PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
ESCOLA POLITÉCNICA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

HAMILTON SANTOS
TALLES BORGES

F.A.D.D.V.M.
DOCUMENTO DE DESIGN

CURITIBA

2013

versão 1.1

HAMILTON SANTOS
TALLES BORGES

F.A.D.D.V.M.
DOCUMENTO DE DESIGN

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientadores:
Prof. Msc. Attilio Zanelatto Neto
Prof. Msc. Everson Mauda
Prof. Msc. Mario Luiz Bernert

CURITIBA

2013

versão 1.1

SUMÁRIO

1INTRODUÇÃO.....5
2ESTUDO BIBLIOGRÁFICO DA TECNOLOGIA UTILIZADA.....5
2.1OBJETIVOS.....5
2.2LINGUAGENS DE PROGRAMAÇÃO.....5
2.3BANCOS DE DADOS.....6
2.4OUTRAS TECNOLOGIAS.....7
2.4.1JSF 7
2.4.2CSS.....7
2.4.3BOOTSTRAP.....7
2.4.4JQUERY.....7
2.4.5JDBC.....7
2.4.6SPRING MVC.....8
2.4.7HIBERNATE.....8
2.4.8PRIME FACES.....9
2.4.9GLASSFISH.....9
2.4.10GITHUB.....9
3ARQUITETURA.....10
3.1VISÃO GERAL.....10
3.2DECISÕES E JUSTIFICATIVAS.....11
4REALIZAÇÃO DOS CASOS DE USO.....12
4.1CASO DE USO 02 – MANTER HISTÓRICO DO PACIENTE.....12
4.1.1Funcionamento geral.....13
4.1.2 Especificação das interfaces visuais.....14
4.1.3 Especificação dos serviços.....14
4.1.4 Especificação da camada de persistência.....15
4.2CASO DE USO 05 – MANTER CATEGORIA.....16
4.2.1Funcionamento Geral.....16
4.2.2Especificação das interfaces visuais.....16
4.2.3Especificações dos serviços.....16
4.2.4Expecificacao da camada de persistência.....18
5COMPONENTES COMUNS.....18
6DIAGRAMA DE CLASSES GERAL DA APLICAÇÃO.....19

7MODELO FÍSICO DE DADOS.....20
8PROTÓTIPO DAS INTERFACES.....20
9CONSIDERAÇÕES FINAIS.....20
REFERÊNCIAS.....21

1 INTRODUÇÃO

A ferramenta de auxilio na decisão do desmame de ventilação mecânica ou F.A.D.D.V.M. é um sistema especialista que irá apoiar o profissional fisioterapeuta intensivista na tarefa de decisão quanto à retirada de um paciente da ventilação mecânica, a fim de reduzir os riscos que a decisão precipitada nesse momento pode causar ao paciente, a intensão do sistema é também reduzir os gastos extras que podem ser causados por esta decisão.

Atualmente a decisão quanto à este procedimento é feita de forma empírica pelo fisioterapeuta, a partir de dados coletados tanto em exames como a gasometria quanto na anamnese, porém, a não padronização deste processo pode causar uma diferença nos resultados da análise de fisioterapeuta para fisioterapeuta.

Além da padronização dos resultados, o sistema tem por objetivo, auxiliar o profissional em início de carreira, que ainda não possui tanta experiência garantindo que certos parâmetros da decisão serão sempre analisados, e analisados de forma correta.

Este documento irá apresentar o sistema em desenvolvimento do ponto de vista técnico, apresentando os detalhes sobre as tecnologias que serão utilizadas, detalhes gerais da arquitetura do sistema, realização de casos de uso e conterá também um protótipo de como ficará a aplicação com o seu comportamento simulado.

2 ESTUDO BIBLIOGRÁFICO DA TECNOLOGIA UTILIZADA

2.1 OBJETIVOS

Esta seção tem por objetivo apresentar um estudo bibliográfico sobre as tecnologias utilizadas neste projeto, permitindo aos envolvidos no projeto uma visão geral da tecnologia utilizada.

2.2 LINGUAGENS DE PROGRAMAÇÃO

Java EE 6, esta linguagem foi definida a pedido do cliente, pois será de mais fácil manutenção e futuros aprimoramentos pelos desenvolvedores e analistas que ficarão responsáveis pelo mantenimento do mesmo.

Ela é uma linguagem com suporte a multicamadas, distribuída e multiusuário, rodando em um servidor de aplicação, podendo ser acessado de qualquer computador com acesso ao mesmo.

A utilização desta linguagem dará ao cliente uma maior flexibilidade, pois é uma linguagem multiplataforma, podendo rodar em qualquer SO, e sendo assim, pode rodar em qualquer configuração de servidor, desde que o mesmo atenda as especificações de performance aceitáveis para o funcionamento do sistema.

A Linguagem é Orientada a Objetos, o que a torna uma ferramenta extremamente poderosa. E é também Multi-Threaded, o que significa dizer que ela suporta processamento paralelo múltiplo podendo ter melhoramentos de performance.

2.3 BANCOS DE DADOS

O banco de dados que iremos utilizar é o **MySQL**, também a pedido do cliente, pois é o banco de dados atualmente utilizado pelo sistema de prontuário eletrônico e o cliente tem planos de integrar os dois sistemas no futuro.

MySQL é um dos bancos de dados opensource mais utilizados devido à sua robustez e flexibilidade, ele é amplamente utilizado por grandes empresas como Wikipedia, Google, Facebook e Twitter e possui, versões gratuitas e pagas, dependendo dos fins do software que irá fazer uso dele e das necessidades da empresa.

Escrito em C e C++ o MySQL é bastante performático, Funciona em diversas plataformas. APIs para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby e Tcl estão disponíveis.

