



**MESTRADO PROFISSIONAL EM TECNOLOGIA MINIMAMENTE
INVASIVA E SIMULAÇÃO NA ÁREA DE SAÚDE**

MARCELO EMANOEL BEZERRA DINIZ

**DESENVOLVIMENTO DA VERSÃO 2.0 DO SIMULADOR VIRTUAL DE
VENTILAÇÃO MECÂNICA XLUN**

**FORTALEZA
2016**

MARCELO EMANOEL BEZERRA DINIZ

DESENVOLVIMENTO DA VERSÃO 2.0 DO SIMULADOR VIRTUAL DE VENTILAÇÃO
MECÂNICA XLUNG

Dissertação apresentada ao Curso de MESTRADO PROFISSIONAL EM TECNOLOGIA MINIMAMENTE INVASIVA E SIMULAÇÃO EM SAÚDE do Centro Universitário Christus, como requisito parcial à obtenção do título de mestre em Área de Concentração: Medicina III. Orientador: Dr. Marcelo Alcantara Holanda

FORTALEZA
2016

Ficha Catalográfica elaborada por Dayane Paula Ferreira Mota – Bibliotecária – CRB-3/1310

D585d Diniz, Marcelo Emanuel Bezerra.
Desenvolvimento da versão 2.0 do simulador virtual de ventilação
mecânica XLUNG / Marcelo Emanuel Bezerra Diniz. – 2016.
68 f.; il. color.

Dissertação (Mestrado) – Centro Universitário Christus - Unichristus,
Mestrado Profissional em Tecnologia Minimamente Invasiva e Simulação
em Saúde, Fortaleza, 2016.

Orientação: Prof. Dr. Marcelo Alcantara Holanda.
Área de concentração: Medicina III.

1. Respiração artificial. 2. Ventilação mecânica. 3. Software. 4.
Desenho de programas de computador. 5. Simulação. I. Título.

CDD 615.836

MARCELO EMANOEL BEZERRA DINIZ

DESENVOLVIMENTO DA VERSÃO 2.0 DO SIMULADOR VIRTUAL DE VENTILAÇÃO
MECÂNICA XLUNG

Dissertação apresentada ao Curso de
MESTRADO PROFISSIONAL EM
TECNOLOGIA MINIMAMENTE INVASIVA
E SIMULAÇÃO EM SAÚDE do Centro
Universitário Christus, como requisito
parcial à obtenção do título de mestre
em Área de Concentração: Medicina III.
Orientador: Dr. Marcelo Alcantara
Holanda

Aprovada em: 06/12/2016

BANCA EXAMINADORA

Dr. Marcelo Alcantara Holanda (Orientador)
Centro Universitário Christus (UNICHRISTUS)

Dra. Andréa Kelly Carvalho
Universidade de Fortaleza (UNIFOR)

Dra. Andrea Soares Rocha da Silva
Universidade Federal do Ceará (UFC)

Dr. Carlos Eduardo Barros Jucá
Centro Universitário Christus (UNICHRISTUS)

À minha família, por sua capacidade de acreditar
e investir em mim.

Agradecimentos

Aos meus pais e avós, pelo amor, incentivo e apoio incondicional. Por me mostrarem que através da educação e do trabalho é possível construir seres humanos melhores.

À minha esposa, que me incentivou bastante e soube entender os vários momentos de ausência devido a imensa dedicação necessária para a realização deste trabalho.

Obrigado meus irmãos e sobrinhos. Em especial ao Mario Henrique, pelas valiosas contribuições com formas melhores de explicar as ideias contidas aqui.

Ao meu orientador, Marcelo Alcantara, por me ensinar um tema novo e tão importante e, ainda por cima, me oferecer a oportunidade de contribuir com a formação de salvadores de vidas.

À minha coorientadora, Andréa Carvalho, por me ajudar a produzir este trabalho da melhor maneira possível. Sem seu apoio esta obra não teria acontecido.

A todos os integrantes do *podcast* semanal Scicast. Por me mostrar a cada novo episódio que a ciência tem que ser divertida.

A esta universidade, seu corpo docente, direção, administração e secretaria.

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação pessoal e profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender. A palavra mestre, nunca fará justiça aos professores dedicados aos quais sem nominar terão os meus eternos agradecimentos.

“É melhor lançar-se à luta em busca do triunfo mesmo expondo-se ao insucesso, que formar fila com os pobres de espírito, que nem gozam muito nem sofrem muito; E vivem nessa penumbra cinzenta sem conhecer nem vitória nem derrota.”

(Franklin Roosevelt)

Resumo

A simulação, como metodologia de ensino e treinamento de habilidades, está estabelecida como meio seguro e efetivo na formação profissional em geral. Na área da saúde tem especial significância frente aos métodos tradicionais, pois proporciona segurança, evitando que os pacientes sejam submetidos a riscos oriundos de um treinamento profissional insuficiente. A Ventilação Mecânica (VM) consiste no método de intervenção terapêutica para o tratamento de pacientes com insuficiência respiratória e é amplamente utilizada em Unidades de Terapia Intensiva (UTI) no mundo inteiro. O Simulador Virtual de Ventilação Mecânica Xlung foi lançado em 2012 e permite a criação de situações semelhantes às que um profissional de saúde pode encontrar no seu cotidiano. Este trabalho objetivou desenvolver a versão 2.0 deste simulador, bem como a criação de simuladores correlacionados ao ensino de ventilação mecânica. Para tanto, foi conduzida uma reestruturação na arquitetura interna do simulador, e um novo protocolo de comunicação para coordenar o seu funcionamento foi elaborado e desenvolvido. Foram planejadas mudanças no leiaute e novas funcionalidades. Ao término, a versão 2.0 do Simulador Virtual de Ventilação Mecânica Xlung foi desenvolvida. Sua arquitetura interna foi reestruturada de maneira a simplificar a manutenção e permitir que os usuários acessem o simulador sem a necessidade de instalação de *plugins*. Foram desenvolvidos testes automáticos de software para os seus principais componentes além de novas funcionalidades e dois outros simuladores: o Physiolung e o Oxylung. O simulador Xlung 2.0 tem sido usado desde seu lançamento em 9 de Junho de 2016 em 66 países para a realização de cerca de 1400 simulações por mês.

Palavras-chave: Respiração Artificial. Ventilação Mecânica. Software. Desenho de Programas de Computador. Simulação. Simulação por Computador. Refatoração. Interface homem-máquina. Ensino.

Abstract

Simulation as a teaching and training method is established as safe and effective mean in the professional curriculum. On health, it has special value compared to traditional methods because it provides safety, avoiding patients to be submitted to preventable risks caused by lack of training. Mechanical Ventilation (MV) is a therapeutic intervention for patients with respiratory failure. It is a widespread and known technique used in Intensive Care Units (ICU) all over the world. The mechanic ventilation virtual simulator Xlung was released in 2012 and allows the creation of similar situations of the day to day care to mechanically ventilated patients. This work intended to develop the 2.0 version of the simulator, as well as other correlated MV teaching simulators. The simulator's internal architecture was restructured and a new communication protocol was devised in order to better coordinate its functionalities. New layout and functions were planned and implemented. In the end, Xlung 2.0 was released in 9th of June, 2016, with its internal architecture restructured to simplify maintenance and to allow users to access the simulator with no need of plugin installation. Automatic software tests were developed for its internal components as well as two other simulators: Physiolung and Oxlung. Since its release the Xlung 2.0 has been used in 66 countries with an average of 1400 simulations per month.

Keywords: Respiration, Artificial. Software. Software Design. Simulation. Computer Simulation. Refactoring. Human-Machine Interface. Teaching.

Lista de ilustrações

Figura 1 – Xlung 1.0	17
Figura 2 – Número de sessões no Portal Xlung por país entre 23/08/2015 e 23/09/2015.	20
Figura 3 – Acessos do Portal Xlung no Brasil entre 23/08/2015 e 23/09/2015.	21
Figura 4 – Pirâmide ágil de testes	31
Figura 5 – Metodologia de desenvolvimento utilizada no projeto	33
Figura 6 – Modelo de comunicação Cliente Servidor com WebSockets	35
Figura 7 – Oxímetro de pulso de dedo	38
Figura 8 – Planejamento da interface do oxímetro no simulador	39
Figura 9 – Cateter de oferta de oxigênio	40
Figura 10 – Fluxômetro regulador de oxigênio	40
Figura 11 – Planejamento do leiaute do cateter de oxigênio.	41
Figura 12 – Máscara de Venturi.	41
Figura 13 – Leiaute planejado para a escolha da $F_{I}O_2$ para a máscara de Venturi.	42
Figura 14 – Máscara de reservatório de O_2	42
Figura 15 – Leiaute planejado da Máscara de reservatório de O_2	43
Figura 16 – Leiaute planejado do nomograma de equilíbrio ácido-base	44
Figura 17 – Curvas de pressão nas vias aéreas e pressão muscular	46
Figura 18 – Tela principal do simulador Physiolog	48
Figura 19 – Oxímetro de pulso de dedo vs. Oxímetro Simulado	49
Figura 20 – Oxímetro de pulso detalhado	49
Figura 21 – Oxímetro de pulso - Configuração dos alarmes vs. alarmes disparando	50
Figura 22 – Menu "Distúrbios Metabólicos"do Paciente	50
Figura 23 – Nomograma ácido-base	50
Figura 24 – Menu "Variação de Altitude"do Paciente	51
Figura 25 – Comparativo dos componentes de monitorização do paciente no Xlung 1.0 vs. Physiolog	51
Figura 26 – Tela principal do simulador Oxylung.	52
Figura 27 – Dispositivos de oxigenoterapia no Oxylung.	52
Figura 28 – Menu "Condições Clínicas"do paciente no Oxylung.	53
Figura 29 – Transferência de dados (KB) necessária para iniciar uma simulação no Xlung 2.0 vs. Xlung 1.0	54

Figura 30 – Comparativo da tela principal Xlung 1.0 vs Xlung 2.0	55
Figura 31 – Tela de configuração do ventilador nos modo ventilatórios VCV e PSV, Xlung 1.0 vs. Xlung 2.0	56
Figura 32 – Tela de gráficos, Xlung 1.0 vs. Xlung 2.0	57
Figura 33 – Comparativo dos gráficos de Volume Ponderal com Zonas de Segurança presentes no Xlung 1.0 vs. Xlung 2.0	57

Lista de Quadros

Quadro 1 – Relação coeficiente cardíaco para o sexo e idade do paciente	39
Quadro 2 – Relação entre Transtorno metabólico e <i>Base Excess</i>	43
Quadro 3 – Relação entre localização geográficas, altitude e pressão atmosférica	44
Quadro 4 – Comparativo dos simuladores em relação às seguintes funcionalidades: alterações clínicas, alterações metabólicas, nomograma de equilíbrio ácido-base, variação de altitude, dispositivos de oxigenoterapia e oxímetro de pulso com alarmes.	58
Quadro 5 – Comparativo dos simuladores em relação às seguintes funcionalidades: pulmão animado, gasometria arterial, parâmetros de monitorização, modos ventilatórios e gráficos.	59
Quadro 6 – Registros de software no INPI	60

Lista de Algoritmos

Algoritmo 1 – Mensagem de início de simulação	36
---	----

Lista de símbolos

<i>AMIB</i>	Associação de Medicina Intensiva Brasileira
<i>CSS</i>	Cascading Style Sheets
<i>DPOC</i>	Doença Pulmonar Obstrutiva Crônica
F_1O_2	Fração de Oxigênio Inspirado
<i>FUNCAP</i>	Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico
<i>GUI</i>	Interface Gráfica com o Usuário, traduzido do inglês "Graphical User Interface"
<i>HTML</i>	Hypertext Markup Language
<i>HTTP</i>	Hypertext Transfer Protocol
<i>INPI</i>	Instituto Nacional de Propriedade Industrial
<i>IOT</i>	Intubação Orotraqueal
<i>IRpA</i>	Insuficiência Respiratória Aguda
<i>JRE</i>	Java Runtime Environment
<i>JSON</i>	Javascript Object Notation
<i>LOC</i>	Linhas de Código
<i>MVC</i>	Model View Controller
<i>OO</i>	Orientação a Objetos
<i>POO</i>	Programação Orientada a Objetos
<i>PAC</i>	Pneumonia Adquirida na Comunidade
<i>PAV</i>	Pneumonia Associada à VM
<i>PEEP</i>	Pressão Positiva ao Final da Expiração
<i>PS</i>	Pressão de Suporte
<i>PTP</i>	Produto Pressão Tempo
<i>SARA</i>	Síndrome da Angústia Respiratória Aguda
<i>SBPT</i>	Sociedade Brasileira de Pneumologia e Tisiologia
<i>SDRA</i>	Síndrome de Desconforto Respiratório Agudo
<i>SVVM</i>	Simulador Virtual de Ventilação Mecânica

<i>UTI</i>	Unidade de Terapia Intensiva
<i>VC</i>	Volume Corrente
<i>VM</i>	Ventilação Mecânica
<i>VNI</i>	Ventilação Não Invasiva

Sumário

1	Introdução	16
1.1	Motivação	19
1.2	Justificativa	22
2	Revisão da Literatura	23
2.1	Ventilação Mecânica	23
2.1.1	Indicações da Ventilação Mecânica	23
2.1.2	Modos Ventilatórios Básicos	24
2.1.2.1	Modo A/C-VCV: Assistido Controlado Ciclado a Volume	24
2.1.2.2	Modo A/C-PCV: Assistido Controlado com Pressão Constante ou Controlada	25
2.1.2.3	Modo PSV: Ventilação com Pressão de Suporte	25
2.1.2.4	Modo CPAP: Pressão Positiva Contínua nas Vias Aéreas	26
2.2	Os princípios SOLID	26
2.3	Refatoração	27
2.3.1	Atividades de refatoração	28
2.3.1.1	Identificação das partes de software a serem refatoradas	28
2.3.1.2	Garantia de preservação do comportamento	30
2.4	Testes Automáticos de Software	30
3	Objetivos	32
3.1	Objetivos Específicos	32
3.2	Objetivos Secundários	32
4	Metodologia	33
4.1	Desenvolvimento e Codificação	34
4.2	Desenvolvimento de uma nova arquitetura para viabilizar o acesso ao simulador sem a utilização de <i>plugins</i>	34
4.2.1	Protocolo de Comunicação	36
4.2.2	Planejamento de melhorias na interface gráfica	36
4.3	Desenvolvimento de novas funcionalidades	37
4.3.1	Oxímetro de Pulso	37
4.3.2	Dispositivos de oxigenoterapia	39
4.3.3	Transtornos Metabólicos	43

4.3.4	Nomograma de Equilíbrio Ácido-base	43
4.3.5	Variações de altitude	44
4.3.6	Trabalho da respiração	45
4.3.7	Pressão de Oclusão aos 100 ms	45
4.3.8	PTP - Produto Pressão Tempo	46
4.3.9	f/VC	46
5	Resultados	48
5.1	Oxylung e Physiolog	48
5.2	Xlung 2.0	53
5.3	Registros de Software	60
6	Discussões	61
6.1	Diferenças percebidas	61
6.2	Limitações	62
6.3	Implicações práticas para o ensino da ventilação mecânica e da fisiologia respiratória	62
6.4	Perspectivas Futuras	62
7	Conclusões	63
	Referências	64
	Apêndices	67
	APÊNDICE A Dispensa do termo de consentimento	68

1 Introdução

A simulação como metodologia de ensino e treinamento de habilidades está estabelecida como meio seguro e efetivo na formação profissional em geral. Na aeronáutica foi introduzida em 1929 e seu uso corresponde a 40% do tempo de formação dos pilotos. Sua utilização é apoiada por resultados evidenciados por redução em 50% dos acidentes aéreos relacionados à falha humana (QUILICI et al., 2012). Na área da saúde tem especial significância frente aos métodos tradicionais, pois proporciona segurança, evitando que os pacientes sejam submetidos a riscos oriundos de um treinamento profissional insuficiente; além de proporcionar aos profissionais em treinamento, variabilidade de casos com diversos cenários clínicos, sem dependerem dos seus surgimentos esporádicos, como acontece na vida real.

A simulação em saúde pode ser dividida em:

- a) simulação mecânica: por meio de equipamentos, manequins e ambientes semelhantes aos que serão encontrados pelos alunos na vida real;
- b) simulação virtual: baseada em programas de computador interativos e associados a resolução de problemas.

