



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

ADAUTO MATUZAKI

**APLICAÇÃO WEB USANDO INTERFACES RICAS PARA A GESTÃO
DE CLÍNICAS DE ENFERMAGEM**

ASSIS
2013

ADAUTO MATUZAKI

**APLICAÇÃO WEB USANDO INTERFACES RICAS PARA A GESTÃO
DE CLÍNICAS DE ENFERMAGEM**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal do Ensino Superior de Assis – IMESA, como requisito do Curso de Bacharelado em Bacharelado em Ciência da Computação.

Orientador: Dr. Almir Rogério Camolesi

Área de Concentração: Informática

ASSIS

2013

FICHA CATALOGRÁFICA

MATUZAKI, Aauto

Aplicação web usando interfaces ricas para a gestão de clínicas de enfermagem/ Aauto Matuzaki. Fundação Educacional do Município de Assis – FEMA – Assis, 2013.

48p.

Orientador: Dr. Almir Rogério Camolesi
Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Aplicações Ricas para Internet. 2. Clinica de enfermagem. 3.JavaFX.

CDD: 001.6
Biblioteca da FEMA

APLICAÇÃO WEB USANDO INTERFACES RICAS PARA A GESTÃO DE CLÍNICAS DE ENFERMAGEM

ADAUTO MATUZAKI

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal do Ensino Superior de Assis, como requisito do Curso de Bacharelado em Ciência da Computação, analisado pela seguinte comissão examinadora:

Orientador: Dr. Almir Rogério Camolesi

Analisador (1): Me. Fábio Eder Cardoso

Analisador (2): Esp. Guilherme De Cleve Farto

ASSIS

2013

AGRADECIMENTOS

Primeiramente a Deus, pela sua infinita bondade e misericórdia, pois sem Ele nada seria possível.

Aos meus pais Ademar e Mitico e minha irmã Adriana pelo incansável apoio, pela ajuda, dedicação, carinho, paciência, enfim, por tudo.

A todos os amigos e colegas que colaboraram comigo, pela amizade e pelo apoio em todos esses anos.

Ao Orientador Prof. Dr. Almir Rogério Camolesi pelo incentivo, pelo apoio, pela paciência e pela orientação.

A todos os professores, pelo conhecimento compartilhado, e por contribuírem para o meu crescimento intelectual.

A todos que direta ou indiretamente contribuíram para a realização deste trabalho os meus sinceros agradecimentos.

“O que importa não é a vitória, mas o esforço,
Não é o talento, mas a vontade,
Não é quem você é, mas quem você quer ser...”

(Autor Desconhecido)

RESUMO

Devido à rápida ascensão da *Internet* e às exigências de um mercado globalizado altamente competitivo, surgiu a necessidade de criar aplicações *web* que proporcionassem interfaces mais interativas e intuitivas, resultando na criação de um novo modelo intitulado como Aplicações Ricas para *Internet* (RIA).

Este trabalho apresenta os principais conceitos desta tecnologia e propõe o desenvolvimento de uma aplicação rica para *Internet*, utilizando a plataforma JavaFX, enfatizando os benefícios que ela pode prover, tanto para o processo de desenvolvimento quanto para a experiência do usuário.

O *software* desenvolvido visa auxiliar os enfermeiros da clínica da Fundação Educacional do Município de Assis no processo de sistematização da assistência em enfermagem e disponibilizar dados científicos para futuras pesquisas e estudos de casos, dentro do contexto de enfermagem.

Palavras Chave: Aplicações Ricas para *Internet*, JavaFX, Aplicações *web*.

ABSTRACT

Due to the rapid rise of the Internet and the demands of a highly competitive global market, the need to create web applications which provide more interactive and intuitive interfaces arose, resulting in the creation of a new model entitled Rich Internet Applications (RIA).

This paper aims to present the main concepts of this technology, and proposes the development of a rich application for the Internet using the JavaFX platform, emphasizing the benefits it can provide both to the development process and to the user's experience.

The software is aimed at helping the nurses of the clinic of the Educational Foundation of the city of Assis in the process of systematization of nursing care, and provide scientific data for future research and case studies within the context of nursing.

Key Words: Rich Internet Applications, JavaFX, web applications.

LISTA DE ILUSTRAÇÕES

Figura 1 - O modelo de interação da uma aplicação <i>web</i> típica	18
Figura 2 - Uma vista geral da história de aplicações de softwares	19
Figura 3 - Funcionalidade de Aplicações Ricas para Internet.....	20
Figura 4 - O modelo de interação de uma aplicação RIA utilizando Ajax.....	21
Figura 5 - Relacionamento entre o Modelo, a View e o Controlador.	23
Figura 6 - Arquitetura JavaFX	26
Figura 7 - Mapa Mental do Sistema	29
Figura 8 - Diagrama de Casos de Uso	31
Figura 9 - Diagrama de Classe	32
Figura 10 - Diagrama Entidade Relacionamento.....	33
Figura 11 - Classes de controle e Arquivos FXML.....	34
Figura 12 - Código FXML da tela de Login	35
Figura 13 - Ferramenta JavaFX Scene Builder	36
Figura 14 - Classe da camada de Controle de Cadastro de Pacientes.....	38
Figura 15 - Tela de Login	39
Figura 16 - Tela Principal	40
Figura 17 - Tela de Pesquisa de Pacientes.....	41
Figura 18 - Tela de Cadastro de Pacientes	41
Figura 19 - Tela de Cadastro de Diagnósticos	42
Figura 20 - Tela de Cadastro de Intervenções de Enfermagem	43
Figura 21 - Tela de Cadastro de Resultados Esperados.....	43
Figura 22 – Tela dados da consulta de enfermagem	44

LISTA DE TABELAS

Tabela 1 - Lista de Eventos.....	29
----------------------------------	----

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface

CSS - Cascading Style Sheets

FEMA – Fundação Educacional do Município de Assis

GPU - Graphics Processing Unit

GUI - Graphical User Interface

HTML - HyperText Markup Language

IDE - Integrated Development Environment

JVM – Java Virtual Machine

MVC – Model-View-Controller

NANDA – North American Nursing Diagnosis Association

NIC – Nursing Interventions Classifications

NOC – Nursing Outcomes Classification

RIA – Rich Internet Application

SAE - Sistematização da Assistência de Enfermagem

SGBD – Sistema de Gerenciamento de Banco de Dados

SQL – Structured Query Language

UC – Use Cases

UI - User Interface

WWW – World Wide Web

UML - Unified Modeling Language

SUMÁRIO

1 – INTRODUÇÃO.....	13
1.1 – OBJETIVOS	14
1.2 – JUSTIFICATIVAS E MOTIVAÇÕES.....	15
1.3 – PERSPECTIVAS DE CONTRIBUIÇÃO	15
1.4 – METODOLOGIA.....	15
1.5 – ESTRUTURAS DO TRABALHO	16
2 – RICH INTERNET APPLICATION (RIA)	17
2.1 – HISTÓRICO DO DESENVOLVIMENTO DE SOFTWARE.....	17
2.2 - CONCEITOS, CARACTERISTICAS E FUNCIONAMENTO	19
3 – TECNOLOGIAS.....	22
3.1 - <i>UNIFIED MODELING LANGUAGE</i> - UML 2	22
3.2 - PADRAO <i>MODEL-VIEW-CONTROLLER</i> (MVC)	22
3.3 - JAVA FX 2.....	24
3.3.1 – Principais características e Recursos	24
3.3.2 – Arquitetura.....	25
3.5 - MYSQL	26
4 – PROPOSTA DO SISTEMA PARA SAE.....	27
4.1 – SAE.....	27
4.2 – PROPOSTA E OBJETIVOS DO SISTEMA.....	28
4.3 – MAPA MENTAL	28
4.4 – LISTA DE EVENTOS.....	29
4.5 – DIAGRAMA DE CASOS DE USO	30
4.6 – DIAGRAMA DE CLASSES.....	31
4.7– DIAGRAMA ENTIDADE RELACIONAMENTO.....	33
5 – DESENVOLVIMENTO	34
5.1 – IMPLEMENTAÇÃO.....	34
5.2 – INTERFACE	39
6 – CONCLUSÃO.....	45
6.1 – TRABALHOS FUTUROS.....	45
REFERÊNCIAS.....	46

1 – INTRODUÇÃO

Com a rápida ascensão da *Internet* e as exigências de um mercado globalizado altamente competitivo, surgiu a necessidade de criar tecnologias inovadoras, com recursos mais interativos, proporcionando a junção de conceitos de aplicações *web* com *desktop* em um mesmo ambiente, resultando na criação de Aplicações Ricas para *Internet* (*Rich Internet Application* - RIA) (LOPES; TAVARES, 2010).

As aplicações ricas para *Internet* representam uma evolução do paradigma cliente-servidor, que durante anos “foi centrado em uma arquitetura empregando um cliente magro”, onde a maior parte dos dados é enviada para serem processados no servidor e o cliente fica responsável apenas pela exibição do conteúdo. Com o aumento do poder computacional por parte do cliente e com a utilização de RIA, é possível que boa parte da aplicação seja executada no sistema local, reduzindo a latência no servidor, além de permitir a execução de funções que aprimoram a experiência do usuário, agregando melhor usabilidade, interatividade e interfaces mais intuitivas (CLARKE; CONNORS; BRUNO, 2010)(LOPES; TAVARES, 2010).

O JavaFX é uma tecnologia que permite facilmente a criação e implementações de aplicações ricas para *Internet*, que se comportam de forma estável em múltiplas plataformas. Fundamentado na tecnologia Java, JavaFX fornece um rico conjunto de gráficos de alto desempenho, com aceleração de *hardware* e motores de mídia (CASTILLO, 2012).

Utilizando-se desta tecnologia, este trabalho propõe o desenvolvimento de um *software* para a clínica de enfermagem da Fundação Educacional do Município de Assis (FEMA), que atua no atendimento à comunidade e é equipada com um consultório e uma sala de procedimentos não invasivos, possibilitando que os alunos da instituição coloquem em prática o conhecimento teórico adquirido no curso. O *software* tem a finalidade de auxiliar no processo de Sistematização da Assistência de Enfermagem (SAE), que atualmente é realizada com o preenchimento de formulários, de forma manual.

Segundo Santos; Nóbrega (2004) a informação é um dos componentes fundamentais no processo de assistência e gerenciamento do serviço de enfermagem. Todos os dados dos pacientes, tais como avaliações, plano de cuidados, condutas, entre outros, são documentados em prontuário pelo enfermeiro (SANTOS; PAULA; LIMA, 2003).

Todavia, nota-se que o volume das informações aumenta nos protocolos de tratamento de enfermagem, e o registro manual não assegura o armazenamento e gerenciamento das mesmas, além de demandar boa parte do tempo (SANTOS; PAULA; LIMA, 2003) (SANTOS; NÓBREGA, 2004).

É importante salientar que as informações dos pacientes necessitam ser compartilhadas entre os profissionais, não só entre enfermeiros, mas com toda a equipe multiprofissional. Quando esses dados encontram-se documentados e organizados sistematicamente, o processo é facilitado, otimizando a comunicação e favorecendo a resolução de problemas. Mas para tanto é necessário um adequado sistema de armazenamento e recuperação de dados (SPERANDIO; ÉVORA, 2005).