Suporte total a multi-threads usando threads diretamente no kernel. Isto significa que se pode facilmente usar múltiplas CPUs, se disponível.

Tabelas em disco (MyISAM) baseadas em árvores-B extremamente rápidas com compressão de índices.

Um sistema de alocação de memória muito rápido e baseado em processo(thread).

Joins muito rápidas usando uma multi-join de leitura única otimizada.

Tabelas hash em memória que são usadas como tabelas temporárias.

Funções SQL são implementadas por meio de uma biblioteca de classes altamente otimizada e com o máximo de performance. Geralmente não há nenhuma alocação de memória depois da inicialização da pesquisa.

2.4 OUTRAS TECNOLOGIAS

2.4.1 JSF

Um framework MVC de aplicações web baseado em Java, que se destina a simplificar o desenvolvimento de interfaces de usuário baseadas em web, iremos usar seus componentes visuais, suas regras de navegação, conceitos de backbeans e regras de validação que serão bastante uteis no projeto.

2.4.2 CSS

Uma linguagem de fácil configuração e entendimento para alteração e definição de parâmetros de aparência em websites, permitindo com facilidade, que alterando apenas um arquivo, se consiga redefinir a aparência de todas as paginas da aplicação web.

2.4.3 BOOTSTRAP

Desenvolvida pela equipe do Twitter para criação de websites, ela possui uma ampla biblioteca de componentes visuais e ferramentas baseadas em HTML, CSS e javascript, para a configuração visual do website.

2.4.4 JQUERY

Biblioteca javascript desenvolvida para simplificar os scripts cliente-side que interagem com o HTML, possui vários componentes visuais interessantes para serem usados em aplicações web.

2.4.5 JDBC

A aplicação utilizará JDBC para se comunicar com o banco de dados, ele é uma API com várias classes e métodos para envio de instruções SQL para qualquer

banco de dados, cada banco de dados fornece um driver que implementa métodos definidos pelo JDBC para realizar a comunicação da aplicação que utiliza o JDBC e o banco de dados.

2.4.6 SPRING MVC

Um framework open source para a plataforma Java criado por Rod Johnson. Trata-se de um framework não intrusivo, baseado nos padrões de projeto inversão de controle (IoC) e injeção de dependência.

No Spring o container se encarrega de "instanciar" classes de uma aplicação Java e definir as dependências entre elas através de um arquivo de configuração em formato XML, inferências do framework, o que é chamado de auto-wiring ou ainda anotações nas classes, métodos e propriedades. Dessa forma o Spring permite o baixo acoplamento entre classes de uma aplicação orientada a objetos.

O Spring possui uma arquitetura baseada em interfaces e POJOs (Plain Old Java Objects), oferecendo aos POJOs características como mecanismos de segurança e controle de transações. Também facilita testes unitários e surge como uma alternativa à complexidade existente no uso de EJBs. Com Spring, pode-se ter um alto desempenho da aplicação.

Esse framework oferece diversos módulos que podem ser utilizados de acordo com as necessidades do projeto, como módulos voltados para desenvolvimento Web, persistência, acesso remoto e programação orientada a aspectos.

2.4.7 HIBERNATE

O Hibernate é um framework para o mapeamento objeto-relacional escrito na linguagem Java, mas também é disponível em .Net como o nome NHibernate. Este framework facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (XML) ou anotações Java (veja Annotation (java)).

O objetivo do Hibernate é diminuir a complexidade entre os programas Java, baseado no modelo orientado a objeto, que precisam trabalhar com um banco de dados do modelo relacional (presente na maioria dos SGBDs). Em especial, no desenvolvimento de consultas e atualizações dos dados.

Sua principal característica é a transformação das classes em Java para tabelas de dados (e dos tipos de dados Java para os da SQL). O Hibernate gera as chamadas SQL e libera o desenvolvedor do trabalho manual da conversão dos dados resultante, mantendo o programa portável para quaisquer bancos de dados SQL.

2.4.8 PRIME FACES

PrimeFaces é uma biblioteca de componentes feita para JSF, se tornando um conjunto de novos componentes visuais em adição aos componentes existentes do JSF, possui uma vasta gama de componentes que facilitarão o desenvolvimento da interface gráfica da aplicação.

2.4.9 GLASSFISH

Utilizaremos o servidor de aplicação Glassfish, por ser totalmente compatível com Java EE, incluindo JSF, EJB e JPA, sendo assim, recomendado e pouco provável de causar problemas com a configuração do nosso projeto, outro motivo importante para termos escolhido Glassfish foi o fato de ele utilizar a licença GPL, mesma licença que pretendemos dar a nosso projeto, sendo assim, necessária em todas as ferramentas utilizadas.

2.4.10 GITHUB

Estamos utilizando também a ferramenta GitHub para versionar nossos arquivos, códigos e documentos do projeto, podendo assim controlar com segurança o versionamento e tendo assim uma maior rastreabilidade dos avanços, além de uma garantia de não haver perda de informação durante o avanço do projeto.

3 ARQUITETURA

3.1 VISÃO GERAL

O padrão arquitetural que utilizaremos é o **MVC**, padrão mais utilizado para aplicações java web, por separar as camadas de apresentação, controle e modelo, sendo assim de fácil compreensão e manutenção.

Com o aumento da complexidade das aplicações desenvolvidas, sempre visando a programação orientada a objeto, torna-se relevante a separação entre os dados e a apresentação das aplicações. Desta forma, alterações feitas no layout não afetam a manipulação de dados, e estes poderão ser reorganizados sem alterar o layout.