O objetivo fundamental da simulação é promover experiências que mimetizem o mais fielmente a realidade. Vale ressaltar que ambas as metodologias, possuem variados graus de fidedignidade. Destaca-se nesse contexto, a simulação virtual, com a possibilidade da quebra da barreira da distância e da escala de abrangência, permitindo que um número muito maior de profissionais tenha um treinamento robusto e de qualidade (LINO et al., 2016). Apesar de todos os benefícios citados, a simulação não substitui o contato real com pacientes e deve ser vista como uma preparação para esta etapa, tornando-a mais segura para ambos, paciente e profissional.

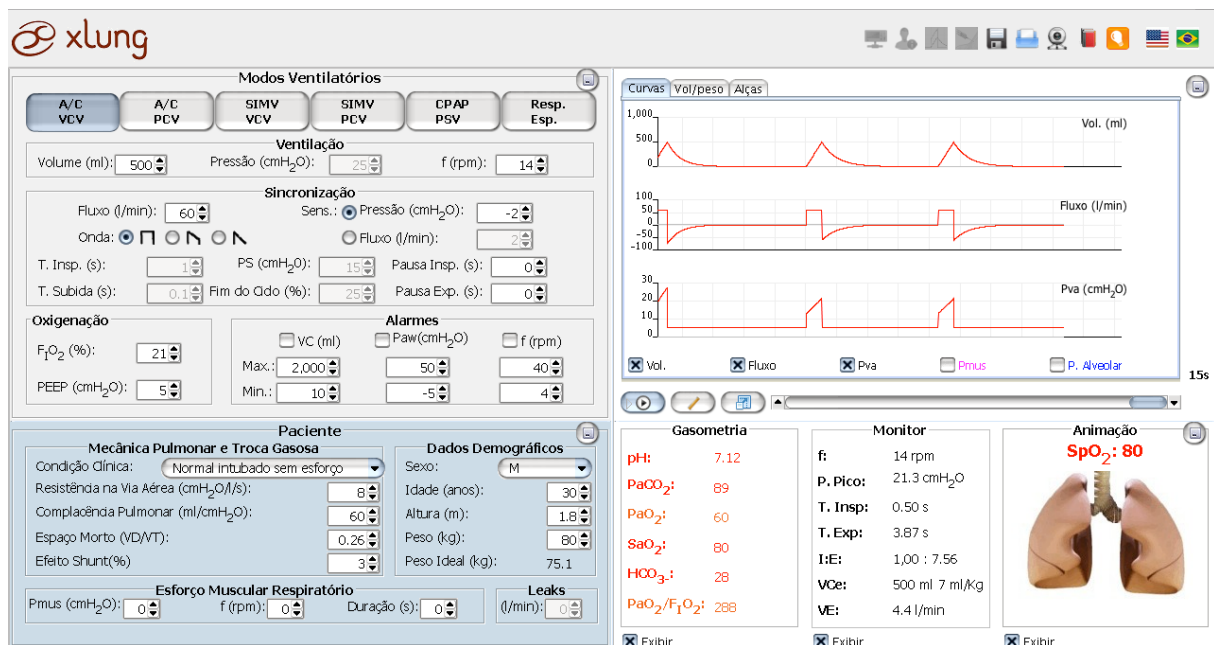
A Ventilação Mecânica (VM) consiste no método de intervenção terapêutica para o tratamento de pacientes com complicações na troca gasosa pulmonar e é amplamente utilizada em Unidades de Terapia Intensiva (UTI) no mundo inteiro. Cerca de um terço dos pacientes internados em UTIs necessitam de suporte ventilatório mecânico invasivo e/ou não invasivo (ESTEBAN; ANZUETO et al., 2002). A utilização correta da VM pode reduzir a mortalidade em algumas condições como a Síndrome da Angústia Respiratória Aguda (SARA), pneumonia e exacerbação da Doença Pulmonar Obstrutiva Crônica (DPOC) entre outras. Além disso, seu uso adequado pode reduzir a permanência do paciente na UTI, acelerar o processo de desmame e extubação (GIRARD et al., 2008), diminuir complicações e, conseqüentemente reduzir os custos

hospitalares.

Apesar da VM ser uma técnica de fundamental importância, há uma percepção de carência de profissionais bem preparados para utilizá-la. Isto, em parte, devido a baixa quantidade de tempo e diversidade de casos clínicos utilizados para o ensino durante a graduação em medicina, fisioterapia e enfermagem. Um estudo avaliou a eficácia do treinamento em Ventilação Mecânica de 347 residentes onde 48% não conseguiram administrar corretamente o volume corrente em pacientes com SARA, 38% falhavam em identificar um paciente apto para o teste de desmame e 27% não conseguiram reconhecer a indicação para Ventilação Não Invasiva (VNI). Do total, apenas 54% diziam estar satisfeitos com o seu treinamento em VM (COX et al., 2003).

A partir desse cenário foi desenvolvido o Xlung (Figura 1), um Simulador Virtual de Ventilação Mecânica (SVVM), com o intuito de auxiliar e disseminar o ensino da técnica, que permite a criação de situações semelhantes às que um profissional de saúde pode encontrar no seu cotidiano (HOLANDA et al., 2012).

Figura 1 – Xlung 1.0



Fonte: <http://www.xlung.net>

Ganhador do Prêmio Brasil de Engenharia em 2011 na seção de Educação Tecnológica categoria livre (SINDICATO DOS ENGENHEIROS NO DISTRITO FEDERAL, 2011) e com trabalho publicado na American Thoracic Society International Conference (HOLANDA et al., 2012), o simulador Xlung foi pensado inicialmente como uma aplicação *desktop offline* que

poderia ser executado em diferentes sistemas operacionais. Para tanto, necessitaria da instalação do Java Runtime Environment (JRE) na máquina do usuário, uma camada extra de software que abstrai os detalhes de implementação específicos de cada sistema operacional. No decorrer do desenvolvimento ocorreram alterações de escopo no projeto e uma nova necessidade foi identificada, o simulador deveria se comportar como uma aplicação *web online* com o intuito de atingir um público muito maior, evitar problemas relacionados ao uso não autorizado e potencializar a sua comercialização. Juntamente com essa mudança surgiu o *website* <<https://www.xlung.net>>, com seções de conteúdo relacionados à VM. O simulador ajudou a quebrar barreiras de distância e limitações relacionadas ao espaço físico e já conta com evidências de sua aplicabilidade no ensino de VM para profissionais de saúde (LINO et al., 2016).

A decisão de mudança de versão *offline* para *online* e a inclusão de seções de conteúdo veio ao encontro de uma tendência de ampliação da escala do ensino. Na capitania das empresas do setor de ensino em larga escala, o Coursera, situado no Vale do Silício norte americano, já possuía em 2013, 2,9 milhões de usuários registrados em todo o mundo. Desses 5,1% eram oriundos do Brasil sendo superados em número apenas pelos Estados Unidos e Índia (WALDROP, 2013). O seguimento de ensino à distância no Brasil obteve o maior índice de crescimento (35%) no período entre 2010 e 2013 de acordo com o Censo de Educação Superior de 2013 (MINISTÉRIO DA EDUCAÇÃO, 2013).

A fim de adequar o simulador à nova realidade foi decidido transformar a aplicação em um *Applet Java* que poderia ser executado em diferentes sistemas operacionais como Windows, Linux e OS X (HOLANDA et al., 2012). *Applets* são aplicações desenvolvidas utilizando a Plataforma Java e são distribuídas através de páginas na *web* (ORACLE, 2015a). Aplicações desse tipo são executadas em processos separados do navegador e podem ter acesso a alguns recursos de hardware como aceleração gráfica e sistema de arquivos. Os *Applets Java* foram introduzidos com a primeira versão da linguagem de programação Java, ainda em 1995. Até o ano de 2010 eram tidos como o padrão *de facto* para aplicações *web* que necessitavam de acesso a tais recursos.

O Java Runtime Environment (JRE) atuava nos navegadores através de uma arquitetura de *plugins*¹. Apesar de facilitar o desenvolvimento em alguns aspectos, os *Applets Java* inserem um nível de complexidade para o usuário final. Para a utilização do simulador o usuário necessita instalar o JRE na sua máquina, o que nem sempre é um processo fácil dado o

¹ Uma camada de software encaixada posteriormente capaz de prover a expansão ou inclusão de funcionalidades antes não previstas.

baixo nível de familiaridade com computadores do público alvo, estudantes e profissionais de saúde. Além disso, a opção pela construção do simulador com o JRE impediu sua utilização em dispositivos móveis como *tablets* e *smartphones*, por não haver suporte nesses dispositivos a esse pré-requisito de funcionamento. Outro fato a ser destacado é que os navegadores mais modernos cada vez mais incentivam a modernização das aplicações através da criação de barreiras ao uso de *plugins* como o JRE, o Adobe Flash e o Microsoft Silverlight. (ORACLE, 2015b; GUSMÃO, 2015; NETFLIX, 2013; AVRAM, 2015)

Esse movimento das empresas que desenvolvem os navegadores ganhou força a partir de 2010, com a carta aberta de Steve Jobs intitulada *Thoughts on Flash* (JOBS, 2010) que defendia a adoção de padrões abertos para a *web* em detrimento do *Adobe Flash*. No texto outros fatores foram levantados, entre eles o alto número de falhas consideradas graves, o consumo excessivo de bateria e a existência de alternativas consideradas melhores para a solução do problema. Desde então, com o apoio de grandes empresas como Apple, Google, Microsoft e demais integrantes do World Wide Web Consortium (W3C), observou-se um aumento no interesse pelos padrões abertos. Tecnologias como o Hypertext Markup Language (HTML), Javascript e Cascading Style Sheets (CSS) se tornaram padrões de mercado para a construção de aplicações *web* (GUSMÃO, 2015; NETFLIX, 2013).

Comprovando a tendência, em Janeiro de 2016, em postagem em seu blog (ORACLE, 2016) a própria Oracle, proprietária da plataforma Java, anunciou o término do suporte aos Applets Java a partir da versão seguinte da plataforma (JDK 9) e a completa remoção em futuras versões.

Diante do que foi exposto, este trabalho dará continuidade ao trabalho inicialmente desenvolvido por Holanda et al. em 2012 através de melhorias, que somadas, constituirão a versão 2.0 do simulador Xlung.

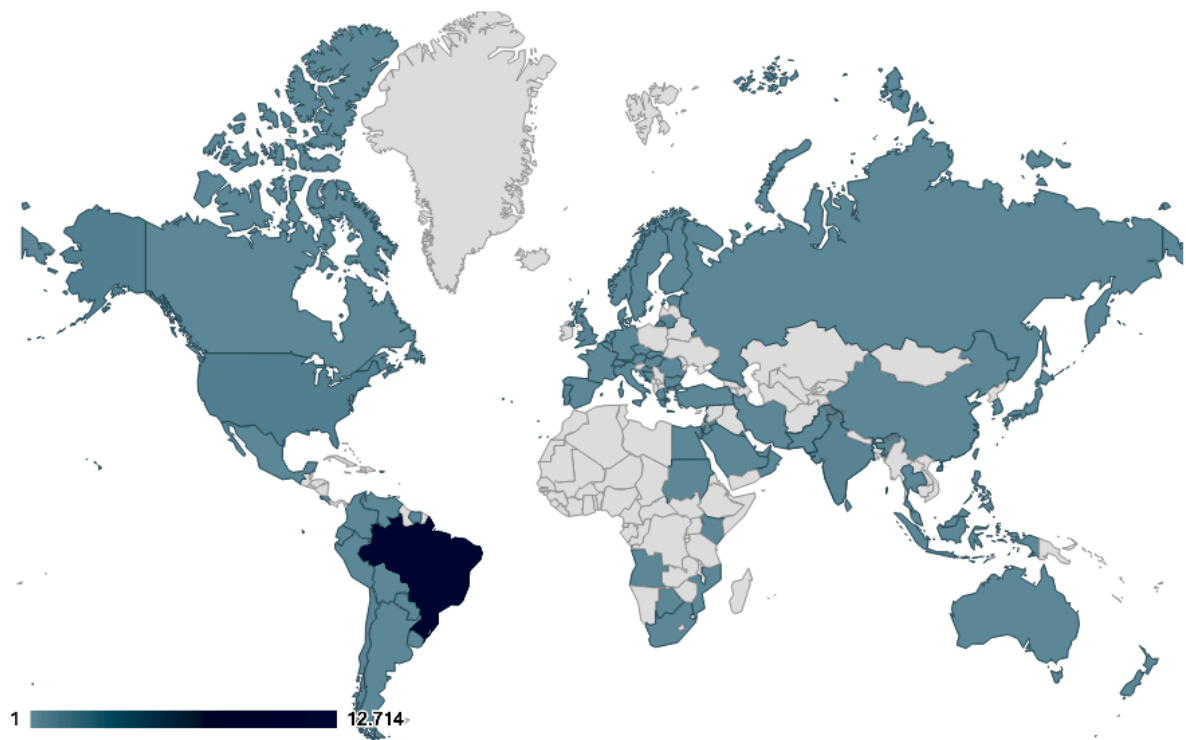
1.1 Motivação

Nascida comercialmente no ano de 2012 com o apoio da empresa Pulmocenter <<http://www.pulmocenter.com.br/>> e da Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) a empresa Xlung, através da sua plataforma de ensino, se destaca hoje, como uma das principais empresas no mundo na promoção do ensino à distância relacionado à VM, em português e inglês. Dispõe de video aulas, exercícios rápidos (*quizzes*) com temas práticos e variados, manual de ventilação mecânica, artigos comentados e finalmente o principal

produto, o simulador de ventilação mecânica Xlung. Contava em 2015 com mais de 22.000 acessos mensais, agrupados em aproximadamente 13.000 sessões de uso, distribuídos entre 31 países conforme pode ser observado na Figura 2. Além de cerca de 3.000 usuários cadastrados no site, possui bastante visibilidade entre estudantes e profissionais de saúde relacionados ao tema.

Além do site, a plataforma conta com um sistema de gerenciamento institucional com, atualmente, 23 instituições parceiras entre universidades, cursos de Ventilação Mecânica e associações médicas, onde os clientes podem coletar estatísticas de uso, gerenciar as suas assinaturas e se comunicar com os assinantes. Esses clientes também têm acesso pleno à plataforma e a todo o seu conteúdo didático.

Figura 2 – Número de sessões no Portal Xlung por país entre 23/08/2015 e 23/09/2015.

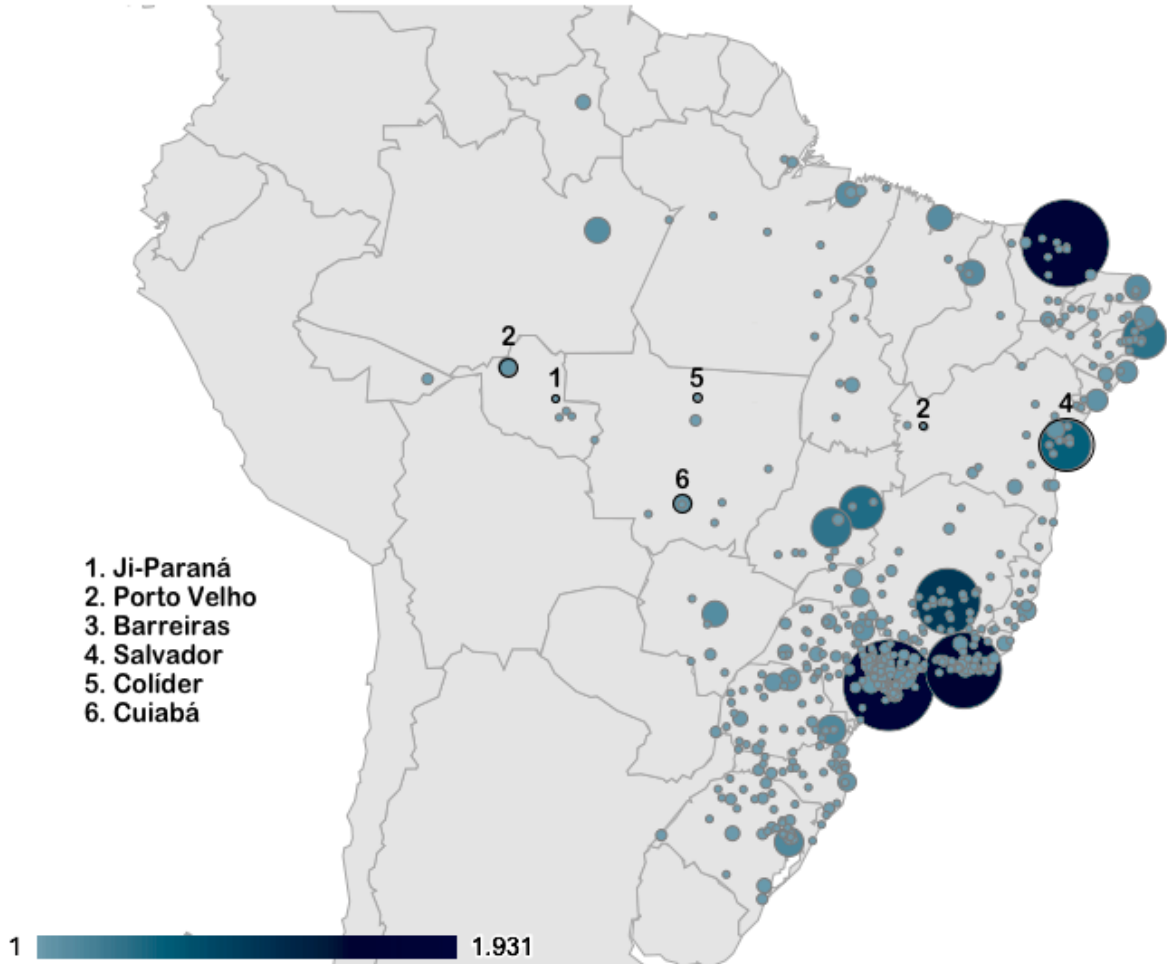


Fonte: Google Analytics

Por ser uma empresa genuinamente cearense e brasileira, é natural que possua mais acessos no Brasil. Entre outras coisas, saltam aos olhos os acessos oriundos de regiões mais distantes dos grandes centros, conforme pode ser visto na Figura 3, como por exemplo, as cidades de Ji-Paraná em Rondônia, distante 373 km da capital Porto Velho; Barreiras no estado da Bahia, distante 863 km da capital Salvador e Colíder no Mato Grosso, distante 650 km da

capital Cuiabá, entre muitas outras.

