

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO**

**Sistema de Treinamento Baseado em Computador para os Profissionais da
Área da Saúde que atuam em
Unidade de Terapia Intensiva Neonatal**

Daniel Rocha Gualberto

Itajubá, novembro de 2016

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO**

Daniel Rocha Gualberto

**Sistema de Treinamento Baseado em Computador para os Profissionais da
Área da Saúde que atuam em
Unidade de Terapia Intensiva Neonatal**

Dissertação submetida ao Programa de Pós-Graduação em
Ciência e Tecnologia da Computação como parte dos requisitos
para obtenção do Título de Mestre em Ciência e Tecnologia da
Computação

Área de Concentração: Matemática Computacional

Orientador: Prof^o. Dr. Alexandre Carlos Brandão Ramos

Coorientadora: Prof^a. Dr^a. Renata Aparecida Ribeiro Custódio

**Novembro de 2016
Itajubá - MG**

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO**

Daniel Rocha Gualberto

**Sistema de Treinamento Baseado em Computador para os Profissionais da
Área da Saúde que atuam em
Unidade de Terapia Intensiva Neonatal**

Dissertação aprovada por banca examinadora em 25 de novembro de 2016, conferindo ao autor o Título de Mestre em Ciência e Tecnologia da Computação

Banca Examinadora:

Prof. Dr. Alexandre Carlos Brandão Ramos (Orientador)

Prof. Dra. Renata Aparecida Ribeiro Custódio (Coorientadora)

Prof. Dr. Nandamudi Lankalapalli Vijaykumar

Prof. Dr. Roberto Claudino da Silva

Novembro de 2016
Itajubá - MG

Dedico este estudo a todos que me apoiaram e compreenderam minha
ausência, pelo tempo que deixamos de estar juntos.
Aos meus pais, João e Zuleides e ao meu irmão Rafael, a eles
todos os créditos.

AGRADECIMENTOS

A Deus pelo dom da vida de modo particular, por essa grande conquista, pela sua proteção e por todas as oportunidades que me foram concedidas.

A toda equipe que cooperou de alguma forma com este trabalho incluindo os colegas e colaboradores do Laboratório de Usabilidade e Fatores Humanos e do Grupo de Visão Sistema de Informação e Aplicações - VISCAP, sem os quais este caminho teria sido muito mais lancinante.

Aos professores, Dr. Alexandre Carlos Brandão Ramos e Dr^a. Renata Custódio, pela dedicação, orientação e paciência neste período de aprendizado.

Aos meus colegas do mestrado, em especial ao Luiz Flávio Felizardo, que me ajudaram neste tempo de pesquisa e estudo.

RESUMO

O objetivo do presente trabalho foi desenvolver um Sistema de Treinamento baseado em Computador (STBC) para o treinamento dos profissionais da área da saúde que atuam em Unidade de Terapia Intensiva Neonatal (UTIN). Foram utilizadas técnicas de Sistemas Tutores Inteligentes (STI) e de Sistemas Baseados em Regras de Produção (SBRP) para o treinamento destes profissionais, para que os mesmos possam aprender e reconhecer as mais diversas situações normais e anormais da interface da incubadora bem como os alarmes associados, o que pode contribuir significativamente com o aumento da segurança do paciente. O sistema desenvolvido através da aplicação de um STI auxilia um instrutor na tarefa de treinamento de um novo usuário ou mesmo de usuários experientes com a finalidade de formação e atualização em relação ao modo de uso da incubadora neonatal em condições comuns e adversas. A ferramenta desenvolvida permite a modelagem de novos casos, bem como a introdução de novos episódios no sistema. Finalmente, o STBC para UTIN, emprega tecnologias que representam o estado da arte na área da Engenharia de Software através de uso de UML para a modelagem do sistema. A sistemática adotada para o desenvolvimento de sistemas de treinamento permite que novos STI sejam desenvolvidos a partir do reuso das classes já desenvolvidas, facilitando o processo de implementação e aumentando a sua eficácia para novos treinamentos.

Palavras-chave: Engenharia de software; Unidades de Terapia Intensiva Neonatal; Incubadora Neonatal; Sistema de Treinamento Baseado em Computador; Sistema de Tutores Inteligentes.

ABSTRACT

The objective of this study was *to* develop a screen-based training system (SBTS) for training of healthcare professionals working in Newborn Intensive Care Units in Brazil. Techniques employed were from Intelligent Tutoring Systems (ITS) and Production Rules Based System (PRBS) for the training of these professionals to learn and recognize the various normal and abnormal situations of the incubator interface as well as the associated alarms, what can significantly contribute to the in-creasing of patient safety. The system developed by applying an ITS assists a teacher in the training task of a new user or even experienced users with the purpose of training and update. The developed tool allows modeling of new cases in the system. Finally, the SBTS for NICU, the systematic adopted for the development of training systems allows that new ITS are developed from the reuse of classes already developed, easing the implementation process and increasing the effectiveness for new training.

Keywords: Component; Software engineering; Newborn intensive care units; Infant incubator; Screen-based training system; Intelligent Tutoring Systems.

LISTA DE FIGURA

Figura 1.1 - Quantidade de publicações por assunto para os termos: “Screen Based Simulators” e “Intelligent Tutoring Systems”	16
Figura 1.2 - Quantidade de publicações, de 1990 até 2016, para os termos: “Screen Based Simulators” e “Intelligent Tutoring Systems”	17
Figura 2.1 - Evolução do IAC até evoluir para STI22	
Figura 2.2 - Arquitetura clássica de um STI27	
Figura 2.3 - Arquitetura de um STI28	
Figura 2.4 - Pré-compilação de regras36	
Figura 2.5 - Compilação da aplicação	377
Figura 3.1 - Diagrama de blocos do STI40	
Figura 3.2 - Regra do JEOPS para calcular o desempenho do aprendiz na avaliação 45	
Figura 3.3 - Regra do JEOPS para fornecer o relatório detalhado Erro! Indicador não definido. 46	
Figura 3.4 - Regra do JEOPS para calcular a quantidade de questão certa e errada.....	46
Figura 3.5 - Arvore de desempenho do aprendiz	47
Figura 3.6 - Regra do JEOPS para emissão do relatório de desempenho da avaliação.....	48
Figura 3.7 - Regra do JEOPS para emissão do relatório de desempenho da avaliação	49
Figura 3.8 - Regra do JEOPS para emissão do relatório de desempenho da avaliação, por assunto.....	50
Figura 3.9 - Regra do JEOPS para emissão do relatório de desempenho da avaliação, por sub-assunto.....	51
Figura 3.10 - Relatório de desempenho da avaliação.....	51
Figura 4.1 - Tela inicial do Sistema de Treinamento de UTIN	52
Figura 4.2 - Tela do sistema após o usuário logar com sucesso.	52
Figura 4.3 - Tela que representa a aba tutorias e exercícios.....	53
Figura 4.4 - Aba tutoriais e exercícios: abrir tutorial	54
Figura 4.5 - Aba tutoriais e exercícios: ouvir alarme	55
Figura 4.6 - Aba tutoriais e exercícios: assistir vídeo.....	55

Figura 4.7 - Aba tutoriais e exercícios: abrir exercício	56
Figura 4.8 - Aba gerenciamento de usuários	56
Figura 4.9 - Gerenciamento de usuários	57
Figura 4.10 - Aba gerenciamento de usuários: adicionar	57
Figura 4.11 - Aba gerenciamento de questões.....	58
Figura 4.12 - Aba gerenciamento de questões: adicionar questão	58
Figura 4.13 - Aba avaliações	59
Figura 4.14 - Tela da avaliação	60
Figura 4.15 - Tela do desempenho do aluno	61
Figura 4.16 - Aba sobre o sistema	61
Figura 5.1 - Recursos teóricos audiovisuais	63
Figura 5.2 - Questionários de mensuração	63
Figura 5.3 - Análise de desempenho do aluno.....	64

GLOSSÁRIO

ECRI	Emergency Care Research Institute
FDA	Food and Drug Administration
IA	Inteligência Artificial
IAC	Instrução Assistida por Computador
JEOPS	Java Embedded Object Production System
PI	Processamento da Informação
RUP	Rational Unified Process
SBRP	Sistema Baseado em Regras de Produção
STBC	Sistema de Treinamento Baseado em Computador
STI	Sistemas Tutores Inteligentes
TMO	Técnicas de Modelagem de Objetos
UML	Unified Modeling Language
UTI	Unidade de Terapia Intensiva
UTIN	Unidade de Terapia Intensiva Neonatal

SUMÁRIO

1. INTRODUÇÃO.....	13
1.1 Justificativa	15
1.2 Objetivos	18
1.2.1 Geral.....	18
1.2.2 Específicos	18
1.3 Classificação da incubadora	19
1.4 Estrutura do trabalho	20
2. FUNDAMENTAÇÃO TEÓRICA	21
2.1 Sistemas Tutores Inteligentes.....	21
2.1.1 História dos STI.....	22
2.1.1.1 Programas lineares.....	22
2.1.1.2 Programas ramificados	23
2.1.1.3 Programas gerativos	24
2.1.2 Características de um STI	25
2.1.3 Arquitetura dos STI.....	27
2.1.1.4 Módulo do Aprendiz.....	28
2.1.1.5 Módulo de Domínio	30
2.1.1.6 Módulo Tutor.....	32
2.1.1.7 Módulo de Comunicação.....	34
2.2 Java Embedded Object Production System - JEOPS.....	35
2.2.1 A evolução do JEOPS	35
2.2.2 Arquitetura do JEOPS.....	35
2.2.3 As regras do JEOPS	37
2.2.4 A escolha do JEOPS	38
3. ESTRUTURAÇÃO DO SISTEMA DE TREINAMENTO.....	40
3.1 Sistemas de Produção.....	41
3.2 Sistemática adotada para a aquisição do sistema	41
3.3 Cenário	43
3.4 Regras para o JEOPS	44
4. IMPLEMENTAÇÃO DO FRAMEWORK.....	52
5. APLICAÇÃO DO FRAMEWORK	62

6. CONCLUSÃO	66
REFERÊNCIAS	67
APÊNDICES	70
Apêndice A - Análise de requisitos	70
Apêndice B - Análise essencial	71
Apêndice C - Caso de uso	72
Apêndice D - Diagrama de sequência	81
Apêndice E - Diagrama de classes	83
Apêndice F - Relatório de Avaliação Heurística – Sistema de Treinamento da Incubadora	83
Apêndice G - Questões que se encontram na avaliação do STI-UTIN.....	89

1. INTRODUÇÃO

A relevância deste trabalho está na crescente preocupação em atingir o alto índice de segurança e qualidade no cuidado com pacientes, o que está intimamente ligado à indústria de tecnologias associadas a este cuidado, tanto para tratamentos, quanto para diagnósticos e treinamento de pessoal.

O crescimento da tecnologia aplicada à medicina tem se tornado cada vez mais importante no cuidado com os pacientes, o mesmo está ligado às alterações, atualizações e a novos projetos de equipamentos médicos (LILJEGREN, 2006). Estas tecnologias aumentam as possibilidades de diagnóstico, tratamento e melhora a qualidade de vida do paciente. Porém, tais avanços podem aumentar consideravelmente a complexidade dos sistemas disponíveis para o cuidado com a saúde e comprometer a segurança, principalmente na questão relacionada aos alarmes que são extremamente comuns em toda esta tecnologia (REASON, 2000). Todos os anos o Emergency Care Research Institute (ECRI), uma organização não-governamental dos EUA, realiza pesquisas na área da saúde e tem como abordagem principal a tecnologia em saúde, a gestão de riscos, a qualidade e o meio ambiente na área da saúde. Em um relatório anual, este mesmo instituto, ECRI, divulga os 10 maiores perigos relacionados à tecnologia. Os relatórios dos anos de 2012, 2013, 2014 e 2015 apontam como o mais importante problema os perigos com alarmes (INSTITUTE, 2015) em que o Food and Drug Administration (FDA)¹ relatou 566 mortes relacionadas a problemas com alarmes, entre os anos de 2005 e 2008 (FUNK *et al.*, 2014) e estima-se que os alarmes são falsos entre 72% a 99% das vezes, gerando dessensibilização, atraso na resposta clínica e consequente perda da monitorização adequada do paciente (BRIDI *et al.*, 2014).

Recentes evidências em pesquisas realizadas nos EUA pela FDA e a Joint Commission² têm demonstrado uma ligação entre a “fadiga de alarme” e a morte de pacientes. Altas taxas de falso alarme têm potencial de levar à fadiga o paciente, levando os enfermeiros a atrasarem a sua resposta aos alarmes, ignorá-los ou desativá-los inteiramente (BONAFIDE *et al.*, 2014). Para reduzir os problemas com os mesmos, a Healthcare Technology Foundation³ recomenda

¹ Órgão governamental dos Estados Unidos da América responsável pelo controle: dos alimentos, suplementos alimentares, medicamentos, cosméticos, equipamentos médicos, materiais biológicos e produtos derivados do sangue humano.

² Organismo de acreditação de unidades de saúde baseada nos Estados Unidos.

³ Fundação com o objetivo de alcançar a melhoria na utilização segura da tecnologia de saúde a fim de utilizar seus conhecimentos coletivos sobre a concepção, a utilização, integração e manutenção de tecnologia de saúde, sistemas e dispositivos.

a formação inicial e contínua sobre os dispositivos médicos à base de alarmes que deveram ser operados (CLARK *et al.* [s.d.]).

A experiência relatada pela ECRI afirma que muitas vezes os problemas resultam de alarmes configurados incorretamente ou inadvertidamente pela equipe de enfermagem da UTI. Estes são afetados tanto pelo *design* da interface do sistema de alarme do dispositivo, bem como do nível da equipe de enfermagem de proficiência, com a configuração e o gerenciamento de alarmes. A formação inicial e contínua eficaz é de importância vital. Muitos sistemas de alarmes, pode parecer simples, mas muitas vezes não são bem compreendidos pela equipe (BOROWSKI *et al.*, 2011), por isso os treinamentos se tornam essenciais, devido à complexidade do ambiente da saúde, onde treinamentos *in loco* não são permitidos por envolverem a segurança dos pacientes.

Pesquisas realizadas por Pul *et al.* (2014) e Brewer *et al.* (2015) na Unidade de Terapia Intensiva Neonatal (UTIN) constatam que o número de alarmes hora/paciente é superior ao da UTI adulta, e que a maioria dos alarmes da UTIN é sobre falha na oximetria de pulso (68%). Este equipamento fica normalmente associado a incubadora neonatal, que neste contexto de suporte à vida, se torna um equipamento protagonista por atender às necessidades de atenção básica do recém-nascido. É de suma importância também associar corretamente os outros equipamentos (monitores fisiológicos, ventilador mecânico, bombas de infusão) ou mesmo os equipamentos que são ou poderão ser utilizados em determinados procedimentos junto ao recém-nascido, interferindo o mínimo possível nas condições ideais para o bebê. Isso tem sido o desafio para a equipe de saúde. Para auxiliar o profissional, a incubadora neonatal também possui alarmes que indicam se o equipamento está fora dos parâmetros escolhidos ou se existe falha no sistema. Portanto, é necessário possuir um conhecimento abrangente sobre a incubadora neonatal e, se necessário é preciso saber corrigir as possíveis situações negativas que possam vir a ocorrer neste conjunto tecnológico, estas atitudes em muitos casos são decisivas para o desenvolvimento adequado e a sobrevivência do neonato.

Desta forma, a fim de contribuir com a redução de perigos e danos à saúde do recém-nascido e visando os funcionários recém-admitidos na UTIN, é necessário que se ofereça um treinamento com ambiente interativo, rico em informações, que abrangem todos os procedimentos operacionais e de emergência, de modo a obter a familiaridade necessária com os sistemas de controle do modelo de UTIN. Baseado nestes aspectos, esta pesquisa objetivou desenvolver um Sistema de Treinamento baseado em Computador (STBC), para o treinamento dos profissionais da área da saúde que atuam em UTIN, este método irá reduzir os custos

envolvidos na fase inicial, uma vez que fornece o treinamento ao funcionário a partir de tutoriais apoiados na utilização de ferramentas de desenvolvimento hipermídia e acompanhamento da evolução do processo de aprendizado. Através da utilização de Sistemas Tutores Inteligentes (STI) e de Sistemas Baseados em Regras de Produção (SBRP), para o treinamento destes profissionais foi possível detectar e reconhecer as diversas situações normais e anormais da interface da incubadora, bem como os alarmes associados, o que pode contribuir significativamente com o aumento da segurança do paciente e a diminuição dos custos de treinamentos para os hospitais.

1.1 Justificativa

Na saúde brasileira a ampliação de sistemas de ensino constitui uma tendência crescente na época presente e boa parte está vinculada às universidades com projetos de pesquisas. Ainda conhecendo as probabilidades e potenciais de uso do computador na educação, existe, em muitas instituições de ensino em saúde, a carência de material educacional com os recursos da informática. Apesar do avanço tecnológico, o ensino de Enfermagem no Brasil ainda ocorre, em sua maioria de forma tradicional, ministrando-se aulas formais com exposição dos conteúdos e com pouca participação efetiva dos alunos (FONSECA *et al.*, 2009).

Existem diversas pesquisas realizadas, tanto na área de STBC como na área de STI. Nas áreas de Engenharia, Ciência da Computação, Medicina e Matemática, um aspecto que deve ser levado em conta é que o STI não apresenta tanta diversificação nos assuntos como o STBC. Isso ocorre, provavelmente, devido a sua inter-relação na área de Ciência da Computação. Essa diferença de assuntos correlacionados, pode ser observada na busca realizada na base científica Scopus de artigos publicados que contém as palavras “*Screen Based Simulators*” (STBC) em comparação aos artigos com as palavras “*Intelligent Tutoring Systems*” (STI), onde observa-se que a maioria das pesquisas é focada nas áreas de Ciência da Computação, Engenharia e Matemática, que englobam um total de 79,46% dos artigos relacionados com as palavras chaves pesquisadas, já as áreas de Enfermagem e Profissionais Saúde representam pouco mais de 0,63% do total, como pode ser observado na Figura 1.1.

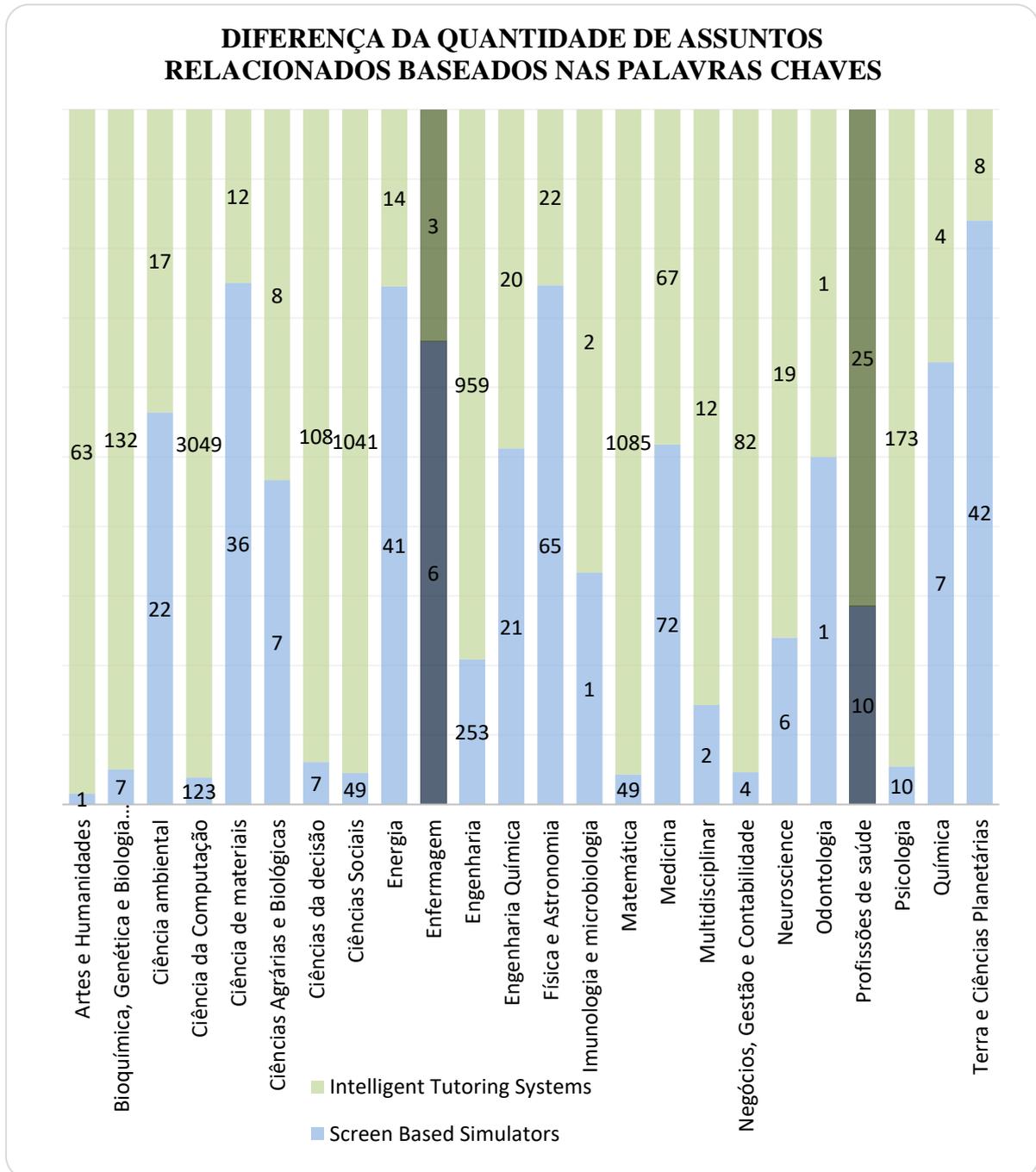


Figura 1.1 - Quantidade de publicações por assunto para os termos: “Screen Based Simulators” e “Intelligent Tutoring Systems”

Fonte: Scopus (2016)

Também é possível notar um aumento das pesquisas na área de STBC e STI como pode ser visto na Figura 1.2, em uma busca realizada em bases científicas do Scopus, de um total de 4465 artigos de 1990 até o ano de 2016, somadas ambas as áreas de pesquisas.

A pesquisa sobre STBC e STI associado a análise e projeto de STBC, voltada para o treinamento de profissionais da área de saúde que trabalham com unidades de cuidados

intensivos para recém-nascidos, foi realizada na base de dados do Scopus quatro buscas, utilizando as palavras chaves associadas:

1. “Intelligent Tutoring Systems” e “newborn intensive care unit”.
2. “Intelligent Tutoring Systems” e “newborn”.
3. “Screen Based Simulators” e “newborn intensive care unit”.
4. “Screen Based Simulators” e “newborn”.

Em ambos os casos, apenas o artigo publicado, em abril de 2016 (GUALBERTO *et al.*, 2016), foi encontrado.

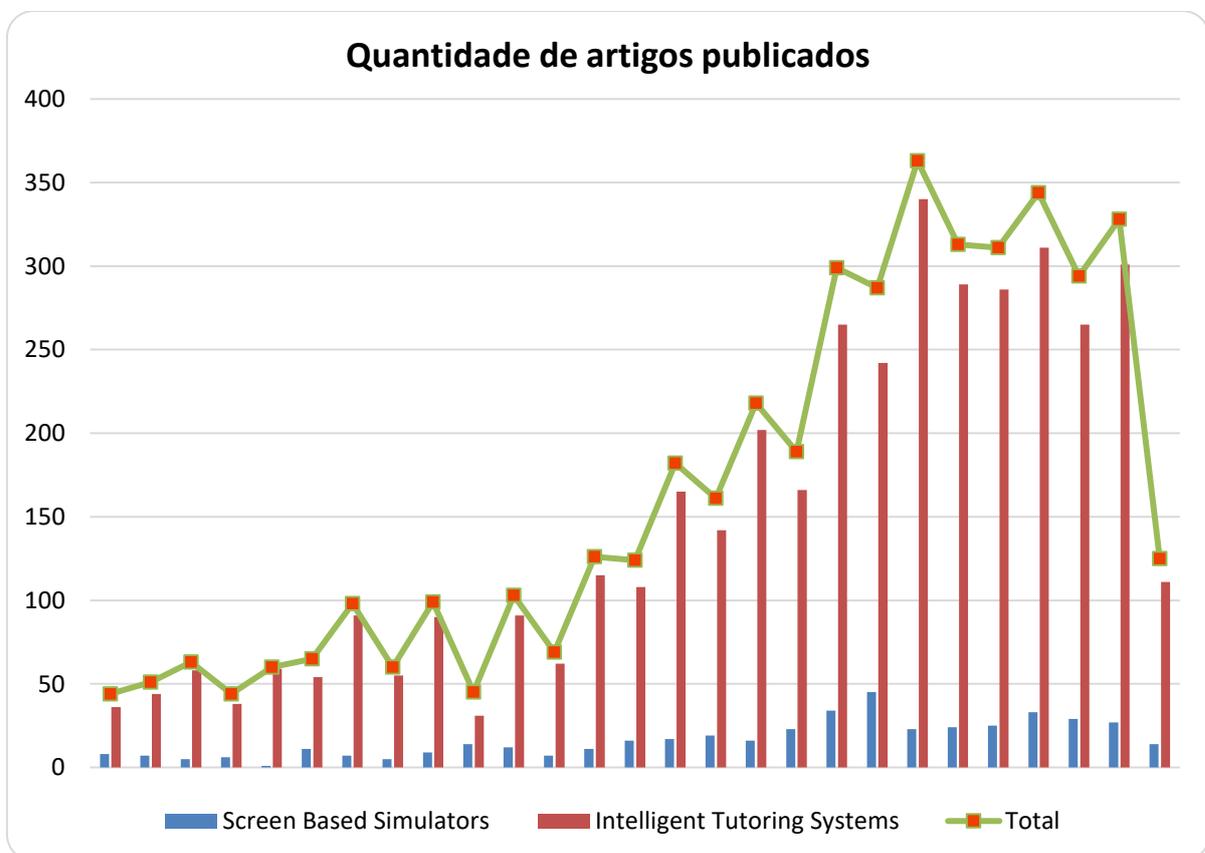


Figura 1.2 - Quantidade de publicações, de 1990 até 2016, para os termos: “Screen Based Simulators” e “Intelligent Tutoring Systems”
Fonte: Scopus (2016)

Refletindo nas inúmeras possibilidades do uso da tecnologia na educação, em especial para abastecer a lacuna existente pela insuficiência de materiais educativos e inovação no processo ensino-aprendizagem da temática de treinamento para (UTIN), notou-se a importância de se desenvolver um software de treinamento para auxiliar no ensino da incubadora neonatal em UTI.

1.2 Objetivos

1.2.1 Geral

- Desenvolver um STBC para ser utilizado no treinamento de profissionais de saúde que atuam na UTIN.

1.2.2 Específicos

- Analisar e desenvolver um STBC;
- Estudar as tecnologias de IAC e STI, e a ferramenta JEOPS, para que possa avaliar o comportamento de diferentes categorias de alunos, durante os exercícios e testes de modo a classificar seu desempenho.