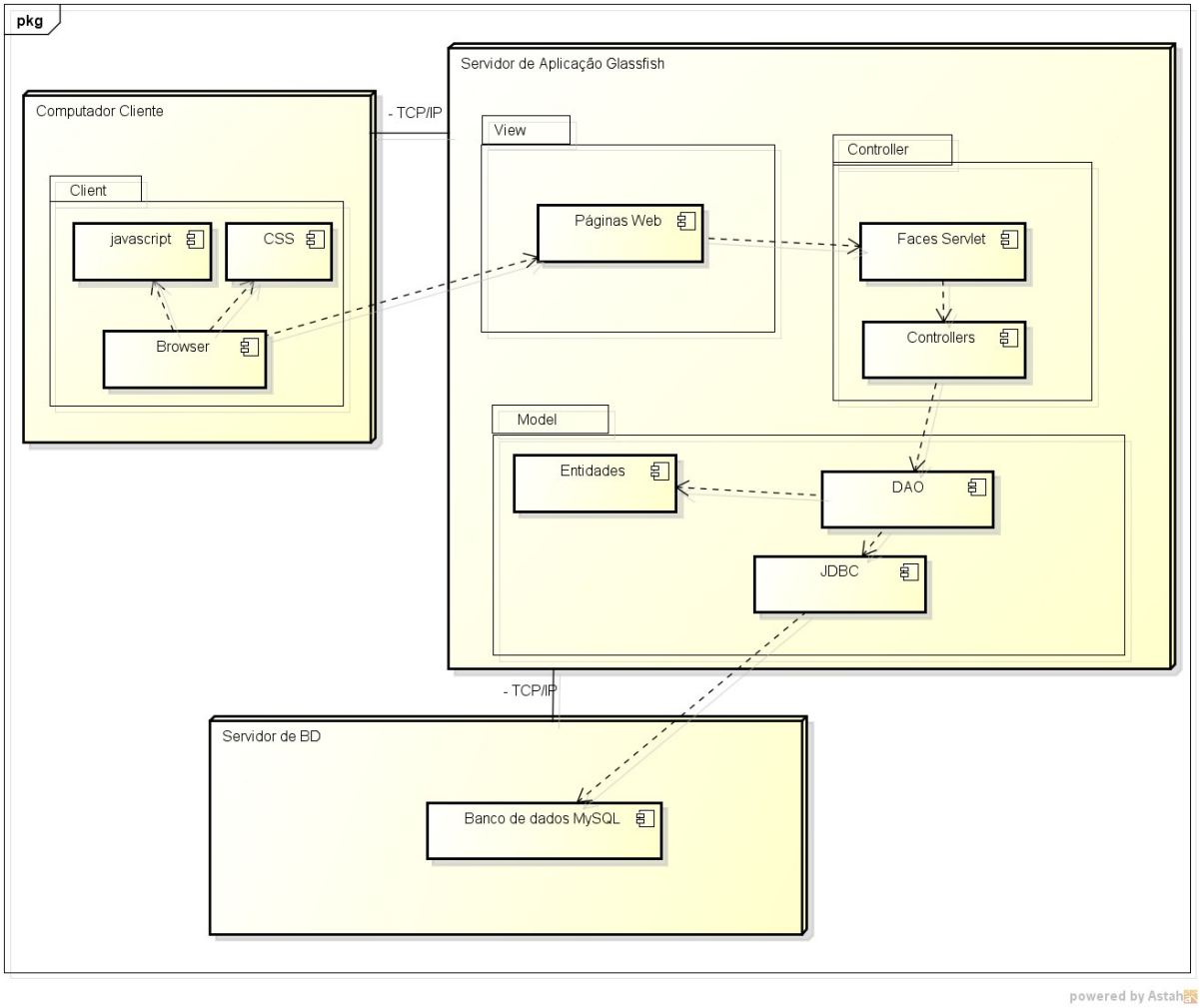
Esse padrão resolve este problema através da separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois: o controlador.

Na camada de apresentação será onde utilizaremos boa parte dos frameworks, CSS, Bootstrap e JQuery serão usados nesta camada para tornar a aparência da aplicação mais agradável e mais fácil de modificar, esta é a camada que o terá interação direta com o usuário, é a camada que é apresentada do browser.

A camada de controle será a camada que irá implementar as regras do sistema e regras de negócio, assim como regras de navegação e validação de dados, Ela é a camada que reage a entrada de dados do usuário, chamando o método apropriado na camada Model, é esta camada que define a navegação na aplicação e altera as informações

A camada modelo será a camada da aplicação que irá representar os valores que estão no banco de dados durante sua utilização e alteração, esta camada que irá se comunicar com o banco de dados e utilizar suas informações para repassá-las às outras camadas do sistema.

Abaixo está o diagrama de implantação da F.A.D.D.V.M. para melhor compreensão do padrão que adotamos.



3.2 DECISÕES E JUSTIFICATIVAS

- Linguagem escolhida em reunião com o cliente, devido ao amplo uso de java e maior facilidade de manutenção do código por outros desenvolvedores que possam assumir o desenvolvimento de possíveis correções e melhoramentos.
- MVC escolhido por ser um framework amplamente usado e considerado uma boa prática para o mantenimento do sistema e para o fácil entendimento do código por programadores que possam precisar analisá-lo.
- JSF foi escolhido por ser o framework mais indicado para aplicações java EE, ele é um dos mais atuais e possui fácil manutenção e compreensão, possui controle de navegação entre paginas,

gerenciamento de objetos, validação de tratamento de erros entre outras vantagens que foram levadas em consideração.