Portanto, o desenvolvimento de um *software* para a clínica de enfermagem proporcionará melhorias no processo de atendimento aos pacientes tais como agilidade, praticidade, organização, padronização, clareza, armazenamento de informações, contribuindo para posteriores pesquisas e estudos de casos.

1.1 – OBJETIVOS

Este trabalho tem por objetivo apresentar conceitos de desenvolvimento de aplicações *web*, utilizando a plataforma JavaFX e propor o desenvolvimento de um sistema para auxiliar no processo de Sistematização da Assistência de Enfermagem da clínica de enfermagem da Fundação Educacional do Município de Assis.

O *software* deverá auxiliar os enfermeiros da clínica no atendimento das fases de coletas de dados, prescrição de intervenções de enfermagem da Sistematização da Assistência de Enfermagem e sua documentação de forma informatizada.

1.2 – JUSTIFICATIVAS E MOTIVAÇÕES

Com a *Internet* se tornando o principal meio de comunicação, a tendência dos *softwares* é migrar para a plataforma *web*, pois fornece algumas vantagens como acesso ao sistema de qualquer dispositivo conectado a *Internet*, praticidade no compartilhamento de informações, entre outras.

O JavaFX traz novas possibilidades para o desenvolvimento de aplicações ricas para *Internet*, provendo o suporte necessário que possibilitam o desenvolvimento de interfaces gráficas mais atraentes, como efeitos visuais, sons e animações.

A construção de sistemas informatizados buscam soluções para aperfeiçoar a administração da informação e apoiar o desenvolvimento do processo de prática da enfermagem e assim, obter resultados adequados para agilizar o atendimento centrado no paciente.

1.3 – PERSPECTIVAS DE CONTRIBUIÇÃO

Este trabalho visa demonstrar e incentivar a utilização da tecnologia JavaFX, contribuindo para o crescimento e desenvolvimento da mesma.

Além de auxiliar os enfermeiros e aprimorar o processo de atendimento de pacientes, o produto resultante deste trabalho visa proporcionar dados científicos para futuras pesquisas dentro do contexto da enfermagem além de estudos de casos.

1.4 – METODOLOGIA

A metodologia de pesquisa adotada para este trabalho foi a experimental. Inicialmente foram realizadas pesquisas sobre o assunto e posteriormente foi definido um estudo de caso com o objetivo de demonstrar a utilização dos conceitos adquiridos. Por fim, foi desenvolvido um sistema com base no estudo de caso

proposto, no intuito de aprofundar o aprendizado e de avaliar o uso das tecnologias estudadas.

1.5 – ESTRUTURAS DO TRABALHO

O trabalho está estruturado da seguinte forma:

Capítulo 1 – Foram apresentados os objetivos, justificativas e motivações, perspectivas de contribuição e a metodologia adotada.

Capítulo 2 - Tem por objetivo explicar o desenvolvimento de *software*, o surgimento das aplicações ricas para a *Internet*, seu conceito, as principais características e seu funcionamento.

Capítulo 3 - Apresenta as principais tecnologias que serão utilizadas para o desenvolvimento deste trabalho.

Capítulo 4 - Visa abordar o processo de Sistematização da Assistência em Enfermagem, propor o desenvolvimento de um sistema que auxiliem os enfermeiros neste processo e apresentar sua modelagem.

Capítulo 5 - Visa demonstrar o processo de desenvolvimento do *software* proposto utilizando a tecnologia JavaFX.

Conclusão.

2 – RICH INTERNET APPLICATION (RIA)

Este capítulo tem por objetivo explicar o desenvolvimento de *software*, o surgimento das aplicações ricas para a *Internet*, seu conceito, as principais características e seu funcionamento.

2.1 – HISTÓRICO DO DESENVOLVIMENTO DE SOFTWARE

Atualmente as aplicações de *software* são utilizadas por milhões de pessoas em todo o mundo, todavia, outrora grande parte delas era utilizada apenas por profissionais treinados. Este amplo e acelerado aumento foi proporcionado graças ao desenvolvimento de interfaces mais amigáveis e intuitivas que oferecem maior interação entre a aplicação e o usuário (SMEETS; BONESS; BANKRAS, 2009).

Com a popularização dos computadores pessoais e seu poder gráfico, começaram a surgir aplicações com interfaces visuais aprimoradas, que necessitavam interagir com um servidor central para processamento e armazenamento de dados. Estas aplicações foram denominadas cliente/servidor, e requerem o desenvolvimento de um *software* tanto para o servidor quanto para o cliente, que poderia utilizar vários sistemas operacionais, com isso aumentando o custo do seu desenvolvimento, manutenção e suporte (SMEETS; BONESS; BANKRAS, 2009).

As aplicações para *Internet* foram fomentadas graças a grande propagação da *Internet* e da *World Wide Web* (WWW), ocorrido em meados da década de 90, e salientava a redução do custo de seu desenvolvimento e proporcionar aplicativos para *desktop* de usuários finais de forma abrangente, independente do sistema operacional utilizado. Apesar de seu êxito, este modelo apresenta deficiências significativas quando se trata de sua “riqueza”, ou seja, oferece uma interface pobre, pouco interativa e limitações de mídia e conteúdo. (ALLAIRE, 2002).

Este modelo é baseado em documentos formatados em *Hypertext Markup Language* (HTML) que ficam centralizadas em servidores de aplicações. Para cada

ação em uma aplicação *web* é realizada uma requisição ao servidor, onde ocorre o processamento e geração da próxima página ou documento, que será retornada e exibida pelo navegador. Todo este processo pode demorar centenas de milissegundos para ser realizado, e durante este período não é possível interagir com a aplicação. A figura 01 demonstra o modelo de interação de uma aplicação *web* típica. (SMEETS; BONESS; BANKRAS, 2009).

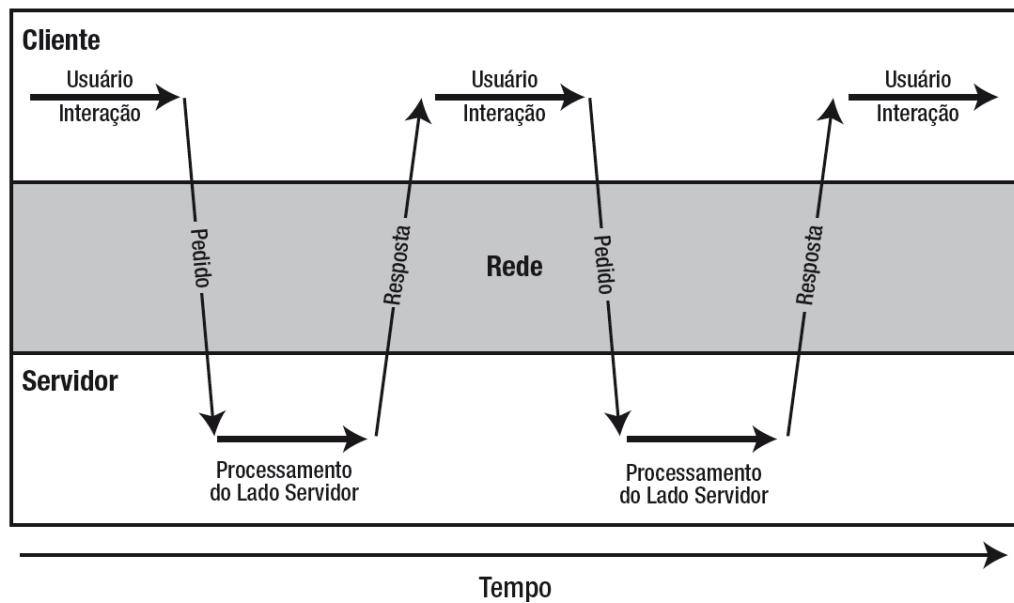


Figura 1 - O modelo de interação de uma aplicação *web* típica (In: SMEETS; BONESS; BANKRAS, 2009, p. 4)

Para atender as exigências de um mercado globalizado altamente competitivo, surgiu a necessidade de criar novos paradigmas de desenvolvimento. O termo RIA foi introduzido pela Macromedia em março de 2002, em um documento técnico de divulgação, estabelecendo um novo modelo de aplicações para a *Internet*, que proporciona a junção da riqueza de aplicações *desktop* com a praticidade de aplicações *web* em um mesmo ambiente, fazendo uso de tecnologias inovadoras, com recursos mais interativos. Sendo assim RIA agrega o que há de melhor em cada plataforma, para tornar a experiência do usuário mais dinâmica e envolvente (ALLAIRE, 2002; LOPES; TAVARES, 2010).

A figura 02 ilustra uma visão geral da evolução do desenvolvimento de *software* considerando o seu custo de desenvolvimento e a experiência proporcionada ao usuário.

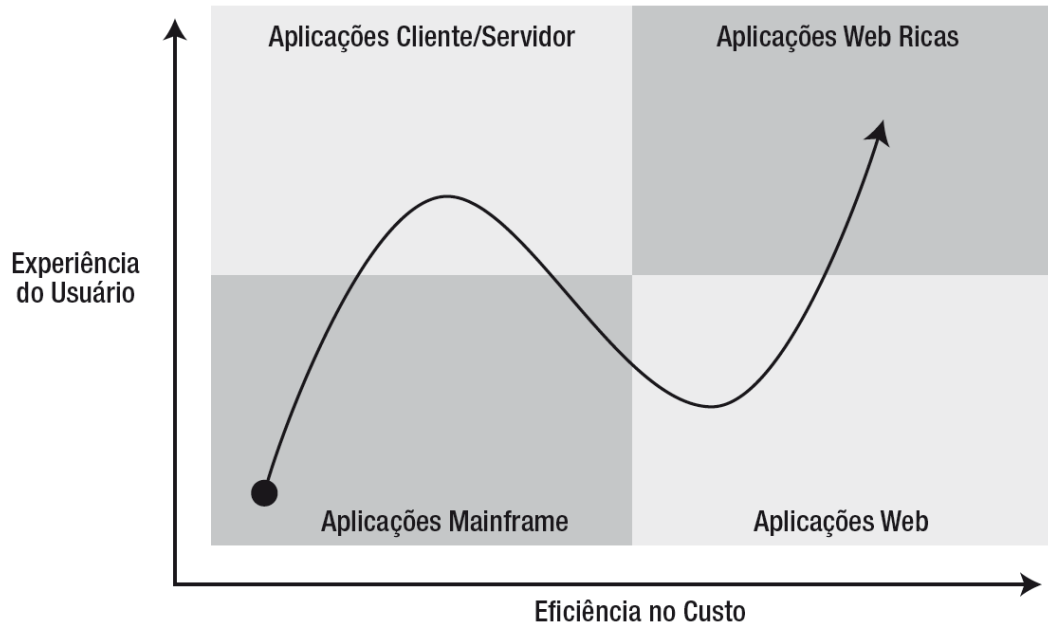


Figura 2 - Uma vista geral da história de aplicações de softwares (In: SMEETS; BONESS; BANKRAS, 2009, p. 1)

2.2 - CONCEITOS, CARACTERÍSTICAS E FUNCIONAMENTO

RIA provê uma solução para aplicações *web*, que combina as melhores funcionalidades da interface de usuário de aplicações *desktop*, como melhor experiência visual, componentes gráficos avançados, o uso de *drag & drop* (arrastar e soltar), animações de objetos e utilização de recursos multimídia, com o amplo alcance e baixo custo de aplicações *web*. Como resultado obtém-se uma aplicação que proporciona ao usuário uma ágil e eficaz experiência, mais intuitiva, com fornecimento de interface interativa, rápido tempo de resposta, e capacidade de trabalhar em modo *online* e *off-line* (DUHL, 2003; LÓPEZ, 2005). A figura 3 demonstra a funcionalidade de RIAs.

