunto.



Figura 5: Exemplo de simulador de paciente existente no mercado<sup>[21]</sup>.

Geralmente a fidelidade da simulação corresponde à capacidade do simulador em “imitar” o ser humano na aparência e reações, permitindo ao aluno apurar suas habilidades profissionais. Com isso em mente é adequado afirmar que a projeto visa abordar o

desenvolvimento de simuladores ainda buscando manter o custo baixo, para ser utilizado em situações onde simulações de baixa fidelidade são suficientes.

## **2.3. Gamificação**

A gamificação é um fenômeno observado nos últimos anos referente a popularização dos games (jogos digitais), a partir desse ponto criou-se a ideia de que talvez seja possível atribuir algumas de suas propriedades aos métodos de ensino utilizados em várias idades e áreas<sup>[13][14]</sup>. Propriedades advindas dos jogos eletrônicos, como por exemplo, as suas capacidades de ambientação, incentivo a resolver problemas, motivar a exploração e estimular a criatividade, isso junto ao propósito de desenvolver métodos de ensino mais eficazes e interessantes que podem ser passados aos estudantes das mais diversas áreas e idades. Essa possibilidade já havia sido cogitada há décadas atrás, porém até então a indústria dos jogos não havia se consolidado como atualmente, quando essa forma de entretenimento faz parte da vida de pessoas em todos os lugares do mundo<sup>[15]</sup>.

Apesar de já estar presente há alguns anos, o aumento de sua relevância nos últimos tempos proporcionou mais estudos e melhorias das teorias dessa área. Originalmente utilizado na área de marketing com o objetivo de atrair e fidelizar clientes, a gamificação leva os elementos encontrados nos jogos como interatividade, competição, objetivos e regras claras, sistema de feedback e recompensas narrativas, tentativa e erro e diversão dentre outros, aos usuários, com a finalidade de obter o mesmo nível de comprometimento e dedicação que são atribuídos a bons jogos pelos jogadores<sup>[16]</sup>.



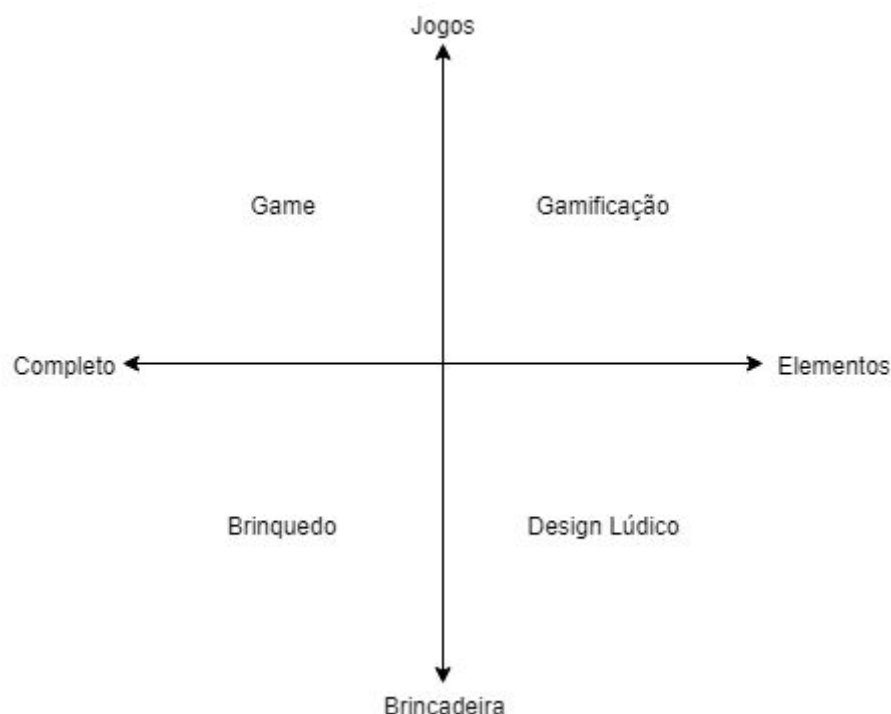


Figura 6: Contextualização da gamificação<sup>[17]</sup>.

Por se tratar de uma derivação dos jogos eletrônicos, a gamificação tem a propriedade de ser composta de diversos elementos interconectados que são combinados para formar um resultado maior que a soma de suas partes, portanto é possível definir a quantidade de elementos de modo que o sistema, em sua completude, proporcione uma experiência próxima a de um game completo. Um exemplo dos elementos mencionado é a utilização de um sistema PBL, que corresponde a pontos, medalhas e sistemas de classificação (Points, Badges and Leaderboards), essa mecânica tem a capacidade de motivar a mudança por fatores externos de forma semelhante às ideias da economia comportamental<sup>[18]</sup>. Outro método envolve o apelo a experiências mais significativas que motivam os indivíduos a exercerem sua função da melhor forma possível<sup>[19]</sup>.

Pelos motivos explicados acima, o objetivo é abordar um determinado problema de modo parecido com o de um game designer, que consegue pensar em formas de engajar o jogador para resolver problemas em mundos virtuais<sup>[20]</sup>, isso por que a gamificação não se resume em criar jogos sobre determinados assuntos mas sim utilizar estratégias e abordagens similares para resolver aqueles problemas nos mundos virtuais em situações do mundo real.

## 3. Desenvolvimento

### 3.1. Metodologia

O planejamento do projeto começou com um levantamento dos pré-requisitos do sistema, pensando primeiramente nas funções de maior prioridade e na necessidade de possuir um protótipo funcional para que os outros elementos pudessem ser incorporados.

Na etapa seguinte foram escolhidas as tecnologias a serem utilizadas, considerando que até então não havia uma plataforma específica padrão a ser utilizada. Inicialmente foi concordado que a linguagem em que a aplicação seria desenvolvida seria JAVA, porém, durante a produção do sistema, foi sugerido outra plataforma que poderia produzir os resultados almejados com mais eficiência e praticidade, por se tratar de um sistema desenvolvido puramente para este tipo de aplicação, a Unity, o que apesar de acabar criando outras dificuldades, em geral possibilitou elevar o nível do projeto sendo desenvolvido e facilitar a agregação de outros projetos.

Por utilizar métodos ágeis, os pré-requisitos podem ser alterados e serem necessárias adaptações, conforme o desenvolvimento.

A terceira etapa foi a etapa de implementação, a cada nova iteração do processo do desenvolvimento contemplava as seguintes etapas:

- Desenvolver lógica;
- Desenvolver testes unitários;
- Escrever documentação;
- Incorporar no sistema principal.

### 3.2. Implementação

A seguinte seção explicará mais detalhes técnicos sobre a implementação do sistema.