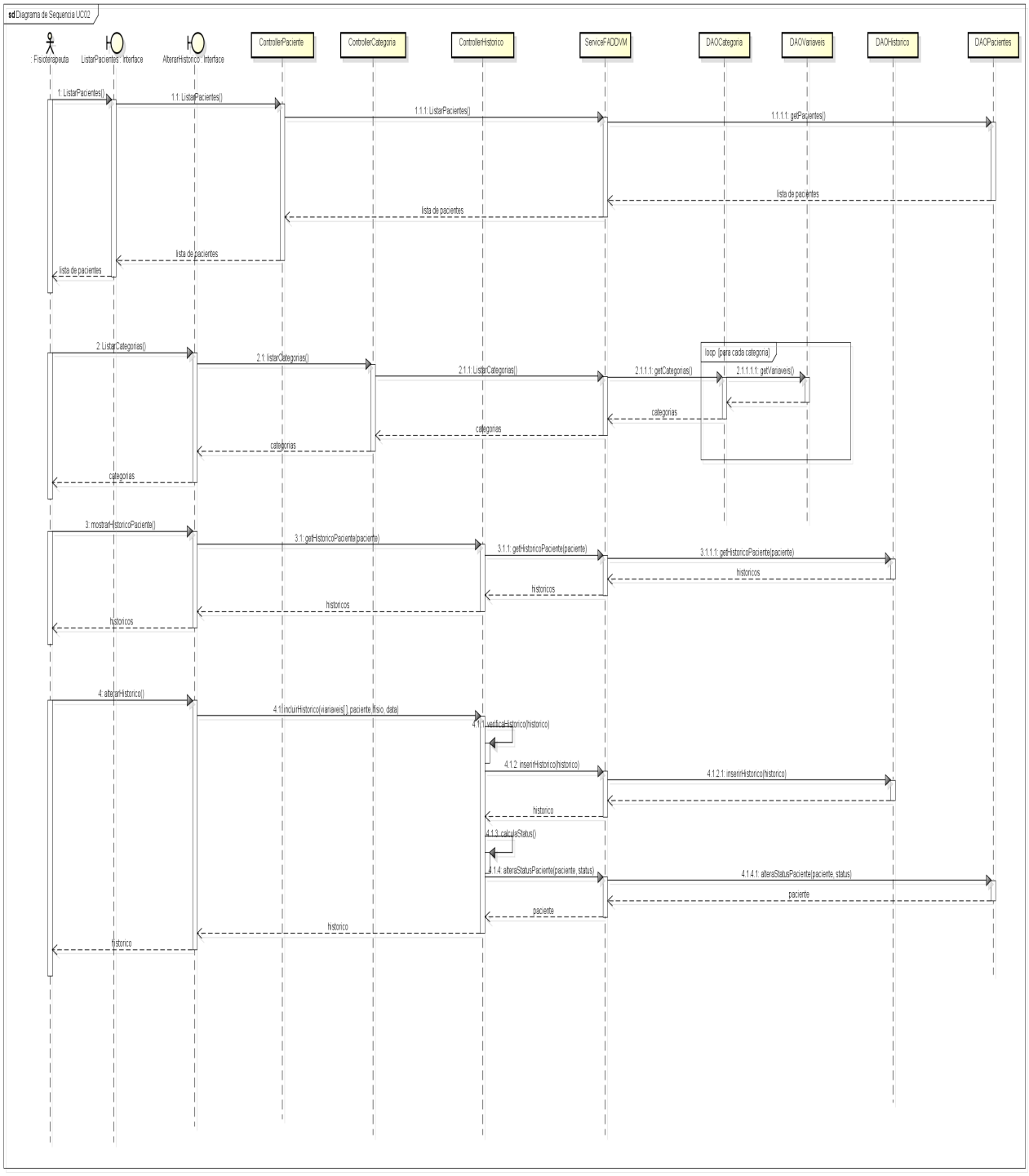
- Utilizaremos os componentes mais simples que encontramos para a manipulação da interface gráfica, utilizando as tags fornecidas pelo JSF assim como leves modificações utilizando CSS, javastcript e componentes fornecidos por JQuery, PrimeFaces e Bootstrap.
- Decidimos por MySQL também em reunião com o cliente, ele foi o BD escolhido para ser utilizado no nosso sistema por ser um robusto banco de dados open source, por ser utilizado no prontuário eletrônico do hospital, sistema este que o cliente pretende integrar com o F.A.D.D.V.M. em um futuro próximo e por atender as licenças que pretendemos utilizar.