Figura 3 – Acessos do Portal Xlung no Brasil entre 23/08/2015 e 23/09/2015.



Fonte: Google Analytics

Com formação em Ciências da Computação pela Universidade de Federal de Campina Grande (UFCG) em 2008 e especialização em Desenvolvimento de Software com ênfase na Arquitetura JavaEE pela Universidade de Fortaleza (UNIFOR) no ano de 2011, passei a integrar a equipe da empresa Xlung e me envolver diretamente com o desenvolvimento do simulador a partir de 2013. No processo de contextualização inicial do simulador foram identificados pontos de melhoria no projeto que serviram de base para a elaboração deste trabalho.

Destacam-se nesses pontos uma mudança de interface gráfica, com o intuito de reduzir a quantidade de informações desnecessárias visíveis, em determinados momentos, para o controle do ventilador mecânico; bem como, a inclusão de equipamentos na simulação: o oxímetro de pulso e dispositivos de oxigenoterapia (cânulas, máscaras de Venturi e de reservatório). Além das funcionalidades acima citadas, acrescentaremos também ferramentas capazes de

auxiliar no ensino de temas relacionados; como o equilíbrio ácido-básico, transtornos metabólicos, efeitos na troca gasosa causados por variações na altitude e cálculo de parâmetros úteis na ventilação mecânica: o trabalho da respiração, a pressão muscular em 100ms (milissegundos) após o início do ciclo respiratório e o Produto Pressão Tempo (PTP).

1.2 Justificativa

Levando em consideração tudo o que foi exposto até aqui, o desenvolvimento de uma nova versão do simulador Xlung permitirá aos usuários continuar utilizando o simulador mesmo após o término do suporte à utilização de *plugins* nos navegadores. O Xlung 2.0 facilitará o acesso dos usuários através da remoção da necessidade de instalação da JRE nas suas máquinas. Aliando a atualização de tecnologias, melhorias no leiaute e novas ferramentas de simulação, irá melhorar a experiência de uso dos usuários atuais e permitirá que mais pessoas interessadas no tema Ventilação Mecânica tenham a possibilidade de aprender e praticar a forma correta de cuidar dos pacientes ampliando o interesse da comunidade da saúde, incluindo empresas relacionadas ao suporte ventilatório.

2 Revisão da Literatura

Para a correta identificação dos requisitos e melhor entendimento do problema, se faz necessário uma revisão dos conceitos que serão implementados. A ventilação mecânica, como tema principal do produto deste trabalho, será abordada na *seção 2.1*. Nas seções seguintes são explicados os temas relacionados ao desenvolvimento e melhoria de software. A *seção 2.2 apresenta Os princípios SOLID*, um conjunto de fundamentos que ajudam a definir, do ponto de vista técnico, o que é considerado um bom projeto de software e, a *seção 2.3 Refatoração*, aborda conceitos relacionados ao processo utilizado na sua reestruturação.

2.1 Ventilação Mecânica

O conceito de respiração artificial foi reconhecido pela primeira vez no século XVI por Vesalius. Entretanto, somente no século XX, a VM passou a ser amplamente utilizada como uma modalidade terapêutica. Bjorn Ibsen aplicou com sucesso ventilação com pressão positiva em uma população de pacientes com paralisia respiratória, causada por pólio durante o surto de Copenhagem em 1952, reduzindo a mortalidade de aproximadamente 85% em Julho de 1952 para 15% em Março do ano seguinte. Esta intervenção se tornou conhecida como o berço da ventilação mecânica moderna utilizada como um método para gerenciar insuficiência respiratória aguda, anunciando o desenvolvimento da Unidade de Terapia Intensiva (UTI) (KARCZ et al., 2012).

2.1.1 Indicações da Ventilação Mecânica

A VM substitui total ou parcialmente a ventilação espontânea e é indicada na Insuficiência Respiratória Aguda (IRpA) ou crônica agudizada. Sua utilização propicia a melhora das trocas gasosas e diminuição do trabalho respiratório, podendo ser utilizada de forma não invasiva, VNI, por meio de uma interface externa, geralmente uma máscara facial, e de forma invasiva, por meio de um tubo endotraqueal ou cânula de traqueostomia (BARBAS et al., 2014).

A insuficiência respiratória pode ser vista como a incapacidade de manter a troca gasosa pulmonar e, portanto, a oferta de oxigênio às células do organismo e a remoção de CO_2 produzido pelo metabolismo celular. A VM é indicada quando a insuficiência respiratória grave se faz presente. Em números, geralmente equivale a uma $PaO_2 < 60$ mmHg mesmo com suplementação de oxigênio, ou seja, $F_{IO_2} > 0.6$ ou 60%; com ou sem aumento progressivo dos

valores da PaCO_2 , acarretando acidose respiratória aguda com valores de pH variando abaixo de 7.25 (KARCZ et al., 2012).

Em 2013 foram publicadas as Recomendações Brasileiras de Ventilação Mecânica (BARBAS et al., 2014) pela Sociedade Brasileira de Pneumologia e Tisiologia (SBPT) em conjunto com a Associação de Medicina Intensiva Brasileira (AMIB). O texto é uma revisão da literatura com o intuito de orientar os profissionais em geral, com base em evidências existentes, para as melhores práticas de VM à beira do leito.

2.1.2 Modos Ventilatórios Básicos

A VM é uma intervenção de suporte avançado de vida, que se utilizada de forma inadequada, pode ocasionar lesões ao pulmão ventilado, especialmente na presença de uma doença pulmonar preexistente. Uma vez iniciada, deve ser aplicada no menor período de tempo possível e de modo "gentil" e fisiológico (KARCZ et al., 2012). A fim de encontrar a forma mais adequada de ventilar o pulmão, foram desenvolvidos os modos ventilatórios.

Pode-se definir como modo ventilatório, o processo pelo qual o ventilador pulmonar mecânico determina, seja de forma parcial ou total, como e quando os ciclos respiratórios mecânicos são ofertados ao paciente. Dessa forma, o modo determina substancialmente o padrão respiratório do paciente durante a ventilação mecânica (HOLANDA, 2014). A seguir, uma descrição breve dos principais modos ventilatórios utilizados.

2.1.2.1 Modo A/C-VCV: Assistido Controlado Ciclado a Volume

O modo ventilatório A/C-VCV permite que o operador do ventilador defina um Volume Corrente (VC) a ser entregue pelo ventilador durante um tempo predeterminado. O VC é atingido em consequência da oferta de um fluxo de gás gerado no ventilador. Um ciclo respiratório pode ser iniciado tanto pelo ventilador, disparo a tempo, ou pelo próprio paciente, disparo a fluxo ou pressão, segundo o parâmetro de sensibilidade. Também é possível ajustar a frequência respiratória (f) mínima e o tempo inspiratório, o fluxo inspiratório e o seu padrão (quadrado ou em rampa), a Fração de Oxigênio Inspirado (F_1O_2) e a Pressão Positiva ao Final da Expiração (PEEP) (KARCZ et al., 2012).

A utilização do modo A/C-VCV é indicada quando se almeja manter um determinado volume minuto ($\text{VC} \times f$). A pressão nas vias aéreas é variável e consequente à combinação dos parâmetros do ventilador e à mecânica ventilatória do paciente. Esse modo é utilizado também

para medir a pressão de pico e a pressão de platô, ou de pausa inspiratória, com o intuito de calcular a resistência e a complacência do sistema respiratório sob fluxo inspiratório constante e quadrado (BARBAS et al., 2014).

2.1.2.2 Modo A/C-PCV: Assistido Controlado com Pressão Constante ou Controlada

No modo ventilatório A/C-PCV um valor predeterminado de pressão nas vias aéreas acima da PEEP é ofertado durante um tempo inspiratório selecionado pelo operador (ciclagem a tempo). As variáveis dependentes são o VC e o fluxo inspiratório. Alterações na resistência das vias respiratórias ou na complacência acarretam variações no volume corrente entregue. Os parâmetros disponíveis são: frequência respiratória (f), delta de pressão aplicada acima da PEEP nas vias aéreas e tempo inspiratório, além da $F_{I}O_2$ e a PEEP, que independem do modo. O ciclo respiratório pode ser disparado pelo paciente, utilizando variações na pressão ou fluxo, ou pelo ventilador, a tempo (KARCZ et al., 2012).

Este modo deve ser utilizado em situação de comprometimento da mecânica do sistema respiratório (complacência baixa e ou resistência alta), pois permitem o controle mais adequado da pressão nas vias aéreas e da pressão alveolar. A aceleração do fluxo inspiratório pode ser controlada através da manipulação do parâmetro *rise time* ou tempo de subida (BARBAS et al., 2014).

2.1.2.3 Modo PSV: Ventilação com Pressão de Suporte

A ventilação com pressão de suporte (PSV) é considerada o modo preferencial durante a ventilação assistida ou espontânea. Sua utilização deve ser iniciada o mais precocemente possível, conforme o quadro clínico. O início do ciclo respiratório é disparado exclusivamente pelo paciente através de variações de pressão ou fluxo. Caracteriza-se por pressão limitada durante toda a fase inspiratória, ciclando para a expiração quando o fluxo inspiratório cai, geralmente, a 25% do pico de fluxo inspiratório. Esse critério de ciclagem (% de ciclagem), em alguns ventiladores, pode ser regulado de 5 a 80%, permitindo redução do tempo inspiratório em pacientes obstrutivos (% de ciclagem > 25%) e aumento do tempo inspiratório em pacientes restritivos (% de ciclagem < 25%). O *rise time* pode ser mais acelerado (menor) em pacientes obstrutivos, permitindo-se diminuir o tempo inspiratório e ajustar um VC mais adequado. Deve-se tomar cuidado especial com a ocorrência de picos de fluxo excessivos (*overshoot*). Nos pacientes restritivos, utilizar *rise time* menos acelerado, com possível ganho de VC (BARBAS et

al., 2014).

2.1.2.4 Modo CPAP: Pressão Positiva Contínua nas Vias Aéreas

O modo ventilatório CPAP se caracteriza pela oferta constante de um nível de pressão nas vias aéreas na inspiração e expiração. Este modo é utilizado na Ventilação Não Invasiva (VNI) por meio de uma interface externa, geralmente uma máscara nasofacial e a ventilação do paciente é feita de forma totalmente espontânea (KARCZ et al., 2012; BARBAS et al., 2014).

Foram apresentados até aqui os aspectos do projeto relacionados à saúde. A seção a seguir abordará os temas relacionados ao desenvolvimento de software e seus desafios.

2.2 Os princípios SOLID

É difícil obter o máximo benefício das linguagens de programação sem seguir alguns princípios básicos relacionados ao gerenciamento de dependências de código. Se feito de forma deficiente produz código fonte que é rígido, frágil e difícil de ser reutilizado. Em contrapartida um gerenciamento adequado permite que o código se mantenha flexível, robusto e reutilizável (MARTIN, 2005).

É possível afirmar que um software contém falhas de projeto quando, mesmo cumprindo os requisitos funcionais, uma ou mais características a seguir estão presentes (MARTIN, 1996):

- a) Rigidez: é difícil realizar mudanças por que cada alteração afeta muitas outras partes do sistema;
- b) Fragilidade: quando uma alteração é realizada, partes inesperadas do sistema quebram;
- c) Imobilidade: é difícil reutilizar partes ou todo o projeto em outra aplicação por que ele não pode ser desligado ou separado da aplicação atual.

Um projeto rígido provoca dificuldade na previsão do tamanho do impacto de alterações. Isto impossibilita a definição de uma estimativa de custo. Gerentes, diante desta imprevisibilidade, se tornam relutantes em autorizar qualquer alteração no sistema e por conta disso, o projeto se torna "engessado". Por sua vez, quando uma única alteração feita provoca novos problemas em áreas que não tem nenhum relacionamento conceitual com a área que foi alterada, esse projeto é considerado frágil. Isso mina a credibilidade das equipes/empresas de projeto e manutenção. Usuários e gerentes se tornam incapazes de determinar a qualidade do

produto, na visão deles, alterações em uma parte da aplicação levam a falhas em outras partes que aparentemente são completamente sem relação. O processo de manutenção passa por um círculo vicioso, onde a correção de problemas gera ainda mais problemas. Por último, em um projeto imóvel, projetistas com a tarefa de investigar e decidir se ele pode ser reutilizado em uma aplicação diferente encontrarão uma quantidade de trabalho desproporcional em relação ao resultado esperado. Não raramente, esses projetos não são reutilizados devido o custo da separação ser muito maior que o custo da construção de um novo projeto sem esses problemas (MARTIN, 1996).

Em uma série de 5 artigos, Martin detalha quais são esses princípios, que passaram a ser conhecidos como "Princípios SOLID" um acrônimo dos títulos dos artigos *The Single Responsibility Principle* (O princípio da responsabilidade única), *The Open Closed Principle* (O princípio do aberto-fechado), *The Liskov Substitution Principle* (O princípio de substituição Liskov), *The Interface Segregation Principle* (O princípio da segregação de interfaces) e *The Dependency Inversion Principle* (O princípio da inversão de dependências), e como utilizá-los para o projeto e construção de software de boa qualidade.

A seção a seguir, trata de como é possível reestruturar um software de modo que sua manutenção, tanto evolutiva quanto corretiva, seja feita de forma segura utilizando um processo chamado de refatoração.

2.3 Refatoração

A necessidade de evolução é uma propriedade intrínseca do software. À medida que é melhorado, modificado e adaptado aos novos requisitos, o código fonte se torna mais complexo e se desvia do *design* inicial, conseqüentemente perdendo qualidade. Devido a isso, a maior parte do custo de desenvolvimento de software é destinada à sua manutenção. Melhores técnicas de desenvolvimento de software não são capazes de resolver esse problema. Sua produtividade aumentada é utilizada para implementar novos requisitos num mesmo período de tempo, tornando novamente o software mais complexo. A fim de lidar com essa espiral de complexidade, se faz necessária a utilização de técnicas capazes de aumentar a qualidade interna do software de forma incremental. O campo de pesquisa que lida com este problema é conhecido por refatoração (MENS; TOURWÉ, 2004).

Refatoração é o processo de mudança de um software, onde sua estrutura interna é melhorada, desde que para isso seu comportamento externo seja mantido (FOWLER et al.,

1999). A técnica é utilizada para melhorar os requisitos internos da qualidade do software (extensibilidade, modularidade, reusabilidade, coesão e coerência), bem como, nos contextos de reestruturação (conversão de código legado e/ou deteriorado, em uma forma mais modular e estruturada) e evolução (MENS; TOURWÉ, 2004).