#### 3.2.1. Patient Simulator

O sistema foi nomeado de **Patient Simulator** (Simulador de Paciente). Antes da implementação inicial foram feitas algumas montagens de como o programa deveria se

parecer em seu estado final, porém após a decisão de migrar do Java para Unity várias mudanças ocorreram como será mostrado a seguir.

### 3.2.2. Esquemáticos

Abaixo se encontra a primeira ilustração a ser feita com o intuito de mostrar as ideias básicas e as principais funcionalidades do programa. A esquerda a barra de ferramentas com os instrumentos que poderiam ser utilizados pelo usuários enquanto que na direita seriam exibidos os sinais vitais do paciente em tempo real mostrando assim o estado do paciente e os resultados obtidos ao administrar os medicamento. A barra superior permitiria o usuario mudar a visão do paciente para cabeça, costas e membros além de mostrar a ficha do paciente a fim de apresentar o histórico e notas sobre o paciente.

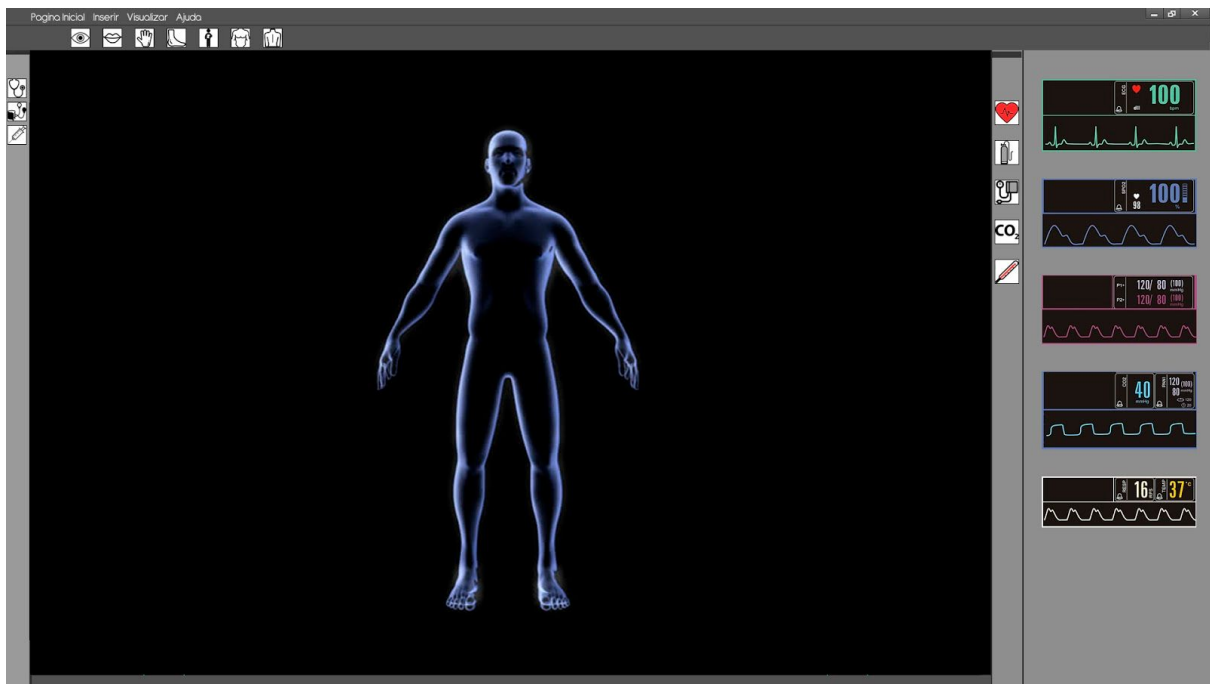


Figura 7: Esquemático da tela principal.

Esta outra imagem mostra uma visão interna do boneco onde poderiam ser vistos os ossos, vasos sanguíneos e órgãos, a fim de mostrar problemas internos.

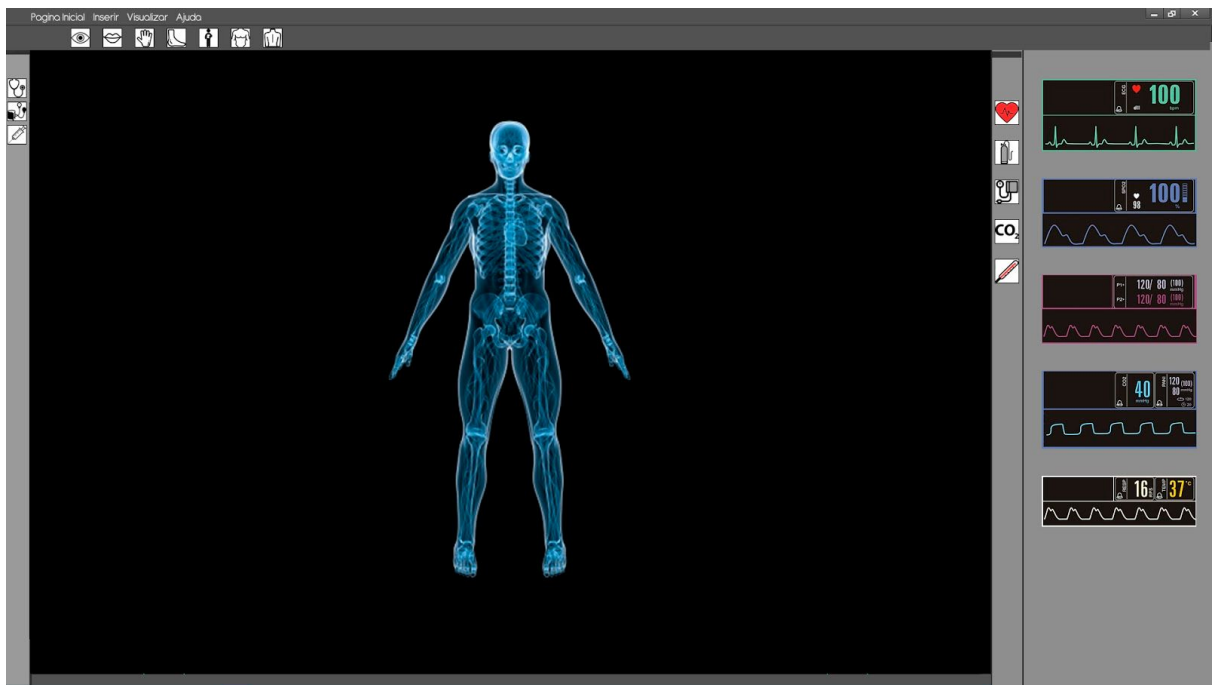


Figura 8: Esquemático da tela com visão interna do boneco.