4 REALIZAÇÃO DOS CASOS DE USO

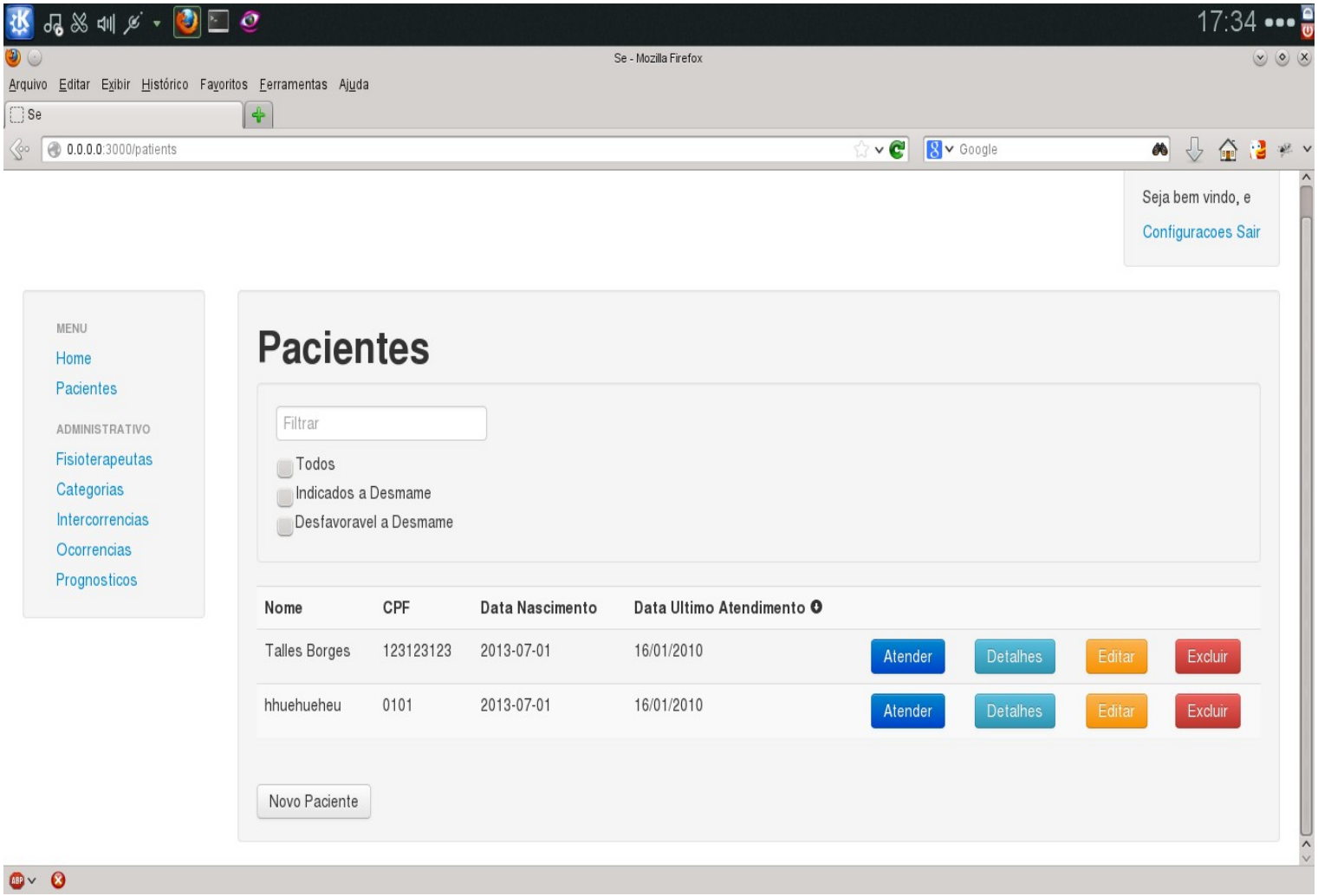
4.1 CASO DE USO 02 – MANTER HISTÓRICO DO PACIENTE

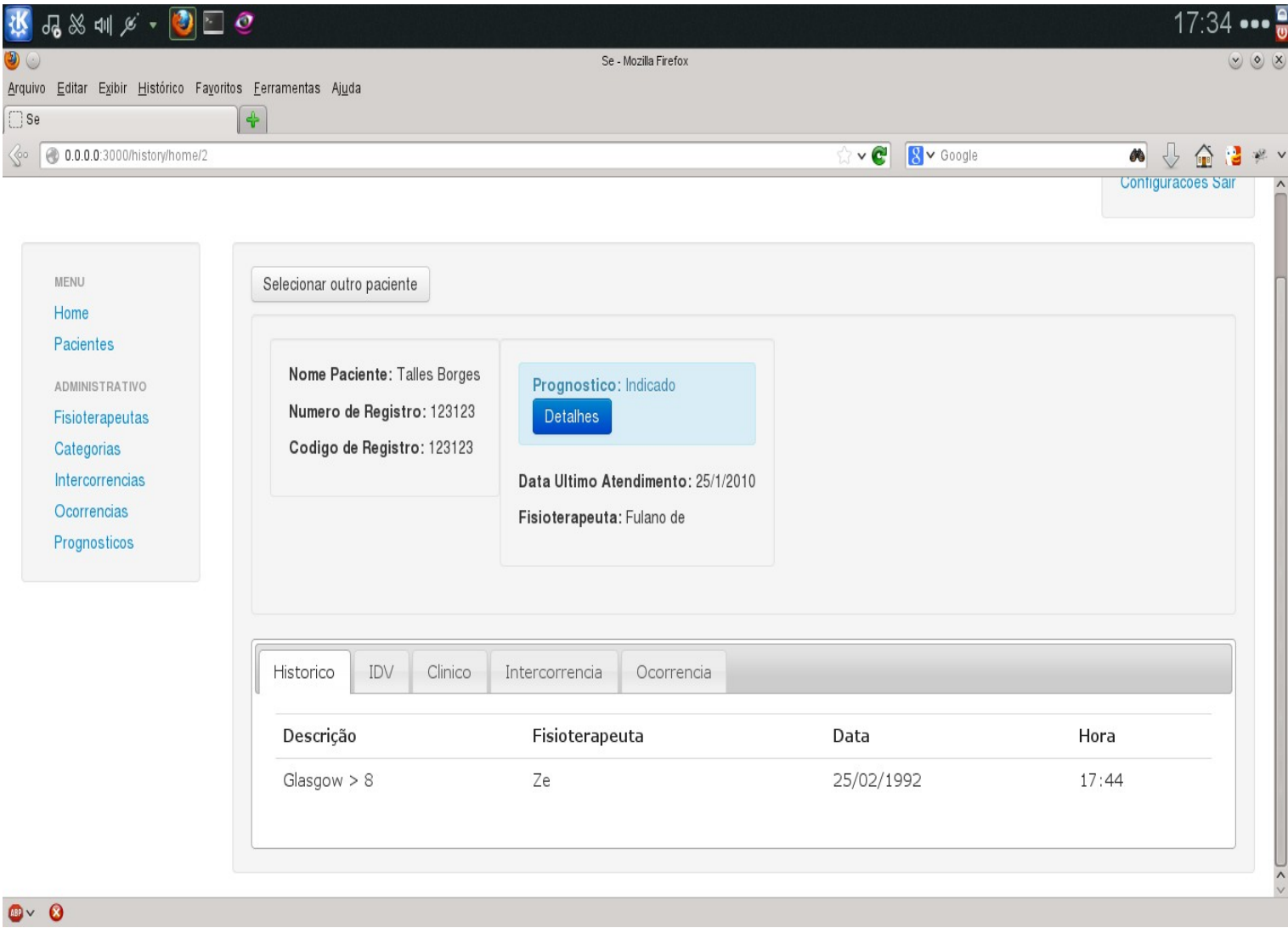
Este caso de uso tem por objetivo, realizar as interações com o histórico do paciente, cadastros de novas entradas no histórico e consultas de valores, assim como alterações ou remoções em caso de entradas incorretas.