2.3.1 Atividades de refatoração

De acordo com Mens e Tourwé (2004) o processo de refatoração consiste das seguintes atividades:

- a) Identificar as partes do software que devem ser refatoradas;
- b) Determinar quais refatorações devem ser aplicadas aos locais identificados;
- c) Garantir que a refatoração a ser aplicada preserve comportamento do software;
- d) Aplicar a refatoração;
- e) Aferir o efeito da refatoração em características de qualidade do software (ex. complexidade, entendimento, *manutenabilidade*¹) ou no processo (ex. produtividade, custo, esforço);
- f) Manter a consistência entre o código do software refatorado e outros artefatos (como documentação, documentos de projeto, especificações de requisitos, testes entre outros).

2.3.1.1 Identificação das partes de software a serem refatoradas

A abordagem mais difundida para detectar partes do programa que necessitam de refatoração é a identificação de *bad smells* (MENS; TOURWÉ, 2004). De acordo com Kent Beck, *bad smells* são "estruturas em código que sugerem (às vezes até gritam por) a possibilidade de refatoração" (FOWLER et al., 1999).

A existência de código duplicado é um típico exemplo de *bad smell*, já que a duplicação deve ser evitada, pois sabe-se que ela dificulta a manutenção. A refatoração a ser proposta nesse exemplo seria a remoção dos códigos duplicados, extraíndo o seu conteúdo para um local intermediário, uma classe ou um método, para fatorar o código em comum (MENS; TOURWÉ, 2004).

Apesar de possível, a análise e inspeção manual do código fonte em busca de oportu-

¹ Termo utilizado na ciência da computação para designar algo que pode sofrer manutenção de forma fácil, tornando alterações mais precisas. Geralmente essa característica está presente quando são utilizados padrões arquiteturais de projeto.

nidades de refatoração é um processo demorado e custoso. Pesquisadores da área geralmente propõem técnicas total ou parcialmente automatizadas para identificação de tais oportunidades (DALLAL, 2015).

Em um estudo de revisão sistemática da literatura, 47 trabalhos foram analisados e foram levantadas seis formas de identificar os pontos de melhoria para refatoração (DALLAL, 2015) são eles:

- a) Identificação orientada por métricas de qualidade;
- b) Identificação orientada por precondições;
- c) Identificação orientada por agrupamento;
- d) Identificação orientada por grafos²;
- e) Identificação orientada por *code slicing*;
- f) Identificação orientada por análise dinâmica.

A identificação por métricas de qualidade leva em consideração estatísticas extraídas através de análise do código fonte para identificar e prever oportunidades de refatoração. São citadas como estatísticas utilizadas: tamanho em Linhas de Código (LOC), análise de coesão e acoplamento, semelhança estrutural e semântica entre métodos de uma ou mais classes entre outras.

A identificação por precondições leva em consideração um conjunto predefinido de condições para determinar a execução ou não de uma técnica de refatoração. Por exemplo, sugere-se que um método seja movido se ele não utilizar recursos locais, raramente é invocado e é referenciado basicamente por outra classe.

A identificação por agrupamento leva em consideração uma medida de similaridade para as linhas de código de um método a fim de identificar a extração dessas linhas em um novo método. Ou ainda, a similaridade de um método com outros na mesma e em diferentes classes.

A identificação por grafos leva em consideração técnicas da teoria dos grafos³. É criada uma representação das dependências do código em questão, na forma de um grafo² e são analisadas, por exemplo, se cada nó do grafo é alcançável para sugerir a extração de uma interface comum.

A identificação por *code slicing* leva em consideração a divisão do grafo de dependências de um programa para encontrar oportunidades de refatoração.

² Estrutura matemática que relaciona um conjunto de vértices V e um conjunto de arestas A através de uma função g da forma $A \rightarrow P(V)$, que associa cada aresta a um subconjunto de um ou dois elementos de V e pode ser interpretado como sendo seus terminais.

³ Ramo da matemática que estuda as relações de um determinado conjunto através do uso de grafos.

Por último, a análise dinâmica utiliza rastros deixados pelos métodos de um programa durante a sua execução, para detectar relacionamentos entre os métodos e identificar oportunidades de aplicação da estratégia *Move Method*.

2.3.1.2 Garantia de preservação do comportamento

Por definição, uma refatoração não deve alterar o comportamento do software. Infelizmente na prática, a precisa definição de comportamento dificilmente é fornecida ou é checada de forma ineficiente.

De forma bem pragmática, é possível se lidar com a preservação do comportamento através de rigorosa disciplina de testes. Se um extenso conjunto de testes forem executados com sucesso antes e depois da refatoração, existe uma boa evidência de que o programa preserva o comportamento. É possível, entretanto, que alguns testes sejam invalidados, mesmo que o comportamento seja preservado, por dependerem da estrutura atual do código que foi modificada pela refatoração (MENS; TOURWÉ, 2004).

2.4 Testes Automáticos de Software

Testes de software podem ser entendidos como um projeto ou implementação de um sistema, que exercita outro sistema, com o intuito de encontrar falhas de especificação ou implementação. Entre os benefícios de se aplicar testes em um software, é possível “descobrir falhas ou defeitos no software que apresenta comportamento incorreto, não desejável ou em não conformidade com sua especificação” (SOMMERVILLE, 2007).

Os testes podem ser classificados de acordo com: a granularidade da unidade sendo testada, a visualização ou não do mecanismo interno sendo testado ou ainda pelo custo de encontrar falhas. Seu custo em relação à utilização de rotinas de testes estritamente manuais é muito menor, possuem uma maior reprodutibilidade e são executados muito mais rapidamente.

De acordo com o objetivo é possível classificar um teste da seguinte forma:

- a) testes unitários: objetivam exercitar as unidades funcionais mais básicas de um sistema, métodos e classes, isoladamente;
- b) testes de integração: objetivam exercitar módulos inteiros de um sistema, além do relacionamento entre eles com outras partes do sistema;
- c) testes de aceitação: exercitam o sistema inteiro através da interface com o usuário.

É possível ainda classificar quanto ao nível de conhecimento da estrutura interna:

- a) testes caixa branca: possuem total conhecimento da estrutura interna do código que está sendo testado. Por conta disso, geralmente é utilizado para melhorar essa estrutura enquanto mantém o comportamento esperado;
- b) testes caixa preta: possuem total desconhecimento da estrutura interna do código que está sendo testado. Por isso, o foco está apenas nas entradas e saídas do sistema sob teste.

O livro *Succeeding with Agile* (COHN, 2009), apresenta o conceito da pirâmide ágil de testes (Figura 4). Onde se percebe uma proporção da distribuição da quantidade de testes de acordo com o objetivo, velocidade de execução e custo de manutenção. A maioria dos testes devem ser de testes unitários seguidos por testes de integração e por último, mas não menos importantes, os testes de aceitação, através da interface com o usuário para a obtenção do melhor custo benefício.

Figura 4 – Pirâmide ágil de testes



Proporção da distribuição da quantidade de testes de acordo com o objetivo, velocidade de execução e custo de manutenção.

Fonte: Imagem retirada do site <http://martinfowler.com/bliki/TestPyramid.html> e adaptada pelo autor

A seção seguinte apresenta os objetivos deste trabalho.

3 Objetivos

Desenvolvimento da versão 2.0 do Simulador Virtual de Ventilação Mecânica Xlung.

3.1 Objetivos Específicos

- a) introduzir testes automáticos de software nas partes principais do projeto;
- b) viabilizar o acesso dos usuários ao simulador através de navegadores sem a necessidade de *plugins*;
- c) desenvolver novas funcionalidades no simulador.

3.2 Objetivos Secundários

Desenvolver outros simuladores correlacionados ao ensino de ventilação mecânica.

4 Metodologia

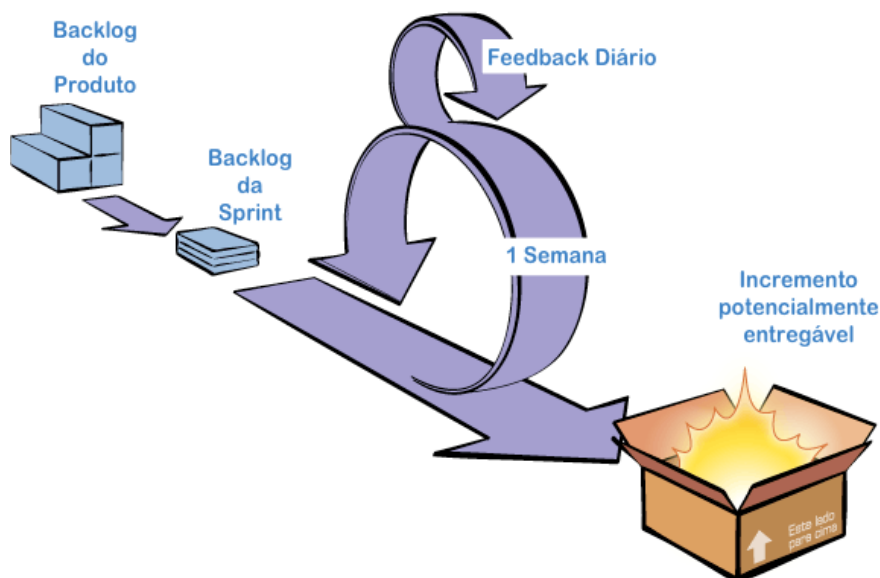
Para a gestão e planejamento do projeto foram utilizadas diversas práticas de desenvolvimento ágil de software. De acordo com a metodologia Scrum, um escopo geral do projeto (*Backlog do Projeto*) foi definido e a execução foi dividida em ciclos de uma semana (*Sprints*), onde várias atividades eram realizadas. O andamento do projeto era acompanhado diariamente em reuniões rápidas entre o desenvolvedor e o orientador. Ao término de cada semana, um pequeno conjunto de funcionalidades (*Backlog da Sprint*), novas ou remodeladas, era introduzido no produto final.

A equipe de desenvolvimento foi composta pelo autor do trabalho desempenhando as atividades de análise, prototipação e codificação, e o orientador desempenhando o papel de *product owner*, responsável por definir e priorizar o escopo do projeto.

As reuniões de planejamento e revisão aconteciam às terças feiras de cada semana, com a validação do produto em um ambiente de testes idêntico ao ambiente final de execução do projeto. A reunião de retrospectiva da *sprint* do Scrum foi intencionalmente removida por uma questão de adaptação à realidade do projeto. A Figura 5 apresenta um diagrama da metodologia de desenvolvimento.

As seções a seguir destacam algumas atividades técnicas realizadas.

Figura 5 – Metodologia de desenvolvimento utilizada no projeto



Fonte: Imagem retirada do site <http://www.desenvolvimentoagil.com.br/scrum/> e adaptada pelo autor

4.1 Desenvolvimento e Codificação

A primeira atividade do projeto foi uma análise exploratória do código fonte, que visava melhorar o entendimento e a contextualização do problema, além de identificar pontos internos de melhoria. Testes automáticos de software foram escritos para garantir que as alterações seguintes não afetassem o comportamento esperado do simulador, conforme defendido por Fowler et al. (1999).

Este processo de melhoria contínua levou em conta os Princípios SOLID (MARTIN, 2005) e a experiência prévia da equipe.

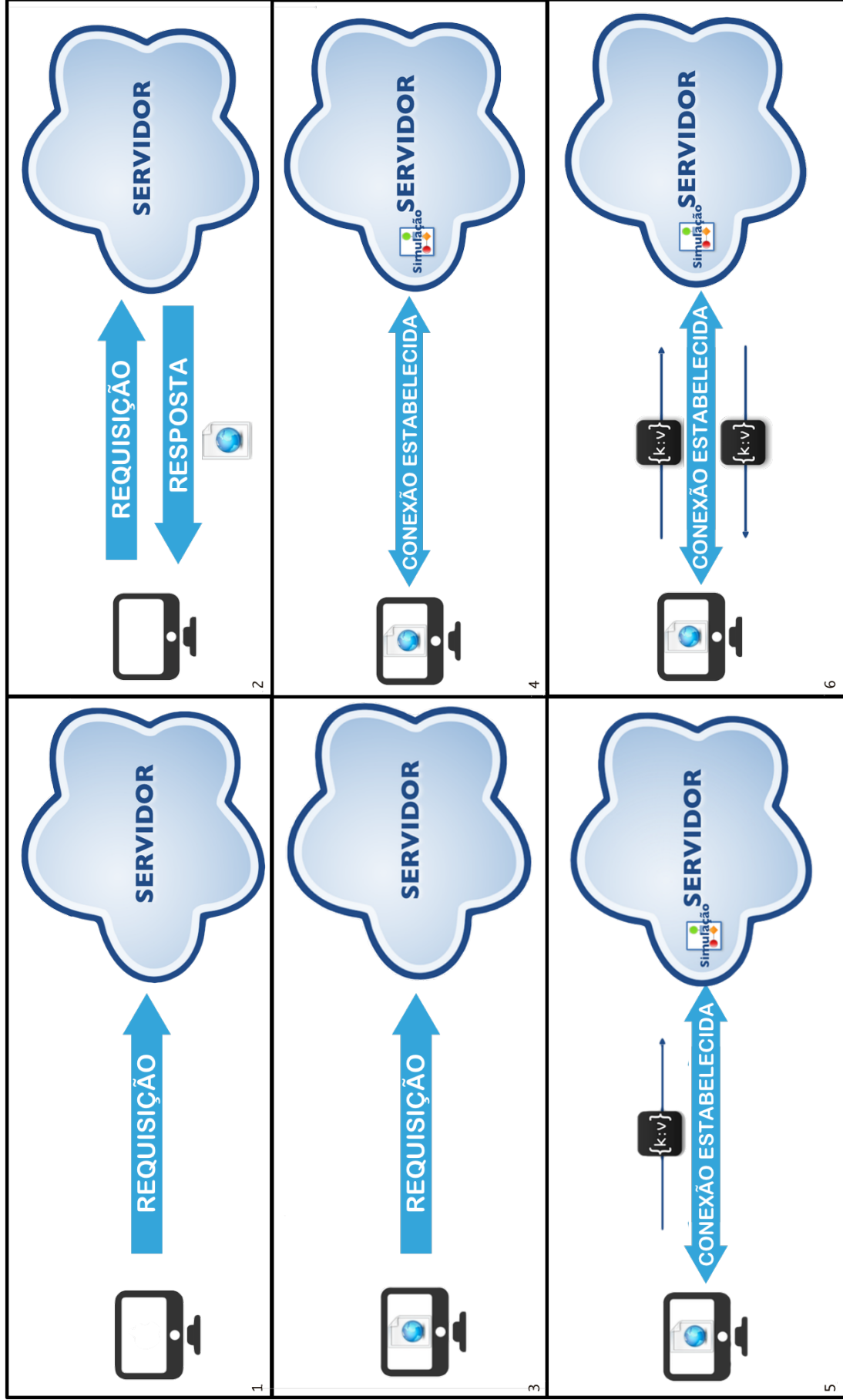
Foram utilizados os conhecimentos gerados para a realização das alterações necessárias no código fonte, sempre de forma a intercalar com a execução dos testes automáticos existentes e a criação de novos testes. Dessa forma, obteve-se um ciclo de *feedback* rápido e contínuo que garantiu que o objetivo final fosse alcançado.

4.2 Desenvolvimento de uma nova arquitetura para viabilizar o acesso ao simulador sem a utilização de *plugins*

Para viabilizar o acesso e o uso do simulador sem a necessidade de um plugin específico foi necessário criar uma nova interface gráfica com o usuário (GUI) utilizando Hypertext Markup Language (HTML) versão 5, a tecnologia padrão para a estruturação e exibição de conteúdo na web. Uma nova arquitetura interna do projeto foi idealizada. A nova versão do simulador não mais executaria a simulação na máquina do usuário. Esta decisão foi tomada por que, apesar de ser possível, não era uma alternativa interessante, por expor completamente a lógica de programação e consequentemente abrir a possibilidade de extravio e pirataria do software.

Um protocolo próprio de troca de mensagens foi idealizado para viabilizar a comunicação entre duas máquinas, de um lado o servidor e, do outro, a máquina do usuário. O protocolo prevê que o servidor é o responsável por realizar os cálculos da simulação e enviar os resultados à máquina do usuário. Esta, através do navegador, possui a responsabilidade de interpretar esses resultados e enviar os comandos que alteram os parâmetros da simulação de volta para o servidor, conforme pode-se ver na Figura 6. Esta troca de mensagens utiliza como meio de comunicação uma tecnologia chamada *WebSockets*, uma ligação direta, de duas vias, entre o navegador do usuário e o servidor de simulação.