### 3.2.3. Implementação em Java

Definidos os primeiros passos foi possível orientar mais facilmente quais seriam as próximas etapas e decidir as primeiras funções a serem desenvolvidas, é possível observar por exemplo, que a visão geral do programa bem como as disposições das ferramentas são semelhantes as do esquemático.

A figura acima é a primeira versão a ser construída, esta versão possui as primeiras funcionalidades programadas como o estetoscópio e o esfigmomanômetro, porém como se pode observar não há um ambiente 3D. Esta versão permitiu desenvolver e criar as ferramentas e lógicas que seriam usadas na versão unity. Nela há somente a imagem do “corpo” do boneco que seria usado como paciente e ao lado a barra de ferramentas.

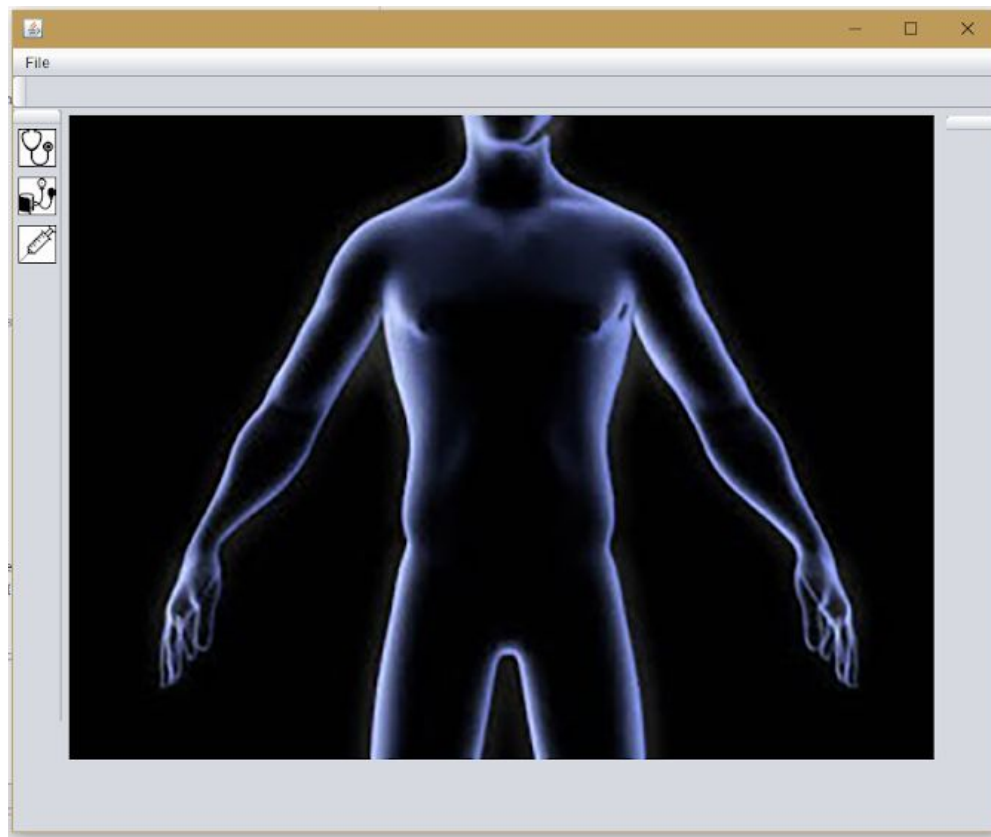


Figura 9: Tela principal do primeiro protótipo em Java.

Nesta versão já havia sido implementado as funções de estetoscópio e esfigmomanômetro ainda que de maneira rudimentar mas funcionais, porém durante o desenvolvimento notou-se que para alcançar os resultados almejados seria necessário utilizar ferramentas mais potentes que permitissem criar uma sistema mais atual, preciso e em pouco tempo, que pudesse ser portado para outras plataformas mais facilmente, para objetivos futuros.

Abaixo se encontra a primeira versão do esfigmomanômetro com a classe correspondente.