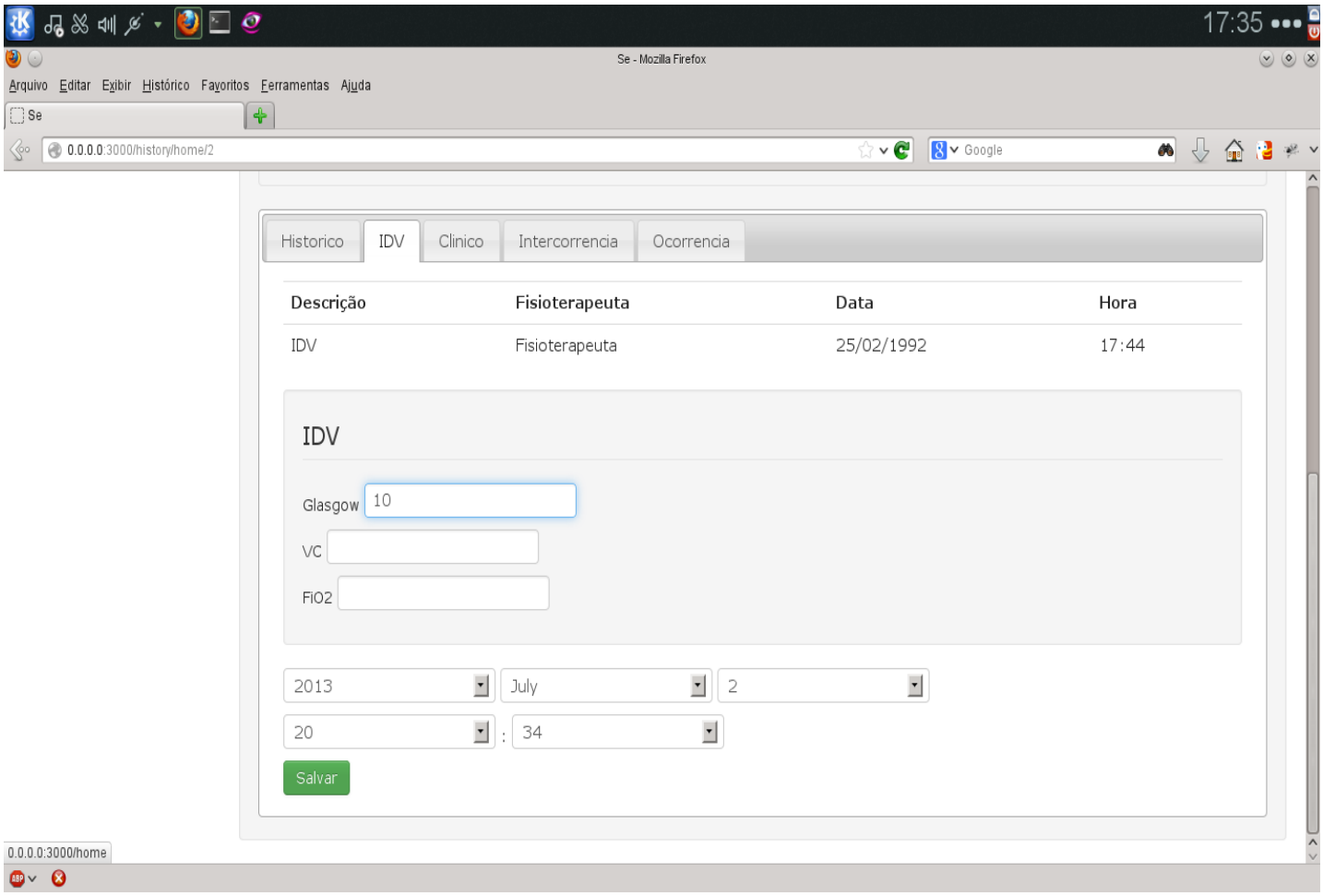
4.1.1 Funcionamento geral



4.1.2 Especificação das interfaces visuais




17:34
Se - Mozilla Firefox
Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda
Se
0.0.0.0:3000/history/home/2
Google
Configuracoes Sair
MENU
Home
Pacientes
ADMINISTRATIVO
Fisioterapeutas
Categorias
Intercorrencias
Ocorrencias
Prognosticos
Selecionar outro paciente
Nome Paciente: Talles Borges
Numero de Registro: 123123
Codigo de Registro: 123123
Prognostico: Indicado
Detalhes
Data Ultimo Atendimento: 25/1/2010
Fisioterapeuta: Fulano de
Historico
IDV
Clinico
Intercorrencia
Ocorrencia
Descrição
Fisioterapeuta
Data
Hora
Glasgow > 8
Ze
25/02/1992
17:44