As Aplicações Ricas para *Internet* utilizam um modelo onde o conteúdo da camada de apresentação, parte gráfica da aplicação, é armazenado e executado no lado do

cliente, reduzindo significativamente a comunicação com o servidor. A aplicação cliente é capaz de realizar operações e cálculos, enviar e receber dados em segundo plano, de forma assíncrona. Desta forma, ao interagir com a aplicação, não se faz necessário que a página seja redesenhada por completo quando solicitado novos dados ao servidor, assim, o usuário obtém uma resposta quase imediata da aplicação e durante este processo pode continuar interagindo normalmente com a mesma (DUHL, 2003)(LÓPEZ, 2005)(LUNA, 2009).

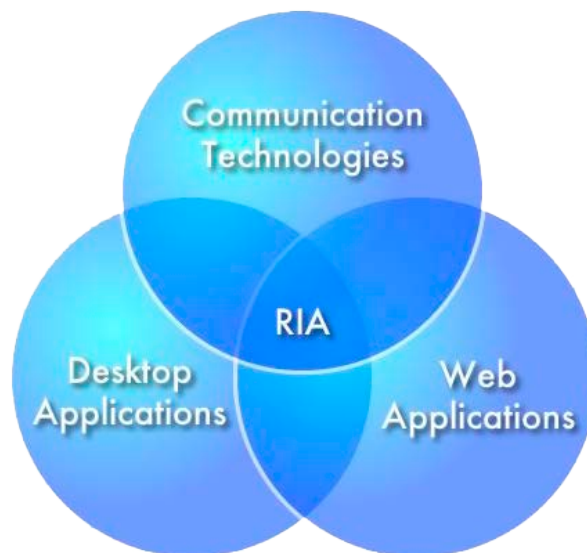


Figura 3 - Funcionalidade de Aplicações Ricas para Internet (In: ADVIT SOLUTIONS, 2012)

A figura 4 apresenta o modelo de interação entre o usuário e uma aplicação RIA, utilizando Ajax. O *Ajax Engine* atua como uma camada intermediária entre a interação do usuário e a comunicação com o servidor, permitindo que o usuário possa interagir com a aplicação, mesmo sendo feitas requisições ao servidor.

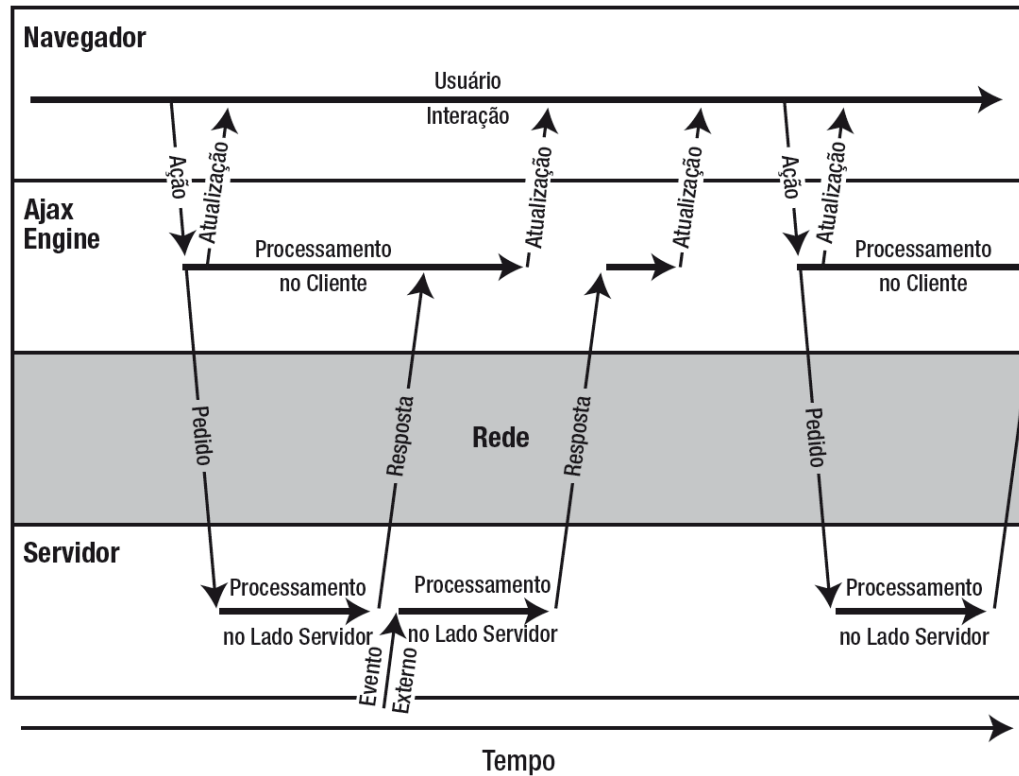


Figura 4 - O modelo de interação de uma aplicação RIA utilizando Ajax (In: SMEETS; BONESS; BANKRAS, 2009, p. 5)

Segundo LÓPEZ (2005), RIAs são caracterizadas por serem aplicações que são iniciadas a partir de uma página *web* ou incluído em seu conteúdo, são executadas de forma assíncrona no cliente, permitem que os usuários realizem operações que são comuns em aplicações *desktop* e sua complexidade de desenvolvimento não difere muito das aplicações *web* tradicionais e requerem a utilização de um “motor” da tecnologia RIA utilizada, que geralmente são *plug-ins*. Além de melhorar a experiência visual, permitir a criação de aplicações mais atrativas, diminuir o consumo de banda utilizado no pela da aplicação e visualização e execução em múltiplas plataformas e dispositivos.

3 – TECNOLOGIAS

Este capítulo tem por objetivo apresentar as principais tecnologias que serão utilizadas para o desenvolvimento deste trabalho.

3.1 - UNIFIED MODELING LANGUAGE - UML 2

A Linguagem de Modelagem Unificada (*Unified Modeling Language - UML*) é uma linguagem visual de modelagem, de propósito geral, utilizada para modelar sistemas computacionais orientados a objetos, e nos últimos anos foi adotada internacionalmente como linguagem padrão, pela indústria de engenharia de *software*. A UML tem o propósito de fornecer múltiplas visões do sistema a ser modelado, antes mesmo de começar a ser desenvolvido, e ajudar engenheiros de *software* a determinarem suas características sob diversos aspectos. Tais características estão relacionadas aos requisitos do sistema, seu comportamento e estrutura lógica (GUEDES, 2009).

Segundo GUEDES (2009), por mais simples que seja o sistema, ele deve ser modelado, visto que cada diagrama UML permite uma melhor análise do sistema no geral, ou parte dele, contribuindo para detecção de falhas e diminuindo possíveis erros durante a fase do desenvolvimento. Outro ponto importante é que os sistemas computacionais costumam sofrer ampliações ou modificações frequentes, e com isso necessitam de uma documentação bem definida e atualizada, contribuindo para ser mantido de forma fácil e com rapidez.

3.2 - PADRAO MODEL-VIEW-CONTROLLER (MVC)

O MVC (*Model-View-Controller*) teve sua origem no fim da década de 80, na Xerox PARC, e é um paradigma de *design* de desenvolvimento de *software*. Seu princípio visa desmembrar o sistema em três camadas distintas, separando as camadas de lógica de negócio da camada de visualização. As camadas implementadas por este

padrão são *Model*, *View* e *Controller*.(GABARDO, 2012)(PALERMO, SCHEIRMAN, BOGARD, 2010).

Gabardo (2012) define as camadas da seguinte forma:

- *Model*: “São os componentes da camada de abstração de dados”. São classes em que sua principal utilização é na gravação e recuperação de dados no banco.
- *View*: É a camada de apresentação, responsável por exibir a interface de usuário.
- *Controller*: É a camada que se encarrega de carregar e realizar a comunicação entre a camada de dados e abstração e a camada de apresentação, efetuando a validação dos dados recebidos.

A figura 05 descreve o relacionamento entre as camadas do padrão MVC. As linhas sólidas indicam uma associação direta, e as linhas tracejadas indicam uma associação indireta.

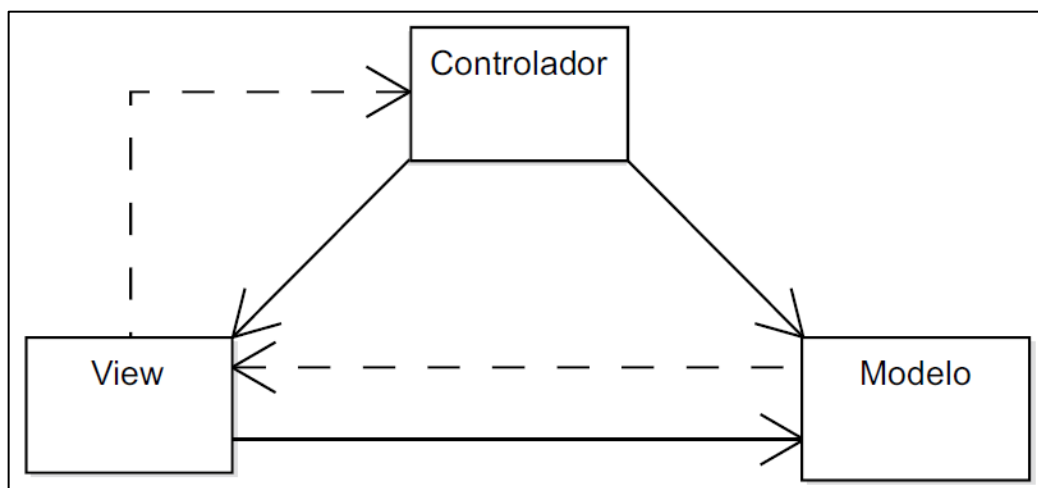


Figura 5 - Relacionamento entre o Modelo, a View e o Controlador. (In: PALERMO, SCHEIRMAN, BOGARD, 2010, p. 32)

Utilizando o padrão MVC em aplicações JavaFX, temos a estrutura de camadas compostas por arquivos FXML que consistem na camada *View*, classes de objetos

que pertencem a camada *Model* e a camada de *Controller*, que contem códigos Java que estabelecem o comportamento da GUI (*Graphical User Interface*) (Hommel, 2012). Ao aplicar este modelo na aplicação em camadas, alguns benefícios são adicionados ao projeto, como fácil manutenção e legibilidade. (FILHO, 2012)

3.3 - JAVA FX 2

O JavaFX é uma tecnologia que permite facilmente a criação e implementações de aplicações ricas para *Internet*, que se comportam de forma estável em múltiplas plataformas, além de proporcionar um ambiente inovador e fornecer um rico conjunto de gráficos de alto desempenho, com aceleração de *hardware* e motores de mídia (CASTILLO, 2012).

Anunciada pela *Sun* em 2007, o JavaFX possuía uma linguagem declarativa de *script* com suporte a uma programação baseada em gráfico e cenário. Atualmente em sua versão 2.0, houve mudanças e melhorias significativas, pois as aplicações agora podem ser escritas em linguagem Java nativa, substituindo o uso do JavaFX *Script*, e seu *runtime*, que é responsável por fornecer as bibliotecas de tempo de execução que são necessárias para executar aplicações JavaFx, também se tornou nativo ao Java 7 (WEAVER, et al., 2010)(CLARKE, CONNORS, BRUNO, 2010) (FILHO, 2012).