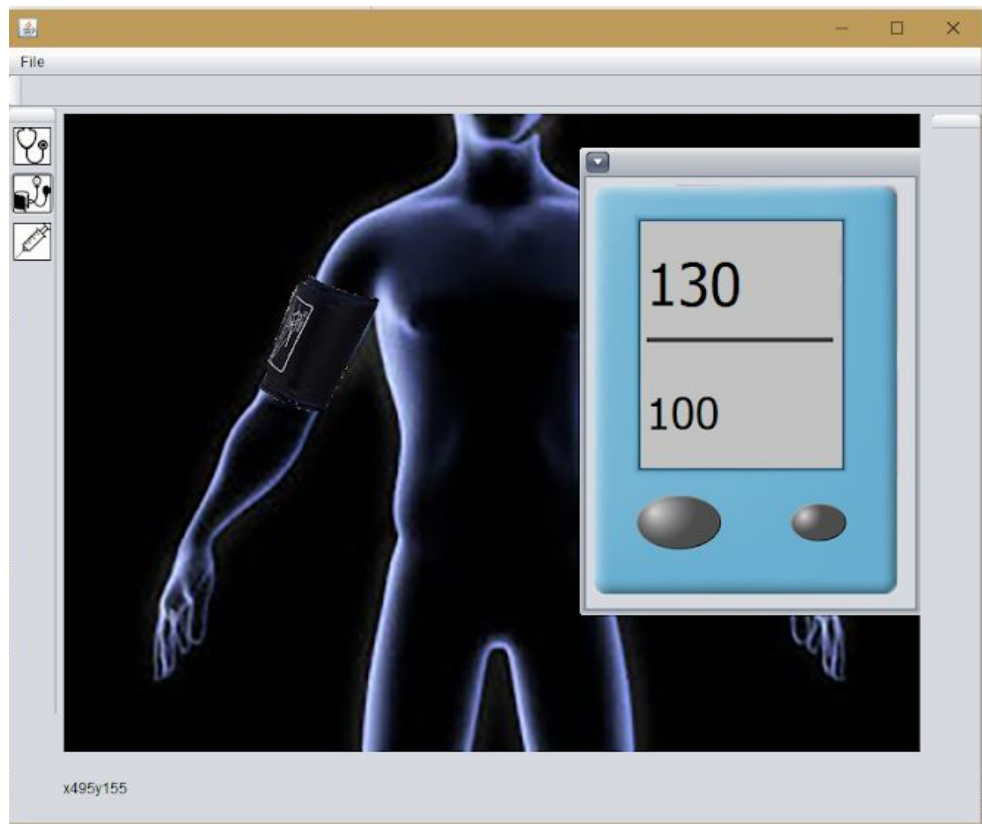


Figura 10: Tela utilizando a ferramenta esfigmomanômetro.

```
public class Telamp extends javax.swing.JInternalFrame {

    Sound Batimento = new Sound(1);
    int Sys =0;
    int Dias = 0;
    int pressaom = 0;
    public Telamp() {
        initComponents();
    }
    void closeTela(){//reseta os parametros ao fechar a tela
        Sys =0;
        Dias = 0;
        pressaom = 0;
        Batimento.closesound();
        String message = ""+ Sys ;
        jLabel5.setText(message);
        String message1 = "" + Dias;
        jLabel6.setText(message1);
    }

    //verifica se a pressao atual esta dentro dos parametros de
    diastolica e sistolica
    private void medindopressao(int pressao){
        if (pressao > 60 && pressao < 120){
            Batimento.closesound();
            Batimento.playSound();
        }else{//se nao esta dentro do limite o som para
            Batimento.closesound();
        }
    }
}
```

```

    }
}
//Diminui a atributo pressao a cada vez que for chamado
public void decresaeP(){
    Dias -=10;
    medindopressao(Dias);
    String message = "" + Dias;
    jLabel6.setText(message);
}
//Aumenta o atributo pressao a cada vez que for chamado
public void increaseP(){

    Dias +=30;
    medindopressao(Dias);
    if (Dias > Sys){
        Sys = Dias;
    }
    String message = ""+ Sys ;
    jLabel5.setText(message);
    String message1 = "" + Dias;
    jLabel6.setText(message1);
}
.
.
.

```

Código 1: Classe do esfigmomanômetro.

Com a evolução do projeto foi notado que a linguagem Java poderia não ser o método ideal para continuar, dado a velocidade a qual se dava a sua construção. A intenção de aumentar a complexidade do programa tornou evidente que o tempo necessário para aprender e desenvolver os elementos desejados iria aumentar demasiadamente , pensou-se então que adotando ferramentas criadas especificamente com o objetivo de construir jogos e aplicações interativas, seria possível elevar o nível do resultado final assim como aumentar a velocidade de construção. Mesmo que o tempo e trabalho que foram dedicados até então seja considerável, a troca de plataforma poderia, de médio a longo prazo, apresentar vantagens que sobrepujam as desvantagens de se começar o projeto novamente. Com essa constatação, procurou-se uma plataforma atual a qual pudesse ser aprendida em pouco tempo e que pudesse oferecer meios para se obter a qualidade desejada.

### 3.2.4. Implementação em Unity

Ao se decidir a mudança de plataforma e linguagem, foi escolhida uma que pudesse oferecer as ferramentas necessárias para a conclusão do projeto e que também não demandasse uma quantidade de tempo maior que o disponível para aprender e pôr em prática, a fim de concluir o projeto em tempo hábil e ainda apresentando boa qualidade. Por esse motivo o motor Unity se encaixou adequadamente, sua maneira intuitiva de programação e a grande disponibilidade de conteúdo fornecido pela própria comunidade, foi de grande ajuda para o desenvolvimento do programa, além do fato de que a linguagem utilizada, C#, é semelhante a outra com que já havia familiarização, C++.

Da versão anterior ainda foi possível transportar as mecânicas e funcionamentos desenvolvidos, adaptando-as para a nova linguagem. A produção desde o modelo conceitual serviu como base desde a primeira versão para direcionar o modelo do espaço e disposição das ferramentas.

Em nova plataforma, a imagem abaixo mostra a cama com o paciente e a mesa de instrumentos assim como a barra de ferramentas. A princípio o ambiente contém somente o boneco paciente que será utilizado e a mesa de instrumentos com as ferramentas que já estão implementadas. Nesse ambiente usuário pode se movimentar livremente pela sala e agir sobre o paciente e as ferramentas, ainda há a possibilidade de utilizar a barra de ferramentas à esquerda no lugar dos itens da mesa.

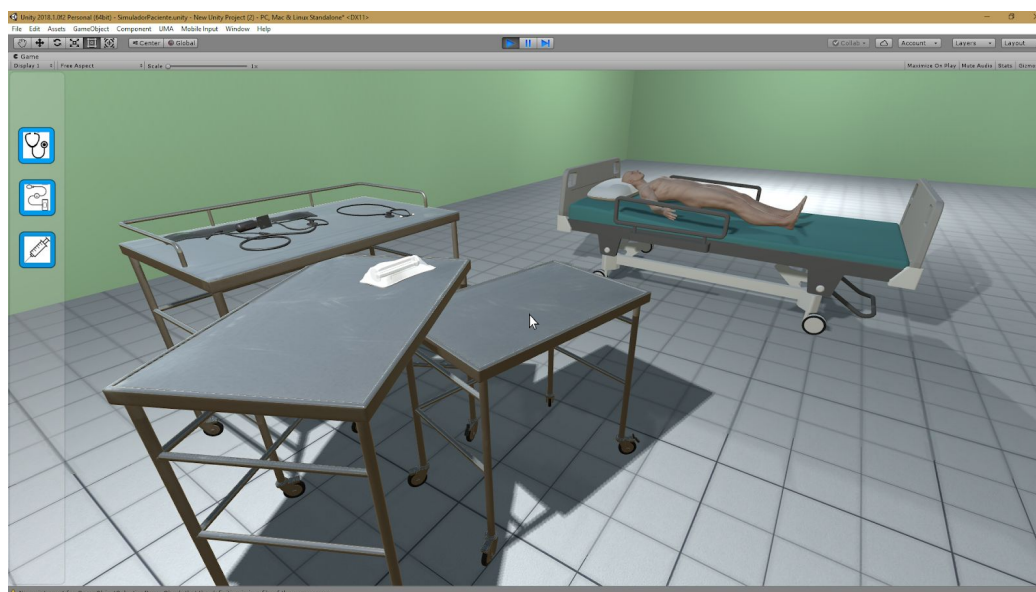


Figura 11: Tela inicial do programa em Unity.



Foram implementadas outras câmeras para facilitar o manuseio das ferramentas, desse modo é possível observar a situação por diversos pontos e escolher os instrumentos pela barra de ferramentas.