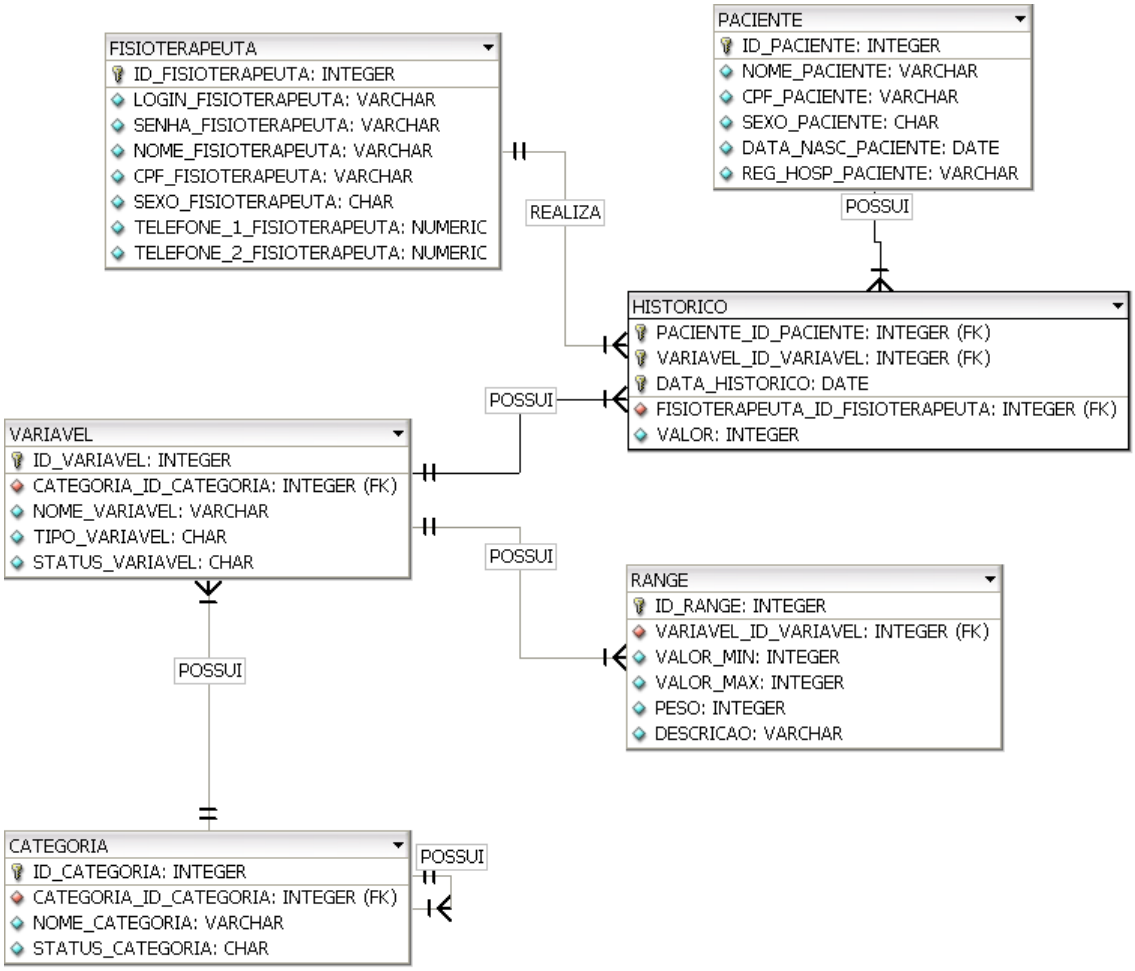
4.1.3 Especificação dos serviços

Descrição	Nome do Serviço	Entrada	Saída	Pré-cond	Pós-cond
Incluir nova entrada no histórico do paciente	incluirHistorico	Variáveis, fisioterapeuta, data, paciente	Histórico	Paciente cadastrado	Entrada de histórico

Mostrar histórico do paciente	mostrarHistorico	Paciente	Históricos do paciente	Paciente cadastrado	Histórico apresentado

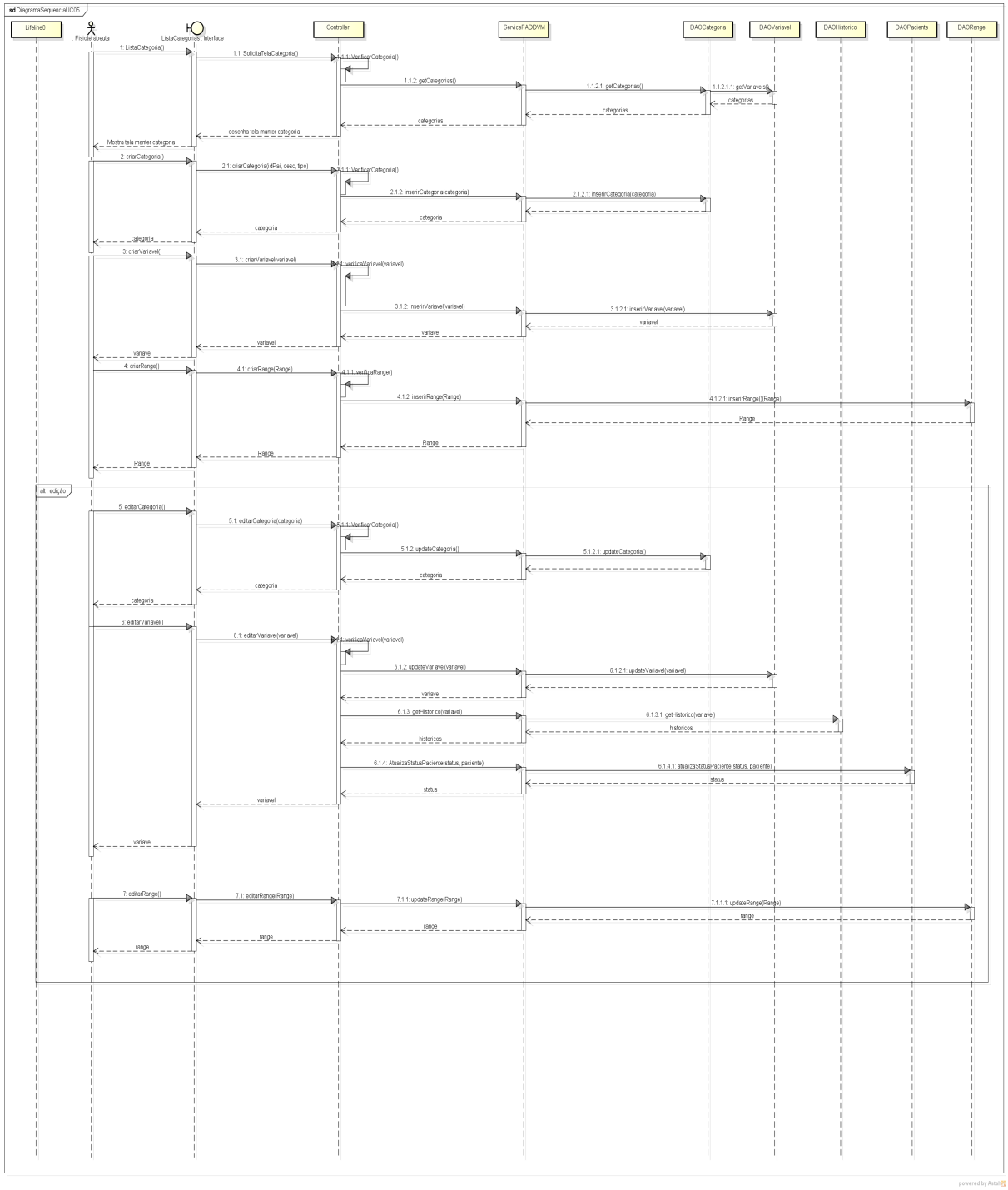
4.1.4 Especificação da camada de persistência