1.3 Classificação da incubadora

Para o escopo deste estudo foi utilizada a Incubadora Neonatal Modelo Vision® Advanced 2286. Em seu manual (FANEM, 2012) é descrito que o ambiente termo neutro é o principal fator para o tratamento de pacientes prematuros. “A incubadora Vision® Advanced 2286, oferece ao recém-nascido um microclima de grande estabilidade e uniformidade em toda a área do colchão, incorporando inovações tecnológicas para tratar e monitorar o paciente sem removê-lo do leito, com mínimo manuseio, em ambientes de baixa luminosidade e de reduzido ruído sonoro, atendendo as mais modernas e eficientes técnicas de incubação de prematuros críticos e com ampla visão do seu monitoramento”.

Com operação extremamente silenciosa, resultando menor incômodo para o paciente, esta incubadora também disponibiliza a função de alta umidade relativa do ar, com servocontrole das temperaturas, bem como funções opcionais de servocontrole de concentração de oxigênio, oximetria de pulso de alta precisão e para baixa perfusão, balança incorporada ao leito radiotransparente, painel de controle lateral e giratório que sintetiza todas as opções de ajuste e monitorização, com gráficos de funções para tratamento e diagnóstico através de menu interativo e botão rotativo.

O equipamento é fabricado em conformidade com as normas NBR IEC 60601-1, Equipamento eletromédico Parte 1 - Prescrições gerais para segurança; NBR IEC 60601-2-19, Prescrições particulares para segurança de incubadoras para recém-nascidos; NBR IEC 60601-1-2 Compatibilidade eletromagnética e NBR IEC 60601-2-49 Prescrições particulares para segurança de monitores multiparamétricos.

1.4 Estrutura do trabalho

A divisão desta pesquisa foi realizada de maneira a facilitar sua leitura, em forma de capítulos e de forma concisa e clara. O presente capítulo (capítulo 1) apresenta uma introdução a partir do contexto do trabalho, da relevância do tema, da justificativa de escolha do tema, dos objetivos gerais e específicos a serem alcançados, bem como a explicação da estrutura do trabalho para que possa ser possível visualizar o que foi realizado.

O capítulo 2 trata da teorização dos elementos envolvidos onde na sua primeira seção será discutido sobre o STI: a história, iniciando do IAC até a sua evolução à STI, as características principais, e sua arquitetura. A segunda seção é sobre Java Embedded Object Production System (JEOPS), o motor de inferência na linguagem Java utilizado nesse estudo.

No capítulo 3 encontra-se o processo de Engenharia de Software para a elaboração do projeto, onde é descrita a estruturação do STI, envolvendo a sistemática adotada e exemplificados os cenários utilizados, a análise de requisitos apurada, os casos de uso, diagramas de sequência e diagramas de classes.

O desenvolvimento do trabalho está descrito nos capítulos 4 e 5. O capítulo 4 descreve a implementação do framework proposto que foi efetivado em Java, empregando classes e objetos disponibilizados pela linguagem. Por fim, no capítulo 5 são mostradas as aplicações desenvolvidas durante os estudos.

O capítulo 6 descreve os detalhes da conclusão do trabalho juntamente com possíveis direcionamentos para trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, os fundamentos necessários para o entendimento do problema e as explicações das escolhas feitas serão listados. Na primeira seção do capítulo tratar-se-á da teoria dos STI, englobando a sua história, características e a explicação das principais arquiteturas e de cada módulo: aprendiz, domínio, tutor e comunicação.

Em seguida, uma explicação do JEOPS, que é o motor de inferência utilizado, começando da sua evolução, o funcionamento de sua arquitetura, as regras que compõem o motor e o que motivou a sua escolha.

2.1 Sistemas Tutores Inteligentes

STI são programas de computador que são projetados para incorporar técnicas de IA, os STI podem similarmente ser pensados como tentativas para produzir o comportamento no computador quando executado por um ser humano. A concepção e o desenvolvimento de tais sistemas de tutores ficam no cruzamento da ciência da computação, psicologia cognitiva e pesquisa educacional; esta área de intersecção é normalmente referida como a ciência cognitiva. Por razões históricas, grande parte da investigação no domínio do software educativo envolvendo Inteligência Artificial (IA), tem sido conduzida em nome de Sistemas de Instrução Assistida por Computador (IAC) foi o passo inicial na história dos STI, que era visto como um programa linear. O problema central com esses sistemas mais antigos era a incapacidade de fornecer *feedback* rico ou individualizado, porque eles não foram projetados para saber o que eles estavam ensinando, quem eles estavam ensinando ou como ensiná-lo.

Com um breve estudo histórico dos primeiros sistemas projetados com objetivos educacionais até os dias atuais, os aspectos da teoria Behaviorista e Construtivismo de Piaget serão apresentados para complementar a metodologia de ensino.

Greeno (1998) cita que os princípios behavioristas⁴ tendem a caracterizar a aprendizagem em termos de aquisição de habilidade, um pressuposto fundamental da prática educativa behaviorista é que as habilidades complexas são aprendidas através da aquisição de

⁴ Behaviorismo, também conhecido como psicologia comportamental, é uma teoria da aprendizagem baseada na ideia de que todos os comportamentos são adquiridos através de condicionamento. Condicionamento ocorre através da interação com o meio ambiente. Behavioristas acreditam que as nossas respostas a estímulos ambientais moldam nossas ações.

componentes mais simples, seguido por combinações destes em mais habilidades comportamentais complexas. Seguindo essa premissa, muitos treinamentos foram organizados em seqüências de objetivos comportamentais.

No construtivismo, segundo Greeno (1998), a prática de ensino bem sucedida nesses treinamentos mais construtivistas são um desafio para muitos tutores, cuja formação e experiência têm sido baseadas nas práticas orientadas à habilidade. O ensino construtivista é mais de improviso, exigindo sensibilidade aos entendimentos que os alunos já possuem, a fim de ajudá-los a construir sobre esses entendimentos.

2.1.1 História dos STI

Os STI têm sua origem na área de IA. No final da década de 50, os pensadores da época acreditavam que os computadores poderiam “raciocinar”, tal qual os seres humanos, no entanto, isso não aconteceu até os dias de hoje. Parecia aceitável admitir que, uma vez criada, as máquinas poderiam pensar, elas seriam capazes de realizar tarefas que estivessem associadas com o pensamento humano, como por exemplo, a instrução, que se refere ao processo de aprendizado qualificado especialmente pelo ensino de métodos de um conteúdo específico, semelhante ao treinamento.

Os Sistemas de IAC foi o passo inicial na história dos STI. A Figura 2.1 mostra a evolução desses sistemas até STI's.



Figura 2.1 - Evolução do IAC até evoluir para STI
Fonte: Nwana (1990)

2.1.1.1 Programas lineares

O caminho para a evolução dos STI se estende por um período de quase quatro décadas. Teve início em 1950 com “programas lineares” simples, que se baseiam no princípio do condicionamento operante (NWANA, 1990).

O material era organizado para dar o passo a passo ao estudante do comportamento desejado e apresentado em uma série de “frames”. A maioria dos quadros apresentava perguntas muito simples (por exemplo, envolvendo apenas o preenchimento de espaços ausentes), e o aluno era informado de imediato se a resposta estava certa ou errada. A principal limitação dos programas lineares encontrava-se na falta do conhecimento individualizado, ou seja, significava que todos os estudantes, independentemente das suas capacidades ou conhecimento prévio do domínio, receberam exatamente o mesmo material e exatamente a mesma sequência, e não forneciam o *feedback* necessário, ignorando as respostas dadas pelos alunos. Este desempenho dos sistemas continha sua base na Teoria Behaviorista, defendida por Skinner (1959). A Teoria Behaviorista propunha que as pessoas trabalhassem por estímulos e que, à igual estímulo corresponde igual resposta. Por este motivo, não precisava-se admitir que os alunos cometessem erros, já que estes lhes dariam um reforço negativo.

Segundo Carbonell (1971), a maioria dos sistemas IAC do tipo orientado a frames, tinha o computador assemelhando-se a um livro de texto, pois o ensino em sequência é extremamente simples, talvez trivial, deve-se considerar utilizar técnicas mais relacionadas com a metodologia de tarefas. Superar essas limitações de cadeia de eventos, que culminaram com os STI de hoje.

2.1.1.2 Programas ramificados

Crowder (1960) superou determinadas limitações dos sistemas skinnerianos deixando de ignorar as respostas dos alunos. Ele propôs usar o resultado da avaliação para controlar o material indicado ao estudante. Os sistemas tinham um número fixo de quadros, mas foram capazes de comentar sobre a resposta de um aluno e, em seguida, usá-lo para escolher o próximo quadro, não sendo mais em sequência. Os programas ramificados ou “programação ramificada” ou ainda, programação em árvore que era mais adequada por ter *feedback* baseado nas respostas dos alunos. Técnicas de correspondência de padrões, a qual consiste em comparar padrões de *strings* foram utilizadas, permitido processar as respostas das alternativas como aceitáveis ou parcialmente aceitáveis e não como totalmente correta ou incorreta, como exigido pelos sistemas skinnerianos.

Uma dificuldade da programação ramificada é o material de ensino exorbitantemente amplo para ser controlável por meio de uma programação simples e por isso a “linguagem de autoria”⁵ foi desenvolvida para a elaboração de materiais IAC.

2.1.1.3 Programas gerativos

No início dos anos 1970, surgiram os sistemas gerativos que vieram a ser também chamados de “sistemas adaptativos”, sendo o principal precursor do STI, com a melhora dos aspectos de individualização e *feedback*, houve uma representação do conhecimento bastante superficial, um novo nível de sofisticação no *design* dos sistemas IAC. Estes surgiram do reconhecimento do fato de que o material didático poderia ser gerado pelo computador. Um sistema gerador tem a capacidade de gerar e resolver problemas significativos. O sistema gera parâmetros para estes formatos e procura em um banco de dados para determinar a resposta correta.

Observa-se que nenhum desses sistemas tem conhecimento do domínio que está ensinando, nem pode responder as perguntas dos alunos quanto ao "*porquê*" e "*como*" a tarefa é executada. Urretavizcaya (2001) indica uma série de situações que ainda necessitam ser corrigidas:

- a) Em vez de concentrar-se em sistemas de construção para os tópicos mais limitados, eram produzidos cursos muito extensos;
- b) Existe uma linguagem técnica que restringe a interação dos usuários com o sistema;
- c) Não continha o "conhecimento" do tutor ou "compreensão" dos próprios alunos, conseqüentemente, eles tendiam a assumir um modelo de conhecimento estabelecido do estudante, e eles não poderiam conceber de modo a diagnosticar o equívoco de um estudante dentro de seu próprio quadro;
- d) Pouca ou nenhuma cooperação entre educadores, psicólogos e cientistas da computação na fase de desenvolvimento dos tutores;
- e) Eles tendem a ser estáticos em vez de dinâmicos. Houve pouca experimentação de sistemas, a fim de melhorá-los.

⁵ Escrita de programas, procedimentos ou regras associadas a documentação, relativo à operação de um sistema de computador e que são armazenados em leitura/escrita na memória.

Em resposta aos problemas IAC enfrentados, a Psicologia Cognitiva começou a questionar as suposições do Behaviorismo (comportamentalismo). Novas teorias da aprendizagem e o Construtivismo de Piaget começavam a ter influência. Chomsky junto com Newell e outros, introduziram as ideias de processamento de informação simbólica (GREENO; COLLINS; RESNICK, 1996), ideias essas que despertaram o interesse da comunidade de IA, em linguística e processamento de linguagem natural. O processamento da informação (PI) tornou-se um paradigma dominante no final anos 70 e começo dos 80.

Os STI nascem como iniciativa no intento de tratar as falhas dos sistemas gerativos e podem ser vistos como os IAC inteligentes dos anos 80. Esta iniciativa foi beneficiada pelo trabalho dos pesquisadores de IA, que tem uma permanente preocupação em melhorar a forma de representação das informações na base de conhecimento dentro de um sistema inteligente. O sistema proposto tinha uma base de conhecimento sobre um assunto e informação geral sobre a linguagem e os princípios de ensino tutorial. O sistema poderia, então, prosseguir um diálogo de linguagem natural com um estudante, por vezes, na sequência da iniciativa do aluno, às vezes tendo sua própria iniciativa, mas sempre gerando suas declarações e respostas em uma forma natural a partir do seu conhecimento geral.

Os modelos psicológicos permitem simulações de desempenho dos alunos que representam até o seu comportamento. Foi reconhecido que um sistema verdadeiramente "inteligente" é uma tarefa não trivial, que precisa de especialistas de várias outras disciplinas; uma necessidade que pode ser fornecida pela comunidade de pesquisa IA, que contém cientistas da computação, psicólogos e pedagogos.

Já, no ano de 1982, Sleeman (1984) e Burton e Brown (1979) revisaram o estado da arte nos sistemas IAC e criaram o termo de STI, para distingui-los dos sistemas IAC antecessores. Estes sistemas facilitam o ensino e aprendizagem fazendo-o mais efetivo, correto e também mais agradável.

2.1.2 Características de um STI

Para ser considerado um sistema inteligente, o sistema deve apresentar três características (JONASSEN; WANG, 1993) que denotam inteligência:

1. O assunto deve ser bem conhecido, o suficiente para que o sistema faça inferências sobre o domínio ou resolva problemas que estejam em seu escopo de atuação;
2. Deve ser capaz de avaliar o aluno;

3. Deve fornecer estratégias pedagógicas que minimizem a diferença entre o aprendiz e o especialista.

Algumas das principais diferenças entre os programas STI e IAC são (ou deveriam ser):

- a) STI fornece uma clara articulação de conhecimentos para um domínio limitado;
- b) STI têm um modelo de desempenho do estudante que é mantido de forma dinâmica e é usado para conduzir a instrução;
- c) O seu criador define o conhecimento e as regras de inferência, mas não a sequência de ensino, que é derivado pelo programa;
- d) STI fornece diagnósticos detalhados de erros ao invés de simplesmente perfurar e a prática;
- e) Os alunos podem fazer perguntas para um STI (esta é a principal característica dos tutores inteligentes).

De uma forma geral, os STI podem ser definidos como um termo amplo, abrangendo qualquer programa de computador que dão suporte às atividades da aprendizagem, com propósitos educacionais e que incorporam técnicas de IA. Oferecem vantagens sobre os IAC, pois podem simular o processo do pensamento humano para auxiliar na resolução de problemas ou em tomadas de decisões.

Os STI permitem um maior grau de individualização na instrução, em particular relaciona a instrução com o entendimento das metas e crenças do aluno, usando as técnicas de IA tais como: o planejamento, a otimização e as buscas, deixando que o sistema decida “inteligentemente” a ordem de apresentação do conteúdo ao aluno, permitindo assim uma interação variada, desde os passivos ou reativos que dependem completamente de esperar que o aluno realize uma ação bem determinada, até os que constantemente apresentam nova informação, passando pelos tutores, caracterizados por observar ao aluno enquanto faz uma tarefa, sem interferir constantemente, mas explicando ou ensinando um conceito em um momento importante ou quando o aluno solicita.

Um STI possui o conhecimento do domínio de forma restrita e claramente articulado, utiliza o conhecimento do aluno para adaptar o ensino, a sequência do ensino não é pré-determinada, realiza processos de diagnóstico mais adaptados ao aluno e permitem a comunicação tutor-aluno (URRETAVIZCAYA, 2001). Têm em sua arquitetura tradicional quatro componentes funcionais: a matéria que ensina (módulo de domínio), as que caracterizam o aluno (módulo do aprendiz) para prover ensino individualizado, as estratégias para o ensino

(módulo tutor ou pedagógico) e o módulo de comunicação, que deve ser bem planejado e de fácil manipulação para favorecer a comunicação tutor-aluno (Figura 2.2).

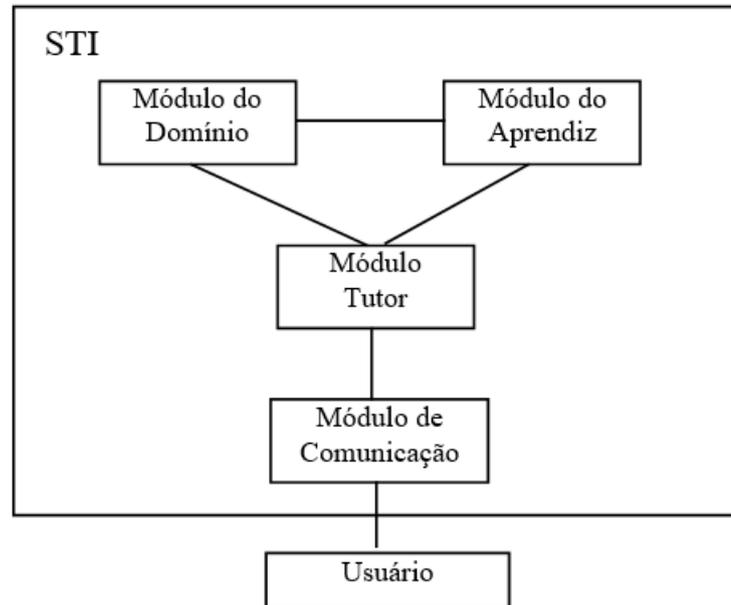


Figura 2.2 - Arquitetura clássica de um STI
 Fonte: McTaggart (2001) *apud* Gualberto et al. (2016)

Este sistema combina dinamicamente as informações dos componentes para tomar decisões adequadas em situações específicas em uma seção de tutoramento (AKHRAS; SELF, 2002).

2.1.3 Arquitetura dos STI

Existe uma variada gama de arquitetura de STI, de fato, é quase uma raridade encontrar dois STIs com base na mesma arquitetura. Isto resulta do caráter experimental do trabalho na área. Com as pesquisas de Mandl e Lesgold (1988) e Wenger (1987), os mesmos identificaram uma arquitetura com quatro componentes (Figura 2.2):

- 1) O módulo de conhecimento especializado.
- 2) O módulo de modelo de estudante.
- 3) O módulo de tutoria.
- 4) O módulo de interface de utilizador.

Wenger (1987) abstrai os componentes das mais tradicionais definições de Engenharia de Software, descrevendo-o como “ferramenta de comunicação de conhecimento”. propõe um estudo em conjunto das disciplinas de IA, Ciência Cognitiva e Educação (WENGER, 1987).

Outra arquitetura geral foi proposta por Clancey (1987), em que são separadas as funções dos módulos: por um lado, o modelo do aluno se constrói e se atualiza através de um diagnóstico, realizado pelo módulo “modelador”, que pode ser considerado o primeiro “especialista”; e por outro lado, o tutor guarda as informações sobre o aluno, mas também o realimenta, para o qual requer uma estratégia e tomada decisões. Essa arquitetura tem sete componentes, conforme pode ser observado na Figura 2.3.

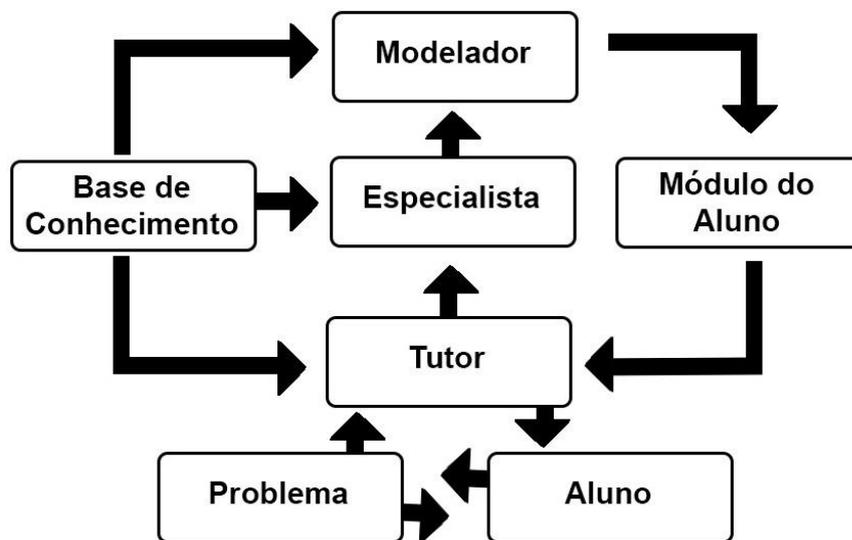


Figura 2.3 - Arquitetura de um STI
Fonte: Baseado na pesquisa de Clancey (1987)

As arquiteturas propostas possuem essencialmente quatro módulos funcionais: o modelo do aluno, o modelo de domínio, o modelo pedagógico e o modelo de interface. É possível adicionar outros componentes a esses módulos, como uma base de conhecimento, um módulo para tomada de decisões ou outros, dependendo dos domínios do STI. A seguir são detalhados os quatro módulos básicos de um STI (COSTA; WERNECK, 1997).

2.1.1.4 Módulo do Aprendiz

O módulo do aprendiz representa o conhecimento e as habilidades do aluno no momento de interação com o STI. A característica fundamental da pesquisa em STI é a atenção meticulosa

ao aluno, visto que essas informações serão utilizadas para a tomada de decisões, sendo constituído de dados estáticos e dinâmicos que serão fundamentais para o tutor comprovar hipóteses sobre o aluno (SELF, 1990).

O modelo do aprendiz é a representação dinâmica do conhecimento do aluno e a habilidade. O modelo fornece ao sistema de tutoria o potencial de guiar e apoiar o aluno durante a aprendizagem. A inclusão de uma representação do conhecimento do aluno permite a avaliação do aluno em níveis adequados. Além disso, “ensinando” o modelo com uma representação do conhecimento incorreto, dá ao sistema a capacidade para diagnosticar, explicar e corrigir os erros do estudante (YACEF, 2005). Se o modelo de estudante é executável, significa que pode explorar o próprio aluno a expandir o seu conhecimento, e então, o sistema pode prever o comportamento dos alunos, que podem ser usados pedagogicamente para fornecer uma experiência de aprendizagem mais eficiente.

Existem diferentes metodologias que são empregadas para a construção do módulo do aprendiz, como por exemplo: comparação da conduta de um aluno com a de um especialista, inclusão de preferências do aluno, indicação de objetivos particulares, e a inclusão de reconhecimento de padrões aplicados à história das respostas fornecidas pelo aluno, esta última adotada nesta pesquisa. O modelo do aluno pode ser representado apoiando-se em alguns modelos de descrição (COSTA; WERNECK, 1997).

- Modelo Overlay: foi arquitetado por comparação da performance do estudante com a ação adotada pelo computador, fundamentada no conhecimento do perito para um problema similar. As falhas neste modelo são creditadas à carência de quaisquer informações presentes na base de domínio.
- Modelo de Buggy: segundo Bertels (1994), o modelo foi batizado com esse nome por relacionar o conhecimento do aluno a uma confusão sobre conhecimento especialista, onde as falhas dos alunos são assumidas pela presença de entendimento errado de um conceito certo. Este modelo agrupa em seu banco de dados os erros típicos. Tendo como vantagem a possibilidade de representar e reconhecer os entendimentos incorretos por parte do aluno ou carência de conceitos.
- Modelo de Simulação: O nome do modelo baseia na correspondência com a verdade que ele oferece, pode prever a resposta do estudante baseado no comportamento modelado.

- **Modelo de Crenças:** O modelo de aluno baseia-se em um conjunto de crenças, ou seja, detém uma proposição ou premissa para a verdade, cogitando o nível de entendimento do estudante sobre um determinado conceito.
- **Modelo de Estereótipo:** Nome que batizou pela ideia ou convicção classificatória preconcebida sobre alguém ou algo, resultante de expectativa, hábitos de julgamento ou falsas generalizações. Em geral, classifica o usuário como novato, intermediário ou avançado em uma dada área de conhecimento. O nível de conhecimento do usuário é representado por um conjunto de pares (tópico, valor), com o usuário podendo ter um ou mais estereótipos ao longo da realização da tarefa.

O modelo usado que tem provocado melhores efeitos é a combinação do Modelo Overlay com o Modelo Estereótipo. Na pesquisa realizada por Rosa Maria e Vera Maria (COSTA; WERNECK, 1997) é ressaltado que o Modelo de Estereótipo se mostra bastante confiável e é mais simples que o Modelo Overlay, porém, menos flexível e poderoso.

Além do apoio dos modelos de descrição, o módulo do aprendiz possui três tarefas essenciais para a sua aplicação, para que obtenha um resultado satisfatório (WENGER, 1987):

1. Coletar dados sobre o aprendiz, que podem ser explícitos (requisitando alguma resposta) ou implícitos (forma de interação);
2. Usar os dados para representar o conhecimento do aluno e processos de aprendizagem;
3. Representar os dados fazendo algum tipo de diagnóstico, tanto no estado do conhecimento do aluno como em termos de uma ótima seleção de estratégias pedagógicas, para apresentar depois a informação do domínio ao aluno.

A efetividade de um STI em proporcionar a educação distinta a cada aluno depende da qualidade e da exatidão do conhecimento sobre o aluno no modelo. Visto que o conhecimento do aluno muda e o modelo deve incorporar essas mudanças dinamicamente, é necessário verificar o nível da refinação do aspecto do conhecimento no sistema e da eficácia dos métodos empregados para extrair e incorporar novos dados sobre o aluno.

O problema da modelagem do aluno já foi considerado intratável, no sentido de que não há uma possibilidade realista de arquitetar modelos de aluno que recebam todos os objetivos dos desenvolvedores de STI.

2.1.1.5 Módulo de Domínio

O módulo de domínio abrange o aspecto do conhecimento especializado de domínio para ser conduzido para o aprendiz, pode ser representado em diferentes formas, tais como regras de produção e redes Bayesianas (SELF, 1990). O módulo especialista tem funções diferentes: ele age primeiro como fonte de conhecimento para ser apresentado ao aprendiz, incluindo a geração de explicações, perguntas e respostas, depois é fornecida uma referência para avaliar a performance, em que deve ser capaz de gerar as soluções para os problemas no mesmo contexto que o aprendiz, de modo que as respostas possam ser comparadas. O módulo de domínio é o elemento especialista do tutor e contém a informação sobre o domínio que se deseja ensinar. É um banco de dados organizado em conhecimentos declarativos e procedurais em um domínio específico (McTAGGART, 2001).

O objetivo deste módulo seria reproduzir essas estruturas de conhecimento na mente do aprendiz. Neste módulo se depara a informação do domínio que deve ser transmitido, sendo classificada em:

- Declarativo: os princípios primários, a abrangência do domínio e sua relação com outros domínios;
- Procedural: o raciocínio utilizado através de uma sequência de instruções de rotina e sub-rotina ou função, que é utilizado para realizar uma tarefa;
- Heurístico: define as operações adotadas provisoriamente, como ideia diretriz, geralmente descrito por um especialista, executadas para resolver problemas.

O modelo de domínio possui a incumbência de disponibilizar a origem da informação a ser demonstrada ao aluno, incluindo fontes para a concepção de perguntas, esclarecimentos e respostas, com o objetivo de fornecer um padrão para o desempenho do aluno.