Figura 6 – Modelo de comunicação Cliente Servidor com WebSockets



1. Cliente realiza requisição inicial utilizando o protocolo Hypertext Transfer Protocol (HTTP); 2. Servidor responde com uma página HTML com a estrutura inicial; 3. Cliente realiza requisição de conexão para a simulação; 4. Servidor estabelece a conexão via *websocket*; 5. Cliente envia mensagem para o servidor; 6. Servidor envia resposta para o cliente.

Fonte: Elaborado pelo autor

4.2.1 Protocolo de Comunicação

As mensagens trocadas entre o navegador e o simulador utilizam o formato de dados amplamente conhecido Javascript Object Notation (JSON), proposto por Douglas Crockford e padronizado pela ECMA International (ECMA INTERNATIONAL, 2013). Cada mensagem representa um comando a ser executado pelo simulador ou pelo navegador. A mensagem possui obrigatoriamente um atributo **name**, que define o comando realizado, e opcionalmente um atributo **data** que representa parâmetros para a execução do comando. O algoritmo 1 exemplifica a mensagem de início de simulação. Os parâmetros **id**, **mode** e **application** referem-se, respectivamente, a: o identificador único da simulação, definido pelo servidor, o modo de ventilação que será utilizado inicialmente e o nome da aplicação.

Algoritmo 1: Mensagem de início de simulação

```

1 {
2     "name": "start",
3     "data": {
4         "id": "2d5b6819-9d38-fab3-d62a-c704d6a302ca",
5         "mode": "VCV",
6         "application": "xlung"
7     }
8 }
```

4.2.2 Planejamento de melhorias na interface gráfica

Com o objetivo de planejar as melhorias na interface gráfica foi utilizado o software de *prototipação* **Balsamiq Mockups 3.0**, que possui uma biblioteca de protótipos comumente utilizados na criação de interfaces gráficas. Além disso, permite a criação de novos componentes a partir dos já existentes. As vantagens da *prototipação* são o *feedback* rápido e o baixo custo de construção.

Foram utilizados para a construção da interface *web* do simulador: uma adaptação da biblioteca de componentes **Bootstrap** (<http://getbootstrap.com/>), o **Play! Framework** (<https://www.playframework.com/>) para a geração e disponibilização da página HTML no servidor e Javascript como linguagem de programação da GUI no navegador.

4.3 Desenvolvimento de novas funcionalidades

A fim de aumentar o realismo das simulações e a gama de tópicos de aprendizado foram inseridos novos dispositivos de simulação. Destacam-se: o oxímetro de pulso, dispositivos de oxigenoterapia (cânulas nasais, máscaras de Venturi e de reservatório), transtornos metabólicos do paciente, nomograma de equilíbrio ácido-básico, efeitos na troca gasosa causados por variações na altitude e cálculo de parâmetros úteis na ventilação mecânica: o trabalho da respiração, a pressão muscular em 100ms (milissegundos) após o início do ciclo respiratório e o Produto Pressão Tempo (PTP) da pressão muscular ou esforço respiratório. As seções a seguir detalham o funcionamento desses dispositivos de simulação e a metodologia utilizada na sua construção e integração com o simulador.

4.3.1 Oxímetro de Pulso

O oxímetro de pulso é um dispositivo responsável por aferir o percentual de oxiemoglobina no sangue periférico de um paciente através da Saturação periférica de Oxigênio (SpO_2). A frequência cardíaca é outra informação comumente exibida.

O oxímetro de pulso é um dispositivo particularmente útil por não ser invasivo, podendo ser utilizado em áreas do corpo translúcidas como o dedo ou o lóbulo na orelha. A Figura 7 exibe um oxímetro de pulso de dedo disponível comercialmente.

Figura 7 – Oxímetro de pulso de dedo



O número 98 corresponde à SpO_2 em % e o número 83, à Frequência Cardíaca (F.C.) em bpm. A barra vermelha corresponde ao sinal de pulso.

Fonte: <http://hbfisio.commercesuite.com.br/oximetro-de-pulso-de-dedo-choicemed-pr-1213-393794.htm>

A SpO_2 é uma informação já calculada na versão 1.0 do simulador. Sendo necessário apenas inferir a frequência cardíaca (FC) de acordo com a Equação 4.1 a seguir, onde, *coeficienteCardiaco* é uma função que depende da SpO_2 , do sexo e idade do paciente. O Quadro 1 define o coeficiente cardíaco para pacientes do sexo masculino e feminino em diferentes faixas etárias (LHUISSIER; CANOUÏ-POITRINE; RICHALET, 2012). A seguir, a Figura 8 exibe o layout elaborado do oxímetro no simulador.

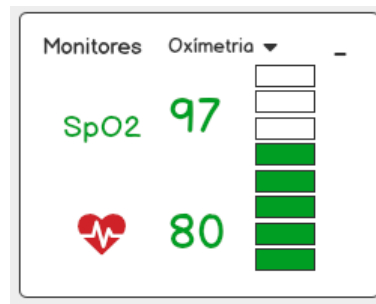
$$FC(SpO_2, paciente) = 80 + [coeficienteCardiaco(paciente) \times (97 - SpO_2)] \quad (4.1)$$

Quadro 1 – Relação coeficiente cardíaco para o sexo e idade do paciente

Idade \ Sexo	Masculino	Feminino
	≤ 30	1.3
$30 < idade \leq 40$	1.2	1.3
$40 < idade \leq 50$	1.1	1.1
$50 < idade \leq 60$	1.07	1.05
caso contrário	1	0.95

Fonte: Elaborado pelo autor

Figura 8 – Planejamento da interface do oxímetro no simulador



O número 97 corresponde à SpO_2 em % e o número 80, à Frequência Cardíaca (F.C.) em bpm. A barra verde corresponde ao sinal de pulso.

Fonte: Elaborado pelo autor.

4.3.2 Dispositivos de oxigenoterapia

A oxigenoterapia consiste na administração de oxigênio numa concentração de pressão superior à encontrada na atmosfera ambiental, para corrigir e atenuar deficiência de oxigênio ou hipóxia, aplicada tanto em situações clínicas agudas quanto crônicas. Para tanto, alguns dispositivos são utilizados para dosar a concentração que o paciente irá receber. No simulador, essa concentração corresponde ao ajuste, Fração Inspirada de Oxigênio, $F_I O_2$.

A administração de O_2 pode ser realizada através de um cateter ou cânula nasal (Figura 9) regulado por um fluxômetro (Figura 10).

Figura 9 – Cateter de oferta de oxigênio



Fonte: Elaborado pelo autor.

Figura 10 – Fluxômetro regulador de oxigênio



Fonte: <http://www.medicalmed.com.br>

A oferta de O_2 utilizando o cateter pode ser regulada pelo usuário do simulador variando de 0 a 5 L/min, onde 0 L/min corresponde a uma $F_I O_2$ de 21%, concentração natural de oxigênio no ar ambiente, e 5 L/min corresponde a aproximadamente 41%. A Figura 11 exibe o leiaute planejado para ser utilizado no simulador.

Figura 11 – Planejamento do leiaute do cateter de oxigênio.

Oxigênio -

= FIO₂ 29%

Cateter nasal de O₂ (L/min)

0
1
2
3
4
5

•

Máscara de Venturi

Máscara de Reservatório

Fonte: Elaborado pelo autor.

Além do cateter, pode ser utilizada a Máscara de Venturi (Figura 12) para a oferta de diferentes concentrações de oxigênio, variando a $F_I O_2$ de 24 a 50%, conforme o conector utilizado.

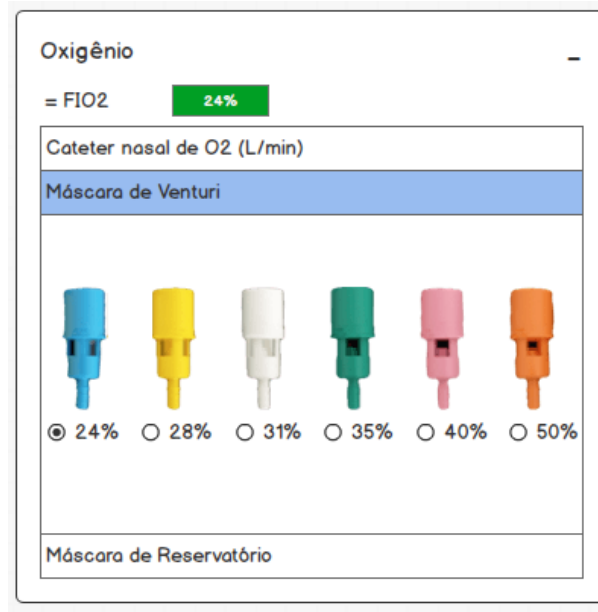
Figura 12 – Máscara de Venturi.



Cada cor do componente de adaptação à máscara corresponde a uma oferta de $F_I O_2$ diferente, variando de 24 a 50%.

Fonte: Elaborado pelo autor.

Figura 13 – Leiaute planejado para a escolha da $F_I O_2$ para a máscara de Venturi.



Fonte: Elaborado pelo autor.

O último dispositivo simulado de oferta de oxigênio é a máscara de reservatório (Figura 14), capaz de ofertar uma $F_I O_2$ aproximadamente de 85 a 90%.

Figura 14 – Máscara de reservatório de O_2 .



Fonte: Elaborado pelo autor.

Figura 15 – Leiaute planejado da Máscara de reservatório de O₂.



Fonte: Elaborado pelo autor.

4.3.3 Transtornos Metabólicos

Para simular os transtornos metabólicos de acidose e alcalose em seus variados graus, foi planejado um menu na configuração do paciente com opções conforme o Quadro 2. Ao selecionar uma das opções do menu o parâmetro *Base Excess* (BE) será alterado e influenciará o HCO_3^- , bem como os parâmetros que dependem dele, gerando uma reação em cadeia, segundo a equação de Henderson-Hasselbalch.

Quadro 2 – Relação entre Transtorno metabólico e *Base Excess*

Transtorno metabólico	<i>Base Excess</i>
Acidose metabólica leve	-5
Acidose metabólica moderada	-10
Acidose metabólica grave	-15
Sem distúrbios metabólicos	0
Alcalose metabólica leve	5
Alcalose metabólica moderada	10
Alcalose metabólica grave	15

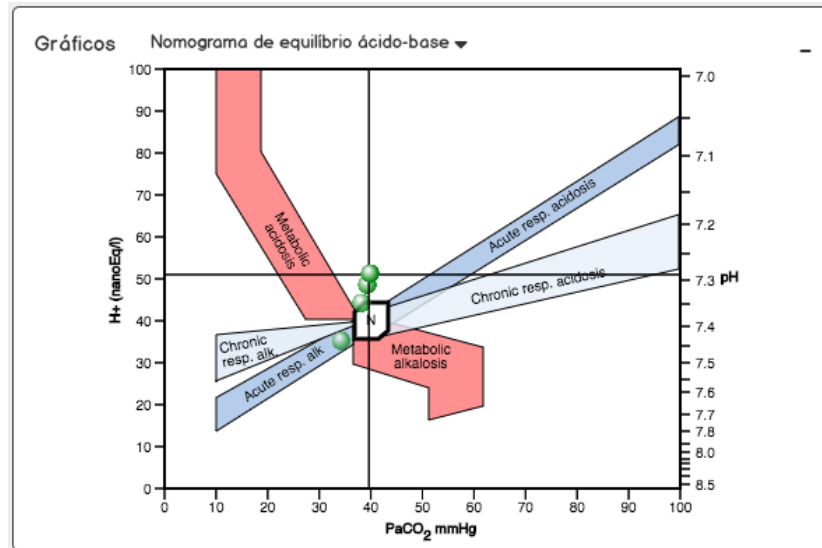
Fonte: Elaborado pelo autor

4.3.4 Nomograma de Equilíbrio Ácido-base

Para visualizar as alterações no PaCO_2 e pH durante o tempo foi introduzido o nomograma de equilíbrio ácido-base. Nele são registrados os 10 últimos valores desses parâmetros.

O gráfico possui ainda, regiões onde é possível visualizar facilmente o estado metabólico do paciente conforme pode ser visto na Figura 16.

Figura 16 – Leiaute planejado do nomograma de equilíbrio ácido-base



Fonte: Elaborado pelo autor.

4.3.5 Variações de altitude

As diversas localizações do planeta possuem altitudes diferentes e por consequência variações na pressão atmosférica. Tais variações devem ser levadas em conta no processo de ventilação mecânica, pois afetam diretamente a PaO_2 e a SaO_2 do paciente. Para demonstrar a importância de tais alterações, foi planejado um menu nas configurações do paciente capaz de alterar a sua localização geográfica, conforme o Quadro 3. Para chegar aos valores de pressão atmosférica foi utilizado a Fórmula Barométrica 4.2 descrita no trabalho U.S. Standard Atmosphere (NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, 1976).

Quadro 3 – Relação entre localização geográficas, altitude e pressão atmosférica

Localização Geográfica	Altitude (m) em relação ao nível do mar	Pressão Atmosférica (mmHg) (valores truncados)
No Mar Morto	-394	796
No nível do mar	0	759
Em São Paulo	760	693
Em Denver	1600	626
Em altitude de cruzeiro num avião	2500	560
Em La Paz	3640	484

Fonte: Elaborado pelo autor

$$P = \begin{cases} P_b \times \left[\frac{T_b}{T_b + L_b \times (h - H_b)} \right]^{\left(\frac{G_0 \times M}{R \times L_b} \right)} & \text{se altitude} < 11000m \\ P_b \times e^x \times \left[\frac{T_b}{T_b + (L_b \times 11000)} \right]^{(G_0 \times M \times R \times L_b)} & \text{se } 11000 \leq \text{altitude} < 20000m \end{cases} \quad (4.2)$$

$$x = \left[\frac{-G_0 \times M \times (h - 11000)}{(R \times T_b) + (L_b \times 11000)} \right] \quad (4.3)$$

Onde:

P_b : Pressão estática (Pascals)

T_b : Temperatura Padrão (K)

G_0 : Aceleração Gravitacional (9.80665 m/s²)

M : Massa molar do ar na terra (0.0289644 kg/mol)

R : Constante universal do gás (8.31432 N·m/(mol·K))

L_b : taxa de lapso a temperatura padrão (K/m) em ISA (Atmosfera Padrão Internacional)

h : Altitude acima do nível do mar (m)

H_b : Altitude na parte de baixo da camada b (m; p. ex. h1 = 11 000 m)

4.3.6 Trabalho da respiração

O trabalho na respiração corresponde à quantidade de energia gasta para inspirar e expirar. É expresso na forma de Joules/L. O trabalho pode ser calculado através da equação $W_b = P \times \Delta V$, onde: P é a pressão aplicada e ΔV é a variação de volume (BEACHEY, 2007). Em estado de repouso, o paciente gasta aproximadamente 5% do consumo total de oxigênio do organismo com os músculos respiratórios. Esse valor pode ser alterado consideravelmente em decorrência de doenças pulmonares.

No simulador, esse parâmetro deve ser exibido junto aos demais medidos pelo ventilador.

4.3.7 Pressão de Oclusão aos 100 ms

A pressão de oclusão aos 100 milissegundos (P01) é um indicador do comando neural ou drive respiratório. É um parâmetro bastante utilizado como fator preditivo de desmame

do paciente. Pode ser aferido através da pressão nas vias aéreas do paciente 100 milissegundos após o início da inspiração.

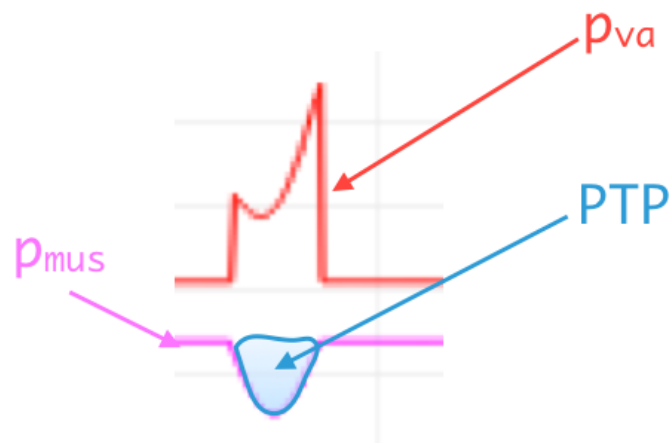
No simulador, esse parâmetro deve ser exibido junto aos demais medidos pelo ventilador.

4.3.8 PTP - Produto Pressão Tempo

O Produto Pressão Tempo (PTP) é uma medida que estima o consumo de oxigênio durante a respiração e o trabalho W_b realizado pela movimentação do diafragma. Esta estimativa pode ser calculada por meio da Equação 4.4 (SASSOON et al., 1991), ou por meio da área da curva rosa na Figura 17.

$$\int p_{mus} dt \quad (4.4)$$

Figura 17 – Curvas de pressão nas vias aéreas e pressão muscular



PTP = Produto pressão tempo medido pela fórmula $\int p_{mus} dt$

Fonte: Elaborado pelo autor.