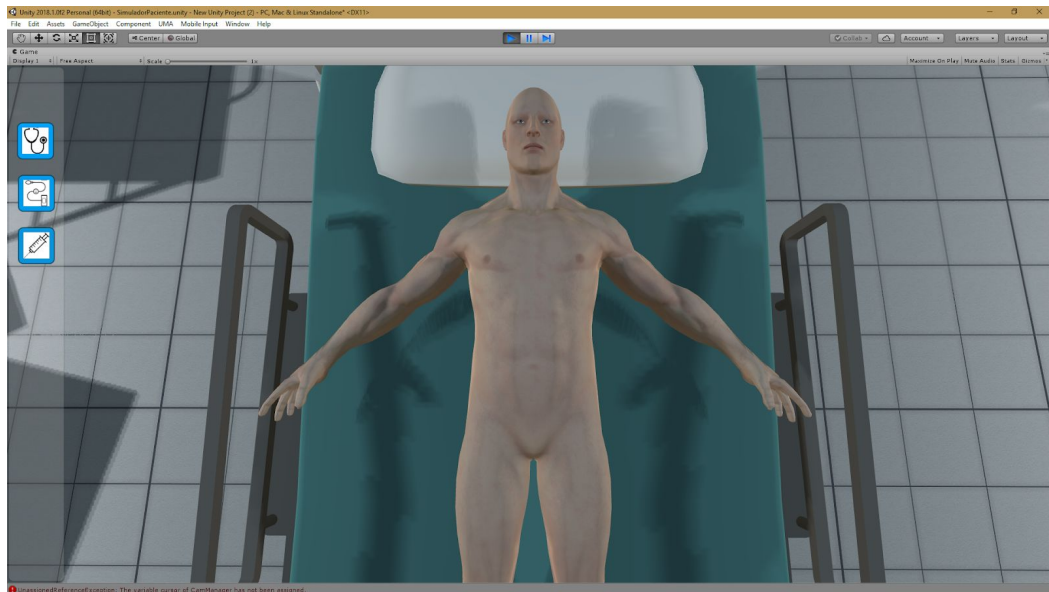


Figura 12: Câmera frontal do boneco paciente.

Internamente o boneco possui órgãos que podem ser usados como referências para localização dos pontos de escuta (esferas brancas). As esferas simulam os diferentes pontos de escuta num corpo humano, no coração as esferas estão posicionadas de maneira correspondente às válvulas cardíacas enquanto que nos pulmões são verificados diferentes pontos para averiguação de seu estado.

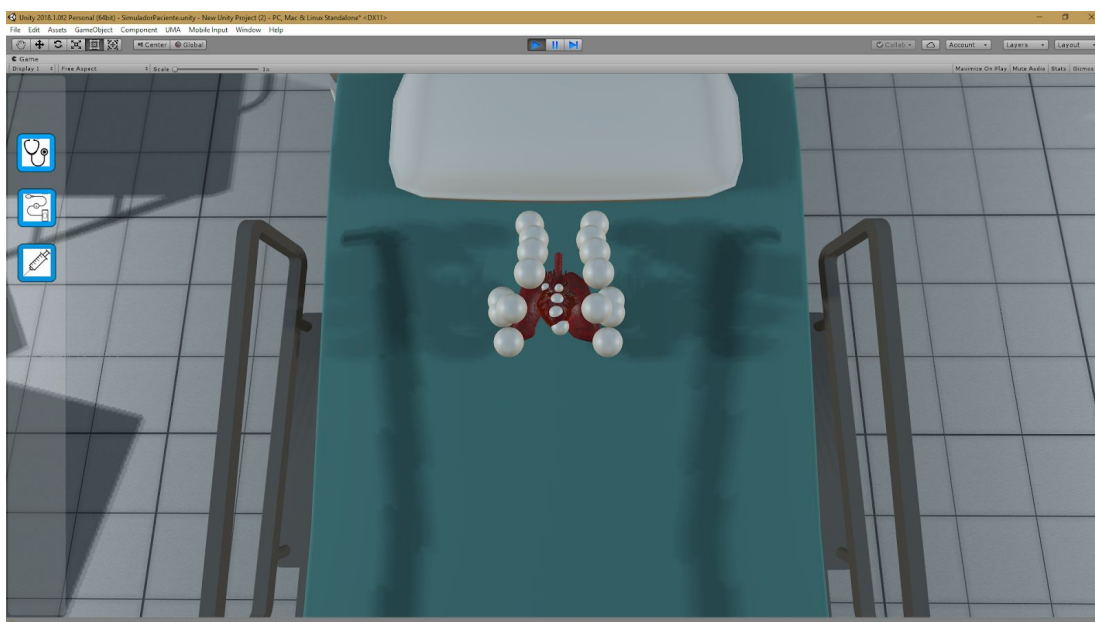


Figura 13: Interior ao boneco são colocados órgãos e pontos de referência.

A ferramenta seringa, permite simular a administração de medicamentos para então acompanhar a reação do paciente, é possível escolher a quantidade de medicamento a ser aplicado e então ao confirmar a ação no botão aplicar o usuário poderá escolher a região a ser administrado o medicamento.