Para a competência em esclarecer o raciocínio utilizado para a resolução de problemas, o modelo emprega-se de heurísticas inteligentes aplicadas na construção da base de conhecimento dos STBC, assim oferecem uma ampla gama de possibilidades em interpretar as respostas não habituais de certos aprendizes, aumentando a destreza do sistema em expressar seu próprio conteúdo de informação. A seriedade do perito para abastecer a base de informações adequadas é indispensável, principalmente pelo fato que o conhecimento a ser armazenado na base de conhecimento é um dos fatores que determina a diferença entre um STI e um IAC (BADER *et al.*, 1988).

Uma divisão categórica dos principais métodos de representação deste modelo é proposta pelos autores Fischetti e Gisolfi (1990), Garcia e Chien (1991) e Maria *et al.* [s.d.]:

- Frames: representação do conhecimento definida como um grafo direcionado no qual os vértices representam conceitos e as arestas representam relações semânticas entre os conceitos, onde os nós que são ligados no frame descrevem objetos, eventos e situações estereotipadas;
- Lógica: usam expressões da lógica simbólica, que preocupa-se basicamente com a estrutura do raciocínio para representar o conhecimento, através de regras de inferência e prova formal a instância do problema;
- Meta-regras: expressam propriedades de outras regras. São regras que concluem qualquer coisa sobre outras regras. Estão fundamentadas no conceito de meta-conhecimento;
- Orientação a objetos: é considerada como uma nova e poderosa ferramenta para a representação do conhecimento. Conglomera a teoria dos frames. A modularidade, a abstração, a passagem de mensagem, a herança, a classes e objetos propiciam um relevante poder à forma como o conhecimento pode ser simulado.
- Regras de produção: é a representação mais efetiva para exposições declarativas do conhecimento. São compostas por regras, que são as condições impostas no sistema.
- Redes semânticas: usa nomes dos objetos e valores em linguagem natural como: em um gráfico e arcos (ou ligações) indicando as relações entre os vários nós.

O modelo de domínio está fortemente relacionado ao modelo do aprendiz, a forma como o modelo de domínio é executada não é necessariamente a forma humana de resolver problemas, pois os humanos aplicam procedimentos apropriados para a resolução de problemas. Novos modelos de domínio têm surgido para simular a resolução humana mais próxima da realidade, agrupando informações reflexivas de fatos, técnicas e desempenho que os seres humanos empregam para estruturar seu conhecimento.

2.1.1.6 Módulo Tutor

Este módulo representa a estratégia de ensino e controle da intervenção pedagógica. O módulo de tutoria está firmemente acoplado com o modelo do aprendiz, no sentido em que todo o modelo do aprendiz é utilizado também no módulo de tutoria, possui as estratégias e táticas

de ensino. As estratégias compõem conhecimento sobre como gerar, a partir das informações de diagnóstico, monitoração e análise, uma sequência de táticas de ensino para apresentar um tópico a um determinado aluno. Uma estratégia de ensino deve definir:

1. Informações sobre quando interromper o curso de raciocínio ou a aprendizagem do aluno;
2. Informações sobre quais tópicos apresentar e a ordem de apresentação;
3. Informações sobre a forma e como apresentar o conteúdo.

Um método bastante utilizado é o chamado método socrático, que prevê situações de diálogo a partir de um fato conhecido do aluno, levando-o a aperfeiçoar este conceito através de explorações de contradições e da formulação de inferências corretas a partir do conhecimento inicial, onde o tutor ensina através de perguntas e diálogos, deixando o aluno tirar suas próprias conclusões. Outro modelo teórico utilizado é o modelo de treinamento (coaching) que utiliza atividades de entretenimento para transmitir conceitos relacionados, onde a aprendizagem é uma consequência indireta. Um terceiro modelo é o de hipertextos, onde o aluno navega em uma estrutura de hipertextos e explora o conteúdo segundo seus interesses, trabalhando de forma mais participativa e dinâmica.

Na metodologia deste módulo é incentivada-se para que exista um ciclo na tutoria, respeitando as seguintes etapas (COSTA; WERNECK, 1997):

1. Aceitar a resposta do aprendiz;
2. Comparar a resposta do aprendiz com a resposta sugerida pelo modelo do conhecimento;
3. Modelar o aprendiz;
4. Diagnosticar o nível de conhecimento do aprendiz;
5. Prover esta informação ao modelo da tutoria;
6. Executar a ordem proveniente do modelo de tutoria;
7. Proporcionar as possibilidades para a próxima execução.

Outras estratégias educacionais para a especificação da apresentação do material a ser ensinado, pode ser através do método de treinamento, em que é explorada a solução de problemas gerais através de jogos educacionais, e o método orientador, o qual executa por interação do aluno, quando este necessita de ajuda ou críticas.

2.1.1.7 Módulo de Comunicação

Com o objetivo de promover a operacionalidade do sistema, bem como torná-lo atrativo e motivador, intermediando a comunicação entre o computador e o aluno, o modelo de comunicação é essencial para o sucesso do sistema. Quando é executado o módulo proporciona um material instrucional e a monitoração do progresso do aluno por meio de suas respostas, realizando o mapeamento entre a representação interna do sistema contida nos módulos e traduzindo-a numa linguagem de interface compreendida pelo aluno.

O módulo oferece diversos recursos visuais em sua apresentação do conteúdo para que motive o aprendizado do aluno, como: sistemas de hipertexto e hipermídia que tem sido utilizado neste modelo de interface. Deve possibilitar facilidade na troca da iniciativa do diálogo onde o aluno possa intervir no discurso do tutor e vice-versa (o tempo de resposta deve estar dentro de limites aceitáveis e o monitoramento deve ser realizado de forma que não carregue o aluno com questionários excessivos).

Dimitracopoulou (1995) distingue dois tipos de interface:

- Primeira pessoa ou de manipulação direta: são interfaces gráficas que indicam a necessidade do uso do mouse. Oferecem ao usuário o efeito de poder trabalhar diretamente sobre o domínio. Utilizam a interação do sistema através de símbolos ou ícones já consagrados em outros softwares para que não necessitem de explicações detalhadas sobre como manuseá-las;
- Segunda pessoa: o controle e as ações do usuário sobre o domínio se fazem através de uma linguagem intermediária, como a linguagem natural e de comando, que executa as ações desejadas.

Diante dos modelos expostos, o elevado grau de interconexão entre os quatro componentes da arquitetura de um STI faz com que as técnicas utilizadas em um modelo sejam aplicadas em outros. O uso de um modelo cognitivo para verificar erros do aprendiz (módulo do aprendiz) pode apresentar conhecimento a ele (módulo de domínio) e também comunicar conhecimento real usado durante a solução de problemas relevantes, melhor que em situações abstratas (módulo pedagógico). As possibilidades de comunicar conhecimento (módulo de comunicação) também fornecem oportunidades para implementar estratégias educacionais, fornecendo fundamentos para o aprendiz durante os primeiros minutos de interação.

2.2 Java Embedded Object Production System - JEOPS

JEOPS é um motor de inferência para a linguagem de programação Java que define as regras para serem executadas sobre objetos. Ele proporciona ao Java o poder dos sistemas de produção, acrescentando um encadeamento de regras de produção de primeira ordem para que a linguagem, através de um conjunto de classes possa proporcionar uma programação declarativa. Com isso é facilitada o desenvolvimento de aplicações inteligentes, tais como agentes de software ou sistemas especialistas.

2.2.1 A evolução do JEOPS

O projeto foi idealizado em 1997 na disciplina de Inteligência Artificial Simbólica, na Universidade Federal de Pernambuco - UFPE, em sua primeira versão, tinha o nome de JEPS - *Java Embedded Production System*.

A medida que o JEPS era utilizado nas pesquisas dos alunos do mestrado, eram propostas correções de bugs e consequentemente um aumento na performance, gerando um novo algoritmo de unificação, em 1998 seu nome foi atualizado para JEOPS.

No ano de 1999 seu código foi refatorado para a linguagem inglesa, com a criação de um manual para os futuros usuários, manteve-se em constante atualização, através de regras pré-compiladas, *parser* otimizado para total integração com a linguagem Java e unificação entre regras e objetos até abril de 2007, onde teve sua última versão disponibilizada. Apesar do longo tempo sem revisões e ampliações, a escolha do JEOPS deve-se ao fato de ser uma ferramenta de domínio público, bem documentada e de aplicação relativamente simples.

2.2.2 Arquitetura do JEOPS

Segundo Santos e Ramalho (2000), as regras de produção são entidades de representação de conhecimento que não são compreendidas por um computador. Aos sistemas de produção compete a responsabilidade da compreensão dos comandos pela linguagem hospedeira. Existem duas opções distintas de se implementar esta conversão:

1. A interpretação das regras de produção através do motor de inferência é responsável por chamar os procedimentos específicos da linguagem, como: operações aritméticas, chamadas a métodos ou criação de novos objetos.

2. A pré-compilação das regras de produção para estruturas reconhecidas pela linguagem hospedeira.

Em computação, o termo compilação é usado para representar o processo de se transformar um programa escrito em uma determinada linguagem de programação em instruções que possam ser compreendidas pelo computador (código de máquina).

A compilação de regras acontece de estilo análogo à compilação de programas. A diferença é que o resultado da compilação das regras é um código fonte da linguagem hospedeira do sistema de produção, que por sua vez será mais uma vez compilado para sua execução. Por isso, acordou-se batizar este processo de pré-compilação das regras.

Na próxima etapa as regras serão compiladas utilizando o compilador da linguagem hospedeira em conjunto com o programa que as utiliza, para que seja gerado o código que será executado.

A Figura 2.4 ilustra este processo: as regras do arquivo texto onde foram escritas as regras (Base de Regras, arquivos “.rule”) são pré-compiladas, gerando o arquivo na linguagem hospedeira (Base de Regras “.java”).

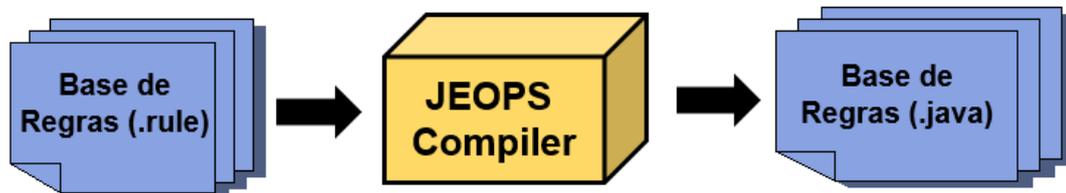


Figura 2.4 - Pré-compilação de regras
Fonte: Santos e Ramalho (2000)

Em seguida, o arquivo gerado e o arquivo que contém a implementação do agente são compilados como programas normais da linguagem Java, como demonstrado na Figura 2.5.

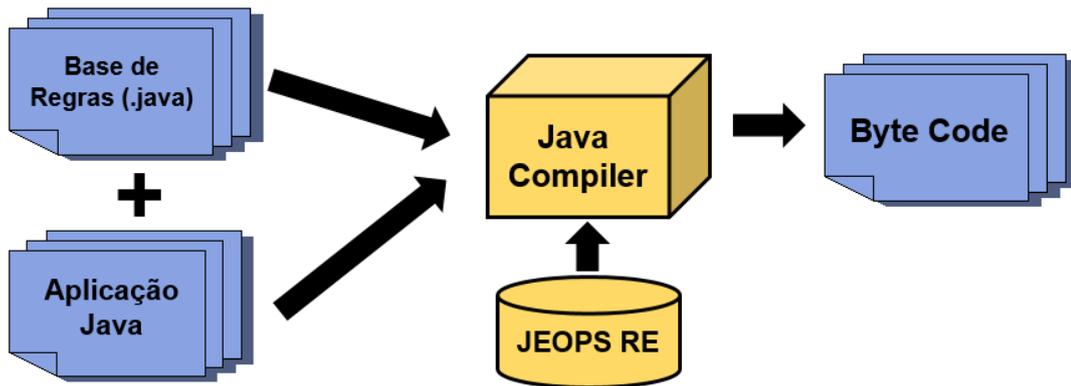


Figura 2.5 - Compilação da aplicação
Fonte: Santos e Ramalho (2000)

A pré-compilação das regras facilita a implementação do motor de inferência. Caso as regras fossem totalmente interpretadas, o próprio motor teria que realizar distintas tarefas que são de responsabilidade do compilador, como a avaliação de expressões aritméticas, a definição e a utilização de variáveis. O compilador gerando o código a ser executado, todas as vantagens obtidas pela otimização realizada pelo compilador são mantidas. Principalmente pelo fato de que o código gerado está num nível bem próximo da linguagem de máquina, é executado de forma mais eficiente que se estivesse em um nível mais alto.

2.2.3 As regras do JEOPS

De acordo com Santos e Ramalho (2000), no JEOPS é a sua base de conhecimentos quem encapsula os seus componentes, de modo a fornecer ao usuário uma porta de entrada única e simples para o motor de inferência. O usuário precisa acessar apenas os métodos principais definidos na classe abstrata `jeops.AbstractKnowledgeBase` para poder utilizá-la. São métodos de inserção e remoção de objetos na memória de trabalho, consulta dos objetos da mesma, reset e ativação da base de conhecimentos. Estes métodos são os seguintes:

- `assert(Object)`: Inserir um objeto na memória de trabalho da base de conhecimentos;
- `retract(Object)`: Remover um objeto da memória de trabalho da base de conhecimentos;
- `objects(String)`: Solicitar informações à base de conhecimentos e retornar todos os objetos da classe passada como parâmetro;
- `flush()`: Remover todos os fatos (objetos) da base de conhecimentos, limpar a história do conjunto de conflitos (se houver);

- `run()`: Ativar a base de conhecimentos, colocando o motor de inferência para funcionar.

A base interna de regras é o componente responsável por responder à base de conhecimentos, se as condições de uma regra são verdadeiras, quantas e quais regras foram definidas no arquivo pelo usuário, quais as declarações de cada regra. A base interna de regras é gerada a partir da pré-compilação do arquivo de regras.

No JEOPS o algoritmo Rete⁶, que é responsável pela memorização de avaliações feitas anteriormente, deixando de lado a utilização da rede para eliminar avaliações redundantes, apresenta dois benefícios em relação à maneira “tradicional” de casamento de padrões, na qual cada novo objeto inserido na base é verificado com todas as demais combinações de objetos já existentes para se encontrar unificações com as condições das regras.

O conjunto de conflitos é o local onde são armazenados todos os pares (regra, conjunto de objetos que a tornam ativa) da base de conhecimentos. O JEOPS, procura dissociar ao máximo a implementação dos conjuntos de conflitos do motor de inferência propriamente dito, de modo que o usuário possa definir suas próprias políticas de resolução de conflitos sem precisar compreender como funciona internamente o motor.

2.2.4 A escolha do JEOPS

Desde sua criação, diversos projetos vêm sendo desenvolvidos contendo como a ferramenta de inferência o JEOPS. As pesquisas atuam em diversas áreas, como: administração de redes, linguística computacional, jogos interativos, comércio eletrônico e recuperação de informação na Internet. Em seguida serão descritos alguns dos sistemas que utilizam o JEOPS:

- Sistema Mobilet: proposta de automatizar esta tarefa do operador de rede, que tem como responsabilidade proceder à limpeza dos discos do sistema, movendo arquivo para nós, com mais espaço, compactando arquivos pouco utilizados ou removendo arquivos que não são mais necessários, através de um agente móvel que percorre continuamente os nós da rede, realizando a verificação de problemas de espaço em disco.
- PubFinder: é um projeto de mestrado que propõe o desenvolvimento de um sistema para a recuperação de documentos de um domínio específico. Neste caso, são páginas

⁶ Rede Rete é o nome do algoritmo utilizado para realizar as inferências entre os fatos e as regras.

eletrônicas na Web que possuem citações a artigos, como por exemplo, páginas pessoais de pesquisadores, páginas de congressos, entre outras.

- Gerador de variações fonéticas: O projeto do gerador de variações fonéticas surgiu a partir do interesse de uma companhia telefônica em resolver este problema, basicamente, o sistema foi desenvolvido para ser acoplado em alguma ferramenta de busca (seja da Internet, seja de algum sistema interno, como é o caso do auxílio a lista), gerando todas as variações fonéticas de uma determinada palavra.
- Ferramenta de Ensino: Os projetos de pesquisa que utilizam da ferramenta JEOPS incluem equipes de robôs, reconhecimento de páginas com cifras musicais e classificação de páginas da rede.

Segundo Santos e Ramalho (2000), os autores das ferramentas descritas anteriormente deram um *feedback* positivo na utilização das mesmas.

3. ESTRUTURAÇÃO DO SISTEMA DE TREINAMENTO

Este sistema consiste de uma aplicação de um STI para o treinamento de profissionais da área da saúde que atuam em UTIN. Sua finalidade é auxiliar o instrutor na tarefa de treinamento de um novo usuário ou mesmo de usuários experientes com a finalidade de formação e atualização em relação ao modo de uso da incubadora neonatal em condições comuns e adversas. O protótipo abrange as seguintes funcionalidades:

- Apresentar aos profissionais da saúde todos os procedimentos operacionais da incubadora neonatal (especificações, funções, operações, alarmes, etc.) a partir de uma interface hipermídia e interativa;
- Avaliar o profissional através de questões que tratam de situações reais de instabilidades utilizando as informações contidas no banco de dados;
- Permitir ao profissional da saúde a interação através de uma interface amigável com o sistema de modo a facilitar o treinamento;
- Avaliar o aprendizado do profissional da saúde, de modo a identificar seus pontos fracos e viabilizar um treinamento nesses pontos.

A Figura 3.1 apresenta um diagrama de blocos do STI para UTIN, onde se pode observar seu relacionamento com os vários módulos.

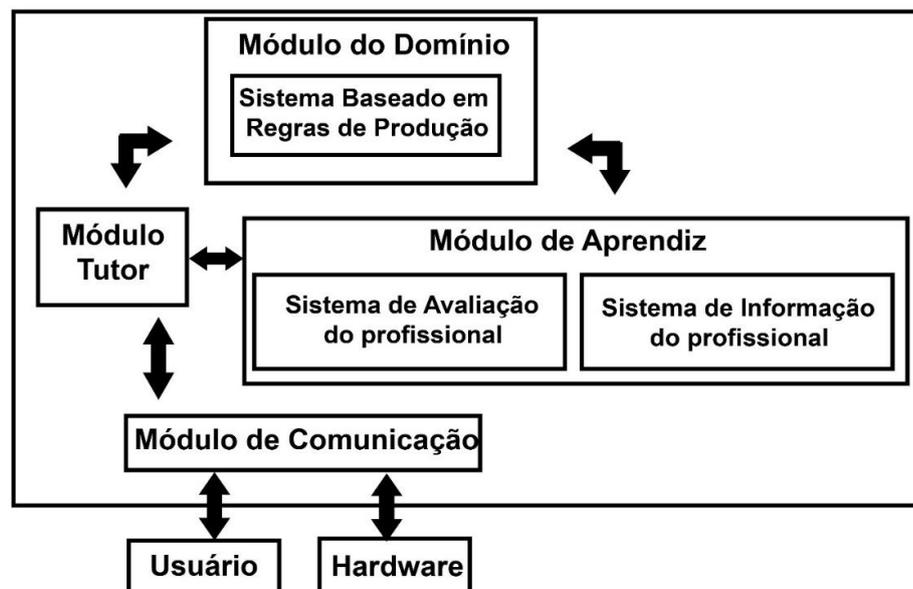


Figura 3.1 - Diagrama de blocos do STI
Fonte: McTaggart (2001) *apud* Gualberto *et al.* (2016)

3.1 Sistemas de Produção

Os sistemas de produção estão entre as formas mais utilizadas de modelar o raciocínio humano representado através do conhecimento declarativo, sendo definido como um conjunto de produções e um algoritmo de encadeamento progressivo⁷. A regra definida é disparada ao ser satisfeita as suas condições por fatos da memória de trabalho.

As regras de produção são utilizadas para conceber uma informação heurística sobre o mundo, explicitando um conjunto de ações que precisam ser efetivadas para uma certa ocasião. A regra é composta por uma parte de condições (também conhecida por parte “se” da regra) e uma parte de ações (a parte “então” da regra). Uma condição apresenta uma lista de símbolos ou variáveis, que necessitam ser padronizadas com os fatos da memória de trabalho. Este processo de unificação é realizado por um motor de inferência.

O ciclo de execução de um sistema de produção com encadeamento progressivo tem três passos principais:

- Unificação: É realizada a unificação dos fatos da memória de trabalho com as declarações das regras de modo a tornar verdade os predicados que compõem as condições das mesmas;
- Resolução de Conflitos: É definido um par <regra, fatos> do conjunto de conflitos, de acordo com a política de resolução definida;
- Disparo da Regra: As ações da regra selecionadas na etapa anterior são executadas com suas variáveis trocadas pelos fatos que modificou a regra disparável.

Este ciclo se repete até que não haja mais nenhuma regra disparável. Estas operações, que convencionou-se chamar de “assert” e “retract” respectivamente, são a base da manipulação de fatos da memória de trabalho.

3.2 Sistemática adotada para a aquisição do sistema

Para a obtenção do sistema foram identificados os problemas e soluções através do contexto geral da sistematização e engenharia de software. A análise de requisitos engloba todas

⁷ O algoritmo de encadeamento progressivo tem os fatos da base sobre os quais é considerado a condição das regras que representam informações cujo valor verdadeiro já foi estabelecido, acrescentando as conclusões das regras disparadas imediatamente na base de fatos.

as tarefas que lidam com investigação, definição e escopo de novos sistemas ou alterações, sendo uma parte importante do processo de desenvolvimento de softwares, que identifica as necessidades ou requisitos de um cliente. Uma vez que os requisitos do sistema tenham sido identificados, os projetistas de sistemas estarão preparados para projetar a solução.

A análise de requisitos é a primeira fase de desenvolvimento de software, dividido em Requisito funcional e Requisito não-funcional. É nesta fase que o analista faz as primeiras reuniões com os clientes e/ou usuários do software, para conhecer as funcionalidades do sistema que será desenvolvido. É nesta fase também que ocorre a maior parte dos erros, pois a falta de experiência dos clientes e/ou usuários faz com que eles nem sempre tenham conhecimento das funcionalidades do software.

O processo de desenvolvimento do Protótipo de um Sistema de Software (como produto) seguiu a Metodologia das Técnicas de Modelagem de Objetos (TMO) (ROSENBERG *et al.*, 1999), adaptada ao Padrão da Unified Modeling Language (UML) (BOOCH *et al.*, 2006) e a Metodologia do Rational Unified Process (RUP) (KRUCHTEN, 2003). Esta última sugere o desenvolvimento de um projeto em 4 fases:

1) Concepção: definir o escopo do projeto.

Nesta fase é estabelecido o caso de negócio do sistema, delimitado o escopo do projeto, identificadas todas as entidades externas (atores), com os quais o sistema vai interagir, identificado todos os casos de uso e todas as documentações pertinentes ao projeto como: visão geral dos principais requisitos, características chave, principais restrições, glossário, estimativa inicial dos riscos e plano de projeto apresentando as fases e iterações. Ao final da fase de concepção atinge-se o primeiro marco principal de progresso: *Lifecycle Objective*.

2) Elaboração: planejar, especificar as características e projetar a arquitetura.

Nesta fase de elaboração são estabelecidas as fundações arquiteturais para a aplicação e desenvolvido um plano de projeto, eliminado os elementos de alto risco para o projeto e obtendo um entendimento superficial de todo o sistema. Como resultado se obtém a descrição de arquitetura do software, um protótipo executável, uma lista de riscos revisada, um plano de desenvolvimento de todo o projeto e um manual de usuário preliminar, atingindo o segundo marco principal de progresso: *Lifecycle Architecture*.

3) Construção: construir o produto.

Nesta fase, o projeto tem o propósito de desenvolver os componentes restantes e as características da aplicação e integrá-los ao produto final, testar todas as características e otimizar custos e qualidade, obtendo com isto um produto integrado em plataforma adequada, manuais de usuários e descrição da versão atual, atingindo o terceiro marco principal de progresso que é a *Initial Operational Capability*.

4) Transição: transferir o produto para dentro da comunidade de usuários.

Seu propósito é transferir o sistema para os usuários, desenvolver novas versões do sistema (em razão de erros ou características que foram proteladas), certificar-se de que um nível aceitável de qualidade foi atingido e que a documentação do usuário está disponível, obtendo atingir o auto apoio do usuário, atingir as linhas bases de desenvolvimento de acordo com a visão do sistema, atingir as linhas base do produto rapidamente e de acordo com o custo estimado, concluindo o quarto marco principal de progresso: *Product Release*.

3.3 Cenário

De acordo com a especificação de requisitos fornecida pelo usuário, foram identificados inicialmente o cenário: enfoca a necessidade de uma solução informatizada para o treinamento de funcionários.

Normalmente quando um novo funcionário é admitido este deve ser treinado, de modo a obter a familiaridade necessária com os sistemas de controle do modelo de UTIN. Este treinamento, atualmente, via de regra, é realizado a partir de aulas convencionais em classes, onde um instrutor, normalmente um funcionário experiente, com o apoio de um pôster com a fotografia (ou desenho) dos instrumentos do modelo da UTIN, apresenta os procedimentos operacionais de rotina, bem como os procedimentos de emergência.

Estas aulas convencionais são onerosas para o hospital, visto que durante esta fase, um funcionário experiente (o instrutor) fica impedido de atuar na UTIN. A criação de um sistema de treinamento de funcionários da área de saúde, que ofereça um ambiente interativo rico em informações que abrangem todos os procedimentos operacionais e de emergência, pode vir a se tornar uma solução viável para as dificuldades, já que existem tecnologias suficientes para a conclusão de tal projeto.

Um STI diminui os custos envolvidos nesta fase, uma vez que fornece o treinamento ao funcionário a partir de tutoriais apoiados na utilização de ferramentas de desenvolvimento

hipermídia, sistemas de gerenciamento de banco de dados e uma ferramenta de acompanhamento da evolução do processo de aprendizado do funcionário.

Seguindo as metodologias de Engenharia de Software, no Apêndice A, encontra-se a Análise de Requisito que é a sentença que expressa as necessidades dos clientes e que condicionam a qualidade do software; a Análise Essencial (Apêndice B) que trata os eventos que o sistema deve reagir produzindo uma ação pré-determinada; os Caso de Uso (Apêndice C) que descrevem os requerimentos funcionais do sistema de forma consistente e clara e os diagramas de Sequência (Apêndice D) com o objetivo de estabelecer a interação dos objetos e seus relacionamentos dentro de um contexto. No Apêndice E, está o Diagrama de classes, que tem o objetivo de descrever os vários tipos de objetos no sistema e o relacionamento entre eles.