3.3.1 – Principais características e Recursos

De acordo com Pawlan, Castillo(2013), as principais características do JavaFX 2 são:

- APIs (*Application Programming Interface*) Java: São projetados para ser uma alternativa amigável para linguagens Java VM;
- FXML: É uma linguagem baseada em XML e fornece uma estrutura pré-definida para construção de interfaces de usuário em aplicações JavaFX. Esta linguagem oferece aos desenvolvedores *web* uma abordagem familiar no design de interfaces, além de não ser compilada, ou seja, não é

necessário recompilar o código para visualizar as alterações.

- *Scene Builder*: Utilizado para criar e projetar interfaces gráficas de usuário (FXML), que pode ser importado para a IDE (*Integrated Development Environment*);
- *WebView*: Um componente que torna possível incorporar páginas *web* dentro de aplicações JavaFX, utilizando a tecnologia *WebKitHTML*;
- Controles internos UI e CSS: É possível desenvolver aplicações cheias de recursos, pois todos os principais controles de interface do usuário necessários são fornecidos pelo JavaFX;
- Suporte *Multitouch*: JavaFX fornece suporte para operações multitoc.
- Motor de mídia de alto rendimento: Suporta a reprodução de conteúdo multimídia da *web*. Ele fornece um quadro de mídia estável que é baseado no *framework* multimídia *GStreamer*.
- Conjunto de controles totalmente repaginados: Botões, gráficos, tabelas, etc.;
- Mecanismo de aceleração gráfica: Aproveita os recursos das GPUs (*Graphics Processing Unit*) modernas;
- *Swing interoperability*: Fácil integração com aplicações e controles *Swing* existente;

3.3.2 – Arquitetura

A arquitetura do JavaFX é constituída por camadas, no topo encontram-se o *Scene Graph* e as APIs públicas do JavaFX. Na parte central se encontram os componentes responsáveis pela renderização gráfica, indicadas pela cor azul, assim como de suporte à multimídia e *web*, *toolkit* de janelas, e na base da arquitetura encontra-se a JVM, conforme ilustra a figura 06. (FILHO, 2012)

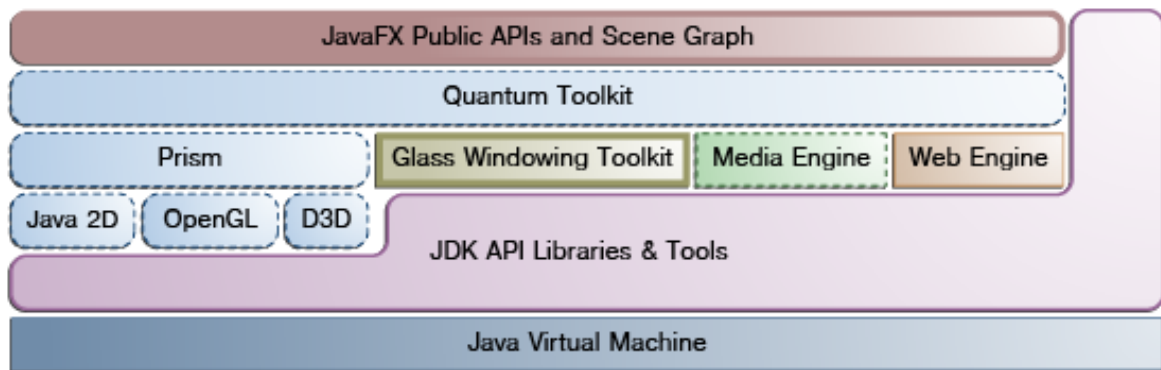


Figura 6 - Arquitetura JavaFX (In: CASTILLO, 2013, p. 1)

3.5 - MYSQL

O MySQL é sistema gerenciador de banco de dados (SGBD) relacional, que utiliza como interface a linguagem *Structured Query Language* (SQL) e uma arquitetura cliente-servidor. O MySQL foi desenvolvido na década de 90, e inicialmente foi projetada para trabalhar com aplicações de pequeno e médio porte. Era mantida pela empresa MySQL AB, que posteriormente foi adquirida pela *Sun Microsystems*. Atualmente possui um alto poder de execução e armazenamento, tem sido o banco de dados mais utilizado em aplicações para *Internet* e é compatível com a maior parte dos sistemas operacionais existentes no mercado (DUBOIS, 2002)(MILANI, 2007).

Segundo Milani (2007) as principais características do MySQL são portabilidade, execução *multithreads*, maior velocidade no acesso aos dados, disponibiliza diversos tipos de tabelas para armazenamento de dados, utiliza conexões com criptografia no tráfego de senhas e funcionalidades de busca no estilo *fulltext search*, além de possuir as funcionalidades implementadas como *stored procedures*, *triggers*, visões, cursores, integridade referencial, replicação e clusterização.

4 – PROPOSTA DO SISTEMA PARA SAE

Este capítulo visa abordar o processo de Sistematização da Assistência em Enfermagem, propor o desenvolvimento de um sistema que auxiliem os enfermeiros neste processo e apresentar sua modelagem.

4.1 – SAE

O processo de sistematização de enfermagem consiste na aplicação de uma metodologia, baseada em princípios científicos, cujo objetivo é identificar problemas referentes à saúde do indivíduo e exercer ações que auxiliem na melhoria, prevenção e recuperação da mesma. Este processo divide-se nas etapas de investigação ou histórico, diagnóstico, intervenção ou implementação e evolução ou avaliação (TANNURE; GONÇALVES, 2008).

As fases são descritas por Tannure;Gonçalves (2008) como:

- A primeira fase do processo refere-se à investigação, que consiste na coleta de informações que determinam o estado de saúde do paciente.
- A segunda etapa é constituída pelo diagnóstico de enfermagem, onde os dados coletados na primeira fase são analisados e interpretados criteriosamente. A taxonomia da NANDA (*North American Nursing Diagnosis Association*) é utilizada para classificar os diagnósticos de enfermagem, sendo um dos sistemas mais utilizados atualmente. Ela é dividida em domínios e classes.
- A próxima etapa é constituída pela implementação da assistência em enfermagem, ela é composta por ações prescritas e necessárias para a obtenção de melhores resultados, como promoção da saúde, minimização de riscos, entre outros. O NIC (*Nursing Intervention Classification*) é a taxonomia utilizada para classificação de intervenções de enfermagem.
- Outra etapa consiste na avaliação ou evolução do processo de enfermagem,

onde é feita a avaliação do progresso do paciente e estabelecidas medidas corretivas, caso necessárias.

- A taxonomia NOC (*Nursing Outcomes Classification*) “contem os resultados esperados para cada diagnóstico de enfermagem da taxonomia NANDA”, e visa obter dados referentes ao estado de saúde do paciente, com base na análise da pontuação obtida pelo uso de escalas (TANNURE; GONÇALVES, 2008).

4.2 – PROPOSTA E OBJETIVOS DO SISTEMA

Para auxiliar no processo de Sistematização da Assistência de Enfermagem da clínica da Fundação Educacional do Município de Assis, foi proposto o desenvolvimento de um sistema, utilizando-se de tecnologia RIA, com intuito de prover maior interação com o usuário possibilitando uma experiência dinâmica e envolvente.

O *software* visa proporcionar melhorias no processo de atendimento aos pacientes, auxiliando os enfermeiros da clínica no atendimento das fases de coletas de dados, prescrição de intervenções de enfermagem, além de oferecer agilidade, praticidade, organização, padronização, clareza no armazenamento de informações, e também contribuir para o fornecimento de dados científicos para futuras pesquisas dentro do contexto da enfermagem além de estudos de casos.

4.3 – MAPA MENTAL

Mapas mentais são diagramas utilizados na organização de informações e ideias, fazendo-se o uso de palavras-chave, cores e imagens com uma estrutura que se irradia a partir de uma ideia central, auxiliando no raciocínio e compreensão.

Com intuito de demonstrar uma visão geral do sistema proposto, foi desenvolvido um mapa mental conforme ilustra a figura 7.

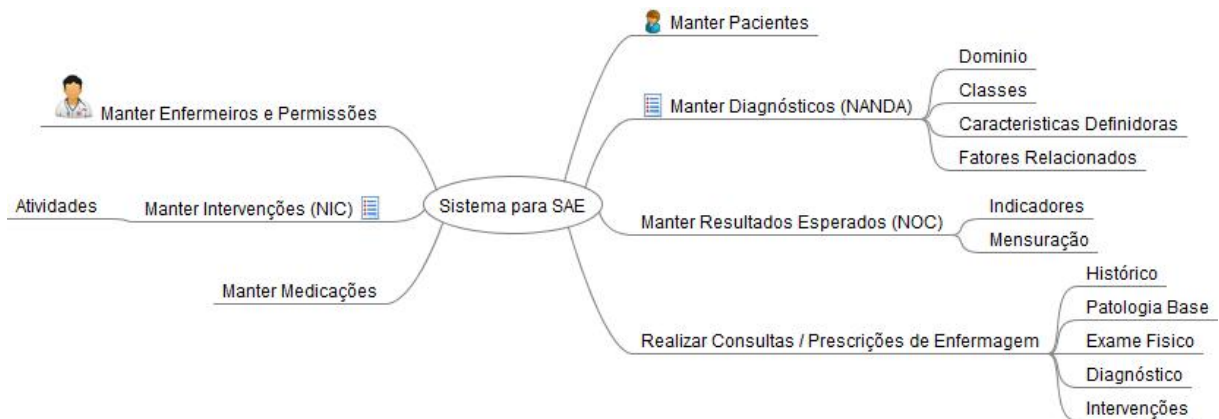


Figura 7 - Mapa Mental do Sistema

4.4 – LISTA DE EVENTOS

A tabela 01 mostra a lista de eventos do sistema.

Lista de Eventos			
Nº	Evento	Descrição	Ator
1	Efetuar <i>Login</i>	Efetuar <i>login</i> no Sistema	Administrador/Usuário
2	Cadastrar Usuários	Efetuar Cadastro de usuários no sistema	Administrador
3	Cadastrar Domínio	Efetuar cadastro de domínio referente ao NANDA	Administrador/Usuário
4	Cadastrar Classes	Efetuar cadastro de Classes referentes ao NANDA	Administrador/Usuário
5	Cadastrar Diagnósticos	Efetuar cadastro de Diagnósticos referentes ao NANDA	Administrador/Usuário
6	Cadastrar Características Definidoras	Efetuar cadastro de Características definidoras referentes ao diagnostico - NANDA	Administrador/Usuário
7	Cadastrar Fatores Relacionados	Efetuar cadastro de Fatores Relacionados referentes ao diagnostico - NANDA	Administrador/Usuário
8	Cadastrar Intervenções	Efetuar cadastro de Intervenções referentes ao NIC	Administrador/Usuário
9	Cadastrar Atividades	Efetuar cadastro de Atividades a serem realizadas referente as Intervenções - NIC	Administrador/Usuário

10	Cadastrar Resultados	Efetuar cadastro dos Resultados Esperados referentes ao NOC	Administrador/Usuário
11	Cadastrar Indicadores	Efetuar cadastro de Indicadores referentes ao NOC	Administrador/Usuário
12	Cadastrar Medicções	Efetuar cadastro de Medicções	Administrador/Usuário
13	Cadastrar Paciente	Efetuar cadastro de Pacientes da Clínica	Administrador/Usuário
14	Cadastrar Patologias	Efetuar cadastro de Patologias Base	Administrador/Usuário
15	Realizar Consultas	Realizar Consultas de enfermagem	Administrador/Usuário
16	Cadastrar Exame Físico	Cadastrar dados dos exames realizados	Administrador/Usuário

Tabela 2 - Lista de Eventos

4.5 – DIAGRAMA DE CASOS DE USO

O Diagrama de Casos de Uso (*Use Case - UC*) é um diagrama UML que tem por objetivo demonstrar, de uma forma geral, os requisitos e o comportamento do sistema. Geralmente é utilizado no levantamento e análise de requisitos, procurando representar os atores, usuários, e as funcionalidades que o sistema deverá prover (Guedes, 2009).