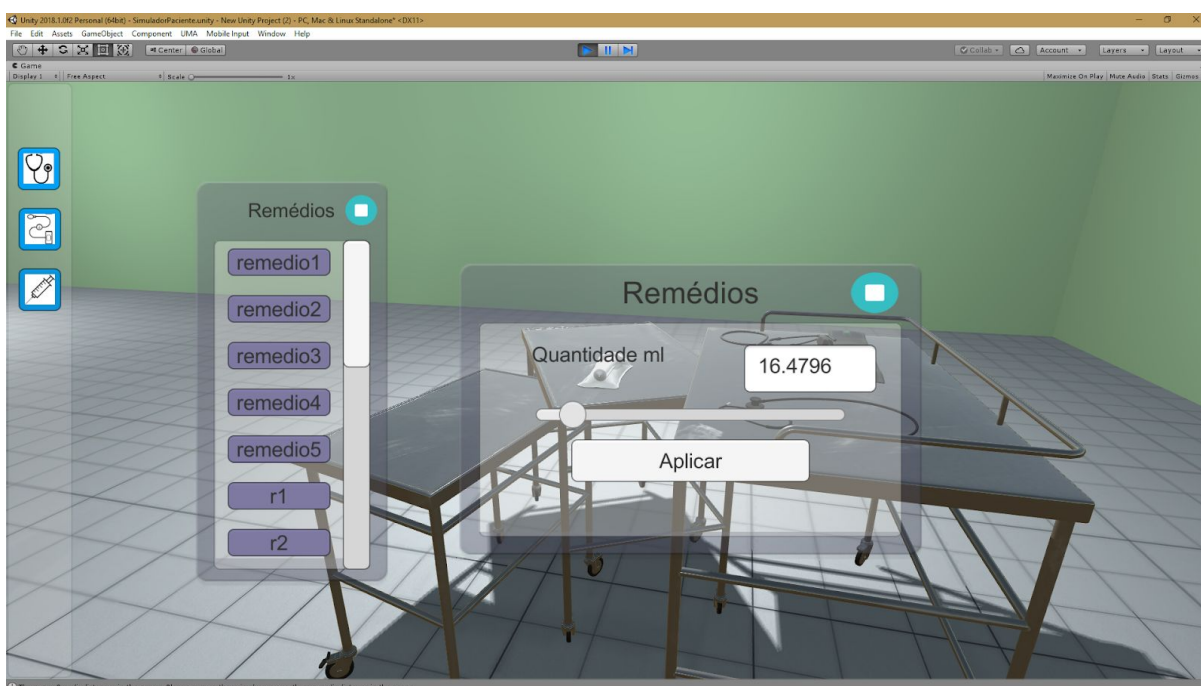


Figura 14: Janela de aplicação de medicamentos.

```
public class PannelRemédios : MonoBehaviour {
```

```

    CSVReader reader;
    public Transform botacontainer, canvas;
    public GameObject panel, Pacienteparametros, botaoprefab, RemedioEspec;

    IEnumerator Start()
    {
        reader = GetComponent<CSVReader>();
        yield return new WaitForEndOfFrame();

        foreach (Remedio r in reader.remedios)
        {
            GameObject go = Instantiate(botaoprefab) as GameObject;
            go.transform.SetParent(botacontainer.transform, false);

            GameObject botoa1 = go.transform.GetChild(0).gameObject;
            GameObject botoa2 = go.transform.GetChild(1).gameObject;

            botoa1.GetComponentInChildren<Text>().text = r.nome;
            botoa2.GetComponentInChildren<Text>().text = r.nome;

            botoa1.GetComponent<Button>().onClick.AddListener(() =>
onButtonClick(r));
            botoa2.GetComponent<Button>().onClick.AddListener(() =>
onButtonClickDestroy());
        }

        public void onButtonClick(Remedio r)
        {
            Destroy(panel);
            Pacienteparametros.SetActive(true);
            Pacienteparametros.GetComponent<CanvasGroup>().alpha = 0f; //this
makes everything transparent
            Pacienteparametros.GetComponent<CanvasGroup>().blocksRaycasts =
false; //this prevents the UI element to receive input events
            panel = Instantiate(RemedioEspec) as GameObject;
            panel.transform.SetParent(canvas.transform, false);
            panel.SetActive(true);
            panel.GetComponent<RemedioSpecs>().remedio = r;
        }
        public void onButtonClickDestroy()
        {
            Destroy(panel);
        }
    }
}

```

Código 2: Script que cria o painel de medicamentos.

A imagem e código seguintes mostram a ferramenta esfigmomanômetro em uso. Para usá-lo o usuário deve selecionar a ferramenta na mesa ou na barra de ferramentas, então ele deve clicar com o botão esquerdo do mouse na braçadeira que aparece no boneco para simular

o aumento de pressão, ao estar dentro de certos limite que correspondem a pressão atual do paciente é possível escutar o som do fluxo sanguíneo e ao ultrapassar o valor máximo o som cessa, sinalizando que a pressão sistólica foi encontrada, para encontrar a pressão diastólica o usuário deve segurar o botão direito do mouse para “esvaziar” o aparelho até que se ache a pressão mínima quando o som cessa novamente.



Figura 15: Utilizando o esfigmomanômetro.

```
public class Bracadeira : MonoBehaviour {
    public Behaviour sound;
    public GameObject pacientep m, Esfigm1, Tela;
    public int flag, current, isys = 0, idia = 0, min, max;
    bool click;
    float lastStep, timeBetweenSteps = 0.5f, temps;

    // Use this for initialization
    void decreasep()
    {
        temps = Time.time;
        click = true;
        if (Time.time - lastStep > timeBetweenSteps)
        {
            current = current - 1;
            lastStep = Time.time;
        }
    }
    void increasep()
    {
        temps = Time.time;
        click = true;
        if (Time.time - lastStep > timeBetweenSteps)
```

```

        {
            current = current + 1;
            lastStep = Time.time;
        }
    }
    void Start()
    {
        max = pacientep.GetComponent<PacienteParametros>().pressaosys;
        min = pacientep.GetComponent<PacienteParametros>().pressaodias;
        sound = (Behaviour)gameObject.GetComponent("AudioSource");
        flag = Tela.GetComponent<Tela>().min;
        gameObject.SetActive(false);
        sound.enabled = false;
    }
    // Update is called once per frame
    void Update()
    {
        max = pacientep.GetComponent<PacienteParametros>().pressaosys;
        min = pacientep.GetComponent<PacienteParametros>().pressaodias;
        if (current <= max && current >= min)
        {
            sound.enabled = true;
        }
        else
            sound.enabled = false;
    }
    .
    .
    .

```

Código 3: Script responsável pelo funcionamento do esfigmomanômetro.

Apesar das vantagens e das ferramentas fornecidas pela IDE, ela ainda possui desvantagens que precisam ser consideradas durante a construção de um projeto. Um dos fatores mais relevantes encontrados é a necessidade de um computador consideravelmente potente para que o sistema funcione, podendo ser este um fator crítico no caso do desenvolvedor não possuir um sistema compatível. Já os requisitos do sistema desenvolvido podem variar bastante dependendo de cada programa.

### 3.3. Comunicação

O sistema ainda foi capaz de se comunicar com um raspberry pi<sup>[xx]</sup> que corresponde a um manequim mecanizado que será responsável por simular mecanicamente o papel de paciente, sendo assim possível definir atributos no software que corresponderá a reações em tempo real do hardware. A comunicação ocorre via TCP/IP se conectando através da rede local em que o servidor se encontra.