3.4 Regras para o JEOPS

As regras de criação são dadas através dos arquivos com extensão *.rules*. Dentro desse arquivo são construídas as regras de produção, que cria os objetos da classe que quer chamar. Com encadeamento progressivo, que visa prover a capacidade de raciocínio JAVA utilizando regras de produção do tipo SE <condição> ENTÃO <ação>. O uso dessas regras em um ambiente baseado em conhecimento visa simular o modelo de cognição humano. Regra Exemplo:

Se:

Aprendiz obteve nota 100 na avaliação

Então:

Obteve desempenho máximo no assunto X

Uma regra em JEOPS é basicamente construída da seguinte forma:

- Regra: define o nome da regra;
- Declarações: criação de um objeto de classe, especifica os objetos que serão utilizados nas condições e/ou ações.
- Condições: São definidos o corpo de uma regra. Os métodos chamados que irá suportar as condições previstas; podendo ser qualquer expressão Java que retorne um valor boolean.

- Ações: podem ser inseridos quaisquer comandos Java, no entanto estes serão executados somente se todas as condições forem verdadeiras.

Para a representação do conhecimento, as regras de produção mostraram-se adequadas para a ferramenta, essa abordagem representa os fatos na memória de trabalho sob a forma de objetos e traz vantagens, tanto ligadas a conceitos de qualidade apresentados pela Engenharia de Software (reusabilidade, modularidade, legibilidade), como relativas à engenharia de conhecimento, permitindo representar de uma forma clara e simples com a ajuda de JEOPS.

Na Figura 3.2 tem um exemplo de como as regras são escritas, e pode-se observar a sintaxe semelhante ao Java, está a ser criado um objeto da classe AVALIAÇÃO_BASE, onde os seus métodos tornam acessíveis para escrever as regras e servem para calcular o desempenho de uma avaliação realizada.

```

5  rule CasoBase {
6
7      declarations
8          AVALIACAO a;
9
10     conditions
11         a.estaCompleto() == true;
12
13     actions
14         a.calculaPercentagens();
15         a.setNoNota();
16         a.setNoNQuestoes();
17         a.adicionaNos();
18
19 }
20

```

Figura 3.2 - Regra do JEOPS para calcular o desempenho do aprendiz na avaliação
Fonte: Elaborado pelo Autor

Nesta regra é declarada a avaliação que está sendo analisada, a condição desta regra é a avaliação esta realizada por completo, então é executado o cálculo de porcentagens de acerto, a nota relacionada a quantidade de acerto do aprendiz, o número de questões total da avaliação, e um nó é adicionado, cada nó é relacionado a um assunto e um sub-assunto, esse nó é representado visivelmente para o instrutor e aprendiz no programa na janela de desempenho.

```

21 rule MudaAssunto {
22
23     declarations
24         AVALIACAO a;
25
26     conditions
27         a.estaIncompleto();
28
29     actions
30         ASSUNTO_BASE ab = new ASSUNTO_BASE();
31         ab.insert(a.getAssunto());
32         ab.run();
33         a.incrementaIndex();
34         modified(a);
35 }
36
21 rule MudaSubAssunto {
22
23     declarations
24         ASSUNTO a;
25
26     conditions
27         a.estaIncompleto();
28
29     actions
30         SUB_ASSUNTO_BASE sab = new SUB_ASSUNTO_BASE();
31         sab.insert(a.getSubAssunto());
32         sab.run();
33         a.incrementaIndex();
34         modified(a);
35 }
36

```

Figura 3.3 - Regra do JEOPS para calcular o desempenho em cada assunto e sub-assunto.
Fonte: Elaborado pelo Autor

Na Figura 3.3 é demonstrada a regra para que muda assunto e o sub-assunto aplicado na avaliação, para que assim seja calculada as estáticas de cada assunto e sub-assunto cadastrado no sistema de banco de dados e que foi aplicado na avaliação.

Ao nível do sub-assunto é aplicado a regra para as questões certas e erradas, como demonstrado na Figura 3.4, com esses dados é calculado o desempenho do aprendiz, em diversos níveis de aprofundamento no conteúdo.

```

36 rule QuestaoErrada {
37
38     declarations
39         SUB_ASSUNTO s;
40
41     conditions
42         s.estaIncompleto();
43         s.getQuestao().estaCorreta() == false;
44
45     actions
46         s.incrementaIndex();
47         s.incrementaNQuestoes();
48         modified(s);
49 }
50
20 rule QuestaoCerta {
21
22     declarations
23         SUB_ASSUNTO s;
24
25     conditions
26         s.estaIncompleto();
27         s.getQuestao().estaCorreta();
28
29     actions
30         s.incrementaIndex();
31         s.incrementaNQuestoes();
32         s.incrementaAcertos();
33         modified(s);
34 }
35

```

Figura 3.4 - Regra do JEOPS para calcular a quantidade de questão certa e errada.
Fonte: Elaborado pelo Autor

O resultado destas regras de inferência é a árvore de desempenho do aluno que é demonstrado na Figura 3.5, onde observa-se:

- O número da avaliação;
- O nome do aprendiz;
- A nota do aprendiz na avaliação;
- O total de questões na avaliação;
- A nota e a quantidade de questões por assunto;
- A nota e a quantidade de questões por sub-assunto.

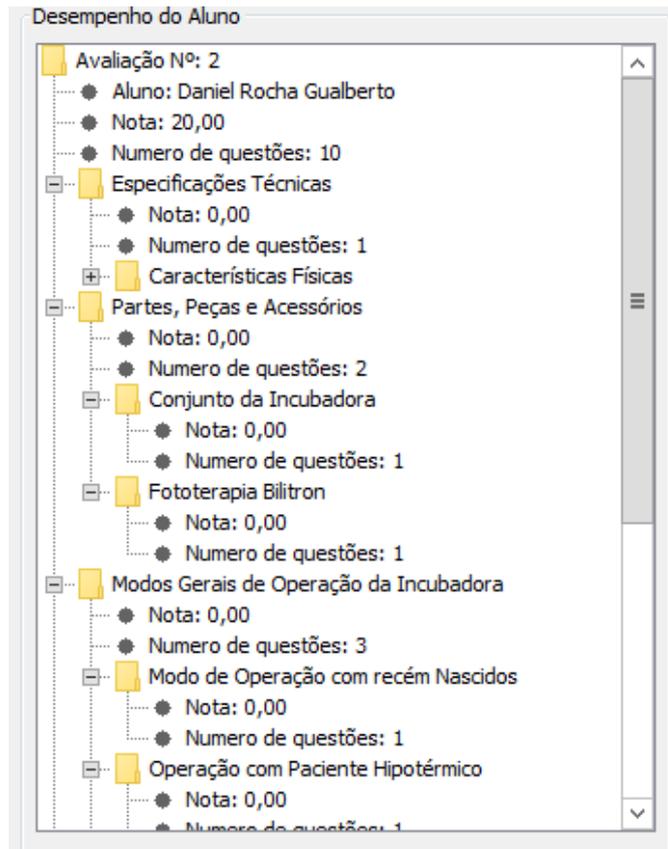


Figura 3.5 - Arvore de desempenho do aprendiz
 Fonte: Elaborado pelo Autor

O objetivo das regras, representado na Figura 3.6, é fornecer o relatório detalhado, com as informações escritas adequadamente ao relatório, sendo dividido em três níveis de desempenho em relação a nota:

- Se a nota do aprendiz for igual a 100, é escrito no relatório que o aluno obteve a nota máxima na avaliação.
- Se a nota do aprendiz for menor que 100 e maior que a média, é escrito no relatório que o aluno obteve desempenho satisfatório na avaliação.
- Se a nota do aprendiz for menor que a média, é escrito no relatório que o aluno obteve desempenho abaixo da média.

```

5  rule regraConhecimentoPleno {
6
7      declarations
8          AVALIACAO a;
9
10     conditions
11         a.getNota() == 100;
12
13     actions
14         a.setRelatorio("Aluno: " + a.getNome() + "\n");
15         a.setRelatorio("Obteve nota máxima na avaliação.\n");
16
17 }
18
19 rule regraConhecimentoMedio {
20
21     declarations
22         AVALIACAO a;
23
24     conditions
25         a.getIndex() == -1;
26         a.getNota() < 100;
27         a.getNota() >= a.getMedia();
28
29     actions
30         a.setRelatorio("Aluno: " + a.getNome() + "\n");
31         a.setRelatorio("Obteve desempenho satisfatorio na avaliação.\n");
32         a.incrementaIndex();
33         modified(a);
34
35 }
36
37 rule regraConhecimentoInsatisfatorio {
38
39     declarations
40         AVALIACAO a;
41
42     conditions
43         a.getIndex() == -1;
44         a.getNota() <= a.getMedia();
45
46     actions
47         a.setRelatorio("Aluno: " + a.getNome() + "\n");
48         a.setRelatorio("Obteve desempenho abaixo da média na avaliação.\n");
49         a.incrementaIndex();
50         modified(a);
51
52 }

```

Figura 3.6 – Regra do JEOPS para emissão do relatório de desempenho da avaliação
 Fonte: Elaborado pelo Autor

Assim como na árvore de desempenho do aluno, no relatório há uma avaliação individual para cada assunto aplicado na avaliação, para que o relatório apresentasse informações mais objetivos, apenas o resultado diferente da nota total é apresentado no relatório, para que seja feito essa análise foi elaborada a regra da Figura 3.7

```

54 rule regraPassaAssuntos {
55
56     declarations
57         AVALIACAO a;
58
59     conditions
60         a.estaIncompleto();
61         a.getNota() < 100;
62
63     actions
64         RELATORIO_ASSUNTO_BASE rab = new RELATORIO_ASSUNTO_BASE();
65         rab.insert(a.getAssunto());
66         rab.insert(a);
67         rab.run();
68         a.incrementaIndex();
69         modified(a);
70
71 }

```

Figura 3.7– Regra do JEOPS para emissão do relatório de desempenho da avaliação
 Fonte: Elaborado pelo Autor

O objetivo das regras, representado na Figura 3.8, é fornecer o relatório detalhado por assunto, sendo dividido em dois níveis de desempenho em relação a nota:

- Se a nota do aprendiz for menor que 100 e maior que a média no assunto “X”, é escrito no relatório que o aluno obteve desempenho satisfatório no específico assunto.
- Se a nota do aprendiz for menor que a média no assunto “X”, é escrito no relatório que o aluno obteve desempenho abaixo da média no específico assunto.

Com o nível de sub-assunto trata-se de um número menor de questões foi elaborado a regra para representar apenas os resultados abaixo da média, conforme demonstrado na Figura 3.9, o objetivo dessas regras é fornecer o relatório detalhado por sub-assunto, no caso do aprendiz apresenta desempenho inferior à média:

- Se a nota do aprendiz for menor que a média no sub - assunto “X”, é escrito no relatório que o aluno obteve desempenho abaixo da média no específico sub-assunto.

```

20 rule regraConhecimentoSatisfatorio {
21
22     declarations
23         ASSUNTO assunto;
24         AVALIACAO av;
25
26     conditions
27         assunto.getIndex() == -1;
28         assunto.getNota() < 100;
29         assunto.getNota() >= assunto.getMedia();
30         assunto.subAssuntoMediaBaixa();
31
32     actions
33         av.setRelatorio("\nObteve desempenho satisfatório no assunto: ");
34         av.setRelatorio(assunto.getTitulo() + "\n");
35         assunto.incrementaIndex();
36         modified(assunto);
37
38 }
39
41 rule regraConhecimentoInsatisfatorio {
42
43     declarations
44         ASSUNTO assunto;
45         AVALIACAO av;
46
47     conditions
48         assunto.getIndex() == -1;
49         assunto.getNota() <= assunto.getMedia();
50
51     actions
52         av.setRelatorio("\nObteve desempenho abaixo da média no assunto: ");
53         av.setRelatorio(assunto.getTitulo() + "\n");
54         assunto.incrementaIndex();
55         modified(assunto);
56

```

Figura 3.8 - Regra do JEOPS para emissão do relatório de desempenho da avaliação, por assunto.
 Fonte: Elaborado pelo Autor

```

rule regraPassaSubAssuntos {
    declarations
        ASSUNTO assunto;
        AVALIACAO av;

    conditions
        assunto.estaIncompleto();
        assunto.getNota() < 100;

    actions
        RELATORIO_SUBASSUNTO_BASE rsb = new RELATORIO_SUBASSUNTO_BASE();
        rsb.insert(assunto.getSubAssunto());
        rsb.insert(av);
        rsb.run();
        assunto.incrementaIndex();
        modified(assunto);
}

rule regraConhecimentoInsatisfatório {
    declarations
        SUB_ASSUNTO subAssunto;
        AVALIACAO avaliacao;

    conditions
        subAssunto.getNota() < avaliacao.getMedia();

    actions
        avaliacao.setRelatorio(" Obteve desempenho abaixo");
        avaliacao.setRelatorio(" da média no sub assunto: ");
        avaliacao.setRelatorio(subAssunto.getTitulo() + "\n");
}

```

Figura 3.9 - Regra do JEOPS para emissão do relatório de desempenho da avaliação, por sub-assunto.
Fonte: Elaborado pelo Autor

O resultado destas regras de inferência é o relatório de desempenho gerado automaticamente que é demonstrado na Figura 3.10.

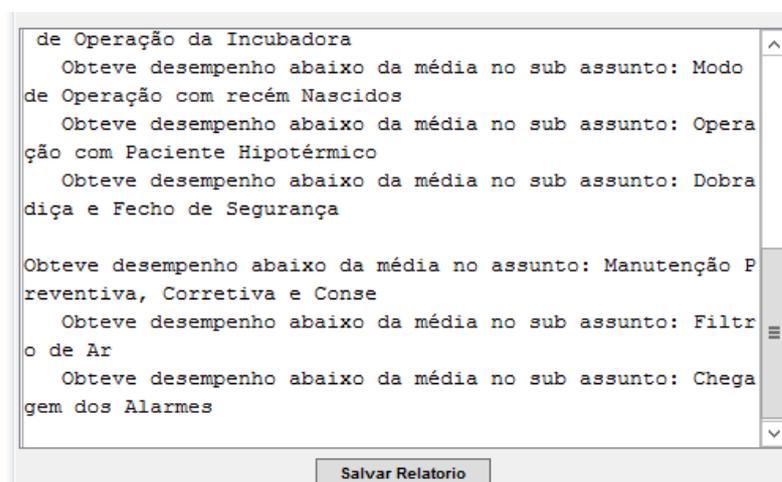


Figura 3.10 - Relatório de desempenho da avaliação.
Fonte: Elaborado pelo Autor

4. IMPLEMENTAÇÃO DO FRAMEWORK

A implementação do framework proposto foi efetivada em Java, empregando classes e objetos disponibilizados pela linguagem. A escolha da linguagem de programação levou em apreço as facilidades oferecidas pela mesma, sua abrangência no mercado e a afinidade do implementador. As diferentes janelas de interação com o usuário foram elaboradas utilizando ferramentas da própria linguagem, objetivando construir uma ferramenta simples e eficaz. A tela inicial da ferramenta construída é apresentada na Figura 4.1.



Figura 4.1 - Tela inicial do Sistema de Treinamento de UTIN
Fonte: Elaborado pelo Autor

Após os dados do usuário serem validados com sucesso pelo sistema é disponibilizada a janela com as opções do programa para iteração do usuário, como pode ser visto na Figura 4.2, as opções são:

- Aba de mensagem de boas-vindas.
- Aba de tutorias e exercícios: nesta área o aprendiz pode estudar através de tutoriais e exercícios.
- Aba de gerenciamento de usuários: nesta área o administrador pode adicionar, alterar, pesquisar ou remover usuários.
- Aba de gerenciamento de questões: nesta área o administrador pode adicionar assuntos, sub assuntos, e novas questões ao programa.

- Aba de avaliações: nesta área é possível realizar as avaliações e pesquisar sobre os desempenhos dos alunos.
- Aba sobre o sistema: ajuda sobre como funciona o sistema.

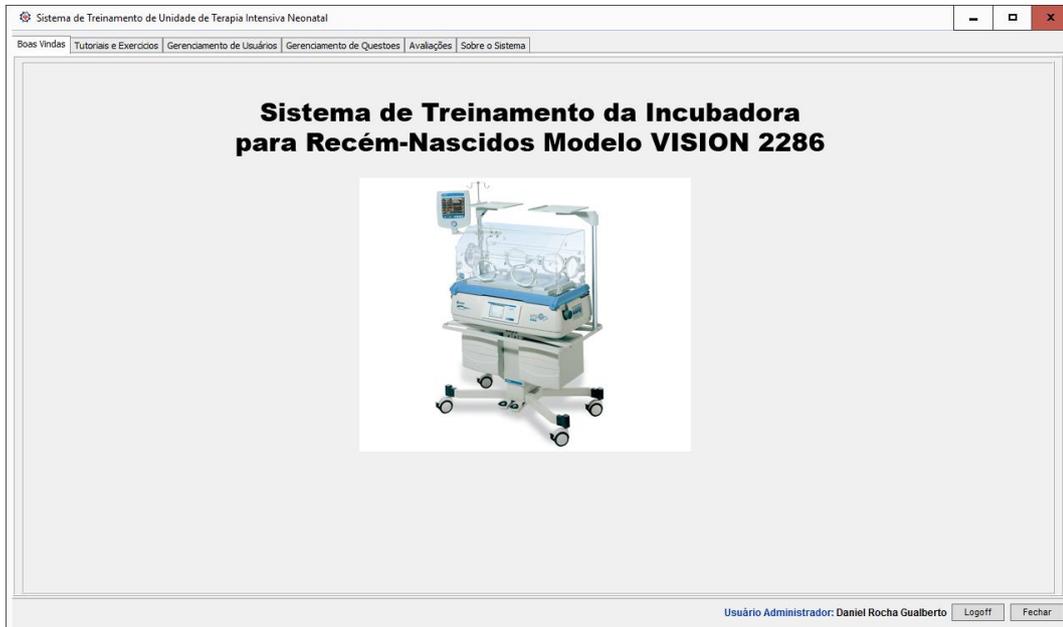


Figura 4.2 - Tela do sistema após o usuário logar com sucesso.

Fonte: Elaborado pelo Autor

Ao clicar na aba 'Tutorial e exercícios' é visualizado os tutoriais, sons de alarmes, vídeos de treinamento e exercícios cadastrado no sistema, conforme a Figura 4.3.

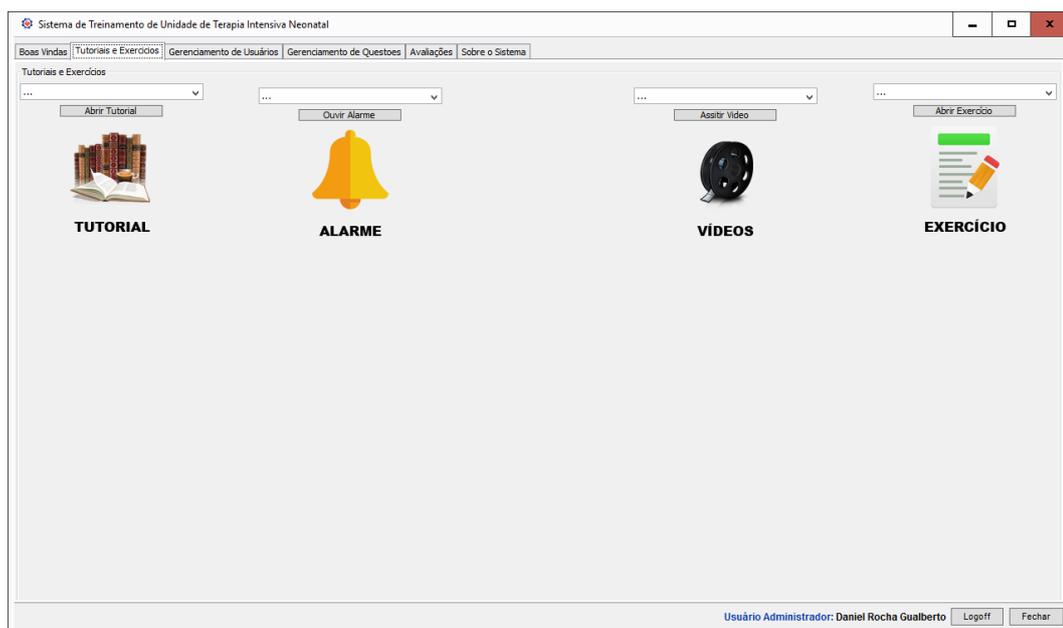


Figura 4.3 - Tela que representa a aba tutoriais e exercícios

Fonte: Elaborado pelo Autor

Ao seleccionar um item do tutorial através da caixa de lista é aberto uma área para sua leitura, com botões de página anterior, próxima página, ir para uma página determinada e fechar o conteúdo, conforme representa a Figura 4.4.



Figura 4.4 - Aba tutoriais e exercícios: abrir tutorial

Fonte: Elaborado pelo Autor

Ao seleccionar um item da caixa de lista de alarmes é executado o áudio, com o som do alarme escolhido (Figura 4.5). Na Figura 4.6, observa-se a caixa de lista de vídeos, que ao ser escolhido, é executado o vídeo através do Player VLC. O Player VLC foi escolhido pela possibilidade de ser instalado em os sistemas operacionais (Windows, MAC e Linux).

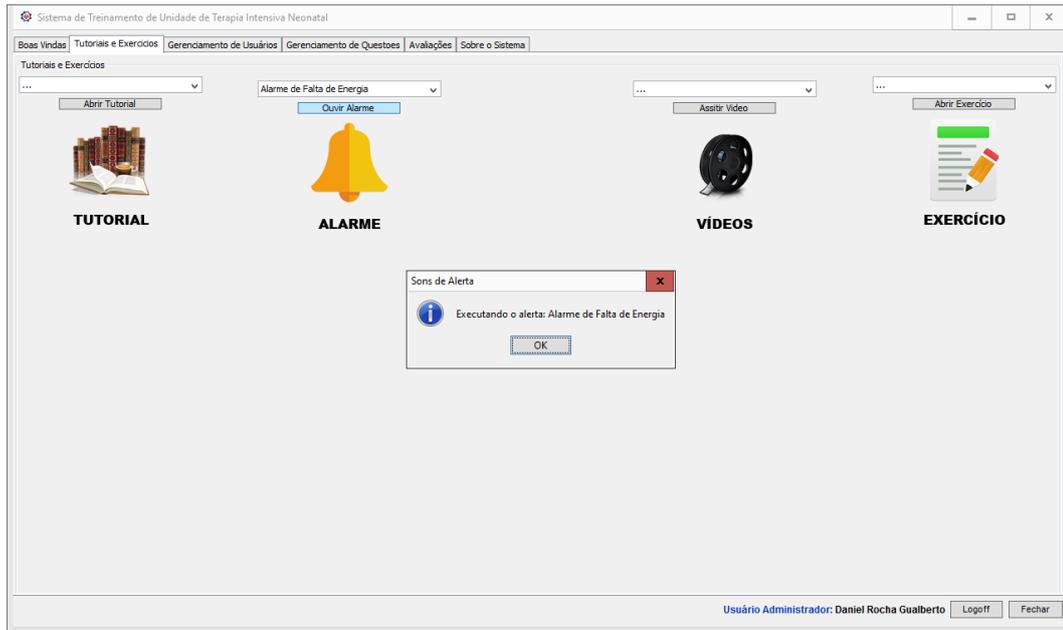


Figura 4.5 - Aba tutoriais e exercícios: ouvir alarme
 Fonte: Elaborado pelo Autor

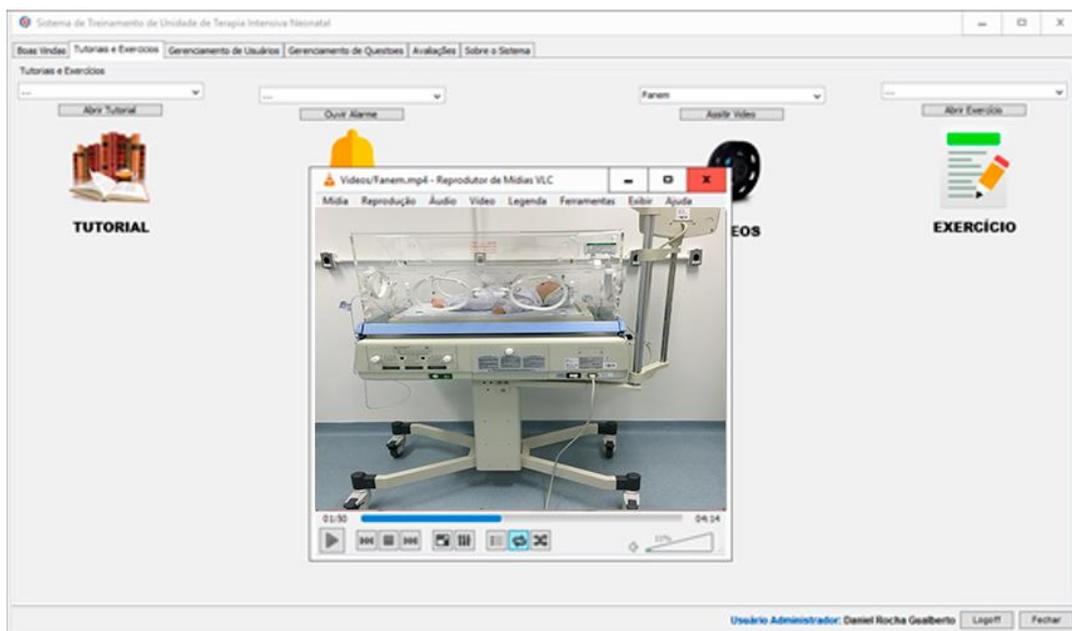


Figura 4.6 - Aba tutoriais e exercícios: assistir vídeo
 Fonte: Elaborado pelo Autor

Se selecionar um item da “caixa de lista de exercício” é aberto uma área para sua leitura, com botões de página anterior, próxima página, ir para uma página determinada e fechar o conteúdo, conforme representa a Figura 4.7.

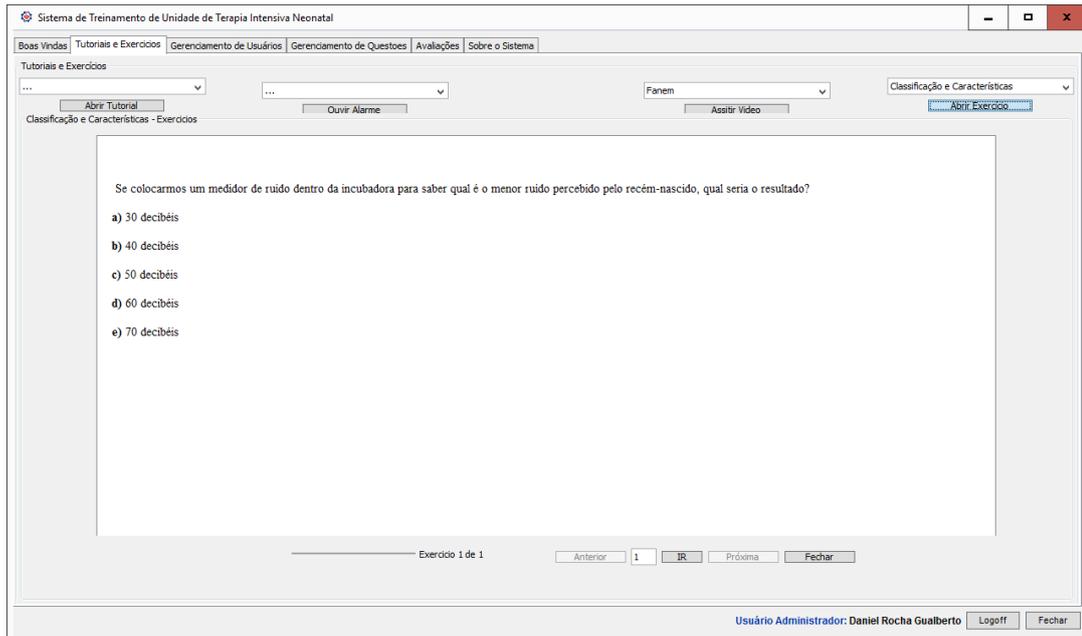


Figura 4.7 - Aba tutoriais e exercícios: abrir exercício
 Fonte: Elaborado pelo Autor

Através da aba gerenciamento de usuário, visto na Figura 4.8, o administrador pode gerenciar os dados dos usuários, nos botões pesquisar, alterar e excluir, ao clicar, é visualizado uma caixa de pesquisa que é necessário digitar o nome do usuário para buscar os dados e depois serem representados (Figura 4.9).