A figura 8 demonstra o diagrama de casos de uso do sistema proposto.

Ao ator enfermeiro são atribuídas funcionalidades para manter pacientes, diagnósticos de enfermagem (NANDA), intervenções de enfermagem (NIC), resultados de enfermagem esperados (NOC), medicamentos, e realizar consultas / prescrições de enfermagem.

O ator administrador consiste no enfermeiro responsável pela unidade, este por sua vez pode manter enfermeiros (cadastrar, alterar, excluir) e dar permissões de acesso aos mesmos, além de herdar todas as funcionalidades do ator enfermeiro.

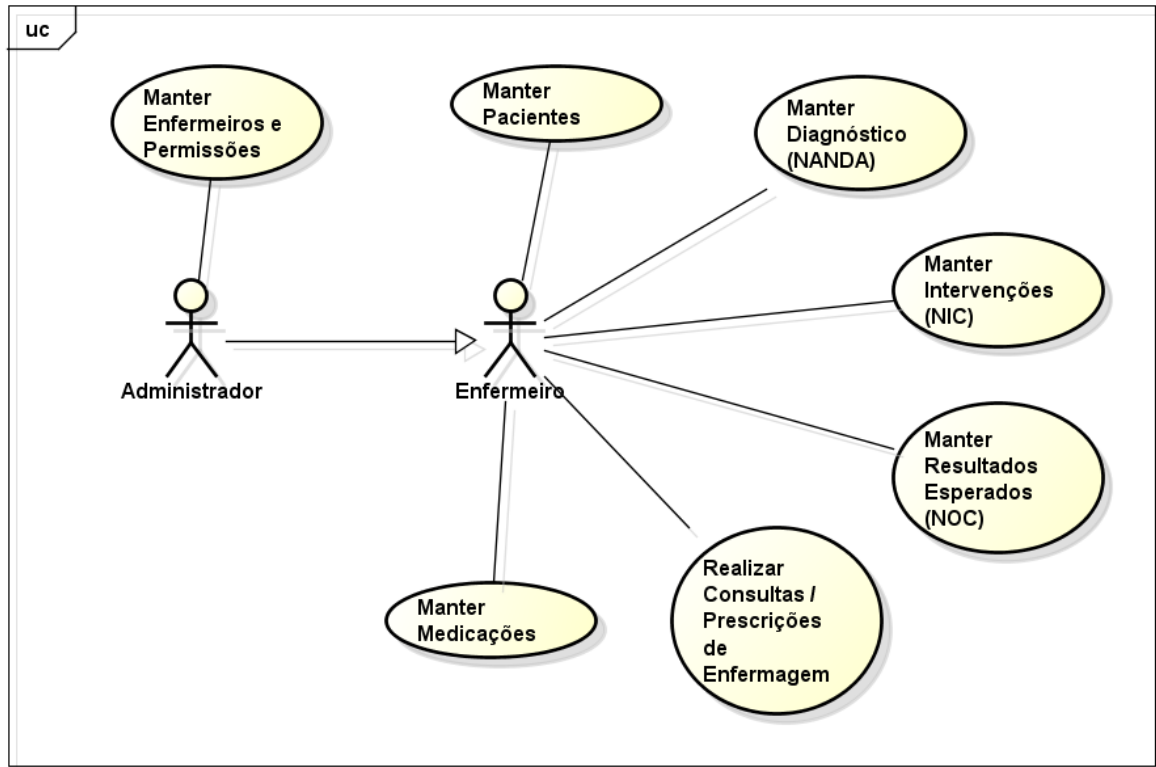


Figura 8 - Diagrama de Casos de Uso

4.6 – DIAGRAMA DE CLASSES

O diagrama de classes é um diagrama UML que define a estrutura de classes a ser implementada no sistema. É considerado um dos diagramas mais importantes, e mais utilizado, pois determinam os relacionamentos entre as classes, seus atributos e métodos (GUEDES, 2009).

A figura 09 ilustra o diagrama de classes do sistema proposto.

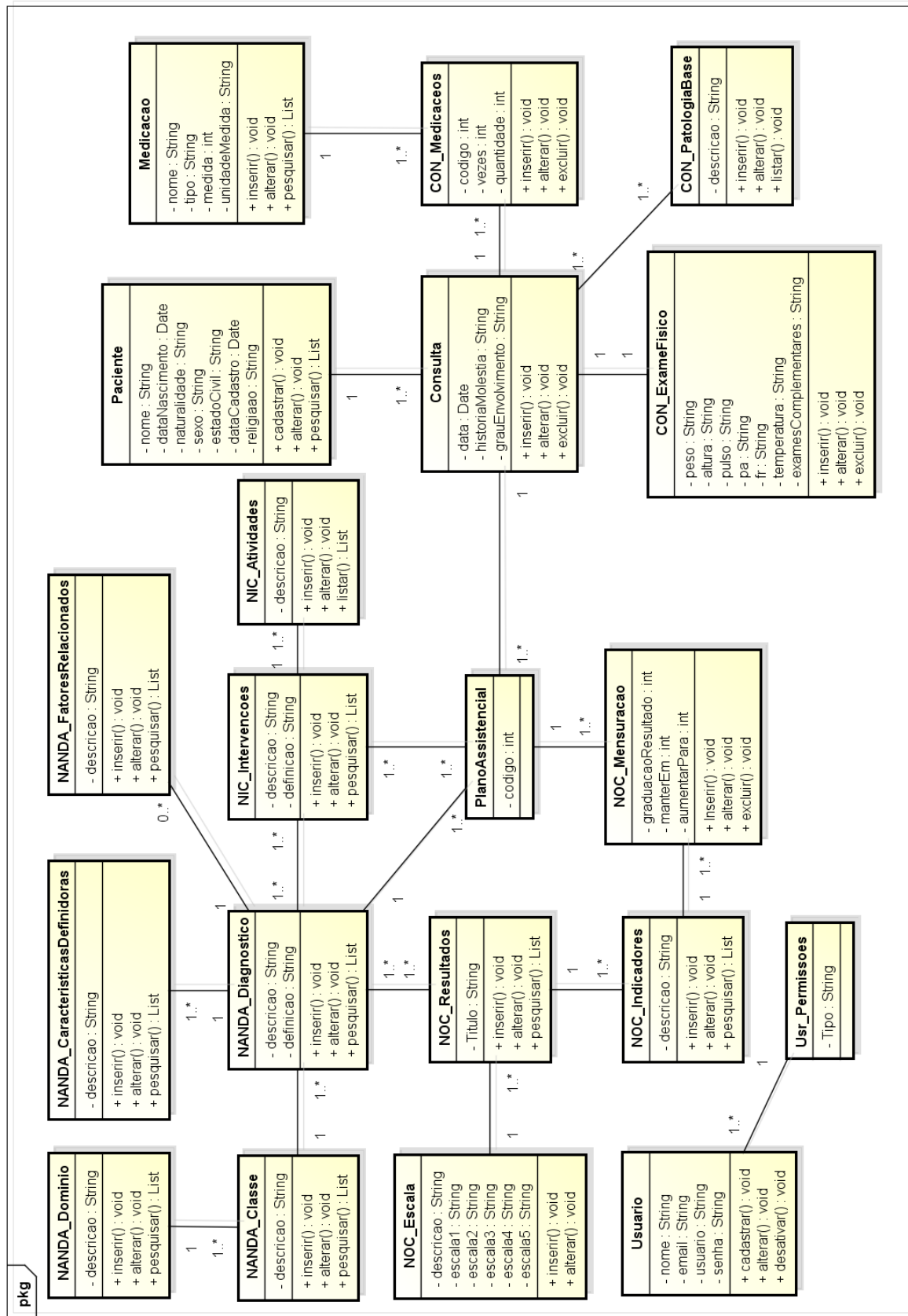


Figura 9 - Diagrama de Classe

4.7- DIAGRAMA ENTIDADE RELACIONAMENTO

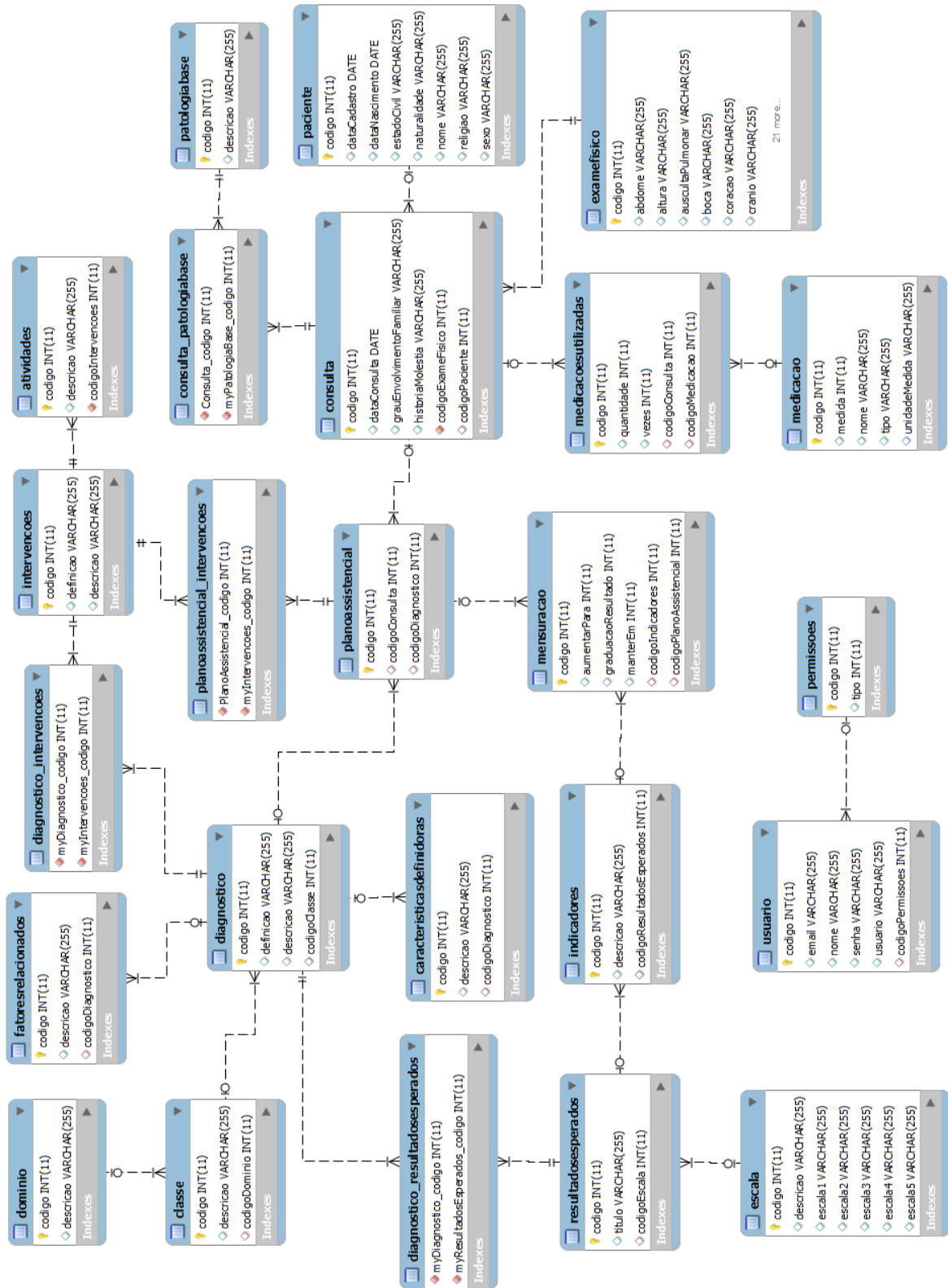


Figura 10 - Diagrama Entidade Relacionamento

5 – DESENVOLVIMENTO

Este capítulo visa demonstrar o processo de desenvolvimento do *software* proposto utilizando a tecnologia JavaFX, para tanto foi utilizada a IDE Netbeans e a ferramenta JavaFX Scene Builder.