No simulador, esse parâmetro deve ser exibido junto aos demais medidos pelo ventilador.

4.3.9 f/VC

Índice de respiração rápida e superficial. Parâmetro que relaciona a frequência respiratória em rpm pelo VC em litros. É um índice preditor de sucesso do desmame e extubação.

No simulador, esse parâmetro deve ser exibido junto aos demais medidos pelo ventilador.

5 Resultados

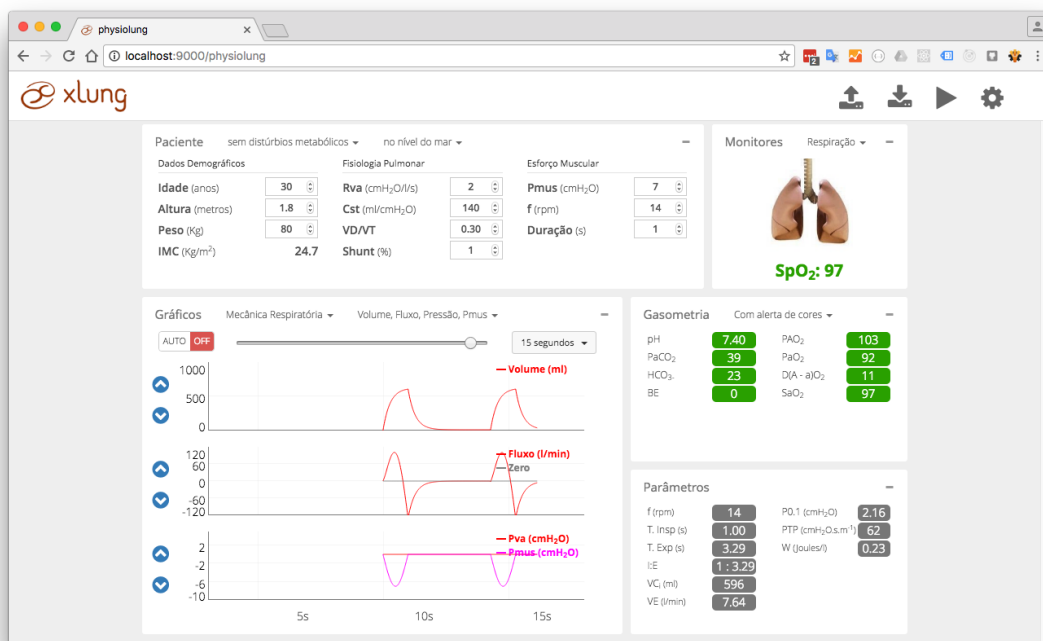
O trabalho de simplificação e reestruturação do simulador Xlung gerou dois subprodutos. Os simuladores Physi lung e Oxylung incrementam a linha de produtos da plataforma Xlung, abrangendo novos públicos, além dos diretamente envolvidos com ventilação mecânica, e, também atingem o objetivo primordial, já que também aprimoram e otimizam a experiência do usuário. As seções seguintes, detalham os resultados de cada um dos produtos.

5.1 Oxylung e Physi lung

Os simuladores Oxylung e Physi lung utilizam o modo de respiração espontânea e surgiram a partir do trabalho de reestruturação e simplificação do Xlung e, por isso, utilizam o mesmo motor de simulação. Foram lançados com um ano de antecedência e são versões voltadas para públicos em estágios diferentes de compreensão da fisiologia respiratória.

O Physi lung (Figura 18) é voltado para o ensino de assuntos relacionados à fisiologia respiratória básica, ensino do mecanismo de controle do equilíbrio ácido-base, e impacto da variação de altitude sobre os gases arteriais.

Figura 18 – Tela principal do simulador Physi lung



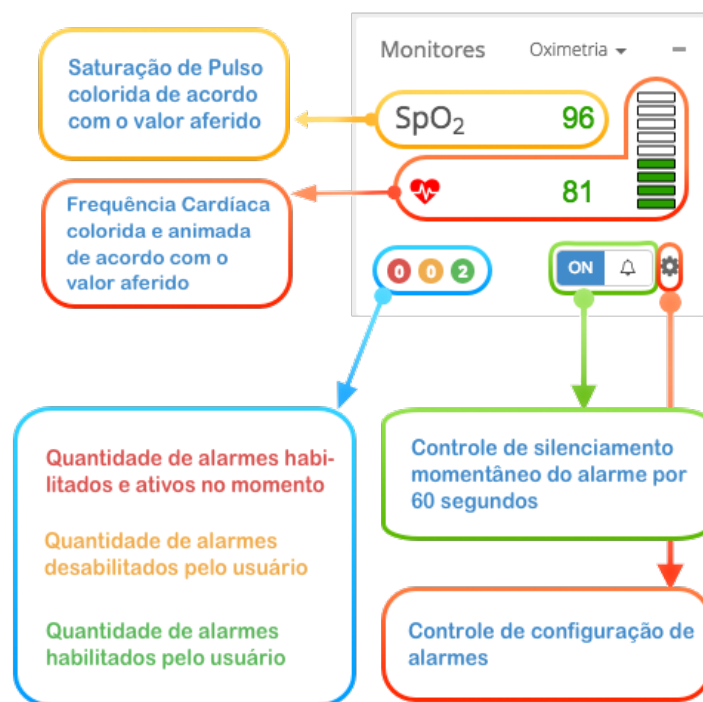
As figuras a seguir destacam os novos dispositivos de simulação introduzidos com o Physiolog: o oxímetro de pulso (Figura 19, Figura 20 e Figura 21), os distúrbios metabólicos do paciente (Figura 22), o nomograma ácido-base (Figura 23) para sua visualização e a variação de altitude (Figura 24). Apesar de não ser possível reproduzir através de imagens e texto, a introdução do oxímetro vem acompanhada do seu som característico, um *beep* repetido responsivo à frequência cardíaca e que sofre interferência no tom, de acordo com as variações da saturação de oxigênio no sangue.

Figura 19 – Oxímetro de pulso de dedo vs. Oxímetro Simulado



Fonte: Elaborado pelo autor

Figura 20 – Oxímetro de pulso detalhado



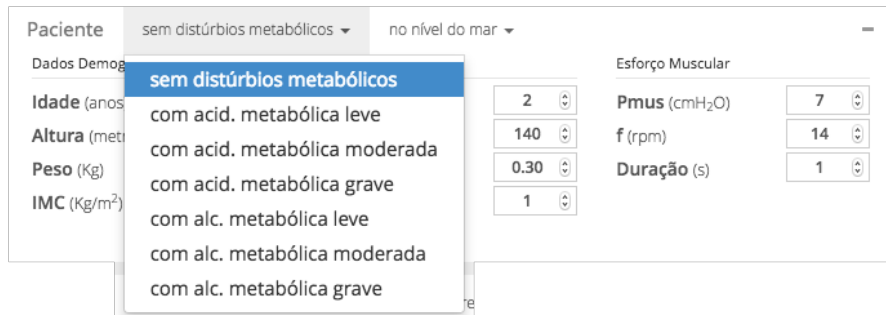
Fonte: Elaborado pelo autor

Figura 21 – Oxímetro de pulso - Configuração dos alarmes vs. alarmes disparando



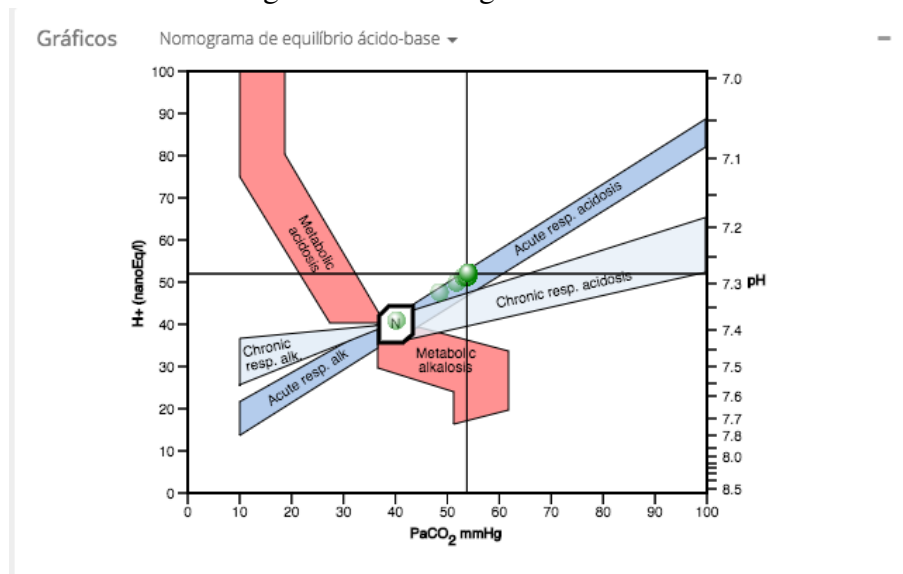
Fonte: Elaborado pelo autor

Figura 22 – Menu "Distúrbios Metabólicos" do Paciente



Fonte: Elaborado pelo autor

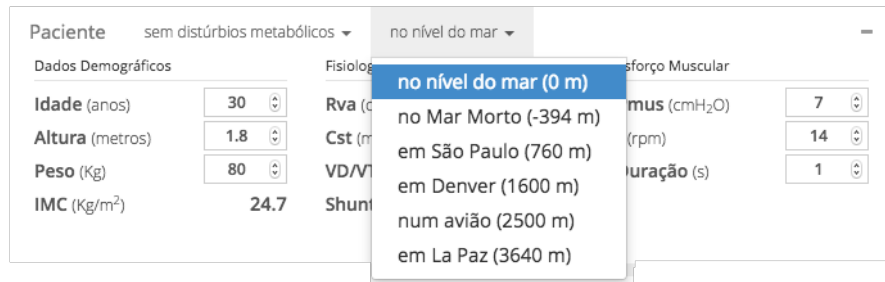
Figura 23 – Nomograma ácido-base



O círculo verde relaciona os valores atuais do pH (direita) e $PaCO_2$ (abaixo).

Fonte: Elaborado pelo autor

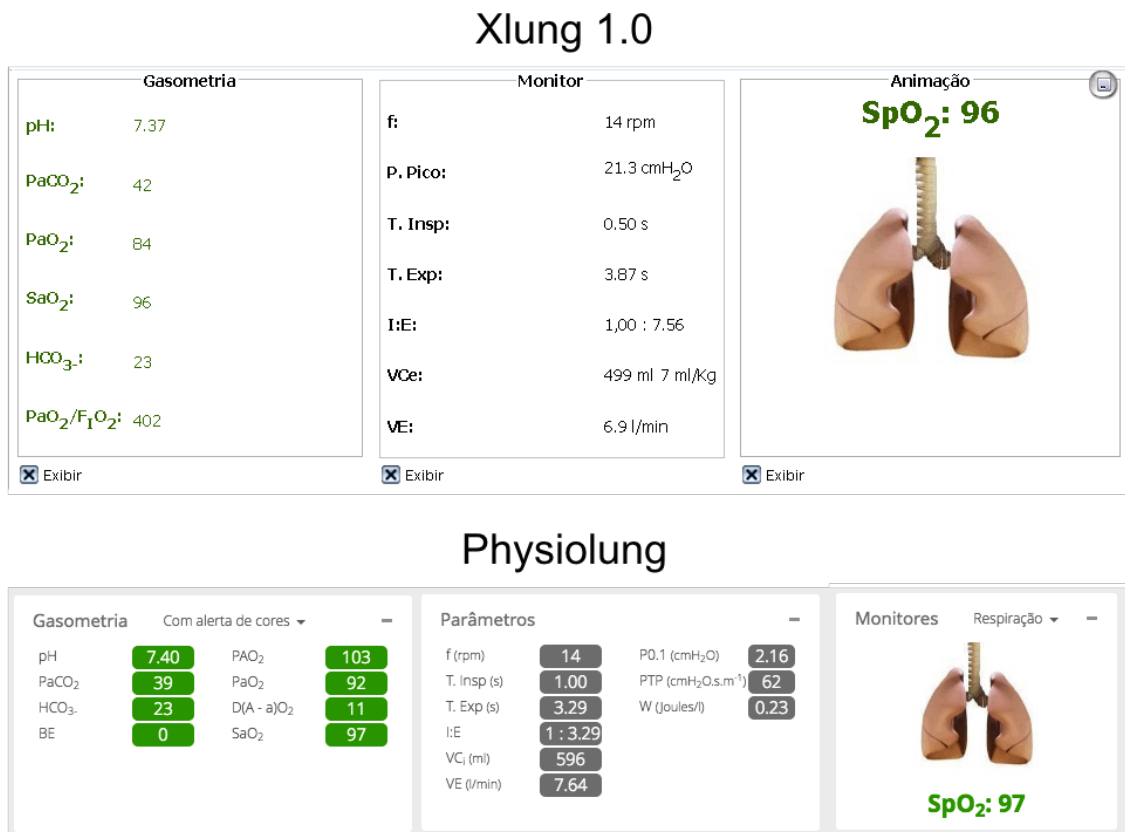
Figura 24 – Menu "Variação de Altitude" do Paciente



Fonte: Elaborado pelo autor

Além dos mecanismos adicionados, foram reestruturados os leiautes dos componentes de monitorização do paciente. A Figura 25 faz uma comparação dos componentes de monitorização do Xlung 1.0 vs. Physiolog.

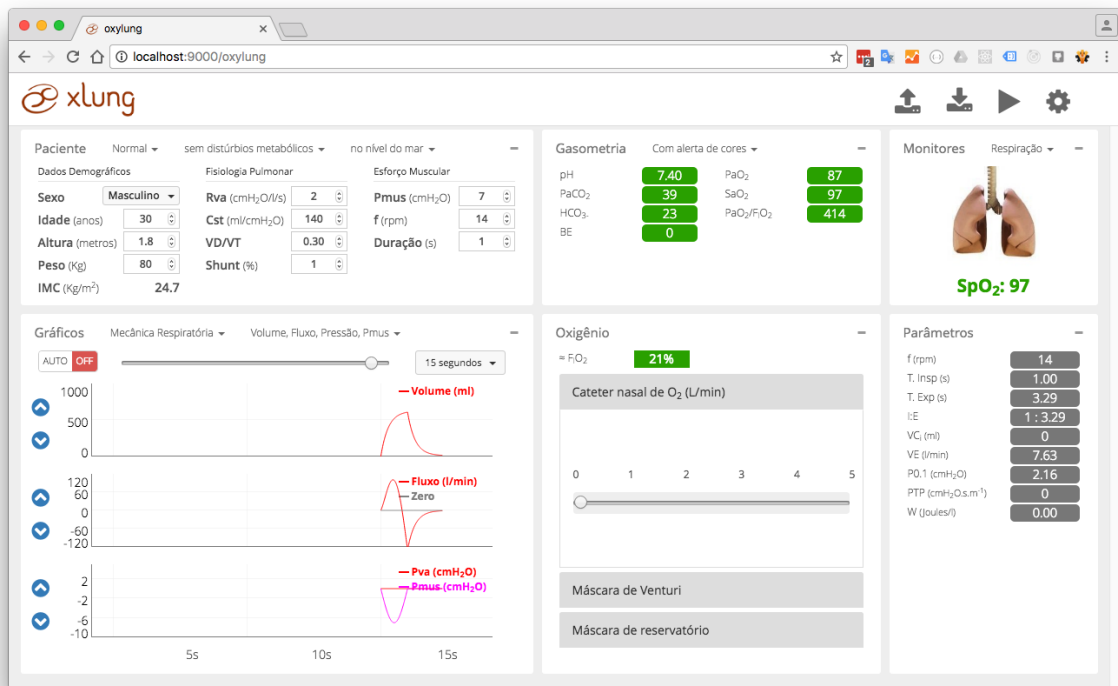
Figura 25 – Comparativo dos componentes de monitorização do paciente no Xlung 1.0 vs. Physiolog



Fonte: Elaborado pelo autor

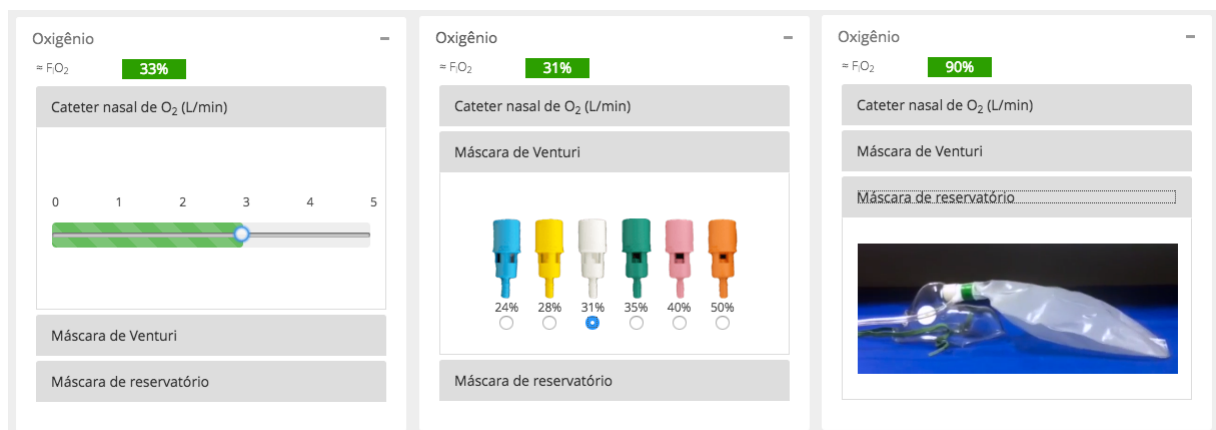
O Oxylung (Figura 26) adiciona à base de recursos do Physiolog os dispositivos de oxigenoterapia (Figura 27) e o conceito de doenças respiratórias (Figura 28).