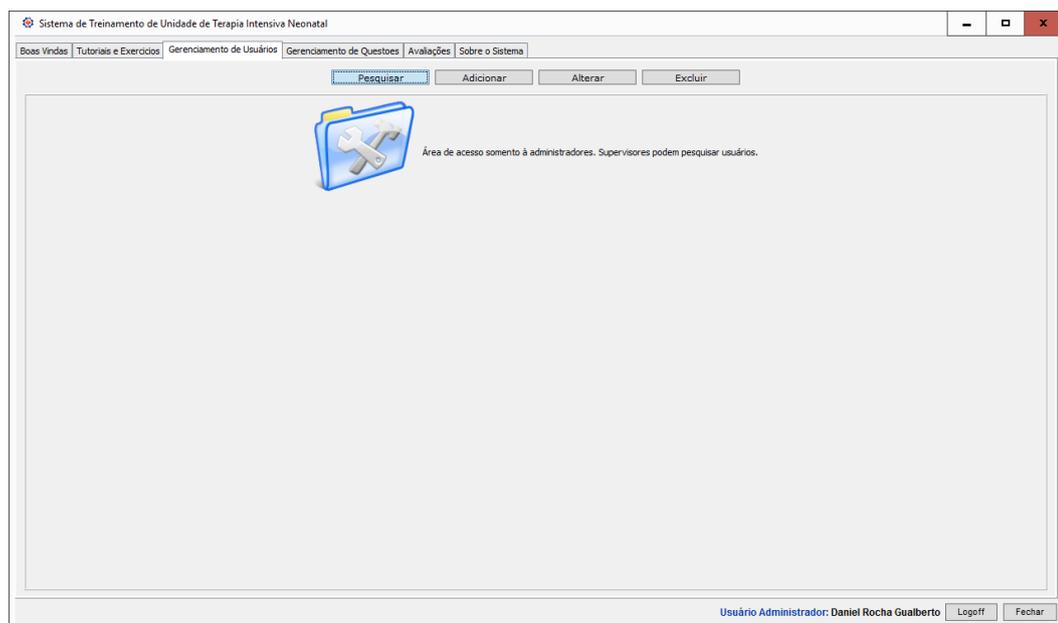


Figura 4.8 - Aba gerenciamento de usuários
 Fonte: Elaborado pelo Autor

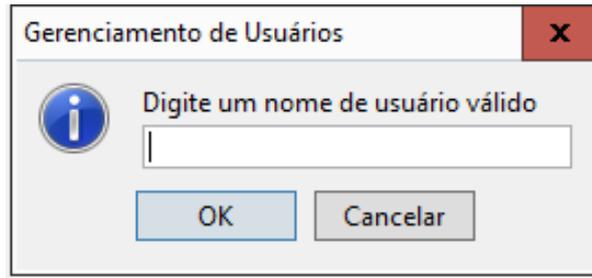


Figura 4.9 - Gerenciamento de usuários
Fonte: Elaborado pelo Autor

Se o administrador clicar em “adicionar” serão disponibilizados quatro campos para preencher com a informação do usuário: o nome do usuário, o login para acesso, a senha do usuário, e o tipo do usuário, que pode ser administrador, instrutor ou aluno, e após o preenchimento deve clicar no botão “Adicionar” localizado no canto inferior direito (Figura 4.10).

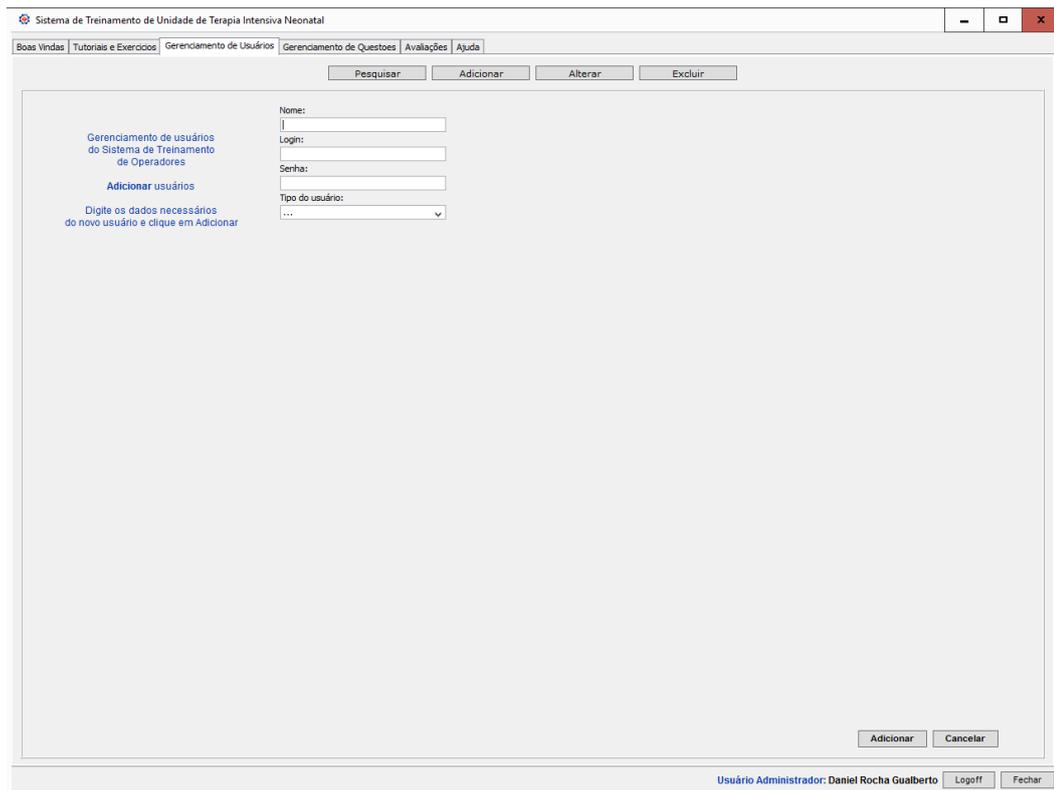


Figura 4.10 - Aba gerenciamento de usuários: adicionar
Fonte: Elaborado pelo Autor

No gerenciamento de questões é disponibilizado opções para que classifique e acrescente novas questões, conforme é demonstrado na Figura 4.11.

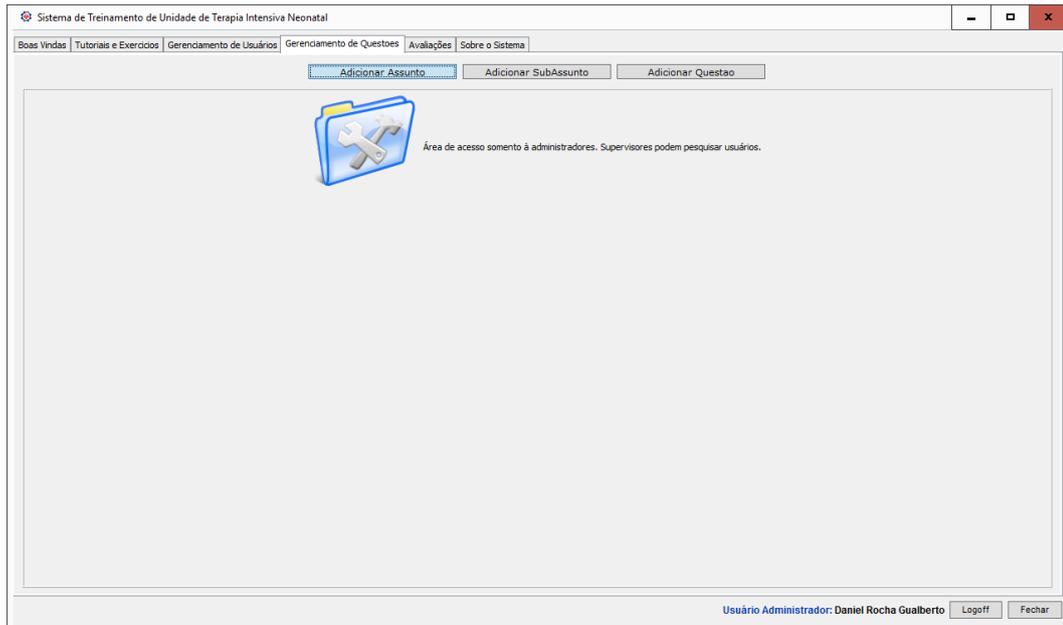


Figura 4.11 - Aba gerenciamento de questões
Fonte: Elaborado pelo Autor

Ao clicar no botão “Adicionar Questão” o administrador deve definir o assunto, o sub assunto, o nome do arquivo “.html” e o gabarito da questão. Ao final deve-se clicar no botão “Adicionar” localizado no canto inferior direito (Figura 4.12).

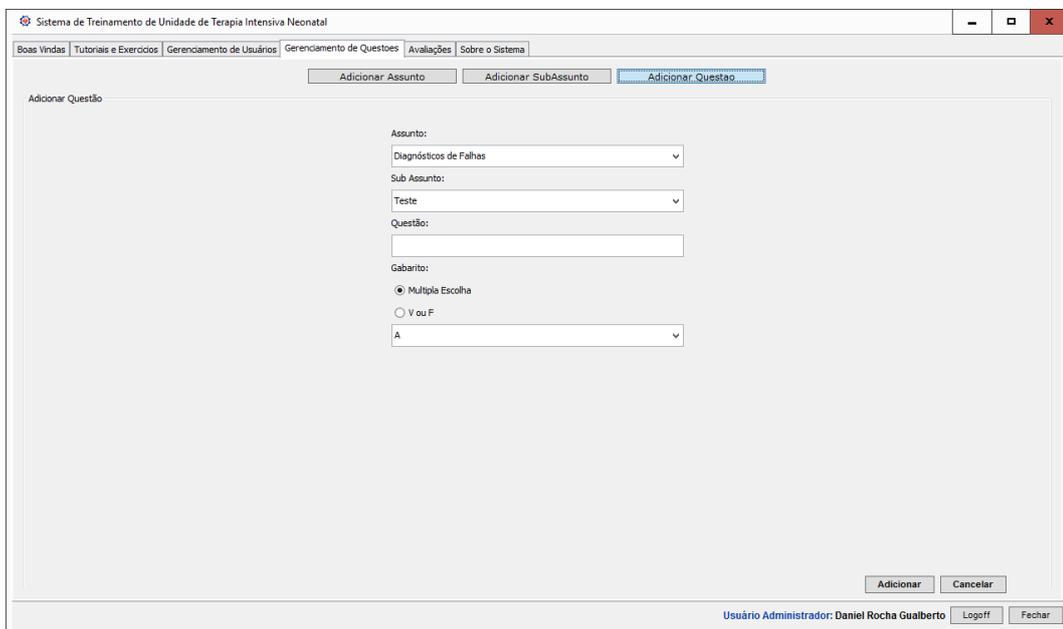


Figura 4.12 - Aba gerenciamento de questões: adicionar questão
Fonte: Elaborado pelo Autor

Na aba avaliações, encontram-se as avaliações disponíveis e o relatório das avaliações já realizadas (Figura 4.13). Quando o aprendiz clicar no botão “Iniciar Avaliação” uma janela

será criada, o qual o instrutor deve definir qual será a quantidade de questões que conterà na avaliação, e preencher o seu login e sua senha, para liberar o início do teste.

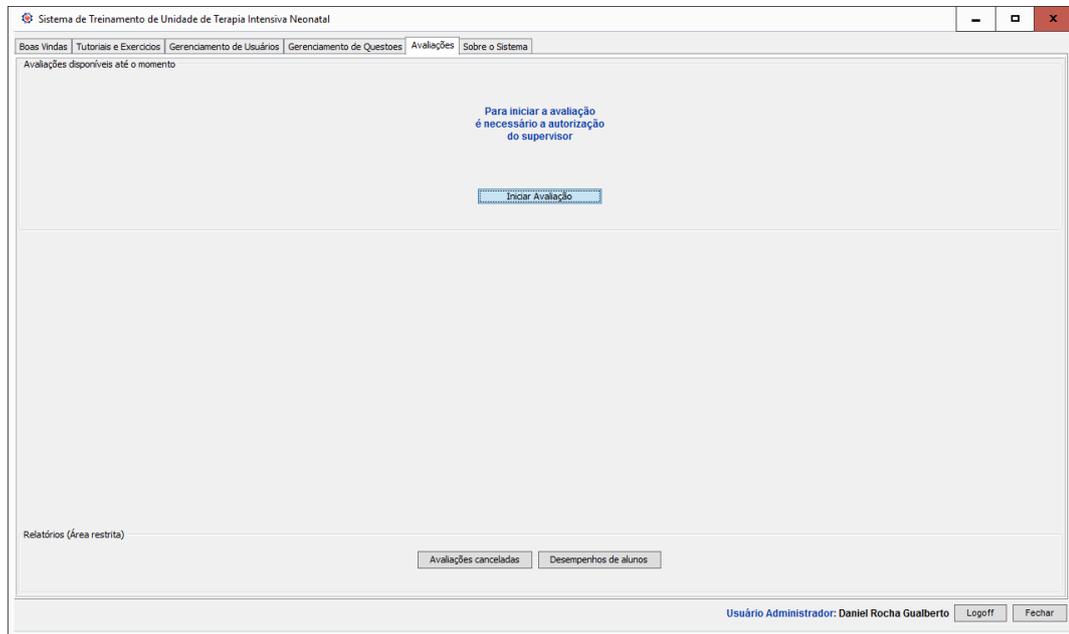


Figura 4.13 - Aba avaliações
Fonte: Elaborado pelo Autor

A janela da avaliação é representada na Figura 4.14. No canto esquerdo o aprendiz visualiza a quantidade de questões, e se ela já foi respondida ou não. No centro da janela fica disponível a questão em conjunto com as alternativas. Ao fim da janela está a área destinada a marcação das questões. Após responder todas as alternativas, deve-se clicar no botão “concluir”, assim é liberado o gabarito de cada questão. A resposta é visualizada na área superior da janela, ao clicar na questão que deseja saber a resposta.

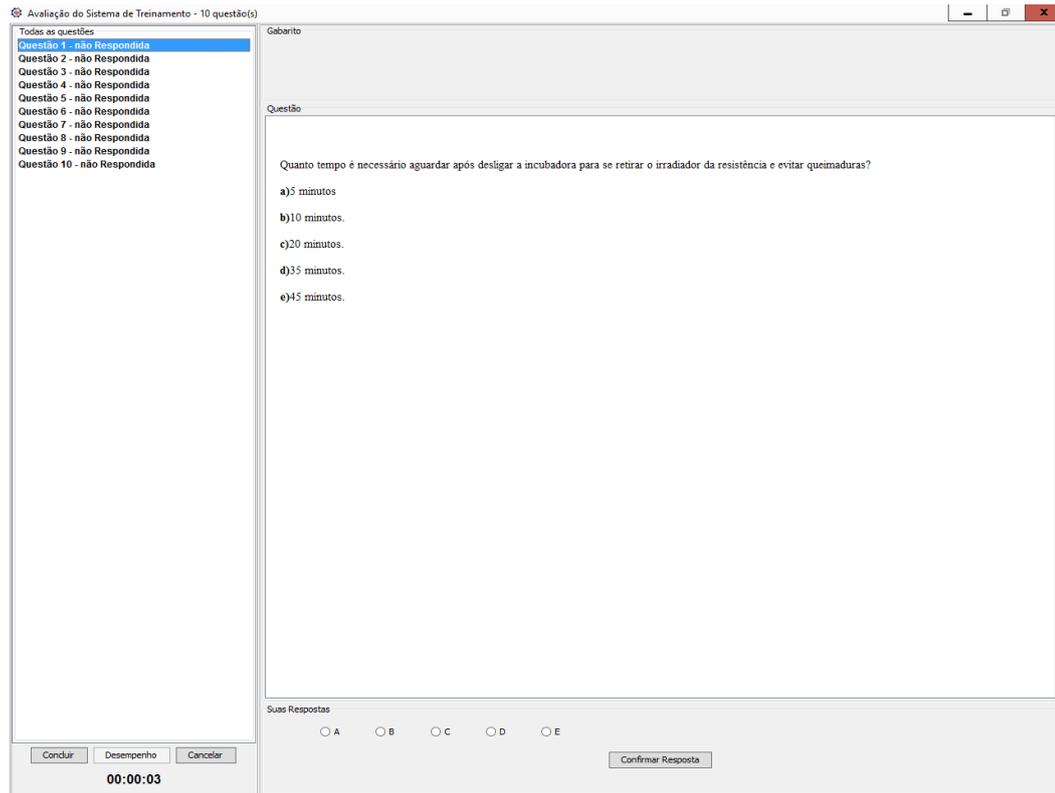


Figura 4.14 - Tela da avaliação
 Fonte: Elaborado pelo Autor

O botão “Desempenho” fica ativado após o término da avaliação. Nele contém todas as informações para o *feedback* para o aprendiz a respeito do seu conhecimento sobre o assunto tratado. Como mostra a Figura 4.15 é possível visualizar o tempo gasto para realizar a avaliação, o total de questões, quantas foram respondidas corretamente e quantas incorretamente e a porcentagem de aproveitamento em relação às questões respondidas. Um relatório é gerado de cada sub assunto tratado na avaliação, comentando como está o aprendiz em relação ao mesmo. À direita está um compilado de todas as informações em relação ao assunto e o sub assunto, com o número de questões e a nota.

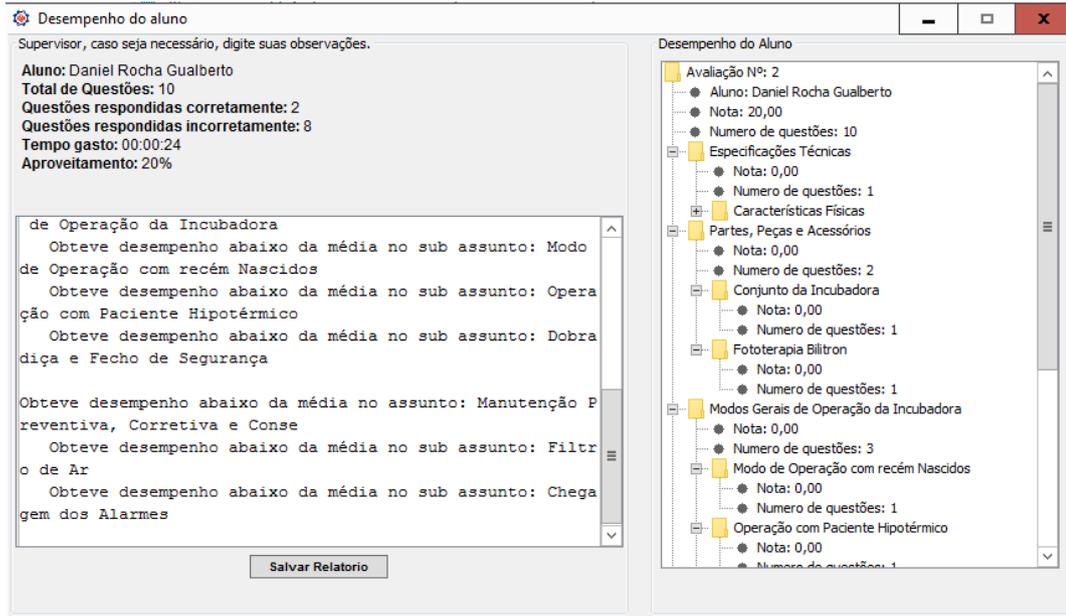


Figura 4.15 - Tela do desempenho do aluno
Fonte: Elaborado pelo Autor

Na aba “Sobre o Sistema”, encontra-se a ajuda do sistema, conforme demonstrada na Figura 4.16.



Figura 4.16 - Aba sobre o sistema
Fonte: Elaborado pelo Autor

5. APLICAÇÃO DO FRAMEWORK

O sistema proposto visa auxiliar o treinamento de um novo operador de incubadora neonatal em sua base de conhecimento. Além disso, o framework deve ser capaz de oferecer o *feedback* em relação ao desempenho do usuário possibilitando determinar seus atributos e também as métricas de similaridade que serão utilizadas na recuperação dos casos e adaptação dos mesmos.

Um relatório dos resultados da aplicação de uma Análise Heurística⁸ para detectar potenciais problemas de usabilidade durante um processo de *design* interativo e a interface do Sistema de Treinamento da Incubadora foi realizado pelos pesquisadores e orientados do Laboratório de Usabilidade e Fatores Humanos da UNIFEI. É uma alternativa de avaliação rápida e de baixo custo, quando comparada a métodos empíricos.

As heurísticas utilizadas para esta avaliação de usabilidade foram as de Nielsen (NIELSEN JAKOB, 1994). Segundo Nielsen “o objetivo da avaliação heurística é encontrar os problemas de utilização na concepção de modo que eles podem ser atendidos como parte de um processo iterativo de design.”. A heurística objetiva buscar a mais perfeita solução para as questões de um certo problema. Isto é, ela por si só pretende a busca do melhor resultado e desempenho sublime em um ambiente onde ela é empregada, tendo como finalidade tornar o sistema mais palatável e de maior compreensão para quem é designado.

Os resultados apontaram os possíveis problemas que o usuário final possa vir a ter. Soluções foram incorporadas nesta versão já corrigida do sistema. No entanto, considerando que este método é realizado por especialistas em usabilidade, a validação final só pode ser feita com o usuário final. O relatório está disponível no Apêndice F deste documento.

O Sistema Tutor Inteligente elaborado visa auxiliar novos operadores da incubadora Neonatal Modelo VISION 2286, a desenvolverem habilidades para manuseio e correta operação e entendimento de suas funções.

O sistema de treinamento conta com o módulo de treinamento teórico e o módulo de avaliação. O primeiro módulo constitui um sistema de ensino, através de leitura, vídeo aulas, e exercícios que visa apresentar ao aluno as características e suas diferentes funções do modelo. Esse módulo mostra ao usuário a documentação e material informativo sobre a incubadora

⁸ A análise heurística consiste em submeter a interface de um determinado aplicativo à avaliação de especialistas em usabilidade, conforme um conjunto previamente determinado de “bons princípios de usabilidade.

neonatal, oferecendo a ele o conhecimento necessário para operar o equipamento com mais segurança.

O módulo de ensino teórico utiliza diferentes recursos audiovisuais para aplicar o ensino, conforme apresentado na Figura 5.1. Cada tela de apresentação de conteúdo é dotada de texto descritivo sobre o equipamento, sons dos próprios alarmes sonoros da incubadora, vídeos com intuito de contribuir com o ensino, e exercícios para desenvolver o conhecimento. Os diferentes recursos de multimídia visam explorar as habilidades potenciais que um aluno em particular possui.

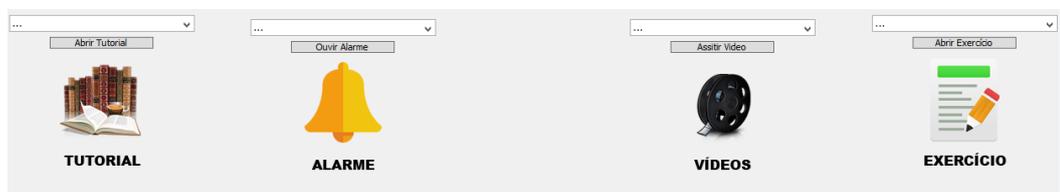


Figura 5.1 - Recursos teóricos audiovisuais
Fonte: Elaborado pelo Autor

No módulo de avaliação encontram-se os questionários de mensuração de aprendizado do aluno, conforme apresentado na Figura 4.14 e está em destaque na Figura 5.2.

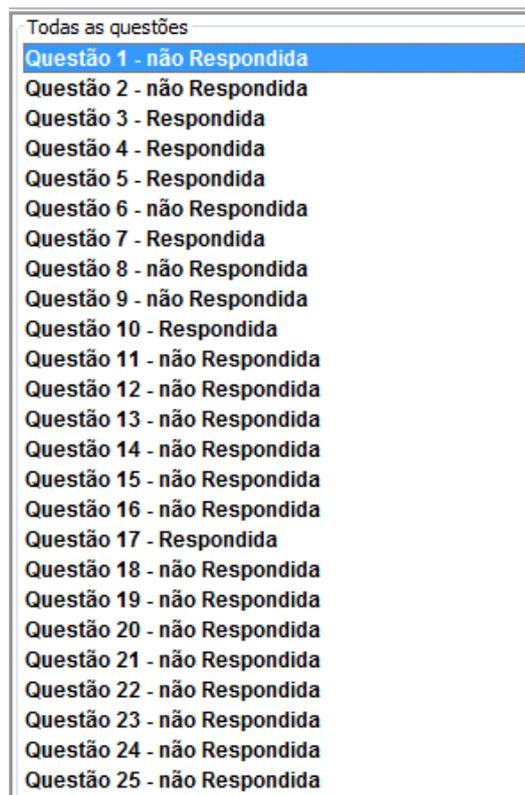


Figura 5.2 - Questionários de mensuração
Fonte: Elaborado pelo Autor

Esses questionários são compostos por questões objetivas de dois tipos, questões de múltipla escolha e questões falso/verdadeiro. Após a realização do questionário é emitida uma análise completa do desempenho do aluno, informando (Figura 5.3).

- O nível de desempenho do aluno no geral;
- O nível de desempenho do aluno em cada assunto;
- O nível de desempenho do aluno em cada sub assunto;
- O número de questões realizadas na avaliação;
- A nota final do aluno;
- A nota do aluno em cada assunto;
- A nota do aluno em cada sub assunto;
- A quantidade de questões realizadas corretamente de cada assunto;
- A quantidade de questões realizadas corretamente de cada sub assunto;
- Tempo gasto;
- Aproveitamento geral do aluno na avaliação.