5.1 – IMPLEMENTAÇÃO

A fase de implementação do *software* foi baseado no conceito de desenvolvimento dividido em camadas, onde temos a camada de apresentação, representada pelos arquivos FXML, a camada de modelo que equivalem às classes utilizadas na manipulação de dados e a camada de controle, que correspondem às classes java responsáveis por estabelecer o comportamento da interface gráfica de usuário e promover a comunicação entre a camada de apresentação e a camada de modelo. A figura 11 apresenta alguns dos arquivos FXML e classes da camada de controle da aplicação.

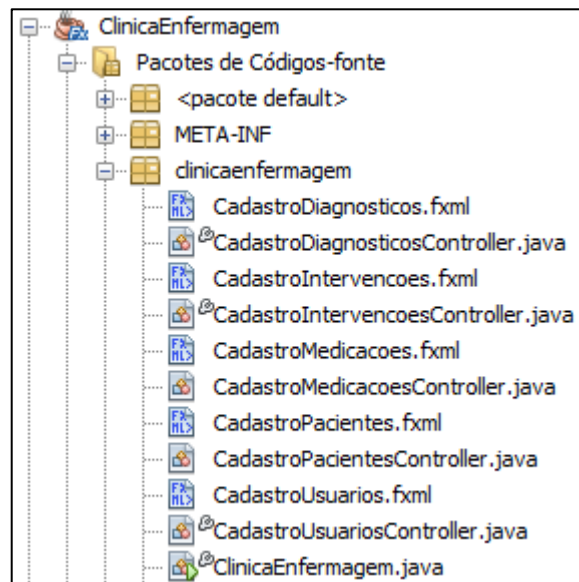


Figura 11 - Classes de controle e Arquivos FXML

A figura 12 ilustra o código do arquivo FXML da tela de *login*, que possui sua estrutura pré-definida baseada em XML.

```

1      <?xml version="1.0" encoding="UTF-8"?>
11     <AnchorPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0"
12         styleClass="mainFxmlClass, background" xmlns:fx="http://javafx.com/fxml"
13         fx:controller="clinicaenfermagem.LoginController">
14         <children>
15             <AnchorPane prefHeight="340.0" prefWidth="600.0" styleClass="top-segment"
16                 AnchorPane.bottomAnchor="60.0" AnchorPane.leftAnchor="0.0"
17                 AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
18                 <children>
19                     <VBox layoutY="69.0" prefHeight="216.0" prefWidth="367.0"
20                         AnchorPane.leftAnchor="117.0" AnchorPane.rightAnchor="114.5">
21                         <children>
22                             <ImageView cache="false" disable="false" fitHeight="65.0"
23                                 fitWidth="250.0" mouseTransparent="false"
24                                 pickOnBounds="true" preserveRatio="true">
25                                 <image>
26                                     <Image url="@logo.png" />
27                                 </image>
28                                 <VBox.margin>
29                                     <Insets bottom="10.0" />
30                                 </VBox.margin>
31                             </ImageView>
32                             <Label text="Usuário:">
33                             <VBox.margin>
34                                 <Insets bottom="5.0" left="5.0" fx:id="x1" />
35                             </VBox.margin>
36                             </Label>
37                             <TextField fx:id="txtUser" prefWidth="-1.0">
38                             <VBox.margin>
39                                 <Insets bottom="20.0" />
40                             </VBox.margin>
41                             </TextField>
42                             <Label text="Senha:" VBox.margin="$x1" />
43                             <PasswordField fx:id="txtPassword" prefWidth="200.0" />
44                         </children>
45                     </VBox>
46                 </children>
47             </AnchorPane>
48             <Button fx:id="btnLogin" mnemonicParsing="false" onAction="#login"
49                 prefWidth="105.0" text="Login" AnchorPane.bottomAnchor="19.0"
50                 AnchorPane.leftAnchor="247.0" AnchorPane.rightAnchor="248.0" />
51         </children>
52         <stylesheets>
53             <URL value="@Estilo.css" />
54         </stylesheets>
55     </AnchorPane>

```

Figura 12 - Código FXML da tela de Login

Para simplificar e agilizar a criação de interfaces gráficas FXML, foi utilizado o JavaFX Scene Builder, que se encarrega de gerar o código FXML para o *layout* de forma automática. Esta ferramenta oferece um ambiente visual simples e intuitivo, que propicia a criação de telas de forma rápida, bastando apenas arrastar e soltar os componentes para uma área de trabalho, possibilitando a modificação de suas propriedades, assim como a aplicação de folhas de estilo (CSS - *Cascading Style Sheets*). A figura 13 exibe a tela principal da ferramenta.

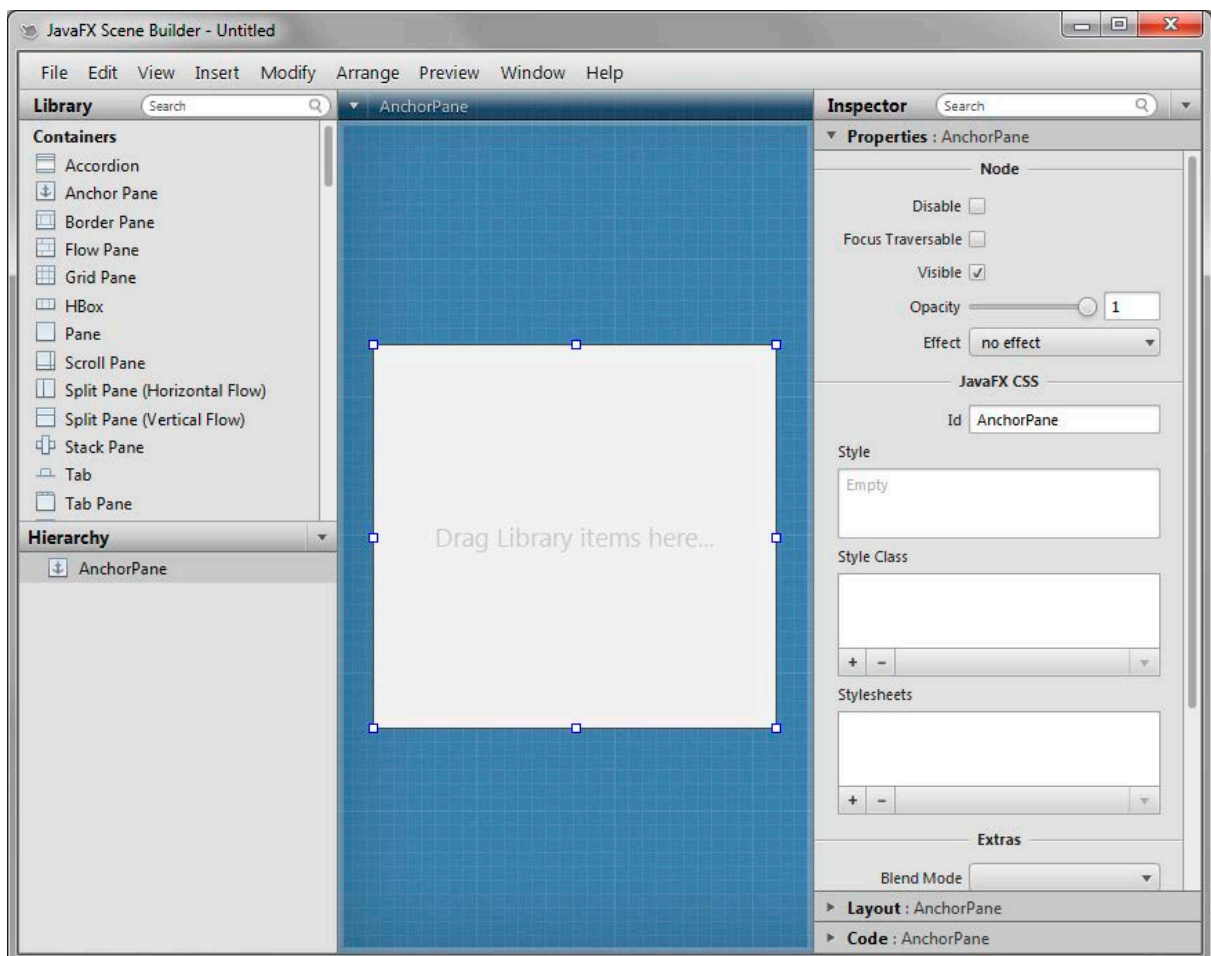


Figura 13 - Ferramenta JavaFX Scene Builder

A ferramenta JavaFX Scene Builder é composta por uma barra de menu, que permite o acesso aos comandos disponibilizados pela ferramenta, o painel de

conteúdo localizado no centro, que é o *container* de cena para os elementos de interface de usuário que compõem o *layout* FXML, o painel de biblioteca onde lista os elementos de interface de usuário do JavaFX, o painel de hierarquia onde é representada a exibição em árvore do *layout* FXML que está construído no painel de conteúdo e o painel de inspeção que contém as propriedades, *layout* e seção de código, e auxiliam no gerenciamento das propriedades do elemento de interface de usuário selecionado.

Em aplicações JavaFX, a classe de controle é uma classe java que contém referências aos controles na interfaces, e também é utilizada para implementar manipuladores de eventos, ou seja, métodos associados aos eventos, para elementos de interface de usuário.