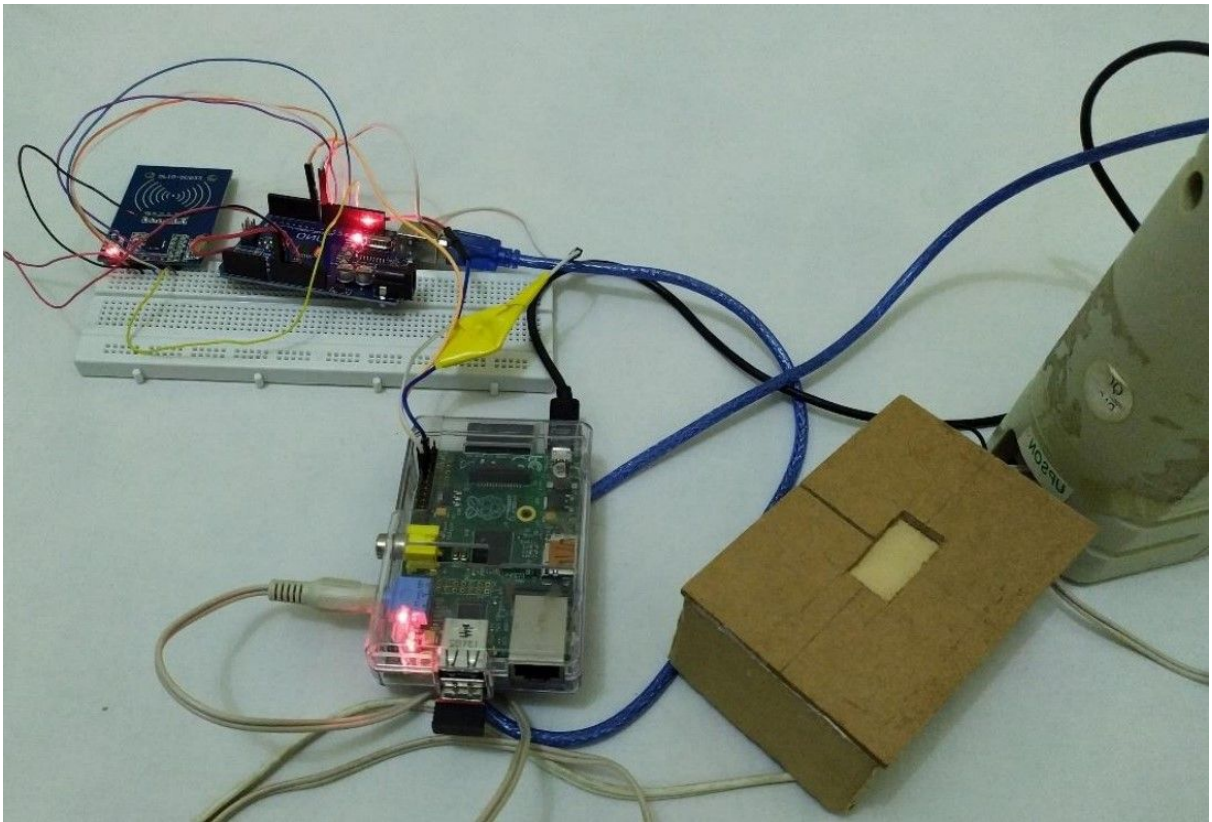


Figura 16: Raspberry pi e componentes de comunicação wifi.

A implementação da parte de comunicação se deu sem muitos problemas, apesar dos testes que foram realizados serem compostos de mensagens simples, isso garante a passagem de mensagem e resposta entre o software e hardware independentes. Viabilizando futuramente a incorporação entre as duas tecnologias.

## 4. Conclusão

### 4.1. Resultados

O objetivo inicial do projeto era criar um protótipo de simulador de paciente que servirá como base para o desenvolvimento de outros componentes com o objetivo de ser utilizado por alunos e professores para auxiliar o ensino na área da saúde. No estado atual o

usuário pode aprender noções de instrumentação com os dos módulos implementados, tais como:

- Ausculta pulmonar;
- Ausculta cardíaca;
- Medição da pressão arterial;
- Reação dos medicamentos nos sinais vitais.

O aluno ou instrutor também pode editar os medicamentos cadastrados facilmente, através de uma planilha evitando a necessidade de um programador para dar suporte.

. Secundariamente o projeto deveria conseguir se comunicar com outros dispositivos de hardware que serão parte do projeto futuramente, o que também foi atingido com êxito. Ademais, o conhecimento e experiência adquiridos eram parte do objetivo desta disciplina, que sinto ter cumprido

Como o projeto final imaginado em sua completude poderá ser muito grande, os objetivos não atingidos envolvem justamente a adição de novas ferramentas. No intuito de entregar uma aplicação funcional, alguns aspectos acabaram ficando mais simples ou limitados do que o almejado. Como exemplo ferramenta de aplicação de medicamentos ainda pode ser aprimorada para reagir de maneira mais realista.

## **4.2. Dificuldades e limitações**

Em geral a maior dificuldade encontrada durante o desenvolvimento foi o desconhecimento da linguagem e plataforma utilizadas. Apesar de ambos serem construídos para oferecer facilidades e serem intuitivos, ainda foi necessário despendar tempo considerável no aprendizado e familiarização com as tecnologias utilizadas.

Também é necessário considerar que a proposta seja um pouco ambiciosa, criar uma interface gráfica com ferramentas funcionais em um semestre, sem o conhecido da linguagem e plataforma e ainda alterando todo o projeto inicial se mostrou um desafio interessante.

Pessoalmente eu não tinha conhecimento e nem experiência com a maior parte do que foi usado nele, então muito do tempo do desenvolvimento foi utilizado para se aprender a utilizá-las de maneira correta, eficiente e funcional.

Organização também foi um pequeno problema, se disciplinar para trabalhar frequentemente, documentar o código ao invés de só implementar imediatamente e planejar

lógicas/pesquisar tecnologias era algo que eu não estava acostumado a fazer mas foi fundamental para o desenvolvimento.

### **4.3. Trabalhos Futuros**

O projeto se mostrou promissor e ainda me encontro motivado para aprimorá-lo e fazê-lo crescer. Para que esteja adequado para ser lançado ainda será necessário muito trabalho, porém acredito que melhorar a qualidade do ensino na área da saúde é um objetivo válido e de grande peso.

Ainda há um grande espaço para melhorias e alguns dos próximos pontos que podem ser trabalhados são:

- Completar a documentação;
- Construir modelos 3D próprios e direcionado às necessidades do projeto;
- Aprimorar a resistência a falhas do sistema e respostas para o usuário;
- Aumentar o número de ferramentas simuladas;
- Adicionar a possibilidade de incluir cenários específicos que seguem padrões pré-programados e que simulam situações reais;
- Portar o sistema para outras plataformas como tablets e VR;
- Desenvolver comunicação via Bluetooth para possibilitar o uso em dispositivos móveis;
- Aplicar testes com usuários reais por questões de usabilidade.

Após a entrega deste trabalho, o repositório será aberto para colaboradores que quiserem participar e melhorar o sistema.

### **4.4. Relacionamento entre o curso e o projeto**

Tendo em mente que o objetivo da graduação é ser capaz de oferecer aos alunos uma base de conhecimento e lógica que os permitirá aprender a resolver novos desafios que ainda não os tenha ensinado, sendo capazes de aprender por conta própria a solucionar problemas de forma competente e profissional, é possível afirmar com certeza que o curso foi muito eficaz, sendo possível aprender todo o conteúdo necessário para desenvolver este projeto.



Buscar melhorar a qualidade de vida das pessoas através da tecnologia é uma atribuição condizente com o de um Engenheiro de Computação e a experiência adquirida neste projeto poderá se mostrar de grande importância em minha carreira profissional.

A organização, planejamento, gerenciamento de tempo, comunicação com o orientador e escrever este relatório também é uma excelente experiência para uma possível futura formação na área acadêmica.

#### **4.5. Considerações sobre o curso de graduação**

No geral creio que o curso de Engenharia de Computação da USP campus São Carlos é um bom curso. A grade acadêmica envolve conteúdos bem diversificados, ocasionalmente podendo até ser considerado abrangente demais, mas que oferece aos alunos a capacidade de se desenvolver em diversas áreas. O corpo docente em sua maioria se mostra preocupado com a qualidade do ensino em aprimorar o método de ensino em sala e as disciplinas são de certa forma bem distribuídos por toda duração do curso.