Figura 26 – Tela principal do simulador Oxylung.



Fonte: <https://xlung.net>

Figura 27 – Dispositivos de oxigenoterapia no Oxylung.



Fonte: Elaborado pelo autor

Figura 28 – Menu "Condições Clínicas" do paciente no Oxylung.

The image shows a software interface for configuring a patient in the Oxylung simulator. A dropdown menu for 'Paciente' is open, listing several clinical conditions: 'Normal' (highlighted in blue), 'DPOC', 'Asma', 'SARA', and 'Personalizado'. The background interface includes several sections:

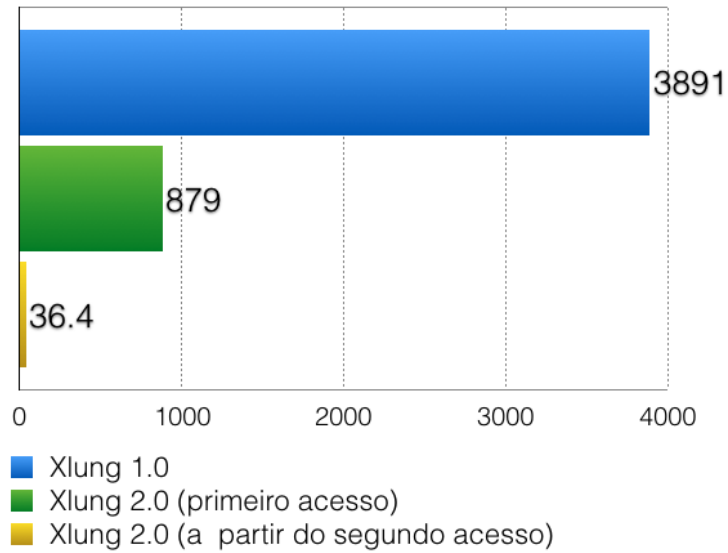
- Paciente:** A dropdown menu currently set to 'Normal'.
- sem distúrbios metabólicos:** A dropdown menu.
- no nível do mar:** A dropdown menu.
- Dados Demográficos:** Fields for 'Sexo', 'Idade (anos)', 'Altura (metr)', 'Peso (Kg)', and 'IMC (Kg/m²)'.
- Esforço Pulmonar:** A section with several numerical input fields with up/down arrows, including values like 2, 140, 0.30, and 1.
- Esforço Muscular:** A section with numerical input fields for 'Pmus (cmH₂O)' (7), 'f (rpm)' (14), and 'Duração (s)' (1).

Fonte: Elaborado pelo autor

5.2 Xlung 2.0

A nova versão do simulador Xlung não depende mais do JRE ou de qualquer *plugin* instalado na máquina do usuário para funcionar. O novo simulador utiliza uma arquitetura cliente-servidor. A simulação de fato ocorre num servidor remoto enquanto o usuário recebe todos os resultados em sua máquina e altera os parâmetros da simulação se assim quiser. Essa separação fez com que o download necessário, 879 KB na primeira execução, para a realização da simulação seja 4,43 vezes menor comparado ao anterior 3.891 KB (3,8 MB). A partir do segundo acesso, essa diferença aumenta ainda mais, pois o navegador do usuário percebe que o conteúdo a ser baixado já está disponível no computador e aquele transferido é de apenas 36,4 KB ou, aproximadamente, 107 vezes menor que o original, conforme pode ser visualizado na Figura 29.

Figura 29 – Transferência de dados (KB) necessária para iniciar uma simulação no Xlung 2.0 vs. Xlung 1.0

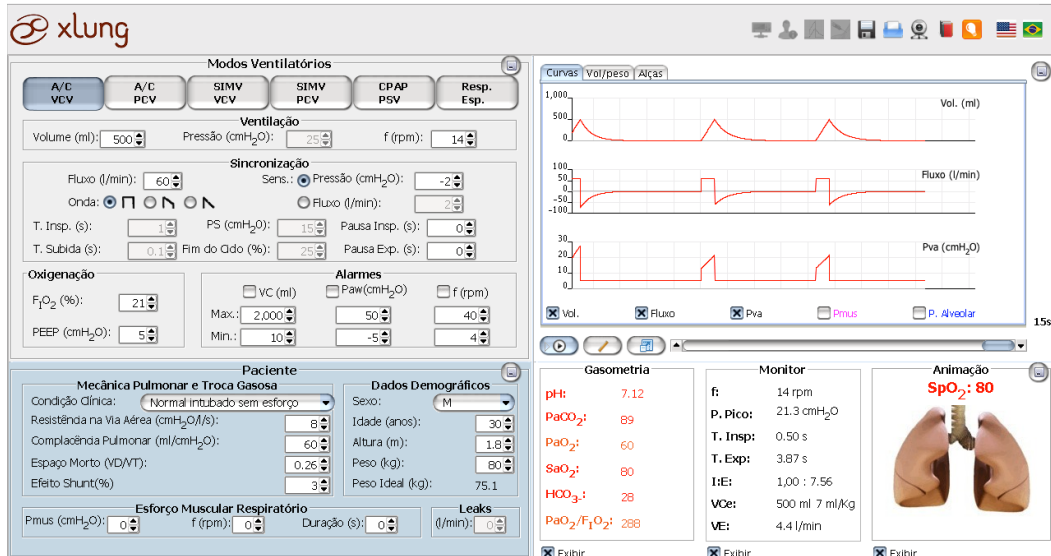


Fonte: Elaborado pelo autor.

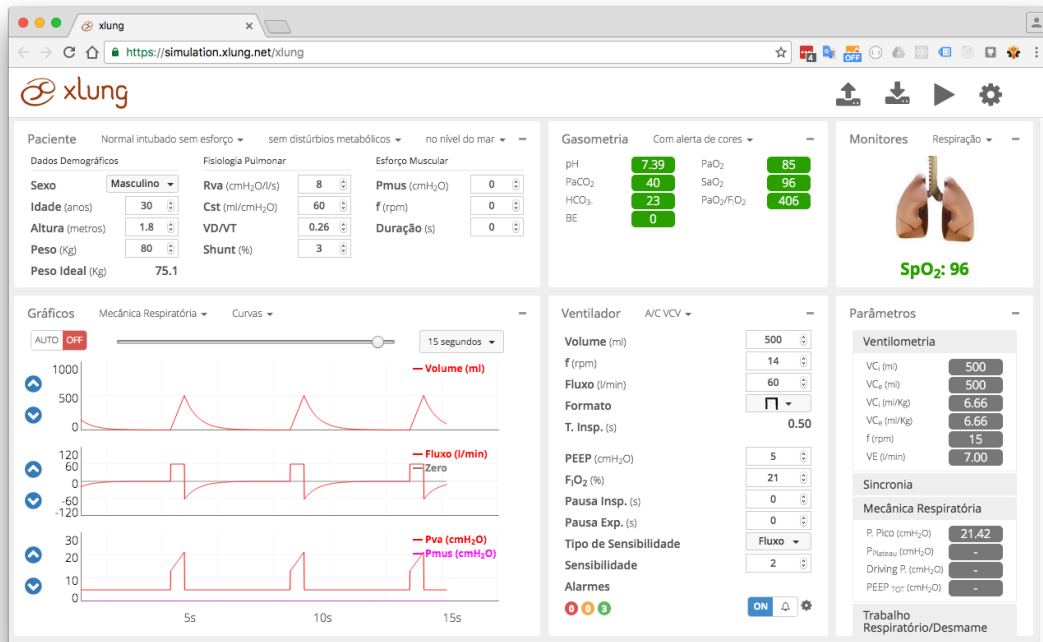
O Xlung 2.0 conta com um novo leiaute (Figura 30). Foi lançado na Plataforma Xlung em 9 de Junho de 2016 e apresentado em Simpósio Satélite no dia seguinte, no V Congresso Luso Brasileiro de Medicina Intensiva realizado em Recife, Pernambuco.

Figura 30 – Comparativo da tela principal Xlung 1.0 vs Xlung 2.0

Xlung 1.0



Xlung 2.0

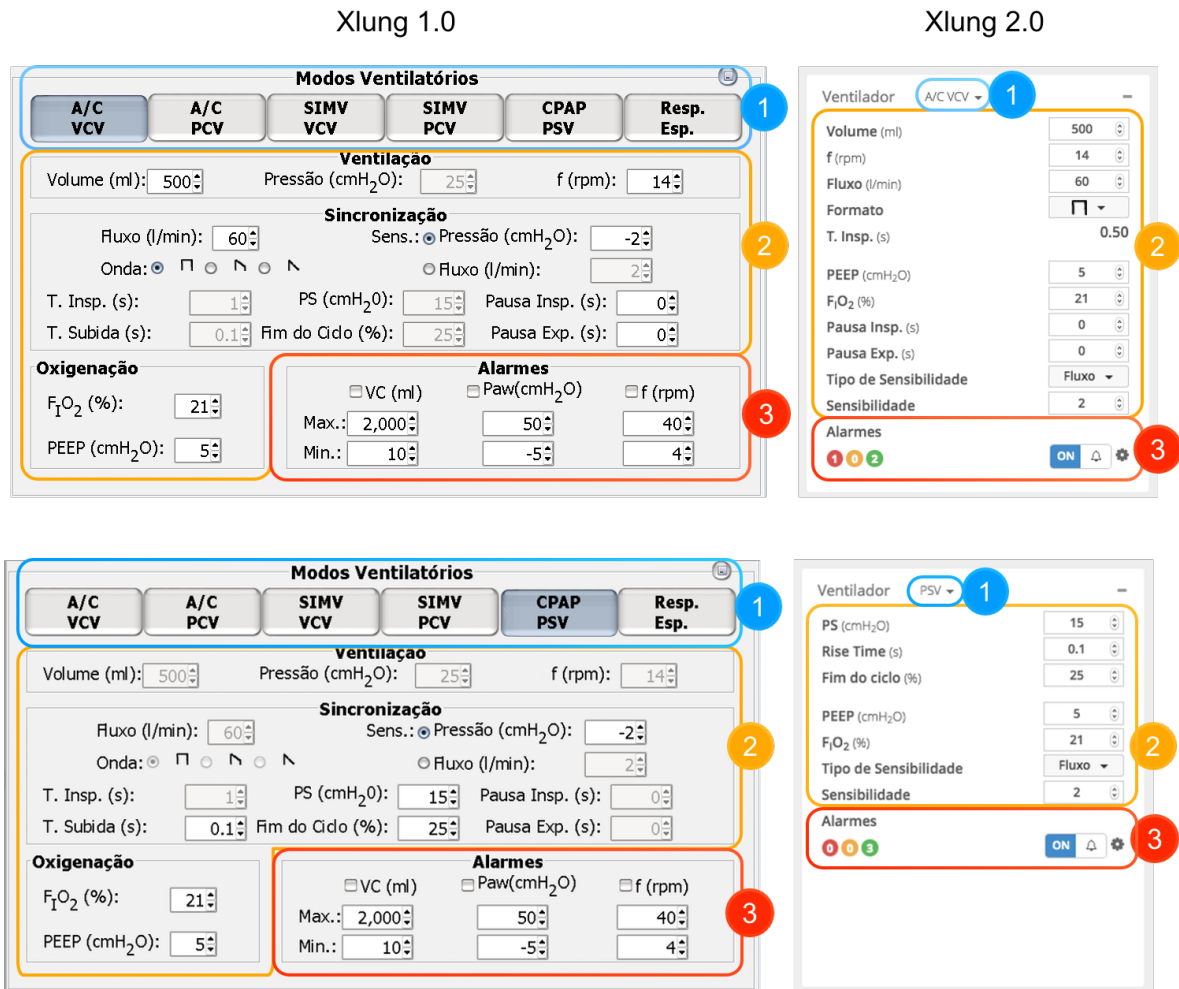


Fonte: <https://xlung.net>

A nova interface objetiva uma reorganização da informação e redução de distrações visíveis na tela. A Figura 31 coloca, lado a lado, a interface do ventilador no Xlung 1.0 (esquerda) e no Xlung 2.0 (direita). A reorganização posicionou, por exemplo, os parâmetros na vertical, com o intuito de facilitar a busca das informações. De forma semelhante, a informação de quantos alarmes estão ativos no momento pode ser visualizada por meio do código de cores (vermelho:

problema, amarelo: alerta e verde: normal), além de poderem ser silenciados, momentaneamente, com apenas um clique.

Figura 31 – Tela de configuração do ventilador nos modo ventilatórios VCV e PSV, Xlung 1.0 vs. Xlung 2.0



1. Os botões de seleção do modo ventilatório foram trocados por um menu de opções.
2. Apresentação dos ajustes específicos do modo ventilatório selecionado;
3. Apresentação dos alarmes do ventilador.

Fonte: Elaborado pelo autor.

A Figura 32 apresenta a um comparativo da tela de gráficos, Xlung 1.0 (esquerda) vs. Xlung 2.0 (direita). De forma semelhante, a Figura 33 apresenta um comparativo dos gráficos de volume ponderal presentes no Xlung 1.0 (esquerda) vs. Xlung 2.0 (direita).

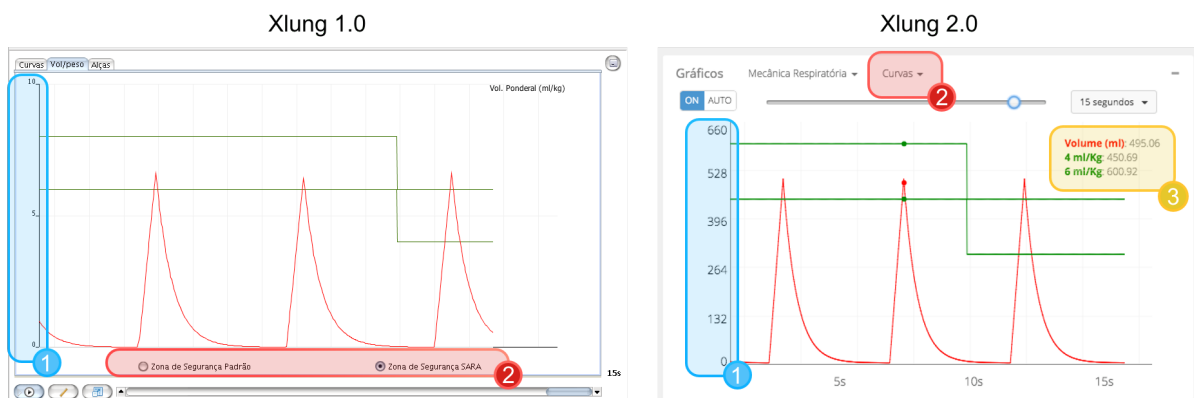
Figura 32 – Tela de gráficos, Xlung 1.0 vs. Xlung 2.0



1. Os botões de exibição de curvas foram agrupados em um menu de opções.
2. A escala automática foi refeita de forma a funcionar constantemente para aumentar o nível de detalhes mesmo com o passar do tempo.
3. A escala manual está ao alcance do usuário quando a escala automática está desligada. Podendo ser alterada individualmente em cada gráfico.
4. A janela de tempo visualizada pode ser alterada pelo usuário sem sair da área de gráficos. No novo layout o gráfico possui intervalos de tempo para localização no eixo do tempo.