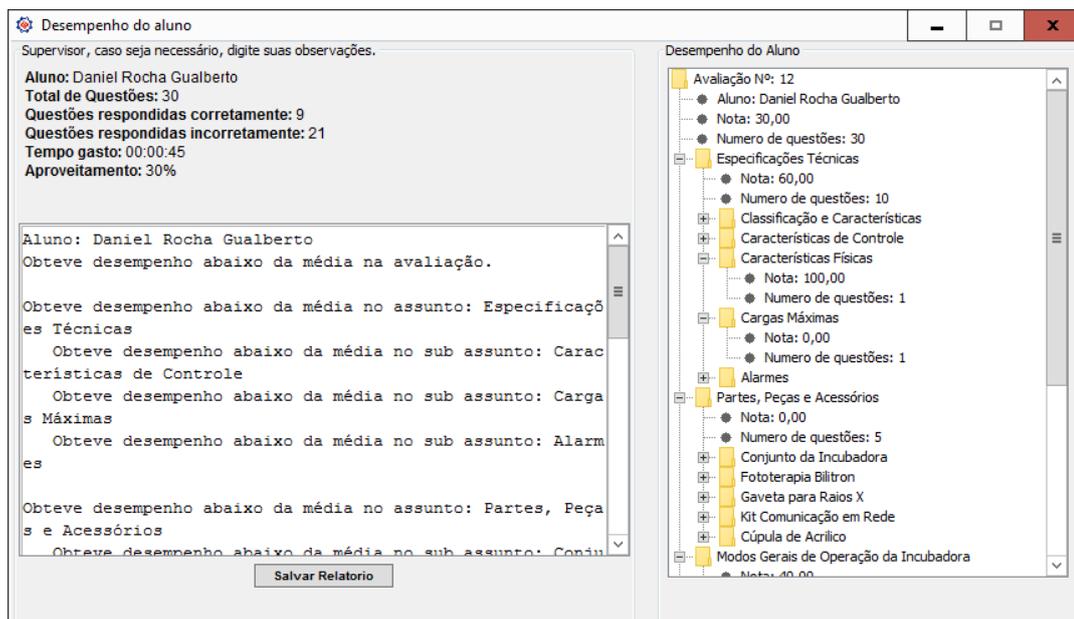


Figura 5.3 - Análise de desempenho do aluno
Fonte: Elaborado pelo Autor

O aluno pode navegar por diferentes tópicos dentro do nível corrente. Além disso, ele pode retornar e rever conceitos já estudados. Todas essas interações entre o aluno e o sistema são armazenadas no sistema na forma de um log para cada usuário em particular. Esse histórico pode ser visto e analisado pelo instrutor, verificando o andamento do ensino dos alunos e também as notas obtidas por eles.

O framework necessitou de dados para a modelagem e inclusão de casos na base de conhecimento (Apêndice G). O sistema descrito ao decorrer deste estudo possibilitou sua verificação através das questões que foram modeladas disponíveis nas avaliações para aplicação aos usuários.

Como as questões podem ser de múltipla escolha ou falso/verdadeiro, o caso é modelado de forma que as possibilidades de respostas estejam no conjunto {a; b; c; d; e} ou {f; v}. Cada uma das alternativas possui um peso associado. Tal valor é utilizado no cálculo da similaridade entre os casos existentes.

Portanto, o sistema proposto pode ser utilizado para organizar diferentes tipos de conhecimento de um Sistema Tutor Inteligente, independente do domínio da aplicação e do número e forma dos atributos utilizados em sua modelagem.

6. CONCLUSÃO

A necessidade de ensino personalizado para os profissionais da área da saúde que atuam em UTIN, empregando tecnologias que representam o estado da arte na área da Engenharia de Software através de uso de UML para a modelagem do sistema, motivou o estudo, a análise, o planejamento e a execução deste sistema de treinamento computadorizado.

A linha de pesquisa em IA de motor de inferência para a linguagem de programação Java foi a utilizada neste estudo. Está em constante modificação, sempre incorporando novos casos e, portanto, maior conhecimento, na forma de casos que podem ser utilizados posteriormente para pesquisa. O framework desenvolvido utiliza essa abordagem de IA para estruturar o conhecimento de diferentes domínios. Tal ferramenta visa auxiliar os desenvolvedores de sistemas a organizarem e a definirem atributos de casos e também valores com os quais cada atributo pode trabalhar.

A sistemática adotada para o desenvolvimento de sistemas de treinamento de funcionários permite que novos STI sejam desenvolvidos a partir do reuso das classes já desenvolvidas, facilitando o processo de implementação e aumentando a sua produtividade.

Espera-se com este STBC colaborar na formação de profissionais da área da saúde que atuam em UTIN, que são recém-admitidos e estão em fase de aperfeiçoamento, de modo que não seja oneroso para os hospitais, e que oposto às aulas convencionais, seja ofertado um treinamento com ambiente interativo rico em informações e que abrangem todos os procedimentos operacionais e de emergência, capacitando para a operação da incubadora neonatal de maneira correta e segura.

Uma série de estudos futuros, podem ser desenvolvidos a partir deste ponto, utilizando como estudo o Framework criado como base, e se adaptar as mais diversas necessidades do pesquisador. A proposta direcionada a uma pesquisa ampla é o desenvolvimento de um Framework de elaboração de um sistema de treinamento, o qual sua função seria gerar programas completos, já com informações inseridas para qualquer sistema de treinamento solicitado.

REFERÊNCIAS

- AKHRAS, F. N.; SELF, J. A. Beyond intelligent tutoring systems: Situations, interactions, processes and affordances. *Instructional Science*, v. 30, n. 1, p. 1–30, 2002. Disponível em: <<http://dx.doi.org/10.1023/A:1013544300305>>.
- BADER, J. *et al.* Practical engineering of knowledge-based systems. *Information and Software Technology*, v. 30, n. 5, p. 266–277, 1988.
- BERTELS, K. A Dynamic View on Cognitive Student Modeling in Computer Programming. *J. Artif. Intell. Educ.*, v. 5, n. 1, p. 85–105, 1994. Disponível em: <<http://dl.acm.org/citation.cfm?id=181425.181430>>.
- BONAFIDE, C. P. *et al.* Video Methods for Evaluating Physiologic Monitor Alarms And Alarm Responses. *Biomedical Instrumentation & Technology*, doi: 10.2345/0899-8205-48.3.220, v. 48, n. 3, p. 220–230, 1 maio 2014. Disponível em: <<http://dx.doi.org/10.2345/0899-8205-48.3.220>>.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: Guia do Usuário*. [S.l: s.n.], 2006. Disponível em: <<http://books.google.com/books?id=ddWqxcDKGF8C&pgis=1>>.
- BOROWSKI, M. *et al.* Medical device alarms. *Biomedizinische Technik/Biomedical Engineering*, v. 56, n. 2, p. 73–83, 1 jan. 2011. Disponível em: <<http://www.degruyter.com/view/j/bmte.2011.56.issue-2/bmt.2011.005/bmt.2011.005.xml>>.
- BREWER, M.; DIAZ-ARRASTIA, T.; TEKLITS, S. Management of Alarm Fatigue in the Neonatal Intensive Care Unit. p. 68, 2015. Disponível em: <http://www.pennstatehershey.org/documents/1699942/10895543/GN2015July_MgmtAlarmFatigueNICU/a9c57dcc-80a9-4df8-9d57-7d40dfeae6a7>.
- BRIDI, A. C.; LOURO, T. Q.; DA SILVA, R. C. L. Clinical Alarms in intensive care: implications of alarm fatigue for the safety of patients. *Revista latino-americana de enfermagem*, v. 22, n. 6, p. 1034–40, 2014. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-11692014000601034&lng=en&nrm=iso&tlng=en>.
- BURTON, R. R.; BROWN, J. S. An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, v. 11, n. 1, p. 5–24, jan. 1979. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0020737379800036>>.
- CARBONELL, J. R. AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction. *Ieee Transactions on Man-Machine Systems*, v. 11, n. 4, p. 190–202, 1971.
- CLANCEY, W. J. *Knowledge-based Tutoring: The GUIDON Program*. Cambridge, MA, USA: MIT Press, 1987.
- CLARK, T. *et al.* *Impact of Clinical Alarms on Patient Safety*. Disponível em: <<http://thehtf.org/documents/White Paper.pdf>>. Acesso em: 30 maio 2016.

COSTA, R. M. E. M. DA; WERNECK, V. M. B. *Sistemas Tutoriais: Aplicação das tecnologias de Hiperfídia e de Inteligência Artificial em Educação*. p. 1–73, 1997. Disponível em: <<http://dc370.4shared.com/doc/3wKGXppZ/preview.html>>.

CROWDER, N. *Automatic Tutoring by Intrinsic Programming*. New York: [s.n.], 1960.

DIMITRACOPOULOU, A. *Le tutorat dans les systèmes informatisés d'apprentissage : étude de la conception et réalisation d'un tutoriel d'aide à la représentation physique des situations étudiées par la mécanique*. 1995. 2 vol. (382, 112) f. phdthesis – 1995. Disponível em: <<http://www.theses.fr/1995PA070089>>.

FANEM. *Manual do Usuário - Incubadora Neonatal, Modelo Vision® ADVANCED 2286*. . Setúbal – Portugal: Cinterqual LDA. , 2012

FISCHETTI, E.; GISOLFI, A. From Computer-aided Instruction to Intelligent Tutoring Systems. *Educ. Technol.*, v. 30, n. 8, p. 7–17, 1990. Disponível em: <<http://dl.acm.org/citation.cfm?id=92704.92705>>.

FONSECA, L. M. M. *et al.* Inovação tecnológica no ensino da semiótica e semiologia em enfermagem neonatal: do desenvolvimento à utilização de um software educacional. *Texto & Contexto - Enfermagem*, v. 18, n. 3, p. 542–548, set. 2009.

FUNK, B. M. *et al.* Attitudes and practices related to clinical alarms. *AJCC*, v. 23, n. 3, p. 9–19, 2014.

GARCIA, O. N.; CHIEN, Y.-T. *Knowledge-based systems : fundamentals and tools*. [S.l.]: Los Alamitos: IEEE computer society press, 1991. p. 495. Disponível em: <<http://lib.ugent.be/catalog/rug01:000281985>>.

GREENO, J. G. The situativity of knowing, learning, and research. *American Psychologist*, v. 53, n. 1, p. 5–26, 1998.

GREENO, J. G.; COLLINS, A. M.; RESNICK, L. B. *Cognition and learning. Handbook of educational psychology. Handbook of educational psychology*. [S.l.: s.n.]. , 1996

GUALBERTO, D. R. *et al.* Software Project and Analysis of a Training Screen Based System for Healthcare Professionals Working in the NICU in Brazil. *Advances in Intelligent Systems and Computing*. v. 448, p. 1219–1230, 2016. Disponível em: <<http://link.springer.com/10.1007/978-3-319-32467-8>>.

INSTITUTE, E. *ECRI Institute Announces Top 10 Health Technology Hazards for 2015*. Disponível em: <<https://www.ecri.org/press/Pages/ECRI-Institute-Announces-Top-10-Health-Technology-Hazards-for-2015.aspx>>. Acesso em: 27 jun. 2016.

JONASSEN, D. H.; WANG, S. The Physics Tutor: Integrating Hypertext and Expert Systems. *Journal of Educational Technology Systems September*, v. 22, p. 19–28, 1993.

KRUCHTEN, P. *Introdução ao RUP: Rational Unified Process*. [S.l.: s.n.], 2003. Disponível em: <<https://books.google.com.br/books?id=MPbVAAAACAAJ>>.

LILJEGREN, E. Usability in a medical technology context assessment of methods for usability evaluation of medical equipment. *International Journal of Industrial Ergonomics*, v. 36, n. 4, p. 345–352, abr. 2006. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0169814106000047>>.

MANDL, H.; LESGOLD, A. (Org.). *Learning Issues for Intelligent Tutoring Systems*. New York, NY: Springer US, 1988. Disponível em: <<http://link.springer.com/10.1007/978-1-4684-6350-7>>.

MCTAGGART, J. Intelligent Tutoring System and Education for the Future. *Literature Review*, n. 512XX, p. 2, 2001.

NIELSEN JAKOB. *Usability Engineering*. [S.l: s.n.], 1994. Disponível em: <<http://proquest.safaribooksonline.com/book/web-development/usability/9780125184069>>.

PUL, C. VAN *et al.* *Alarm Management in an ICU Environment*. . Veldhoven, The Netherlands: [s.n.], 2014

REASON, J. Human error: models and management. *BMJ*, v. 320, n. 7237, p. 768–770, 18 mar. 2000. Disponível em: <<http://www.bmj.com/cgi/doi/10.1136/bmj.320.7237.768>>.

ROSENBERG, D.; STEPHENS, M. *Use Case Driven Object Modeling*. [S.l: s.n.], 1999. v. 3.

SANTOS, C.; RAMALHO, G. L. JEOPS – The Java Embedded Object Production System. p. 53–62, 2000.

SELF, J. Bypassing the intractable problem of student modelling. *Intelligent Tutoring Systems: At the crossroads of artificial intelligence and education*, n. 41, p. 107–123, 1990.

SKINNER, B. F. Teaching Machines. *IRE Transactions on Education*, v. 2, n. 1, p. 14–22, 1959. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4322064>>.

SLEEMAN, D. . Intelligent tutoring systems: a review. *Stanford, CA: Stanford University, School of Education & Department of Computer Science*, 1984.

URRETAVIZCAYA, M. Sistemas Inteligentes en el ámbito de la Educación. *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, v. 5, n. 12, p. 5–12, 2001.

Disponível em:

<<http://dialnet.unirioja.es/servlet/articulo?codigo=1254881&info=resumen&idioma=SPA>>.

WENGER, E. *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987.

YACEF, K. The Logic-ITA in the classroom: A medium scale experiment. *International Journal of Artificial Intelligence in Education*, v. 15, n. 1, p. 41–60, 2005. Disponível em: <<https://telearn.archives-ouvertes.fr/hal-00257107/>>.

APÊNDICES

Apêndice A - Análise de requisitos

Entende-se por requisitos de software, as sentenças que expressam as necessidades dos clientes e que condicionam a qualidade do software. Em função disto, é classificado os requisitos como funcionais (funções que o sistema deve fornecer, como deve reagir a entradas específicas e como deve se comportar) e não funcionais (são as restrições sobre o serviço). Os requisitos funcionais para o STI que deverá:

- Permitir ao administrador a seleção de opção de cadastro, onde poderá:
 - Incluir e excluir informações dos aprendizes;
 - A inclusão e exclusão das informações contidas no banco de dados do sistema;
 - Oferecer a correlação entre as informações contidas no banco de dados e as informações de avaliação do aprendiz emitidas;
- Permitir ao aprendiz a visualização dos tutoriais e dos testes sobre os diversos procedimentos da UTIN;
- Permitir ao aprendiz e administrador a visualização da avaliação sobre a incubadora.

Esses requisitos funcionais de usuário definem recursos específicos que devem ser fornecidos pelo sistema.

Apêndice B - Análise essencial

A análise essencial sugere a divisão do sistema por eventos. A medição do valor de um sistema está na sua competência de executar com eficácia todas as interações a que for submetida. Assim, um sistema é construído para responder aos eventos. A cada evento o sistema deve reagir produzindo uma ação pré-determinada.

Usa-se uma tabela de eventos como base para representar a divisão do sistema. Os eventos geram fluxos de dados (eventos) para o sistema. O modelo essencial é construído sem considerar restrições de implementação (assume uma tecnologia perfeita) – o que permite uma solução ideal para o problema. Na análise essencial um sistema de informação é visto como um sistema de resposta planejado.

Para representar esta premissa básica de descrever o sistema de maneira independente de restrições tecnológicas e encontrar o conjunto de requisitos essenciais do sistema, foi elaborado um quadro com os eventos do STI-UTIN (Quadro 1).

Nº	Descrição	Entrada	Saída	Caso de uso
1	Usuário entra no sistema.	Dados do usuário: nome, senha e tipo.	Mensagem 1.	UserLogin
2	Administrador adiciona um novo usuário no sistema.	Dados do novo usuário: nome, senha e tipo.	Mensagem 2.	AddUser
3	Administrador altera um usuário existente.	Dado(s) do usuário existente a ser alterado.	Mensagem 3.	ChangerUser
4	Administrador faz pesquisa de um usuário existente.	Dado(s) do usuário a pesquisar.	Informações do usuário: dados, acessos, operações.	SearchUser
5	Administrador elimina um usuário.	Dado(s) do usuário a eliminar.	Mensagem 4.	DeleteUser
Nº	Descrição	Entrada	Saída	Caso de uso
6	Instrutor faz pesquisa de uma avaliação existente.	Seleção da avaliação a pesquisar.	Informações da avaliação.	SearchPerformance
7	Aprendiz entra no sistema para estudar um tutorial sobre a incubadora.	Seleção do tutorial a estudar.	Informações do tutorial.	StudyTutorial
8	Aprendiz realiza uma avaliação.	Seleção da avaliação a realizar.	Nota da avaliação.	DoTest
9	Aprendiz salva avaliação.	Nota da avaliação.	Mensagem 14.	SaveTeste
10	Instrutor pesquisa o desempenho de um aprendiz.	Seleção do aprendiz a pesquisar o seu desempenho.	Resultados das avaliações do aprendiz.	SearchPerformance

Quadro 1 - Quadro de Eventos do STI - UTIN

Fonte: Elaborado pelo Autor

Apêndice C - Caso de uso

O objetivo do caso de uso é descrever os requerimentos funcionais do sistema e fornecer uma descrição consistente e clara sobre as responsabilidades que devem ser cumpridas pelo mesmo, e oferecer as possíveis situações do mundo real para o teste do sistema.

Caso de uso: geral do STI-UTIN

A visão geral das funções do STI-UTIN é apresentada na Figura 1 e no Quadro 2, onde é descrito o caso de uso geral das funções.

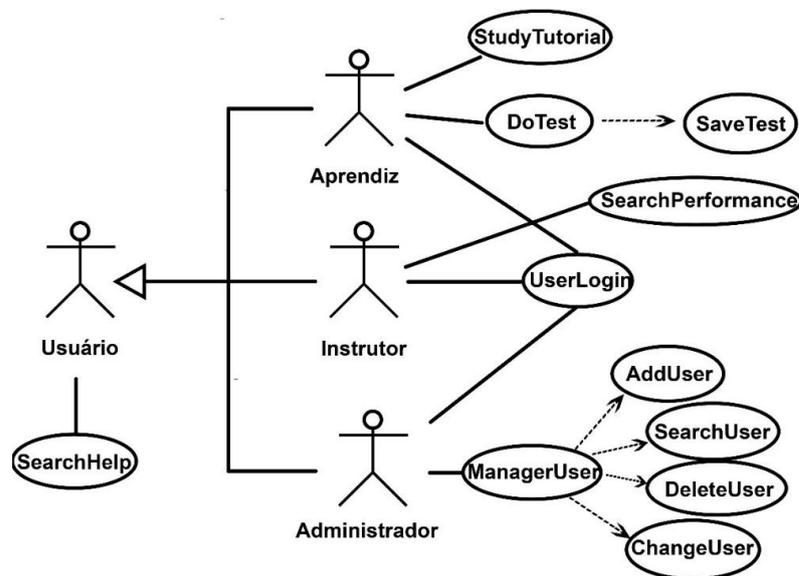


Figura 1 - Visão Geral das funções do STBC
Fonte: Gualberto *et al.* (2016)

Casos de uso: “UserLogin”; “AddUser”; “ChangeUser”; “SearchUser”; “DeleteUser”; “SearchTutorial”; “SearchTest”; “SearchTutorial”; “SearchHelp”; “StudyTutorial” “DoTest”; “SaveTest”; “SearchPerformance”

Ator Primário: Usuário

Descrição dos atores:

Administrador: Encarregado de executar as operações gerenciais do sistema. Vela pelo correto funcionamento do sistema e a sua conexão com o banco de dados.

Aprendiz: Realiza o seu treinamento com a ajuda dos tutoriais, como também fazer as avaliações para avaliar a sua aprendizagem.

Instrutor: Sua função mais importante é a de acompanhar o desempenho do aprendiz.

Descrição: O STBC apresenta os sistemas de tutoriais, gerenciamento e avaliação. Através destes itens o aprendiz poderá iniciar seus estudos onde poderá, através de um ambiente multimídia interativo obter

<p>todos os conhecimentos relacionados ao modelo de UTIN assim como ser avaliado pelas informações que lhe foram apresentadas.</p>
<p>Pré-condições: O usuário deve ser devidamente cadastrado no sistema para poder fazer uso do mesmo.</p>
<p>Fluxo Principal:</p> <ol style="list-style-type: none"> 1. Ao ser acionado, o sistema apresenta ao usuário a tela inicial do sistema. 2. O usuário entra com login e senha válida. 3. O sistema apresenta ao usuário todos os procedimentos que sua permissão o compõem. 4. Usuário com permissão de Aprendiz (A1). 5. Usuário com permissão de Instrutor (A2). 6. Usuário com permissão de Administrador (A3). 7. O sistema encerra o caso de uso.
<p>Fluxo Alternativo:</p> <p>A1: [Usuário com permissão de Aprendiz]</p> <ol style="list-style-type: none"> 1. É apresentada ao aprendiz a teoria de todos os procedimentos da UTIN e pequenos testes que permitirá consolidar o que lhe foi apresentado. 2. O aprendiz seleciona o tutorial que deseja estudar. 3. O sistema mostra as informações do tutorial. 4. O aprendiz passa para a etapa de avaliação onde é avaliado quanto ao seu grau de aprendizado. 5. O sistema de avaliação oferece o nível de aprendizado do aprendiz e mostra o resultado da avaliação. <p>A2: [Usuário com permissão de Instrutor]</p> <ol style="list-style-type: none"> 1. O instrutor busca o aprendiz de quem deseja pesquisar o desempenho. 2. O sistema mostra o desempenho do aprendiz selecionado. <p>A3: [Usuário com permissão de Administrador]</p> <ol style="list-style-type: none"> 1. É apresentado ao administrador as funções essenciais ao controle de usuário (adicionar, alterar, pesquisar e excluir um usuário).
<p>Pós-condições: Aprendiz passa por todas as etapas importantes para conhecer e consolidar o conhecimento de UTIN e, ao obter resultado de avaliação satisfatória, poderá começar a operar no setor de UTIN.</p>
<p>Requisitos funcionais satisfeitos: O sistema oferece, através de um ambiente interativo, as informações para os procedimentos operacionais e de emergência para operar a incubadora. O sistema avalia o desempenho do aprendiz.</p>

Quadro 2 - Descrição dos casos de uso geral do STI-UTIN

Fonte: Elaborado pelo Autor

Caso de uso: UserLogin

A visão da funcionalidade de Login do usuário do STI-UTIN é apresentada na Figura 2 e no Quadro 3 é descrito o caso de uso.

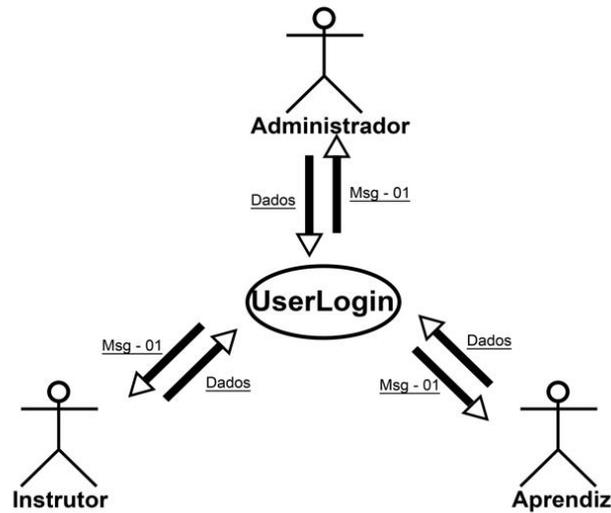


Figura 2 - Representação do caso de uso: UserLogin
Fonte: Elaborado pelo Autor

Caso de Uso: “UserLogin”
Ator Primário: Administrador, Instrutor e Aprendiz
Descrição: Usuário realizar o login no sistema
Pré-condições: O usuário deve ser devidamente cadastrado no sistema para poder fazer uso do mesmo.
Fluxo Principal: <ol style="list-style-type: none"> 1. Ao ser acionado, o sistema apresenta ao usuário a tela inicial do sistema; 2. O usuário entra com login e senha válida; 3. O sistema valida os dados; 4. O sistema emite a Msg-01: Usuário “X” logado no sistema.
Fluxo Alternativo: <p>A1: [Usuário sem cadastro no sistema]</p> <ol style="list-style-type: none"> 2. O sistema invalida os dados. 3. O sistema emite a mensagem: “Usuário X: Login ou senha inválidos! ”. 4. O sistema abandona o caso de uso.

Quadro 3 - Descrição do caso de uso: *UserLogin*

Fonte: Elaborado pelo Autor

Caso de uso: AddUser

A visão da funcionalidade de adicionar um usuário ao STI-UTIN é apresentada na Figura 3 e no Quadro 4 é descrito o caso de uso.

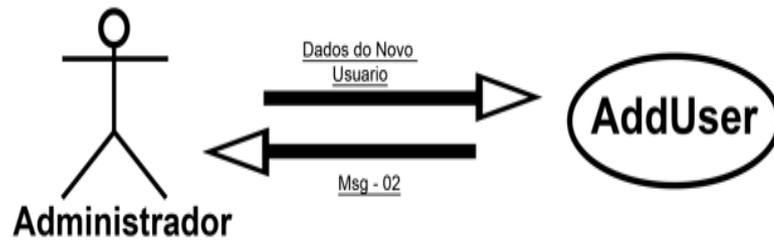


Figura 3 - Representação do caso de uso: AddUser
 Fonte: Elaborado pelo Autor

Caso de uso: “AddUser”
Ator Primário: Administrador
Descrição: O administrador deseja adicionar um novo usuário para utilizar o sistema
Pré-Condições: Existir um usuário como administrador e ter as informações para o cadastro do novo usuário.
Fluxo Principal: <ol style="list-style-type: none"> 1. O administrador seleciona a opção “Adicionar”. 2. O administrador digita os dados do novo usuário. 3. O sistema registra o novo usuário no sistema. 4. O sistema emite a Msg-02: “Usuário “X” adicionado ao sistema com sucesso. ”
Fluxo Alternativo: A1:[Usuário já cadastrado] <ol style="list-style-type: none"> 3. O sistema verifica que o usuário já foi cadastrado. 4. O sistema emite a mensagem: “Dados coincidentes com outro usuário. Digite novamente. ”

Quadro 4 - Descrição do caso de uso: AddUser
 Fonte: Elaborado pelo Autor

Caso de uso: ChangeUser

A visão da funcionalidade de alterar as informações de um usuário já cadastrado no STI-UTIN é apresentada na Figura 4 e no Quadro 5 é descrito o caso de uso.

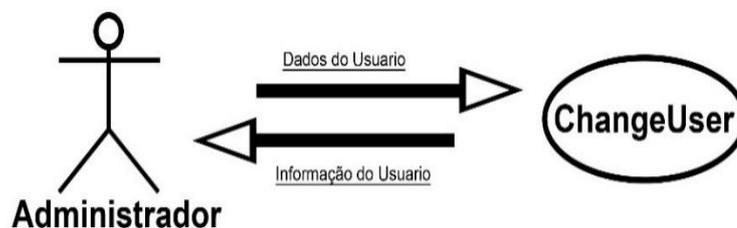


Figura 4 - Representação do caso de uso: ChangeUser
 Fonte: Elaborado pelo Autor

Caso de uso: “ChangeUser”;
Ator Primário: Administrador
Descrição: O administrador deseja alterar informações do usuário.