A figura 14 exibe o código da classe de controle de cadastro de pacientes. Nas linhas 16 a 31 são feitas as declarações das referências aos controles existentes na interface (Arquivo FXML), que serão acessados e manipulados pelo controlador, e para tanto é utilizada a anotação @FXML.

```

1  package clinicaenfermagem;
2  import ...
15 public class CadastroPacientesController implements Initializable {
16     @FXML
17     TextField txtNome;
18     @FXML
19     TextField txtDataNascimento;
20     @FXML
21     TextField txtNaturalidade;
22     @FXML
23     TextField txtReligiao;
24     @FXML
25     ComboBox cbxSexo;
26     @FXML
27     ComboBox cbxEstadoCivil;
28     @FXML
29     TextField txtDataCadastro;
30     @FXML
31     Label lblError;
32     private Paciente paciente = new Paciente();
33
34     @FXML
35     public void salvar(ActionEvent event) {
36         try {
37             loadObjectFromForm();
38             PacienteDAO pDao = new PacienteDAO();
39             pDao.save(paciente);
40         } catch (Exception e) {
41             lblError.setText("Erro ao salvar dados: " + e.getMessage());
42         }
43     }
44
45     @Override
46     public void initialize(URL url, ResourceBundle rb) {
47         try {
48             txtDataCadastro.setText(Conversores.dateParaString(new Date()));
49             if (rb != null) {
50                 paciente = (Paciente) rb.getObject("paciente");
51                 loadFieldsFromObject();
52             }
53         } catch (Exception ex) {
54         }
55     }
56
57     public void resetForm() {...}
63
64     public void loadFieldsFromObject() throws Exception {...}
73
74     public void loadObjectFromForm() throws Exception {...}
83 }

```

Figura 14 - Classe da camada de Controle de Cadastro de Pacientes

5.2 – INTERFACE

Ao iniciar o aplicativo a primeira tela a ser exibida é a de *Login*, para que o usuário possa realizar a autenticação no sistema. A figura 15 exibe a tela de *login* da aplicação.

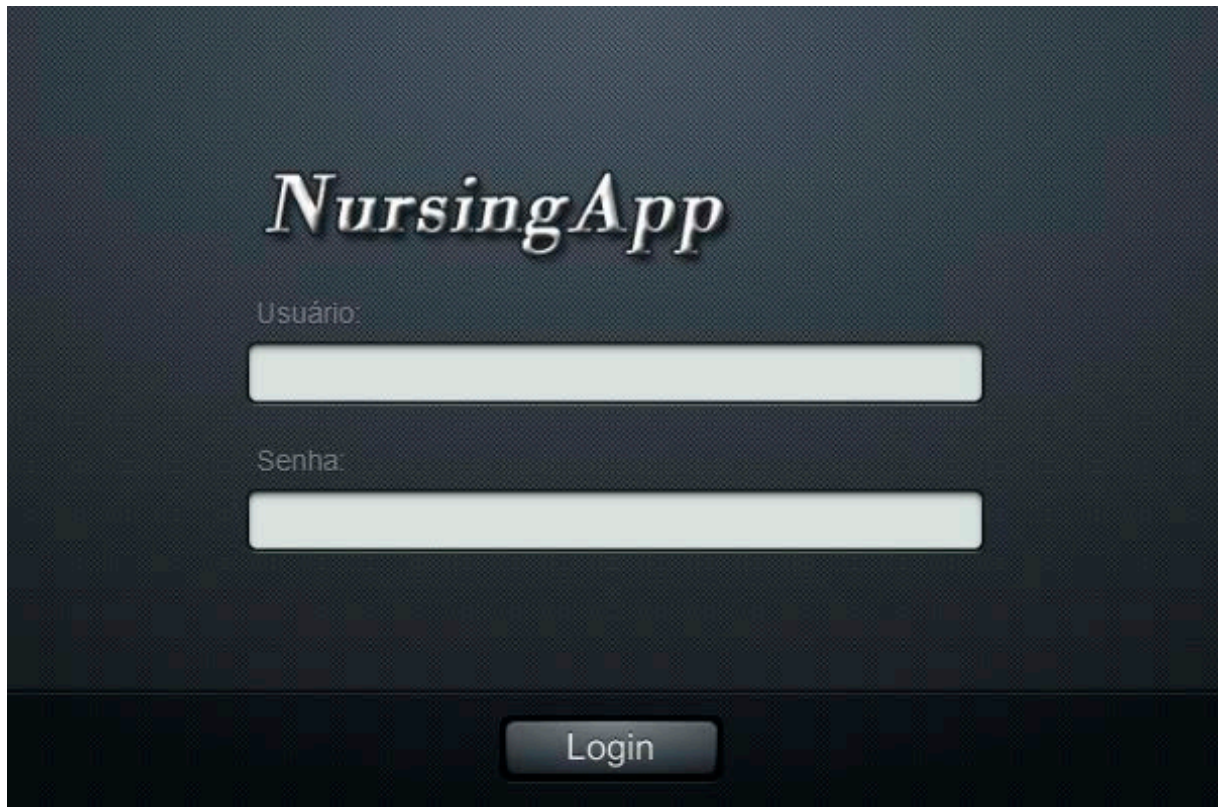


Figura 15 - Tela de Login

Após realizada a autenticação, é exibida a tela principal do programa, conforme ilustrado na figura 16. Nela o usuário administrador possui permissão total de todas as funcionalidades do aplicativo, que podem ser acessadas pelo menu principal, e compreendem o gerenciamento de usuários, pacientes, medicamentos, consultas de enfermagem assim como diagnósticos, intervenções, e resultados esperados.

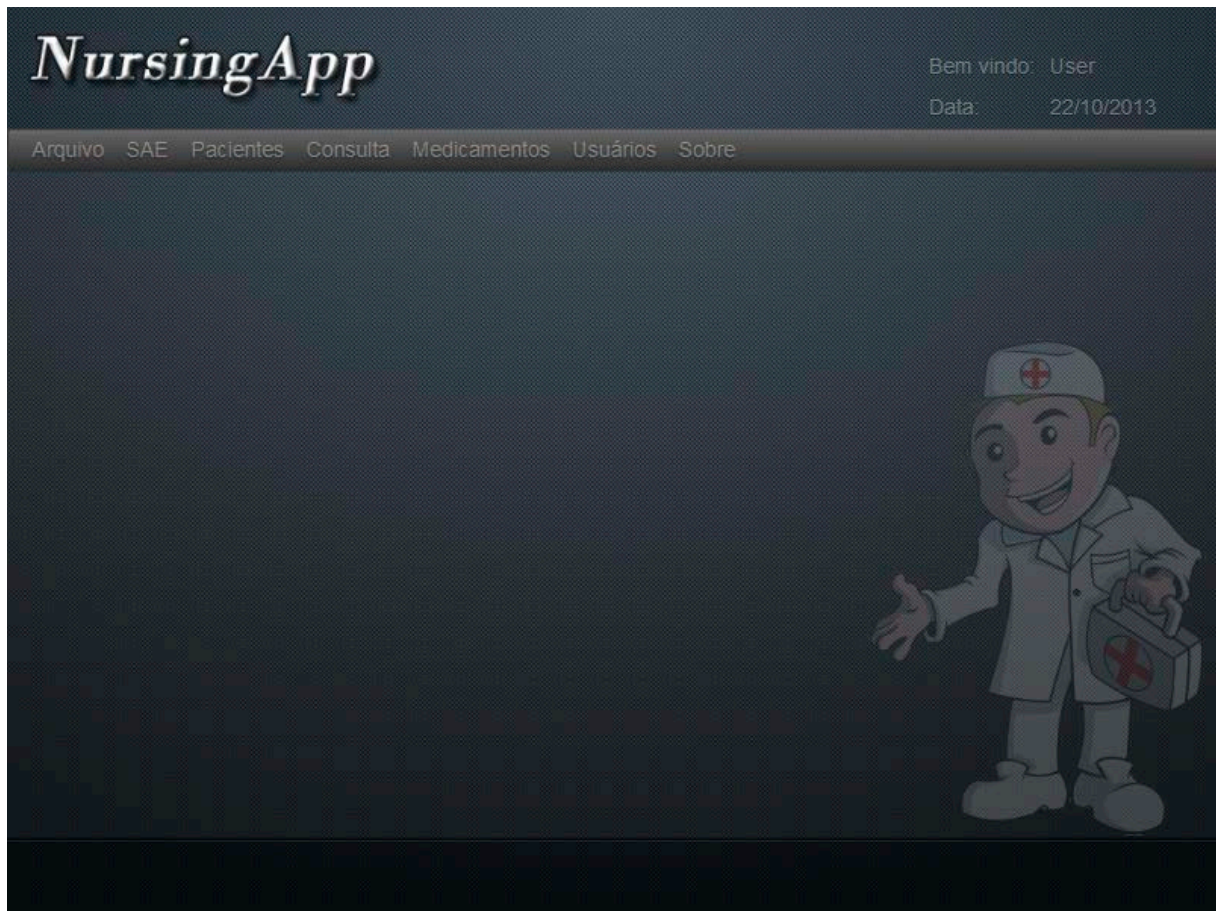


Figura 16 - Tela Principal

A figura 17 exibe a tela de consulta de pacientes, nela são listados todos os pacientes cadastrados no sistema, com possibilidade de filtragem por nome e ordenação de forma crescente ou decrescente pelos campos exibidos na tabela. Ao selecionar um dos pacientes, os seus dados são exibidos de forma detalhada na parte direita da tela de consulta.

Além da pesquisa de usuários, esta tela ainda apresenta algumas funcionalidades, como a possibilidade de cadastrar novos pacientes e também editar e excluir os pacientes já cadastrados.

Ao selecionar as opções de edição ou cadastro de novos pacientes, é exibida outra tela, conforme ilustra a figura 18. Nesta há os campos para o preenchimento das principais informações referentes ao paciente que serão salvas pelo sistema no banco de dados.

Paciente:

Nome	Sexo	Data Nascim...
Paciente 01	Masculino	1990-01-01
Paciente 02	Feminino	1990-01-01
Paciente 03	Masculino	2000-12-13
Paciente 04	Feminino	1950-10-10

DETALHES:

Nome: Paciente 01
D.N.: 01/01/1990
Sexo: Masculino
Est. Civil: Solteiro
Natural.: Pres. Prudente
Religião:
Data Ca... 05/11/2013

Figura 17 - Tela de Pesquisa de Pacientes

CADASTRO DE PACIENTES

Data Cadastro:

Nome:

D.N.: Sexo:

Naturalidade: Estado Civil:

Religião:

Figura 18 - Tela de Cadastro de Pacientes

A tela a seguir ilustra o formulário de cadastro de diagnósticos de enfermagem. Ela possui os campos para que sejam informadas a descrição e a definição do diagnóstico, assim como algumas tabelas para cadastros de características definidoras, fatores relacionados, intervenções e resultados esperados. Ao clicar no botão de adicionar (“+”) em características definidoras e fatores relacionados, é exibida uma janela de diálogo de entrada, para que sejam cadastrados novos itens. Ao clicar no botão de adicionar em intervenções e resultados esperados é exibida uma janela para consulta para seleção dos mesmos, possibilitando também o cadastro de um novo item, caso ainda não esteja cadastrado.

CADASTRO DE DIAGNÓSTICOS

Descrição:

Definição:

Características Definidoras	+	Intervenções	+
Esquecimento de efetuar uma ação em horário planej...	-	Controle da Demência	-
Experiências de esquecimento		Monitoração Neurológica	
Incapacidade de aprender novas habilidades		Controle Hídrico	

Fatores Relacionados	+	Resultados Esperados	+
Alterações ambientais excessivas	-	Cognição	-
Anemia		Concentração	
Distúrbios neurologicos		Estado Neurológico	

Cancelar Salvar

Figura 19 - Tela de Cadastro de Diagnósticos

A figura 20 exhibe a tela de cadastro de intervenções, com os campos de descrição, definição e uma lista de atividades.