Fonte: Elaborado pelo autor.

Figura 33 – Comparativo dos gráficos de Volume Ponderal com Zonas de Segurança presentes no Xlung 1.0 vs. Xlung 2.0



1. A nova escala utiliza a mesma unidade de medida do gráfico de volume (ml) presente nos ventiladores comuns.
2. As opções: Zonas de Segurança Padrão, Zona de Segurança SARA e Nenhuma Zona de Segurança foram transferidas para o menu junto das demais curvas.
3. Com a simulação em pausa, os valores exatos de cada uma das curvas são exibidos de acordo com o local em que o mouse esteja apontando.

Fonte: Elaborado pelo autor.

Os quadros 4 e 5 apresentam um comparativo do conjunto de funcionalidades

disponíveis nos três simuladores desenvolvidos.

Quadro 4 – Comparativo dos simuladores em relação às seguintes funcionalidades: alterações clínicas, alterações metabólicas, nomograma de equilíbrio ácido-base, variação de altitude, dispositivos de oxigenoterapia e oxímetro de pulso com alarmes.

Funcionalidade	Simulador		
	Physiolung	Oxylung	Xlung 2.0
Alterações Clínicas			
Normal intubado sem esforço			✓
Normal intubado com esforço			✓
Normal não intubado			✓
Normal		✓	
DPOC		✓	✓
Asma		✓	✓
SARA		✓	
SARA leve			✓
SARA moderada			✓
SARA grave			✓
Alterações Metabólicas	✓	✓	✓
Nomograma de equilíbrio ácido-base	✓	✓	✓
Variação de Altitude	✓	✓	✓
Dispositivos de Oxigenoterapia		✓	
Oxímetro de pulso com alarmes	✓	✓	✓

Fonte: Elaborado pelo autor

Quadro 5 – Comparativo dos simuladores em relação às seguintes funcionalidades: pulmão animado, gasometria arterial, parâmetros de monitorização, modos ventilatórios e gráficos.

Funcionalidade	Simulador		
	Physiolung	Oxylung	Xlung 2.0
Pulmão Animado	✓	✓	✓
Gasometria Arterial			
pH	✓	✓	✓
PaCO ₂	✓	✓	✓
HCO ₃ ⁻	✓	✓	✓
BE	✓	✓	✓
PAO ₂	✓		
PaO ₂	✓	✓	✓
D(A - a)O ₂	✓		
SaO ₂	✓	✓	✓
PaO ₂ /F ₁ O ₂		✓	✓
Parâmetros de Monitorização			
f (rpm)	✓	✓	✓
T. Insp (s)	✓	✓	✓
T. Exp (s)	✓	✓	✓
T _{TOT} (s)			✓
I:E	✓	✓	✓
VC _i (ml)	✓	✓	✓
VC _i (ml/Kg)			✓
VC _e (ml)			✓
VC _e (ml/Kg)			✓
VE (l/min)	✓	✓	✓
P. Pico (cmH ₂ O)			✓
P. Plateau (cmH ₂ O)			✓
Driving Pressure (cmH ₂ O)			✓
PEEP (cmH ₂ O)			✓
P0.1 (cmH ₂ O)	✓	✓	✓
PTP (cmH ₂ O.s.m ⁻¹)	✓	✓	✓
W (Joules/l)	✓	✓	✓
f/VC (rpm/l)			✓
Modos Ventilatórios			
Respiração Espontânea	✓	✓	✓
A/C VCV			✓
A/C PCV			✓
PSV			✓
CPAP			✓
Gráficos			
Volume Sem Zona de Segurança (ml)	✓	✓	✓
Volume Com Zona de Segurança Padrão (ml)			✓
Volume Com Zona de Segurança SARA (ml)			✓
Fluxo (l/min)	✓	✓	✓
Pressão nas vias aéreas (cmH ₂ O)	✓	✓	✓
Pressão alveolar (cmH ₂ O)	✓	✓	✓
Pressão muscular (cmH ₂ O)	✓	✓	✓

Fonte: Elaborado pelo autor

5.3 Registros de Software

Este trabalho gerou três registros de software junto ao Instituto Nacional de Propriedade Industrial (INPI) listados a seguir no Quadro 6.

Quadro 6 – Registros de software no INPI

Software	Nº do protocolo de registro	Data de entrada no INPI
Physiolung	BR 51 2016 001287 0	04/10/2016
Oxylung	BR 51 2016 001289 7	04/10/2016
Xlung	BR 51 2017 000056 5	01/02/2017

Fonte: Elaborado pelo autor

6 Discussões

O desenvolvimento dos novos simuladores Physiolog, Oxylung e Xlung 2.0 ocorreu em tempo hábil para o seu lançamento como produtos da Plataforma Xlung e para os registros de propriedade intelectual. Apesar de não ter sido previsto inicialmente, o desenvolvimento do Physiolog e Oxylung surgiu como uma etapa natural do processo de reestruturação e atualização do Xlung 1.0.

6.1 Diferenças percebidas

O Physiolog se diferencia do Xlung 1.0 por seu novo leiaute e por não ter a presença do ventilador mecânico, o que era considerado uma dificuldade inicial para usuários sem conhecimento do assunto. Com isso foi possível focar no assunto fisiologia respiratória básica sem antes ter que explicar todo funcionamento do simulador de VM.

O Oxylung também se diferencia do Xlung 1.0 pelo novo leiaute, por não conter um ventilador mecânico e por adicionar os dispositivos de oxigenoterapia além das funcionalidades presentes no Physiolog.

O Xlung 2.0 possui como principais diferenças para o usuário: a não necessidade de instalação do Java Runtime Environment (JRE) ou qualquer outro *plugin*, novo leiaute e novas funcionalidades.

A utilização da JRE permitiu que o projeto fosse inicialmente viabilizado mas, causava um grande número de chamados técnicos para solução de problemas relacionados ao *plugin*, além de causar uma má impressão no usuário. O novo leiaute, visou despoluir as informações por meio de reorganização das informações principais e remoção das informações que não são necessárias naquele instante da simulação. A velocidade de carregamento é muito superior à da versão anterior por ter menos dados a serem transferidos, especialmente a partir da segunda simulação (aproximadamente 107 vezes menos). Além dessa diferença, por utilizar HTML como tecnologia base para construção da interface gráfica, o simulador pode exibir uma tela de carregamento tão logo a página seja transferida (em menos de um segundo) enquanto o restante do programa é transferido. Dessa forma, o usuário tem uma percepção de que o simulador já está em execução, mesmo que o usuário ainda não possa interagir com ele.

6.2 Limitações

Não houve tempo hábil para uma validação de usabilidade do produto. Além disso, algumas funcionalidades do Xlung 1.0 ainda não foram implementadas no Xlung 2.0 como por exemplo a simulação de vazamentos além dos gráficos Fluxo \times Volume e Pressão \times Volume.

6.3 Implicações práticas para o ensino da ventilação mecânica e da fisiologia respiratória

Após o lançamento dos novos simuladores observa-se uma perspectiva de aumentar a ubiquidade do ensino da VM no Brasil e em outros países.

Acompanhando métricas de acesso e uso do simulador verificou-se que os mesmos obtiveram 5.682 acessos em 3 meses, média de 1.894 acessos/mês, distribuídos em 66 países em todos os continentes.

6.4 Perspectivas Futuras

Como perspectiva futura é provável a criação de um módulo de exercícios de simulação virtual online, incorporado como uma nova funcionalidade agregando valor à simulação virtual. Este produto já é foco de pesquisa e desenvolvimento da empresa Xlung e do Programa de Pós-Graduação da Unichristus.

São consideradas ainda como oportunidades de trabalho futuros, a validação de usabilidade dos simuladores, além do desenvolvimento de algumas funcionalidades da versão anterior do Xlung, que ainda não foram migradas para a versão 2.0.

7 Conclusões

Ao término deste trabalho:

- a) Foi desenvolvida a versão 2.0 do Simulador Virtual de Ventilação Mecânica Xlung;
- b) Testes automáticos de software foram introduzidos nos principais componentes do simulador;
- c) A nova arquitetura permite que os usuários do simulador acessem o mesmo sem a necessidade de instalação de *plugins*;
- d) Novas funcionalidades foram desenvolvidas;
- e) Dois outros simuladores correlacionados ao ensino de ventilação mecânica foram desenvolvidos: o Physiolung e o Oxylung.

Referências

- AVRAM, A. **Firefox Will No Longer Support Plug-ins Except for Flash**. 2015. Disponível em: <<http://www.infoq.com/news/2015/10/firefox-npapi>>. Acesso em: 28 de Outubro de 2015.
- BARBAS, C. S. V.; ÍSOLA, A. M.; FARIAS, A. M. de C.; CAVALCANTE, A. B.; GAMA, A. M. C.; DUARTE, A. C. M. et al. Recomendações brasileiras de ventilação mecânica 2013. parte i. **Revista Brasileira de Terapia Intensiva**, v. 26, p. 89 – 121, 2014.
- BEACHEY, W. **Respiratory Care Anatomy and Physiology: Foundations for clinical Practice**. [S.l.]: MOSBY Elsevier Inc., 2007. ISBN 978-0-323-02740-3.
- COHN, M. **Succeeding with Agile: Software Development Using Scrum**. [S.l.]: Addison-Wesley Professional, 2009. ISBN 978-0321579362.
- COX, C. E.; CARSON, S. S.; ELY, E. W. et al. Effectiveness of medical resident education in mechanical ventilation. **American Journal of Respiratory and Critical Care Medicine**, v. 167, p. 32 – 38, 2003.
- DALLAL, J. A. Identifying refactoring opportunities in object-oriented code: A systematic literature review. **Information and Software Technology**, Elsevier, v. 58, p. 231 – 249, Fevereiro 2015.
- ECMA INTERNATIONAL. **The JSON Data Interchange Format**. 2013. Disponível em: <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>>. Acesso em: 2 de Setembro de 2015.
- ESTEBAN, A.; ANZUETO, A. et al. Characteristics and outcomes in adult patients receiving mechanical ventilation a 28-day international study. **JAMA - The Journal of American Medical Association**, v. 287, p. 345 – 355, 2002.
- FOWLER, M.; BECK, K.; BRANT, J.; OPDYKE, W.; ROBERTS, D. **Refactoring – Improving the design of existing code**. [S.l.]: Addison-Wesley Professional, 1999. v. 1.
- GIRARD, T. D.; KRESS, J. P.; FUCHS, B. D. et al. Efficacy and safety of a paired sedation and ventilator weaning protocol for mechanically ventilated patients in intensive care (awakening and breathing controlled trial): a randomised controlled trial. **Lancet**, v. 371, p. 126 – 134, 2008.
- GUSMÃO, G. **Nova versão do Chrome é a primeira a desativar plugins Java e Silverlight por padrão**. Editora Abril, 2015. Disponível em: <<http://info.abril.com.br/noticias/tecnologia-pessoal/2015/04/nova-versao-do-chrome-e-a-primeira-a-desativar-plugins-java-e-silverlight-por-padrao.shtml>>. Acesso em: 22 de Setembro de 2015.
- HOLANDA, M. A. **Modos Ventilatórios Básicos**. 2014. Disponível em: <<https://xlung.net/manual-de-vm/modos-ventilatorios-basicos>>. Acesso em: 10 de Fevereiro de 2016.
- HOLANDA, M. A.; FILHO, M. L.; REIS, R. C. et al. Efficacy of a computer simulator program (xlung™) for teaching mechanical ventilation in complex clinical scenarios to medical students. In: **American Thoracic Society International Conference**. San Francisco: American Journal of Respiratory and Critical Care Medicine, 2012.
- JOBS, S. **Thoughts on Flash**. Apple Inc., 2010. Disponível em: <<http://www.apple.com/hotnews/thoughts-on-flash/>>. Acesso em: 16 de Setembro de 2015.

KARCZ, M.; VITKUS, A.; PAPADAKOS, P. J.; SCHWAIBERGER, D.; LACHMANN, B. State-of-the-art mechanical ventilation. **Journal of Cardiothoracic and Vascular Anesthesia**, v. 26, p. 486 – 506, 2012.

LHUISSIER, F. J.; CANOUÏ-POITRINE, F.; RICHALET, J.-P. Ageing and cardiorespiratory response to hypoxia. **The Journal of Physiology**, The Physiology Society, v. 590.21, p. 5461 – 5474, 2012.

LINO, A. J.; GOMES, C. G.; SOUSA, C. N. D. S. V.; CARVALHO, K. A.; DINIZ, B. M. E.; JUNIOR, B. A. V.; HOLANDA, A. M. A critical review of mechanical ventilation virtual simulators: Is it time to use them? **JMIR Med Educ**, v. 2, n. 1, p. e8, Jun 2016. Disponível em: <<http://mededu.jmir.org/2016/1/e8/>>.

MARTIN, R. C. **The Dependency Inversion Principle**. 1996. Disponível em: <<http://www.objectmentor.com/resources/articles/dip.pdf>>. Acesso em: 23 de Setembro de 2015.

MARTIN, R. C. **The Principles of OOD**. 2005. Disponível em: <<http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>>. Acesso em: 23 de Setembro de 2015.

MENS, T.; TOURWÉ, T. A survey of software refactoring. **IEEE Transactions on Software Engineering**, IEEE, v. 30, p. 126 – 139, 2004.

MINISTÉRIO DA EDUCAÇÃO. **Resumo Técnico Censo da Educação Superior 2013**. 2013. Disponível em: <http://download.inep.gov.br/download/superior/censo/2013/resumo_tecnico_censo_educacao_superior_2013.pdf>. Acesso em: 05 de Setembro de 2016.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. U.S. Standard Atmosphere . 1976.

NETFLIX. **HTML5 Video at Netflix**. 2013. Disponível em: <<http://techblog.netflix.com/2013/04/html5-video-at-netflix.html>>. Acesso em: 22 de Setembro de 2015.

ORACLE. **Applets**. 2015. Disponível em: <<http://www.oracle.com/technetwork/java/applets-137637.html>>. Acesso em: 29 de Setembro de 2015.

ORACLE. **Java e o Browser Google Chrome**. 2015. Disponível em: <https://www.java.com/pt_BR/download/faq/chrome.xml>. Acesso em: 22 de Setembro de 2015.

ORACLE. **Moving to a Plugin-Free Web**. 2016. Disponível em: <https://blogs.oracle.com/java-platform-group/entry/moving_to_a_plugin_free>. Acesso em: 10 de Fevereiro de 2016.

QUILICI, A. P.; ABRÃO, K. C.; TIMERAM, S.; GUTIERREZ, F. et al. **Simulação clínica: do conceito à aplicabilidade**. [S.l.]: Editora Atheneu, 2012.

SASSOON, C. S. H.; LIGHT, R. W.; LODIA, R.; SIECK, G. C.; MAHUTTE, C. K. Pressure-time product during continuous positive airway pressure, pressure support ventilation, and t-piece during weaning from mechanical ventilation. **American Review of Respiratory Disease**, ATS Journals, v. 143, n. 3, p. 469 – 475, 1991.

SINDICATO DOS ENGENHEIROS NO DISTRITO FEDERAL. **Prêmio Brasil de Engenharia 2011 – Engenharia para todos**. 2011. Disponível em: <<http://tpqb.eq.ufrj.br/download/premio-brasil-de-engenharia-2011-mencao-honrosa.pdf>>. Acesso em: 23 de Setembro de 2015.

SOMMERVILLE, I. **Engenharia de Software**. [S.l.]: Pearson Adison Wesley, 2007. v. 8.

WALDROP, M. M. Campus 2.0. massive open online courses are transforming higher education — and providing fodder for scientific research. **Nature**, Macmillan Publishers Limited, v. 495, p. 160 – 163, 2013.

Apêndices

APÊNDICE A – Dispensa do termo de consentimento

Nós, Marcelo Emanuel Bezerra Diniz, Andrea Kelly Carvalho, Marcelo Alcantara Holanda, pesquisadores do Centro Universitário Christus, solicitamos a dispensa do TCLE na pesquisa intitulada DESENVOLVIMENTO DA VERSÃO 2.0 DO SIMULADOR VIRTUAL DE VENTILAÇÃO MECÂNICA XLUNG em razão de não necessitar o envolvimento direto ou indireto de terceiros e não expor ou necessitar de material de natureza confidencial.

Fortaleza, 8 de março de 2017

Marcelo Emanuel Bezerra Diniz

Dr. Marcelo Alcantara Holanda

Dra. Andréa Kelly Carvalho