Pré-Condições: O usuário deve ser devidamente cadastrado no sistema para ser encontrado na busca de usuários.
Fluxo Principal: <ol style="list-style-type: none"> 1. O administrador seleciona a opção “Alterar”. 2. O administrador busca o usuário. 3. O sistema mostra os dados do usuário na tela. 4. O administrador faz as alterações nos dados do usuário. 5. O sistema registra as alterações nos dados do usuário. 6. O sistema emite a Msg-03: “Dados do usuário alterados com sucesso! ”
Fluxo Alternativo: A1: [Usuário não encontrado] <ol style="list-style-type: none"> 3. O sistema verifica que esse usuário não está cadastrado. 4. O sistema emite a mensagem: “Usuário não encontrado”. 5. O sistema abandona o caso de uso.

Quadro 5 - Descrição do caso de uso: ChangeUser
 Fonte: Elaborado pelo Autor

Caso de uso: SearchUser

A visão da funcionalidade de localizar um usuário cadastrado STI-UTIN é apresentada na Figura 5 e no Quadro 6 é descrito o caso de uso.

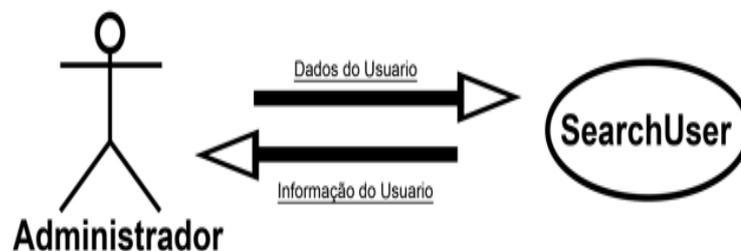


Figura 5 - Representação do caso de uso: SearchUser
 Fonte: Elaborado pelo Autor

Caso de uso: “SearchUser”
Ator Primário: Administrador
Descrição: O administrador deseja pesquisar um usuário para visualizar, alterar ou excluir as informações atreladas ao usuário.
Pré-condições: O usuário deve estar cadastrado no sistema para poder ser encontrado na pesquisa.
Fluxo Principal: <ol style="list-style-type: none"> 1. O administrador seleciona a opção “Pesquisar”. 2. O administrador busca o usuário. 3. O sistema mostra as informações do usuário na tela. 4. O administrador abandona o caso de uso.
Fluxo Alternativo: A1: [Usuário não está cadastrado]

1. O sistema verifica que esse usuário não está cadastrado.
2. O sistema emite a mensagem: “Usuário não encontrado”.
3. O sistema abandona o caso de uso.

Quadro 6 - Descrição do caso de uso: SearchUser

Fonte: Elaborado pelo Autor

Caso de uso: DeleteUser

A visão da funcionalidade de deletar um usuário cadastrado no STI-UTIN é apresentada na Figura 6 e no Quadro 7 é descrito o caso de uso.



Figura 6 - Representação do caso de uso: DeleteUser

Fonte: Elaborado pelo Autor

Use Cases: “DeleteUser”
Ator Primário: Administrador
Descrição: O administrador deseja excluir um usuário do sistema.
Pré-Condições: O usuário deve estar devidamente cadastrado no sistema para poder ter suas informações excluídas.
Fluxo Principal: <ol style="list-style-type: none"> 1. O administrador seleciona a opção “Excluir”. 2. O administrador busca o usuário que deseja eliminar. 3. O sistema faz a eliminação do usuário. 4. O sistema emite a Msg-04 “Usuário excluído com sucesso! ”
Fluxo Alternativo:
A1: [Usuário não cadastrado] <ol style="list-style-type: none"> 1. O sistema verifica que esse usuário não está cadastrado. 2. O sistema emite a Msg-04: “Usuário não encontrado”. 3. O sistema abandona o caso de uso.

Quadro 7 - Descrição do caso de uso: DeleteUser

Fonte: Elaborado pelo Autor

6.1.1 Caso de uso: StudyTutorial

A visão da funcionalidade do aprendiz para estudar um tutorial é apresentada na Figura 7 e no Quadro 8 é descrito o caso de uso.

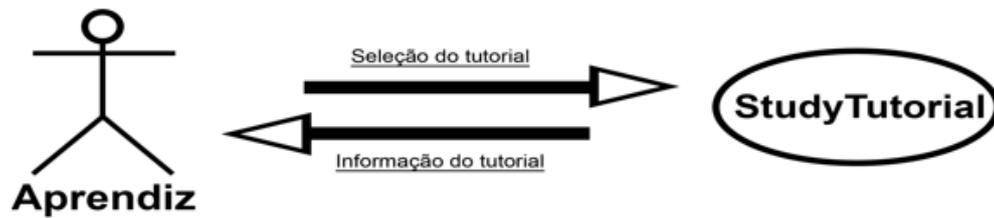


Figura 7 - Representação do Caso de Uso: StudyTutorial
 Fonte: Elaborado pelo Autor

Caso de uso: “StudyTutorial”
Ator Primário: Aprendiz
Descrição: O aprendiz deseja selecionar um tutorial para estudar.
Pré-Condições: O tutorial deve estar devidamente cadastrado no sistema para poder fazer uso do mesmo.
Fluxo Principal: <ol style="list-style-type: none"> 1. O aprendiz seleciona o tutorial que deseja estudar. 2. O sistema mostra as informações do tutorial. 3. O aprendiz abandona o caso de uso.

Quadro 8 - Descrição do caso de uso: StudyTutorial
 Fonte: Elaborado pelo Autor

Caso de uso: DoTest

A visão da funcionalidade do aprendiz para realizar uma avaliação no STI-UTIN é apresentada na Figura 8 e no Quadro 9 é descrito o caso de uso.

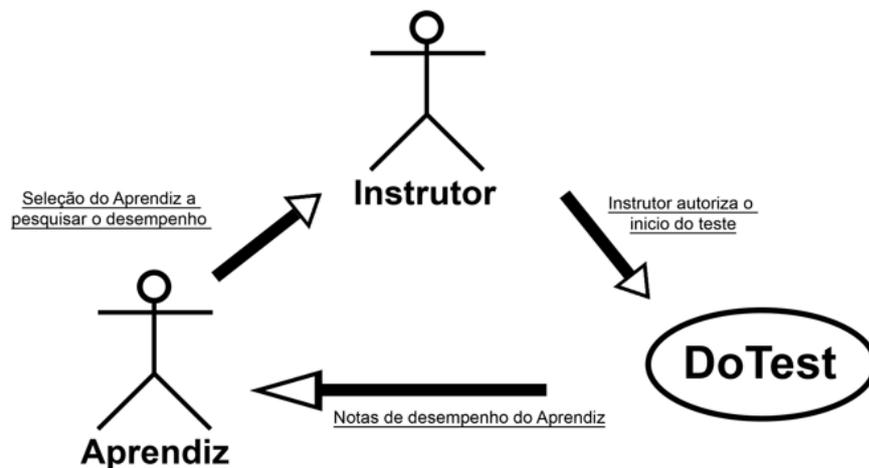


Figura 8 - Representação do caso de uso: DoTest
 Fonte: Elaborado pelo Autor

Use Cases: “DoTest”
Ator Primário: Aprendiz. Instrutor

Descrição: O aprendiz deseja realizar a avaliação para testar o seu conhecimento, é necessário que um Instrutor o acompanhe na sua avaliação.
Pré-Condições: O aprendiz deve ser devidamente cadastrado no sistema para poder fazer uso do mesmo. O instrutor deve acompanhar o aprendiz na realização da avaliação.
Fluxo Principal: <ol style="list-style-type: none"> 1. O aprendiz seleciona o tutorial que deseja estudar. 2. O instrutor autoriza o início da avaliação. 3. O sistema mostra as informações da avaliação. 4. O aprendiz realiza a operação e conclui o caso de uso.

Quadro 9 - Descrição do caso de uso: DoTest

Fonte: Elaborado pelo Autor

Caso de uso: SaveTest

A visão da funcionalidade de salvar a informação de desempenho do aprendiz é apresentada na Figura 9 e no Quadro 10 é descrito o caso de uso.

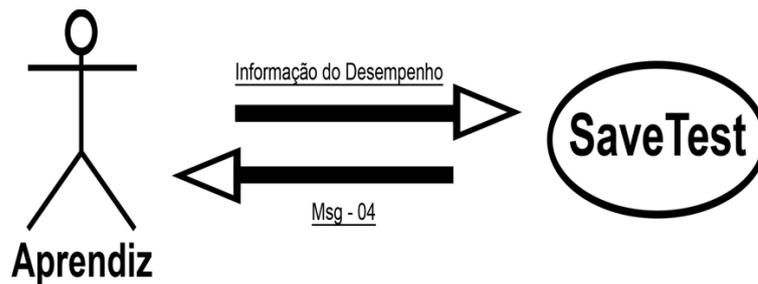


Figura 9 - Representação do caso de uso: SaveTest

Fonte: Elaborado pelo Autor

Use Cases: “SaveTest”;
Ator Primário: Aprendiz, Instrutor
Descrição: Salva a informação do desempenho do aprendiz
Pré-Condições: O aprendiz deve ter concluído a realização da avaliação já cadastrado no sistema.
Fluxo Principal: <ol style="list-style-type: none"> 1. O aprendiz realiza a avaliação. 2. O sistema devolve uma nota para o aprendiz. 3. O aprendiz seleciona a opção “Salvar Avaliação”. 4. O sistema salva a avaliação e emite a Msg-04: “Nota salva no seu histórico de desempenho”. 5. O sistema abandona o caso de uso.

Quadro 10 - Descrição do caso de uso: SaveTest

Fonte: Elaborado pelo Autor

Caso de uso: SearchPerformance

A visão da funcionalidade de salvar a informação de desempenho do aprendiz é apresentada na Figura 10 e no Quadro 11 é descrito o caso de uso.

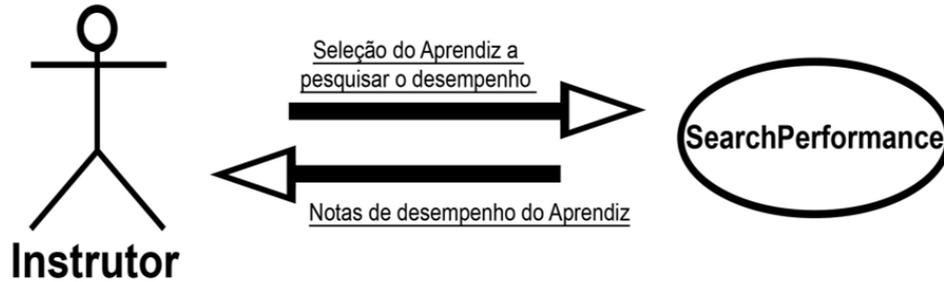


Figura 10 - Representação do caso de uso: SearchPerformance
Fonte: Elaborado pelo Autor

Use Cases: “SearchPerformance”
Ator Primário: Instrutor
Descrição: O instrutor deseja visualizar o desempenho de um aprendiz que já realizou uma avaliação.
Pré-Condições: Informações no histórico do sistema das informações do desempenho de aprendizes que já realizaram avaliações.
Fluxo Principal: <ol style="list-style-type: none"> 1. O instrutor busca o aprendiz de quem deseja pesquisar desempenho. 2. O sistema mostra o desempenho do aprendiz selecionado. 3. O instrutor abandona o caso de uso.
Fluxo Alternativo: <p>A1: [Aprendiz não existe no sistema]</p> <ol style="list-style-type: none"> 1. O sistema invalida a existência do aprendiz. 2. O sistema emite a mensagem: “Aprendiz não existe no sistema”.

Quadro 11 - Descrição do caso de uso: SearchPerformance
Fonte: Elaborado pelo Autor

Apêndice D - Diagrama de sequência

Para representar o comportamento do aprendiz, o ator alvo, com o sistema, é utilizado o digrama de sequência. Dentro de um contexto, são mostrados objetos participando em interações de acordo com suas linhas de vida e as mensagens que trocam, com o objetivo de estabelecer os objetos que interagem e seus relacionamentos dentro de um contexto (caso de uso). O diagrama de sequência para interação do aprendiz com os tutoriais, está ilustrado na Figura 11.

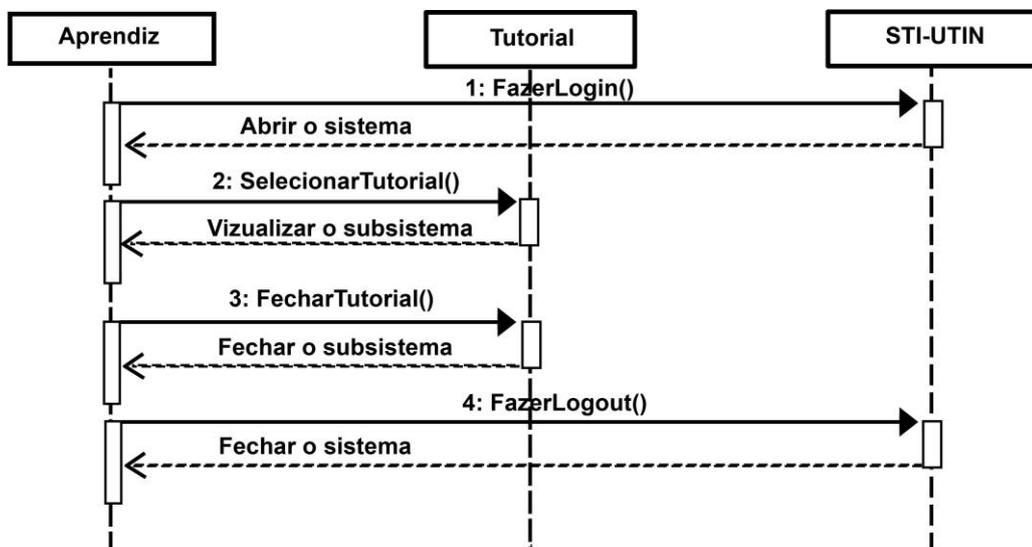


Figura 11 - Diagrama de sequência para interação do aprendiz com os tutoriais
Fonte: Elaborado pelo Autor

A Figura 12 apresenta o diagrama de sequência para interação do aprendiz com as avaliações.

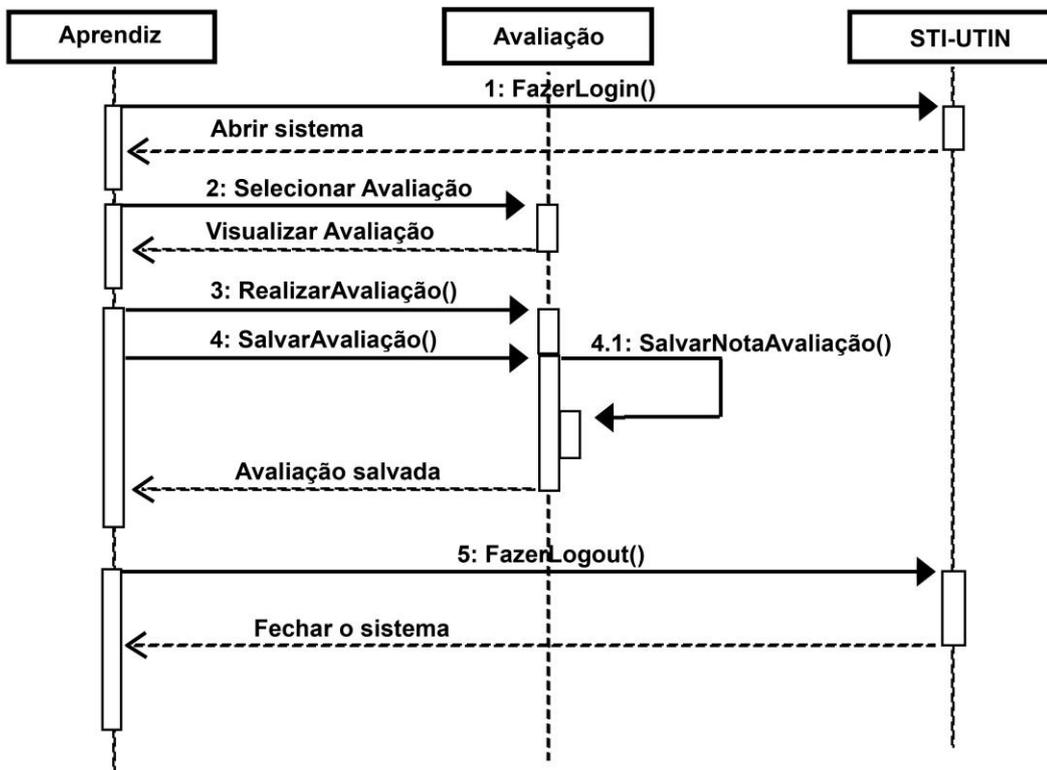


Figura 12 - Diagrama de seqüência para interação do aprendiz com as avaliações.
 Fonte: Elaborado pelo Autor

Apêndice E - Diagrama de classes

O diagrama de classes tem objetivo de descrever os vários tipos de objetos no sistema e o relacionamento entre eles. Este oferece a perspectiva de implementação, abordando os detalhes de implementação, tais como navegabilidade, tipo dos atributos, etc. Para o STBC, o diagrama de classes, está representado na Figura 13.

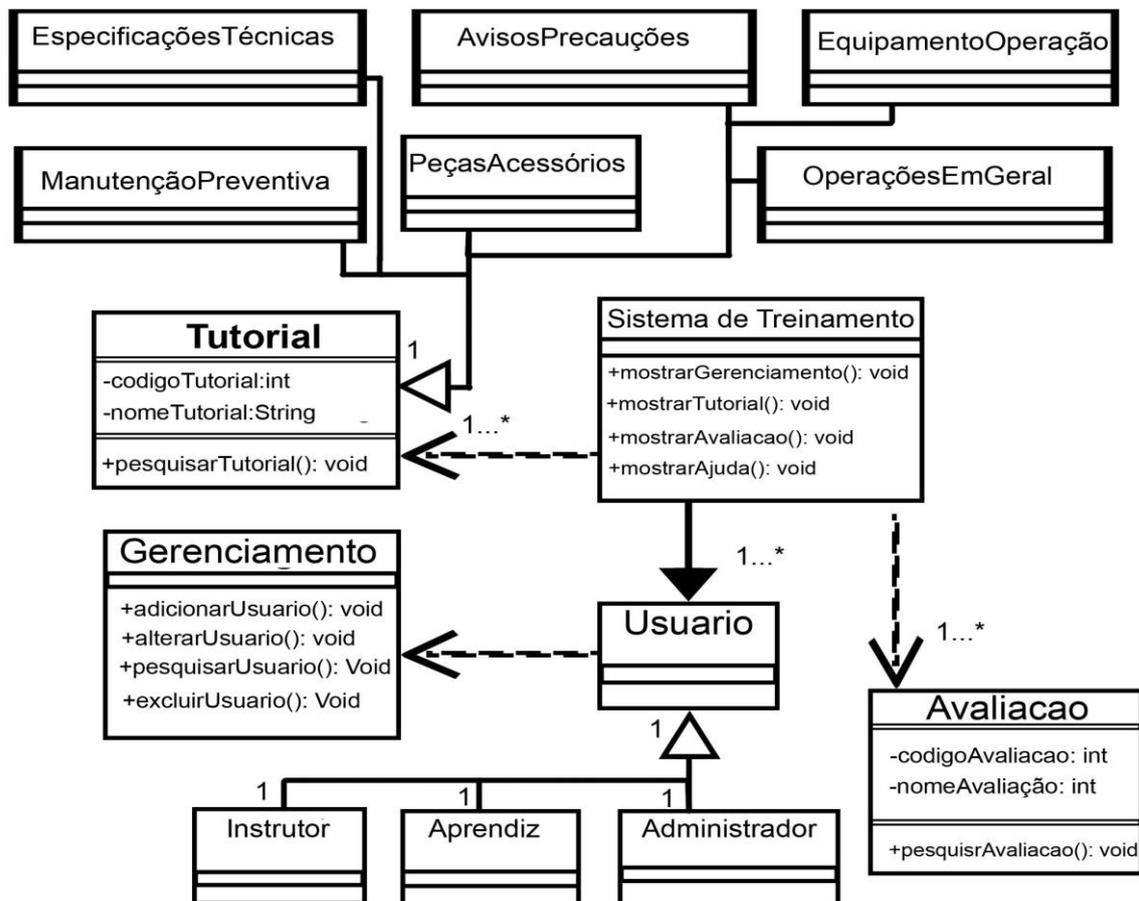


Figura 13 - Diagramas de classes do STI-UTIN

Fonte: Elaborado pelo Autor

Apêndice F - Relatório de Avaliação Heurística – Sistema de Treinamento

da Incubadora

Relatório de Avaliação Heurística Sistema de Treinamento da Incubadora

Junho de 2016

1. Introdução

Este relatório apresenta os resultados da aplicação de uma Avaliação Heurística para detectar potenciais problemas de usabilidade sobre a interface do Sistema de Treinamento da Incubadora que está sendo desenvolvido.

A avaliação foi realizada como parte do projeto orientado pelo professor Carlos Alexandre Brandão Ramos, e feita pelos pesquisadores e orientados do Laboratório de Usabilidade e Fatores Humanos da UNIFEI.

Os resultados podem não abordar todos os problemas possíveis em sua totalidade encontrados em análises mais criteriosa, porém, são relevantes para que os desenvolvedores já possam melhorar o produto levando em consideração pontos de vista distintos.

Quanto aos problemas de usabilidade listados, foi feito de acordo com a visão de especialistas, portanto algumas validações podem envolver os usuários finais do sistema (essa validação na fará parte do escopo desse relatório).

2. Método da Avaliação

O método utilizado foi a de Avaliação Heurística, método criado para encontrar problemas de usabilidade durante um processo de design iterativo, é uma alternativa de avaliação rápida e de baixo custo, quando comparada a métodos empíricos.

Usabilidade é um atributo de qualidade relacionado à facilidade de uso de algo, ou seja, refere-se a rapidez com que os usuários podem aprender a usar alguma coisa, a eficiência deles ao usá-la, o quanto lembram daquilo, seu grau de propensão a erros e o quanto gostam de utilizá-las (Nielsen, Loranger, 2007).

Benefícios: A aplicação deste método permite uma rápida resposta sobre a atual interface do produto, comparando-a com as heurísticas na visão dos especialistas. Pode ser aplicada desde o início do processo de desenvolvimento, mesmo em protótipos não funcionais do produto. Pode compor a documentação do Processo de Engenharia de Usabilidade do fabricante no requisito 5.8 da norma ABNT NBR IEC 60601-1-6:2011.

Limitações: O resultado de uma Avaliação Heurística não pode ser apresentado como validação de usabilidade do produto, visto que não avalia o produto com seu usuário final. O resultado se limita a opinião e experiência dos especialistas em usabilidade envolvidos. Podem ser apontados problemas que não irão se concretizar durante a utilização do usuário ou que poderão ser considerados de baixa prioridade pela empresa.

As diretrizes de usabilidade utilizado na avaliação do Sistema de Treinamento da Incubadora foram as heurísticas de Nielsen (1995) especificadas na Tabela 1.

	Heurística	Descrição
1	Visibilidade do status do Sistema	Informações claras, constantes e imediatas sobre o status do sistema. Ex: Aviso afirmando que um e-mail já foi enviado, barra de progresso do download, etc.
2	Correspondência entre o Sistema e o usuário	Conceitos, termos, tarefas e procedimentos devem ser próximos à realidade do usuário com palavras, frases e conceitos familiares no dia a dia do usuário. Ex: Disquete para indicar ícone de gravar, cor vermelha para perigo ou parar.
3	Controle e liberdade do usuário	Interface reativa, o usuário deve controlar o sistema, ter a opção de desfazer um comando.
4	Consistência e Padrões	Usuário não deve ter dúvida se as mesmas ações, palavras, símbolos ou situações têm significado diferente.
5	Prevenção de erros	O sistema deve informar ao usuário as consequências de suas ações para evitar enganos.
6	Conhecimento ao invés de memorização	O usuário deve reconhecer de imediato as possíveis ações, ao invés de decorá-las.
7	Flexibilidade e eficiência do uso	Ações devem possuir diferentes formas de serem acionadas, deve conter opção de atalhos e customização para usuários avançados.
8	Estética e design minimalista	Deve conter a quantidade de informação necessária, layout agradável e leve.

9	Mensagem de erro	O sistema deve ter mensagem de erro informativa e orientativa auxiliando o usuário na correção do erro. Deve utilizar linguagem adequada ao usuário e educada.
10	Ajuda e documentação	O sistema deve oferecer ajuda em todas as ações, com conteúdo informativo.

Tabela 1 - Heurísticas de Nilsen

E de acordo com cada problema de usabilidade encontrado, o mesmo foi classificado de acordo com a escala referente ao grau de severidade especificado na tabela 2.

Severidade	Descrição
0	Não apresenta problemas de usabilidade. Não é necessário reparo.
1	Problema superficial. Não necessita ser reparado a menos que exista tempo disponível.
2	Problema de usabilidade de pequeno impacto. Deve-se dar baixa prioridade para o reparo desse problema.
3	Principal problema de usabilidade. Deve-se dar alta prioridade para o reparo desse problema.
4	Problema catastrófico. O reparo do problema é obrigatório e deve ser feito antes mesmo do seu lançamento.

Tabela 2 - Grau de Severidade

3. Objeto de estudo

O produto avaliado é um software de treinamento para incubadora de U.T.I Neonatal que será utilizado por profissionais da área da saúde, como, enfermeiros, médicos, auxiliares e técnicos de enfermagem. O software busca treinar os usuários para que manuseio da incubadora seja feito da forma mais segura e correta.

4. Extensão da Avaliação

A equipe fez uma descrição das tarefas que podem ser realizadas no software e avaliou cada uma das tarefas separadamente especificando qual heurística foi violada e qual foi o grau de severidade do problema encontrado.

5. Resultados

A Tabela 3 demonstra os problemas de usabilidade encontrados no Sistema de Treinamento da Incubadora após a realização da Avaliação Heurística, junto de sugestões feitas para melhoria a ser considerada pelos desenvolvedores.

Os números utilizados na coluna "Heurística Violada" se refere as heurísticas especificadas na Tabela 1, e os números utilizados na coluna "Severidade" se refere ao grau de severidade especificado na Tabela 2.

5.1 Tabela 3 - Relação dos problemas encontrados durante a Avaliação Heurística, especificada de acordo com as tarefas analisadas.