CADASTRO DE INTERVENÇÕES

Descrição:

Definição:

Atividades

- Monitorar a função cognitiva usando um instrumento padronizado de investigação
- Incluir os familiares no planejamento, na prestação e na avaliação dos cuidados, durante o tempo desejado
- Identificar os padrões usuais de comportamento em atividades como sono, uso de medicamentos, eliminação, ingestão de alim...
- Determinar a história física, social e psicológica do paciente, seus hábitos comuns e rotinas

Figura 20 - Tela de Cadastro de Intervenções de Enfermagem

A figura abaixo ilustra o formulário para cadastro de resultados esperados e possui os campos de descrição, definição, seleção da escala e uma lista de indicadores.

CADASTRO DE RESULTADOS ESPERADOS

Descrição:

Definição:

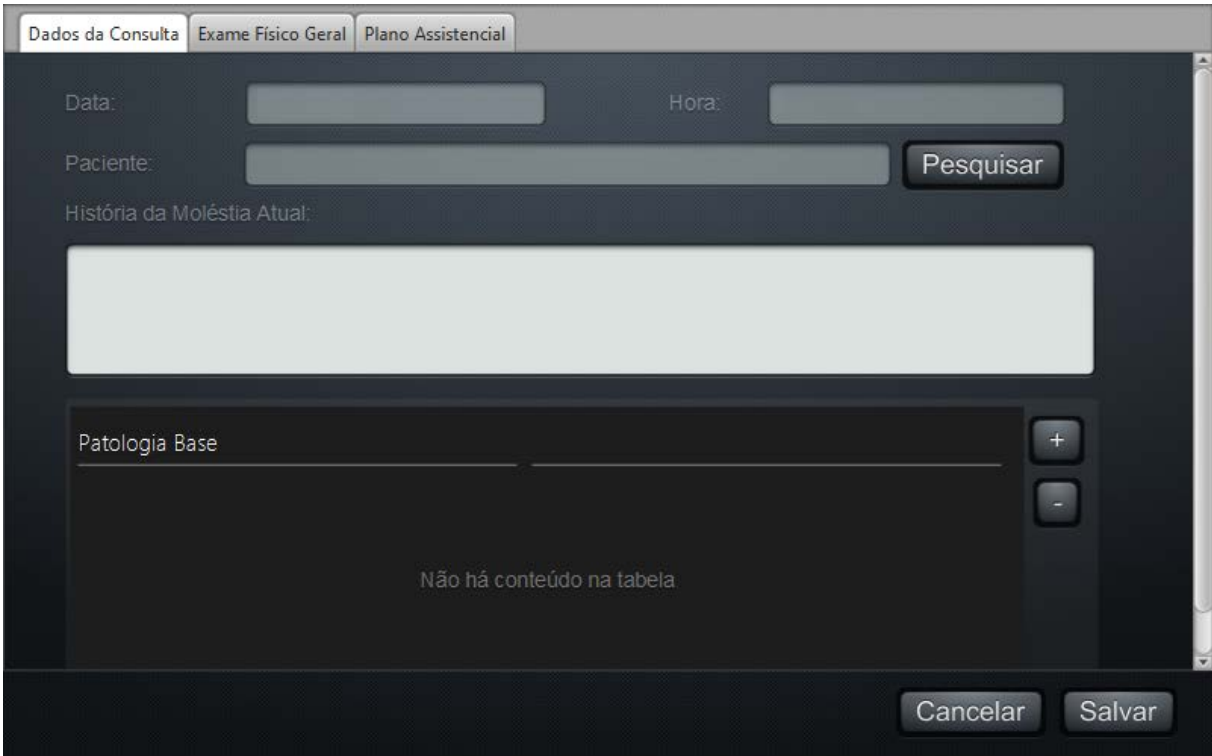
Escala:

Indicadores

- Comunica-se com clareza e adequação para a idade e a capacidade
- Compreende o significado dos eventos e das situações
- Atenção
- Concentração
- Orientação cognitiva
- Demonstra memória imediata

Figura 21 - Tela de Cadastro de Resultados Esperados

Uma das telas mais importantes deste sistema é a tela de consulta de enfermagem, conforme ilustrado na figura 22. Nela é realizado o procedimento de sistematização da assistência em enfermagem, que compreende as fases de coletas de dados, como anamnese e exame físico e plano assistencial, que compreende o diagnóstico, prescrição de intervenções de enfermagem e estabelecimento e acompanhamento dos resultados esperados referentes ao estado de saúde do paciente.



A imagem mostra a interface de usuário de um sistema de consulta de enfermagem. No topo, há uma barra de navegação com três abas: "Dados da Consulta" (selecionada), "Exame Físico Geral" e "Plano Assistencial". Abaixo, há campos de entrada para "Data:" e "Hora:". Um campo "Paciente:" é seguido por um botão "Pesquisar". Abaixo disso, há um campo "História da Moléstia Atual:" com uma área de texto vazia. Na parte inferior, há uma seção "Patologia Base" com uma tabela vazia e o texto "Não há conteúdo na tabela". À direita da tabela, há botões "+" e "-". No canto inferior direito, há botões "Cancelar" e "Salvar".

Figura 22 – Tela dados da consulta de enfermagem

6 – CONCLUSÃO

Aplicações ricas para *Internet* representam uma evolução do paradigma cliente-servidor, onde o conteúdo da camada de apresentação é armazenado e executado no lado do cliente, reduzindo significativamente a comunicação com o servidor. RIA também proporciona ao usuário uma experiência mais dinâmica, com interfaces mais interativas e intuitivas.

A tecnologia JavaFX facilita a criação de aplicações ricas para *Internet*, pois permite a criação de interfaces gráficas de forma simples e rápida, utilizando a ferramenta JavaFX Scene Builder, que se encarrega de gerar o código do *layout* de forma automática.

Além de ser uma tecnologia inovadora, ela proporciona um rico conjunto de gráficos de alto desempenho, e permite a criação de aplicações que se comportam de maneira estável em múltiplas plataformas. Por ter como sua base principal a linguagem java, torna o desenvolvimento do *software* amigável e familiar aos desenvolvedores Java.

O *software* foi desenvolvido com o intuito de explorar e demonstrar o estudo relacionado ao uso desta tecnologia e também contribuir para a clínica de enfermagem, de forma a auxiliar no processo de Sistematização da Assistência em Enfermagem, proporcionando um atendimento ágil e prático ao paciente, além de tornar o armazenamento de informações padronizado, organizado e claro e prover uma base de dados que poderá ser utilizada posteriormente em pesquisas e estudos de casos.

6.1 – TRABALHOS FUTUROS

Devido ao crescimento expressivo do mercado de dispositivos móveis e a praticidade proporcionada pelos mesmos, como sugestão de trabalhos futuros fica o desenvolvimento de um módulo mobile, para que os alunos enfermeiros da clínica de enfermagem da FEMA possam realizar visitas e atendimentos domiciliares.

REFERÊNCIAS

ADIVT SOLUTIONS. Rich Internet Applications. 2013. Formato JPGE. Disponível em: <<http://advitsol.com/advitsolutions/website-design-advit/rich-internet-apps.html>> Acesso em: 06 jun. 2013

ALLAIRE, J. Macromedia Flash MX - A next - generation rich client. Macromedia white paper. San Francisco, 2002, 14 p.

CASTILLO, C. JavaFX Overview. Oracle. Redwood City, 2012, 8 p.

CASTILLO, C. JavaFX Architecture and Framework. Oracle, 10 p. 2013.

CLARKE, J.; CONNORS, J.; BRUNO, E. JavaFXTM: Desenvolvimento de Aplicações de Internet Ricas. Rio de Janeiro: Alta Books, 2010. 325 p.

DUBOIS, P. MySQL Cookbook. Sebastopol: O'REILLY, 2002. 949 p.

DUHL, J. Rich Internet Applications. IDC, Framingham, 2003, 33 p.

FILHO, L. C. Q. Por dentro do JavaFX 2. Revista Java Magazine. Disponível em <<http://www.devmedia.com.br/por-dentro-do-javafx-2-revista-java-magazine-105/25041>>. Acesso em: 01 jul. 2013.

GABARDO, A. C. PHP e MVC: com codelgniter. 1ª ed. São Paulo: Novatec, 2012, 288 p.

GUEDES, G. T. A. UML2: Uma abordagem prática. 1º. Ed. São Paulo: Novatec, 2009, 488 p.

HOMMEL, S. JavaFX: Implementing JavaFX Best Practices. Oracle. 13 p. 2012.

LOPES, J. S.; TAVARES, R. O. JAVAFOX: Uma abordagem ao desenvolvimento de aplicações RIA. Belo Horizonte: FUMEC, 2010. 53 p. Trabalho de Conclusão de

Curso (Graduação em Ciência da Computação) – Faculdade de Ciências Empresariais, Universidade FUMEC, 2010.

LÓPEZ, X. F. Rich Internet Applications. Catalunya: Universitat Politecnica de Catalunya, 2005. 61 p. Treball de fi de Carrera, Escola Politècnica Superior de Castelldefels, 2005.

LUNA, A. M. Geração de interfaces RIA dirigida por ontologias. Rio de Janeiro: PUC-Rio, 2010. 179 p. Dissertação (Mestrado em Informática) - Pontifícia Universidade Católica do Rio de Janeiro, 2009.

MILANI, A. MySQL: Guia do Programador. São Paulo: Novatec, 2007. 400 p.

ORACLE. JAVAFX 2.0: The premier platform for rich enterprise client applications. 2011, 3 p.

PALERMO, J.; SCHEIRMAN, B.; BOGARD, J. ASP.NET MVC em Ação. São Paulo: Novatec, 2010, 432 p.

PAWLAN, M.; CASTILLO, C. JavaFX Overview. Oracle, 6 p. 2013.

SANTOS, S. R.; NÓBREGA, M. M. L. A busca da interação teoria e prática no sistema de informação em enfermagem – enfoque na teoria fundamentada nos dados. Revista Latino-Americana de Enfermagem, v. 12, n. 3, 2004.

SANTOS, S. R.; PAULA, A. F. A.; LIMA, J. P. O enfermeiro e sua percepção sobre o sistema manual de registro no prontuário. Revista Latino-Americana de Enfermagem, v. 11, n. 1, 2003.

SMEETS, B.; BONESS, U.; BANKRAS, R. Programando Google Web Toolkit do iniciante ao profissional. 1. ed. Rio de Janeiro: Alta Books, 2009, 192 p.

SPERANDIO, D. J.; ÉVORA, Y. D. M. Planejamento da assistência de enfermagem: proposta de um software-protótipo. Revista Latino-Americana de Enfermagem, v. 13, n. 6, 2005.

TANNURE, M. C.; GONÇALVES, A. M. P. SAE, Sistematização da Assistência de Enfermagem: Guia Prático. Rio de Janeiro: Guanabara Koogan, 2008, 168 p.

WEAVER, J. L. et al. Plataforma Pro JavaFX - Desenvolvimento de RIA para Dispositivos Móveis e para Área de Trabalho por Scripts com a Tecnologia Java. Rio de Janeiro: Ciência Moderna, 2010, 619 p.