Entretanto a quantidade de conteúdo e carga horária excessiva são umas das reclamações mais frequentes levantadas pelos alunos, enquanto que ao mesmo tempo, algumas disciplinas ou assuntos que aparentemente não apresentam grande relevância no cenário da tecnologia atual são vistos em sala. Das disciplinas que utilizam tecnologias ultrapassadas a maior parte delas provém da parte da elétrica e que ao mesmo tempo necessitam de grande quantidade de tempo e esforço sem que haja a percepção de um sentimento de retorno prático esperado de tal custo. Custo este que vários estudantes gostariam de utilizar para outras atividades como projetos pessoais, grupos extracurriculares e iniciações científicas, que podem ter um grande impacto na formação do indivíduo não apenas do ponto profissional mas também pessoal.

Além disso, gostaria de acrescentar que durante a graduação eu tive a oportunidade de fazer parte de um programa de intercâmbio que me permitiu cursar uma instituição de ensino no exterior, com a duração de um ano, apesar de ser um curso diferente ao que curso no Brasil, sendo aquele na realidade Engenharia de Software, essa experiência foi capaz de me expor um método de ensino consideravelmente diferente ao vivido aqui, tendo um enfoque mais prático, com menos tempo gasto em sala com assuntos teóricos, mas exigindo que o aluno busque por si, parte do conteúdo, através de trabalhos e até mesmo uma disciplina em

que pequenos grupos de aluno são montados com o objetivo de criar um projeto maior seguindo o acompanhamento próximo de um professor.

A meu ver este método possui a vantagem de oferecer ao aluno uma técnica de aprendizado mais dinâmica e possibilitar a criação projetos com certa relevância que ao mesmo tempo apresenta ao aluno situações mais próximas às encontradas na vida real como também pode trazer um sentimento de realização por cumprir tarefas mais complexas e que podem ser utilizadas em situações reais nem seu futuro.

Por fim, esta avaliação tem com objetivo, retratar alguns dos problemas encontrados durante a graduação e oferecer uma alternativa à um curso que, apesar de alta qualidade, muitas vezes acaba por desmotivar e afastar uma parcela de alunos pelo desgaste emocional e físico mas que talvez pudessem ter um futuro promissor na área.

## 5. Referências

- [1] BRANDÃO, C. F.S., COLLARES, C. F., MARIN, H. F. 2014. A simulação realística como ferramenta educacional para estudantes de medicina. Educação em Ciências da Saúde.
- [2] MONDREDINHO AR, SILVA RM. Percepção dos pacientes sobre a sua participação como instrumento de aprendizado nas aulas práticas de semiologia. Arq. Catarin. Méd. 2006;35(3):35-41.
- [3] IGLESIAS AG, PAZIN-FILHO A. Emprego de simulações no ensino e na avaliação. Medicina (Ribeirão Preto) 2015;48(3):233-40.
- [4] S. LAPKIN and T. LEVETT-JONES, “A cost–utility analysis of medium vs. high-fidelity human patient simulation manikins in nursing education,” Journal of clinical nursing, vol. 20, no. 23-24, pp. 3543–3552, 2011.
- [5] M. T. NETO, P. GARCIA, and J. VIDEIRA-AMARAL, “Simulação e ensino-aprendizagem em pediatria. iia parte: Experiência de um centro de simulação de técnicas,” 2010.
- [6] Ken ARNOLD, James GOSLING, David HOLMES. 2005. The Java™ Programming Language, Fourth Edition.
- [7] JOHN K. Haas. 2014 .A History of the Unity Game Engine.
- [8] DEITEL, P. Deitel, H. 2016. Visual C# How to Program.
- [9] HIGGS, J. Clinical reasoning in the health professions. Elsevier Health Sciences, 2008
- [10] CENDAN, J.; LOK, B. The use of virtual patients in medical school curricula. Advances in physiology education, v. 36, n. 1, p. 48-53, 2012.
- [11] FORTE, M. et al. Portfólio Reflexivo Eletrônico na Unidade Educacional de Prática Profissional do Curso de Medicina da UFSCar. In: X Workshop de Informática Médica, Belo Horizonte. Anais do XXIX Congresso da Sociedade Brasileira de Computação. Porto Alegre: Sociedade Brasileira de Computação, p. 1566-75, 2010.

- [12] R. C. N. BAPTISTA, J. C. A. MARTINS, M. F. C. R. PEREIRA, and A. MAZZO, “Students’ satisfaction with simulated clinical experiences: validation of an assessment scale,” *Revista latino-americana de enfermagem*, vol. 22, no. 5, pp. 709–715, 2014.
- [13] PIRES RAMOS, V.; PEREIRA MARQUES, J. Dos jogos educativos à gamificação. *Revista de Estudios e Investigación en Psicología y Educación*, n. 01, p. 319-323, 15 dic. 2017.
- [14] WILSON RIBEIRO FERNANDES, Carlos; LUIZ PEREIRA RIBEIRO, Erick. GAMES, GAMIFICAÇÃO E O CENÁRIO EDUCACIONAL BRASILEIRO. *CIET:EnPED*, [S.l.], maio 2018. ISSN 2316-8722. Disponível em : <http://cientenped.ufscar.br/submissao/index.php/2018/article/view/344>>. Acesso em: 01 jun. 2019.
- [15] Malone, T. W. and Lepper, M. R. 1987 Making learning fun: A Taxonomy of intrinsic motivations for learning. In *Aptitude, Learning, and Instruction: Vol. 3*. Lawrence Erlbaum, Hillsdale, NJ.
- [16] ZICHERMANN, Gabe; CUNNINGHAM, Christopher. *Gamification by Design. Implementing Game Mechanics in Web and Mobile Apps*. Canada: O’Reilly Media, 2011.
- [17] DETERDING, S.; DIXON, D.; KHALED, R.; NACKE, L. From Game Design Elements to Gamefulness: Defining “Gamification”. In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (MindTrek '11)*. ACM, Nova Iorque, EUA. Disponível em: . Acesso em: 15 abr 2013.
- [18] LADLEY, Paul. *Gamification, Education and Behavioural Economics. Games-ED Innovation in Learning*, 2011.
- [19] WERBACH, Kevin; HUNTER, Dan. *For The Win: How Game Thinking Can Revolutionize Your Business*. Filadélfia, Pensilvânia: Wharton Digital Press, 2012
- [20] MCGONIGAL, Jane. *Reality Is Broken: Why Games Make Us Better and How They Can Change The World*. Nova Iorque: The Penguin Press, 2011.
- [21] CIVIAM. Página inicial disponível em: <http://civiam.com.br>