Tarefas	Problemas	Local	Heurística Violada	Consequência	Severidade	Sugestão
1 Realizar login no sistema	O termo "logar" pode não ser familiar	Tela login	2	Hesitar em clicar no botão	1	Mudar para "entrar"
	O botão "fechar" não tem sentido	Tela login	1,2	Não saber a função exata do botão e fechar algo sem querer	2	Realocar o botão
	Mensagem de erro com escrita confusa	Tela login	1	Não há consequência mas o usuário pode notar a escrita.	1	Mudar mensagem para: Login ou Senha incorretos, tente novamente.
3 Selecionar tutorial	"Tutoriais e exercícios disponíveis..." logo acima do campo com três pontos leva a crer que serão encontrados os tutoriais e os exercícios.	Tela tutoriais	8	Pode confundir o usuário	2	Retirar a frase "Tutoriais e exercícios disponíveis até o momento"
	Palavra tutorial está no singular, mas na realidade existem vários tutoriais dentro da pasta	Logo de tutoriais	4	A palavra no singular não atende as expectativas do usuário	1	Mudar para tutoriais
4 Visualizar página 18 do tutorial	O sistema não oferece atalho para ir direto a página desejada, deve-se passar uma por uma	Barra de operações inferior em tutoriais	7	Falta de eficiência e flexibilidade de uso	1	Colocar um campo entre botões (Anterior e Próxima) para que o usuário possa inserir a página que deseja
5 Voltar a aba tutoriais	Para voltar, a janela que cobre deve ser fechada, minimizada ou deve move-la, e ainda, se fechar, irá fechar somente a janela aberta?	Janela de tutorial	1	Não ficou claro como voltar e acessar outras abas	3	Abri a janela de modo que as abas do menu fiquem visível, ou, criar um botão com a função de voltar ao menu principal
6 Selecionar exercícios	A palavra tutoriais cria uma expectativa de haver apenas tutoriais e não também exercícios.	Aba tutoriais	1	Não encontrar ou demorar para encontrar exercícios	2	A aba pode passar a se chamar "Tutoriais e Exercícios" ou criar uma aba para cada
	O ícone do laboratório fica encoberto pela janela de seleção	Aba tutoriais	8	O usuário percebe a interface como esteticamente	2	Deixar o ícone do laboratório acima da janela. Idem para tutoriais.
7 Visualizar número de exercícios disponíveis e visualizar o exercício 13	Os exercícios não estão numerados	Janela de exercícios	1	O usuário não sabe exatamente em qual exercício ele está	2	Numerar os exercícios
	O sistema não oferece atalho para ir direto a página desejada, deve-se passar uma por uma	Barra de operações inferior em exercícios	7	Falta de eficiência e flexibilidade de uso	1	Colocar um campo entre botões (Anterior e Próxima) para que o usuário possa inserir a página que deseja
	Informações confusas, é um tutorial ou exercício? Pois o título da página é, "Tutorial : auxiliar - 16 exercícios disponíveis	Título da janela de exercícios	5	O usuário não tem certeza de que está realizando o exercício	2	Trocar título por: "Exercício 1 de 16", por exemplo.
8 Selecionar ajuda	Opção não fica disponível em telas secundárias como a de exercícios, por exemplo.	Janela de exercícios	1,7	Para selecionar ajuda, em certos casos, o usuário precisa fechar a página atual, decorrendo em falta de eficiência.	2	Inserir ícone de ajuda nas telas secundárias.
9 Selecionar tópico "3.2.2 Exercícios"	Não há um índice com link para cada parte do Manual	Aba ajuda	7	O usuário terá que rolar a página até achar o tópico, falta eficiência	1	Criar hiperlinks para cada tópico.

10	Fazer avaliação geral	A expressão "avaliação geral" indica que existe apenas uma avaliação a se fazer, mas o texto "avaliações disponíveis.." indica que existem várias possíveis, embora não haja uma lista.	Aba avaliação geral	1	Não é possível localizar uma lista	2	Trocar por avaliações se a ideia for deixar uma listagem delas.
		Não fica claro a necessidade de um supervisor por perto	Aba avaliação geral	5	Podem induzir o usuário a tentar realizar a avaliação sem supervisor	2	Melhorar a informação, demonstrando a necessidade de um supervisor
11	Visualizar avaliações canceladas	Para voltar, a janela que cobre deve ser fechada, minimizada ou deve move-la, e ainda, se fechar, irá fechar somente a janela aberta?	Janela de avaliações canceladas	1	Não ficou claro como voltar e acessar outras abas	3	Abri a janela de modo que as abas do menu fiquem visível, ou, criar um botão com a função de voltar ao menu principal
12	Visualizar o relatório 7, quem era o supervisor, o aluno, a hora e data da avaliação e o tema da avaliação cancelada	Não fica claro que a hora/data é o dia da avaliação ou do cancelamento da mesma	Janela de relatórios de avaliações canceladas	1	Podem gerar confusão sobre o que a data e hora se referem	1	Mudar texto para "avaliação cancelada em.. data .. hora"
13	Iniciar uma avaliação	A ordem dos comandos está invertida, primeiro ecoia o número de questões e depois confirmação do supervisor	Iniciar avaliação	1	O comando na ordem correta agiliza o processo	1	Inverter a ordem
		Orientação sobre a autorização do supervisor é pouco explicativa	Iniciar avaliação	5	O usuário pode ficar confuso quanto aos requisitos para iniciar a avaliação	2	Colocar informações mais claras
15	Responder questão 1	Não há identificação do número da questão no texto, somente na coluna lateral a esquerda	Janela de avaliação	1,7	Usuário pode demorar para identificar o número da questão	2	Colocar número na frente da questão
		Aparecem campos sem função na tela (para responder a questão)	Janela de avaliação	8	Podem confundir o usuário	2	Retirar campos que não serão utilizados
16	Ir para a questão seguinte	A palavra "computar" pode não estar de acordo com a linguagem utilizada pelo usuário	Janela de avaliação	2	Usuário pode ficar inseguro em clicar no botão	3	Mudar, por exemplo, para confirmar ou confirmar resposta
		O programa não vai para questão seguinte automaticamente	Janela de avaliação	1	Usuário pode ficar confuso quanto a como ele deve ir para a próxima questão	3	Acrescentar a informação de que ele deve selecionar a questão ou acrescentar a função de passar automaticamente.
		Podem ir para a questão seguinte sem responder a anterior?	Janela de avaliação	1	Usuário pode ficar confuso se pode fazer a avaliação fora da ordem	3	Acrescentar a informação de que pode realizar avaliação fora da ordem
		Não possui atalho para mudança de tela como tinha na página de exercícios e tutoriais	Janela de avaliação	4	Não existe um padrão de acordo com as outras páginas	3	Acrescentar botão para ir a próxima questão
17	Conferir se todas as questões foram concluídas e concluir avaliação	Podem concluir o teste sem responder todas as questões?	Janela de avaliação	5,9	Usuário pode finalizar teste sem responder todas as questões	2	Deve ter uma mensagem de erro para caso isso ocorra
		Na janela que confirma a conclusão, o texto pode ser mais adequado, por exemplo, avaliação está com letra maiúscula no meio do texto	Confirmar conclusão	8	Usuário pode observar que o programador não se atentou aos detalhes	2	Alterar para: "Sua avaliação foi concluída com sucesso. Para conferir suas respostas clique em GABARITO."
18	Conferir o gabarito	Não há orientação de como conferir gabarito, não existe um botão gabarito	Janela de avaliação	1	O usuário pode necessitar de suporte	3	Criar o botão "gabarito"
19	Visualizar o tempo de duração da avaliação	O tempo continua passando mesmo depois de finalizar o teste, será que ele se refere ao tempo de duração da avaliação?	Janela de avaliação	1,5	Usuário pode confundir a função do temporizador	3	Parar o contador assim que a avaliação for concluída e adicionar explicação sobre o que ele se refere
20	Visualizar desempenho de aluno na avaliação	Algumas informações estão confusas, o sistema não fornece o valor de cada questão, o que contém nas pastas além da avaliação 1?	Janela de relatório de avaliação	6, 8	O usuário demora para visualizar seu desempenho	3	Colocar as informações mais importantes, como valor de questões, mais claramente
21	Salvar relatório e voltar para menu inicial	A palavra "somente" deixa em dúvida o que de fato acontecerá. Permanecerei nesta tela?	Janela de relatório de avaliação	1	Usuário fica em dúvida sobre status do sistema	3	Trocar por "salvar relatório"
		Se confirmar relatório ele fica salvo? Pode sair sem confirmar ou salvar?	Janela de relatório de avaliação	3, 5 e 9	Usuário pode sair sem salvar	3	Especificar como ocorre as ações e ter mensagens de erro
23	Sair do sistema	O termo "logout" pode não ser familiar	Tela inicial	2	Hesitar em clicar no botão	1	Mudar para "sair"
		Existência de botões logout e fechar próximos	Tela inicial	1, 2	Podem confundir o usuário de acordo com sua necessidade	2	Mudar nome dos botões ou esclarecer a função de cada um

Apêndice G - Questões que se encontram na avaliação do STI-UTIN

I. Ajuste do Relógio Calendário

Questão 1: Por que é importante manter a incubadora com data e hora ajustada?

- a) Para que a incubadora alarme avisando a data de troca de filtro de ar.
- b) Para registrar na ficha/prontuário do paciente os eventos/procedimentos realizados junto ao recém-nascido.
- c) Para que o tempo do gráfico de tendência esteja atualizado.
- d) Todas as alternativas anteriores
- e) Nenhuma das alternativas anteriores.

II. Alarmes

Questão 2: Por quanto tempo é possível inibir o alarme?

- a) 60 minutos
- b) 30 minutos
- c) 20 minutos
- d) 15 minutos
- e) 10 minutos

Questão 3: Quantos tipos de som os alarmes da Incubadora VISION 2286 possui?

- a) 1
- b) 2
- c) 3
- d) 4
- e) 5

Questão 4: Além do sinal sonoro e acender o LED vermelho, o alarme também mostra na tela uma mensagem visual indicando o que está alterado e precisa ser corrigido. Qual a ação correta ou conduta ao perceber um alarme?

- a) Apenas inibir o som do alarme rapidamente.
- b) Ler a mensagem antes de inibir o som.
- c) Corrigir o que está alterado e inibir o som.
- d) Ler a mensagem, inibir o som e corrigir o que está alterado.
- e) Nenhuma das anteriores.

Questão 5: Todos os alarmes disparam uma mensagem visual na tela imediatamente ao serem detectados. Quais alarmes disparam o sinal sonoro apenas após 15 segundos aparecer a mensagem visual?

- a) Alta temperatura do AR e PELE e Baixa temperatura do AR e PELE.
- b) Alta temperatura do AR e PELE e Baixa temperatura do AR e UMIDADE.
- c) Alta temperatura do AR, PELE, UMIDADE e OXIGÊNIO.
- d) Baixa temperatura do AR, PELE, UMIDADE e OXIGÊNIO.
- e) Nenhuma das anteriores.

Questão 6: Todos os alarmes disparam uma mensagem visual na tela imediatamente ao serem detectados. Quais alarmes disparam o sinal sonoro após 2 minutos aparecerem a mensagem visual?

- a) Alta temperatura do AR e PELE e Baixa temperatura do AR e PELE.
- b) Alta temperatura do AR e PELE e Baixa temperatura do AR e UMIDADE.
- c) Alta temperatura do AR, PELE, UMIDADE e OXIGÊNIO.
- d) Baixa temperatura do AR, PELE, UMIDADE e OXIGÊNIO.
- e) Nenhuma das anteriores.

Questão 7: Como são ajustados os limites de alarme da incubadora VISION 2286?

- a) Manualmente.
- b) Automaticamente.
- c) Manualmente e automaticamente.
- d) Não existem limites de alarme.
- e) Nenhuma das anteriores.

III. Bateria Recarregável

Questão 8: Qual a função da bateria recarregável na incubadora?

- a) Alimentar o alarme de falta de energia elétrica da rede.
- b) Alimentar o alarme de alta temperatura da PELE.
- c) Alimentar o alarme de alta temperatura do AR.
- d) Alimentar o alarme de baixa temperatura da PELE.
- e) Alimentar o alarme de baixa temperatura do AR.

IV. Características de Controle

Questão 9: Quanto tempo a incubadora leva para aquecer ao ser ligada?

- a) 10 minutos.
- b) 20 minutos.
- c) 30 minutos.
- d) 40 minutos.
- e) 50 minutos.

Questão 10: De quanto em quanto tempo deve ser feita a calibração dos sensores de OXIGÊNIO?

- a) 1 dia.
- b) 2 dias.
- c) 4 dias.
- d) 6 dias.
- e) 7 dias.

Questão 11: Qual o tempo de operação de um reservatório de água com capacidade de 1 litro funcionando ininterruptamente com UMIDADE RELATIVA de 80%?

- a) 6 horas.
- b) 8 horas.
- c) 12 horas.
- d) 18 horas.
- e) 24 horas.

V. Características Físicas

Questão 12: Quantas portinhas existem na incubadora VISION 2286?

- a) 2.
- b) 3.
- c) 4.
- d) 5.
- e) 6.

Questão 13: Quantas mangas Íris possui a incubadora VISION 2286?

- a) 1.
- b) 2.
- c) 3.
- d) 4.
- e) 5.

VI. Cargas Máximas

Questão 14: Qual é a carga máxima que pode ser colocada sobre a prateleira da incubadora?

- a) 2 kg.
- b) 4 kg.
- c) 6 kg.
- d) 8 kg.
- e) 10 kg.

VII. Checagem dos Alarmes

Questão 15: Em qual temperatura o alarme de ALTA TEMPERATURA DE SEGURANÇA é ativado quando ajustado para temperatura inferior à 37°?

- a) Ativado em 37°.
- b) Ativado em 39°.
- c) Ativado em 40°.
- d) Ativado em 41°.
- e) Ativado em 42°.

Questão 16: Em qual temperatura o alarme de ALTA TEMPERATURA DE SEGURANÇA é ativado quando ajustado para temperatura superior à 37°?

- a) Ativado em 37°.
- b) Ativado em 39°.
- c) Ativado em 40°.
- d) Ativado em 41°.
- e) Ativado em 42°.

Questão 17: A mensagem visual FALTA SENSOR corresponde a qual condição?

- a) Quando o plug do sensor do paciente está desconectado ou o fio do sensor está com mau contato.
- b) Quando o plug do sensor do paciente está desconectado.
- c) Quando o fio do sensor está com mau contato.

- d) Quando o sensor de pele do recém-nascido está desalojado.
- e) Quando falha o sensor de temperatura do AR.

Questão 18: A mensagem visual SENSOR DESALOJADO corresponde a qual condição?

- a) Quando o plug do sensor do paciente está desconectado ou o fio do sensor está com mau contato.
- b) Quando o plug do sensor do paciente está desconectado.
- c) Quando o fio do sensor está com mau contato.
- d) Quando o sensor de pele do recém-nascido está desalojado ou acontece uma variação brusca de temperatura.
- e) Quando falha o sensor de temperatura do AR.

Questão 19: O que está acontecendo quando na mensagem visual do alarme está escrito FALHA no campo referente ao modo AR?

- a) Significa uma falha no sensor de temperatura de AR da incubadora.
- b) Significa uma falha no sensor de temperatura de PELE do paciente.
- c) Significa uma falta o sensor de temperatura de auxiliar da incubadora.
- d) Significa uma falha interna no microprocessador de sistema.
- e) Nenhuma das anteriores.

Questão 20: O que está acontecendo quando na mensagem visual do alarme está escrito FALHA NO SISTEMA?

- a) Significa uma falha no sensor de temperatura de AR da incubadora.
- b) Significa uma falha no sensor de temperatura de PELE do paciente.
- c) Significa uma falta o sensor de temperatura de auxiliar da incubadora.
- d) Significa uma falha interna no microprocessador de sistema.
- e) Nenhuma das anteriores.

VIII. Classificação e Características

Questão 21: Se colocarmos um medidor de ruído dentro da incubadora para saber qual é o menor ruído percebido pelo recém-nascido, qual seria o resultado?

- a) 30 decibéis
- b) 40 decibéis
- c) 50 decibéis
- d) 60 decibéis
- e) 70 decibéis

IX. Conjunto da Incubadora

Questão 22: A incubadora VISION 2286 possui ajuste de inclinação do leito?

- a) Não.
- b) Sim, apenas um ajuste (Trendelenburg).
- c) Sim, possui dois ajustes (Trendelenburg e proclive).
- d) Sim, possui três ajustes (Trendelenburg, proclive e horizontal baixa).
- e) Sim, possui quatro ajustes (Trendelenburg, proclive, horizontal alta e baixa).

X. Cúpula de Acrílico

Questão 23: A parede dupla interna da incubadora VISION 2286 é formada por três partes, sendo uma instalada na cúpula e as outras duas instaladas nas portas de acesso frontal e traseira. Para que servem estas paredes duplas?

- a) Para melhorar o controle de temperatura dentro da incubadora.
- b) Para melhorar o controle de umidade dentro da incubadora.
- c) Para melhorar a transparência da cúpula e facilitar a visualização do recém-nascido.
- d) Para manter os níveis de oxigênio dentro dos parâmetros ajustados.
- e) Nenhuma das anteriores.

XI. Dobradiça e Fecho de Segurança

Questão 24: É possível abrir totalmente a cúpula para limpeza?

- a) Não, a cúpula pode ser limpa apenas através das portinholas.
- b) Não, a cúpula não permite abri-la totalmente para limpeza, apenas parcialmente.
- c) Sim, a cúpula se abre totalmente através de uma trava lateral.
- d) Sim, a cúpula se abre totalmente, mas apenas a assistência técnica é autorizada a abri-la.
- e) Nenhuma das anteriores.

XII. 20- Filtro de Ar

Questão 25: Com quanto tempo de uso o filtro de ar deve ser trocado?

- a) 1 mês
- b) 2 meses
- c) 3 meses
- d) 4 meses
- e) 5 meses

Questão 26: Por que é importante trocar o filtro de ar a cada 3 meses?

- a) Para não afetar a concentração de oxigênio dentro da incubadora.
- b) Para não causar a formação de dióxido de carbono (CO²).
- c) Para não aumentar a temperatura do ar.
- d) Para não afetar a concentração de oxigênio e causar a formação de dióxido de carbono (CO²).
- e) Nenhuma das anteriores.

XIII. Fototerapia Bilitron

Questão 27: A fototerapia quando muito próxima da cúpula da incubadora pode afetar alguns parâmetros. Quais?

- a) Temperatura da parede da cúpula
- b) Temperatura do ar da incubadora
- c) Temperatura da pele do recém-nascido.
- d) Temperatura do ar da incubadora e temperatura da pele do recém-nascido.
- e) Temperatura da parede da cúpula, temperatura do ar da incubadora e temperatura da pele do recém-nascido.

Questão 28: A fototerapia pode interferir no desempenho de qual sensor?

- a) Sensor do oxímetro (SpO₂)
- b) Sensor auxiliar
- c) Sensor de pele
- d) Sensor de circulação de ar
- e) Sensor de temperatura do ar

Questão 29: Como se pode proteger o sensor do oxímetro da interferência do aparelho de fototerapia?

- a) Cobrindo a área do sensor com material brilhante
- b) Não colocando o sensor enquanto se aplica a fototerapia
- c) Diminuindo a temperatura do ar da incubadora
- d) Diminuindo o tempo de aplicação da fototerapia
- e) Cobrindo a área do sensor com material opaco

XIV. Gaveta para Raios X

Questão 30: como é possível acessar a gaveta das placas de Raio X?

- a) Abrindo uma portinhola
- b) Abrindo duas portinholas
- c) Abrindo o painel de acesso frontal
- d) Abrindo o painel de acesso traseiro
- e) Abrindo toda a cúpula

XV. Kit Balança

Questão 31: Qual a principal vantagem de ter uma balança integrada com a gaveta para Raio X?

- a) Evitar maiores manuseios do recém-nascido
- b) Precisão da balança
- c) Economia de tempo para realizar os procedimentos
- d) Melhor uso do espaço disponível sobre o leito
- e) Nenhuma das anteriores

XVI. Kit Comunicação em Rede

Questão 32: É possível integrar a incubadora a uma rede para monitoração geral da unidade em um único ponto?

- a) Não, a incubadora não permite esta comunicação em rede.
- b) Sim, todas as incubadoras já vêm com este sistema de comunicação em rede.
- c) Sim, as incubadoras podem ser integradas a uma rede para monitoração, mas precisam do kit comunicação em rede.
- d) Não, a incubadora só pode ser monitorada visualmente.
- e) Nenhuma das anteriores.

XVII. Limpeza e Conservação

Questão 33: Para limpeza da incubadora após alta do paciente, por exemplo, é necessário:

- a) Separar totalmente a cúpula da base da incubadora.
- b) Separar a cúpula, da base e do suporte da incubadora.
- c) Manter a cúpula acoplada à base e ao suporte da incubadora.
- d) Enviar para assistência técnica para desmontagem.
- e) Nenhuma das anteriores.

Questão 34: Com quais produtos se pode realizar a limpeza da cúpula de acrílico da incubadora?

- a) Desinfetantes que contenham álcool e quaternário de amônia.
- b) Produtos abrasivos.
- c) Hipoclorito de sódio.
- d) Sabão neutro ou detergente enzimático e quaternário de amônia.
- e) Nenhuma das anteriores.

Questão 35: Quanto tempo é necessário aguardar após desligar a incubadora para se retirar o irradiador da resistência e evitar queimaduras?

- a) 5 minutos
- b) 10 minutos.
- c) 20 minutos.
- d) 35 minutos.
- e) 45 minutos.

XVIII. Manuseio da Parede Dupla

Questão 36: Qual a maneira correta de se abrir as portinholas?

- a) Com as pontas dos dedos.
- b) Com as palmas das mãos.
- c) Com o dorso da mão.
- d) Com os cotovelos.
- e) Nenhuma das anteriores.

XIX. Manutenção Preventiva

Questão 37: De quanto em quanto tempo é necessário lavar o filtro de oxigênio?

- a) 1 mês.
- b) 3 meses.
- c) 5 meses.
- d) 6 meses.
- e) 12 meses

XX. Modo de Operação com recém-nascidos

Questão 38: Quanto tempo antes de colocar o recém-nascido na incubadora a mesma deve ser ligada?

- a) 5 minutos.
- b) 10 minutos.

- c) 20 minutos.
- d) 30 minutos.
- e) 60 minutos.

XXI. Operação com Paciente Hipotérmico

Questão 39: Quando a incubadora está no Modo Pele, a incubadora aquece até quantos graus acima da temperatura ajustada para pele do recém-nascido?

- a) 0,5 °C.
- b) 1,0 °C.
- c) 1,5 °C.
- d) 2,0 °C.
- e) Nenhuma das anteriores.

XXII. Operação com Sensores em Geral

Questão 40: Como deve ser posicionado o sensor de PELE do paciente?

- a) Sob o paciente.
- b) Na região axilar.
- c) Na região retal.
- d) Em contato direto com a pele.
- e) Nenhuma das anteriores.

XXIII. Operação do Leito Inclinado

Questão 41: Onde estão posicionados os manipuladores para elevação do leito?

- a) No painel frontal.
- b) No painel lateral direito.
- c) No painel lateral esquerdo.
- d) Nas laterais da incubadora.
- e) No painel traseiro.

XXIV. Painel de Controle - Monitor Display

Questão 42: Onde o Painel de Controle pode ser ligado?

- a) Através do botão na parte traseira da incubadora.
- b) Através do botão na parte lateral direita da incubadora.
- c) Através do botão na parte lateral esquerda da incubadora.
- d) Através do botão na parte anterior da incubadora.
- e) Através do botão na parte anteroinferior da incubadora.

XXV. Painel de Controle – Monocromático

Questão 43: o Painel de controle possui 3 LEDS que correspondem a quais alarmes?

- a) Falta de energia elétrica, Alarmes diversos, Temperatura >37°C.
- b) Falta de energia elétrica, Alarmes diversos, Temperatura >39°C.
- c) Falta de energia elétrica, Alarmes baixa temperatura, Temperatura >39°C.
- d) Falta de sensor, Alarmes diversos, Temperatura >39°C

- e) Falta de energia elétrica, Alarmes diversos, Paciente febril.

Questão 44: Quais as duas ações necessárias para ligar a incubadora?

- a) Ligar a incubadora e o Pannel de controle através de sua chave geral localizada na parte traseira.
- b) Ligar a incubadora através de sua chave geral localizada na parte anteroinferior do equipamento e o Pannel de controle através de sua chave na parte traseira.
- c) Ligar a incubadora através de sua chave geral localizada na parte lateral do equipamento e o Pannel de controle através de sua chave na parte anteroinferior.
- d) Ligar a incubadora através de sua chave geral localizada na parte traseira do equipamento e o Pannel de controle através de sua chave na parte anteroinferior.
- e) Ligar a incubadora através de sua chave geral localizada na parte lateral direita do equipamento e o Pannel de controle através de sua chave na parte lateral esquerda.

Questão 45: Qual ação deve ser realizada se a tela do Pannel de controle estiver escura?

- a) Enviar para o fabricante.
- b) Enviar par assistência técnica.
- c) Chamar a coordenação da unidade.
- d) Verificar o botão de contraste localizado ao lado do botão liga/desliga na parte anteroinferior da incubador.
- e) Nenhuma das alternativas anteriores.

Questão 46: O que indica uma “roda girando” no display?

- a) Indica qual o Modo de operação foi selecionado.
- b) Indica que a incubadora está funcionando.
- c) Indica que a incubadora está aquecendo.
- d) Não tem significado algum.
- e) Nenhuma das alternativas anteriores.

Questão 47: Qual a faixa de temperatura que pode ser mantida no Modo de Temperatura AR?

- a) 10 à 37°C ou de 37 à 39°C.
- b) 20 à 37°C ou de 37 à 40°C.
- c) 20 à 37°C ou de 37 à 39°C.
- d) 20 à 37°C ou de 37 à 41°C.
- e) 20 à 39°C ou de 37 à 41°C.

Questão 48: Os pontos ou limites de alarme são selecionados de que forma?

- a) Automaticamente.
- b) Manualmente.
- c) Limite superior automaticamente e inferior manualmente.
- d) Limite inferior automaticamente e superior manualmente.
- e) Nenhuma das alternativas anteriores.

Questão 49: o que é necessário para colocar a incubadora em Modo AR?

- a) Apertar o botão LIGA na tela de Ajuste da Temperatura da PELE.
- b) Não há necessidade de fazer nenhuma ação para colocar em Modo AR.
- c) Após configurar a temperatura desejada apertar o botão RETORNO.
- d) Após configurar a temperatura desejada apertar o botão LIGA na tela de Ajuste da UMIDADE.

e) Após configurar a temperatura desejada apertar o botão Liga na tela de Ajuste da Temperatura do AR.

Questão 50: Quando se liga a incubadora, por quanto tempo o alarme permanece inibido até que se atinja a temperatura desejada?

- a) 10 minutos
- b) 15 minutos
- c) 20 minutos
- d) 30 minutos
- e) 40 minutos

Questão 51: No primeiro ciclo de aquecimento a incubadora aquece 1°C a mais do que a temperatura ajustada. Em quanto tempo ela retorna para temperatura ajustada?

- a) 5 minutos
- b) 10 minutos
- c) 15 minutos
- d) 20 minutos
- e) 30 minutos