Na persistência de histórico foram utilizadas todas as tabelas listadas na imagem abaixo.

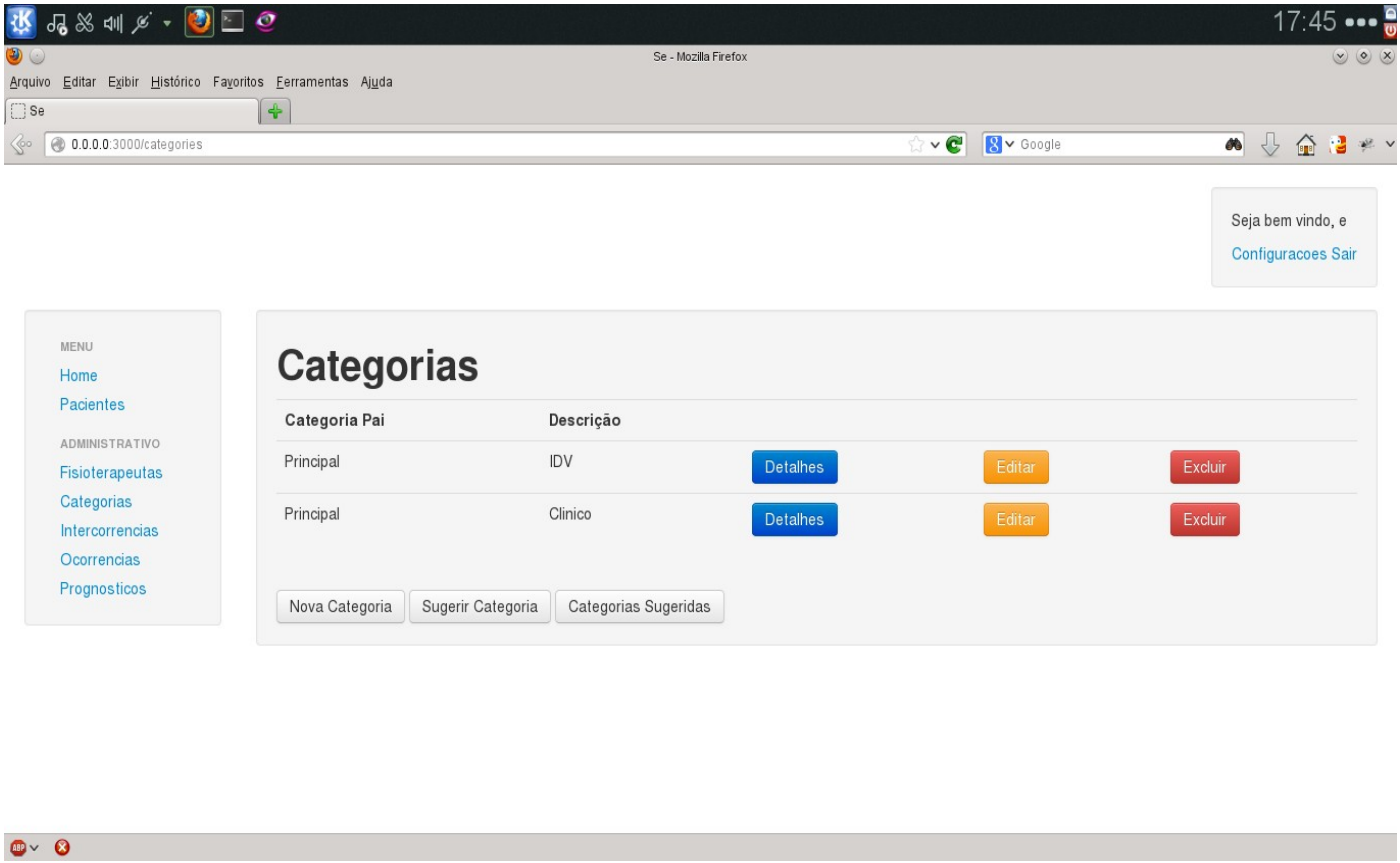


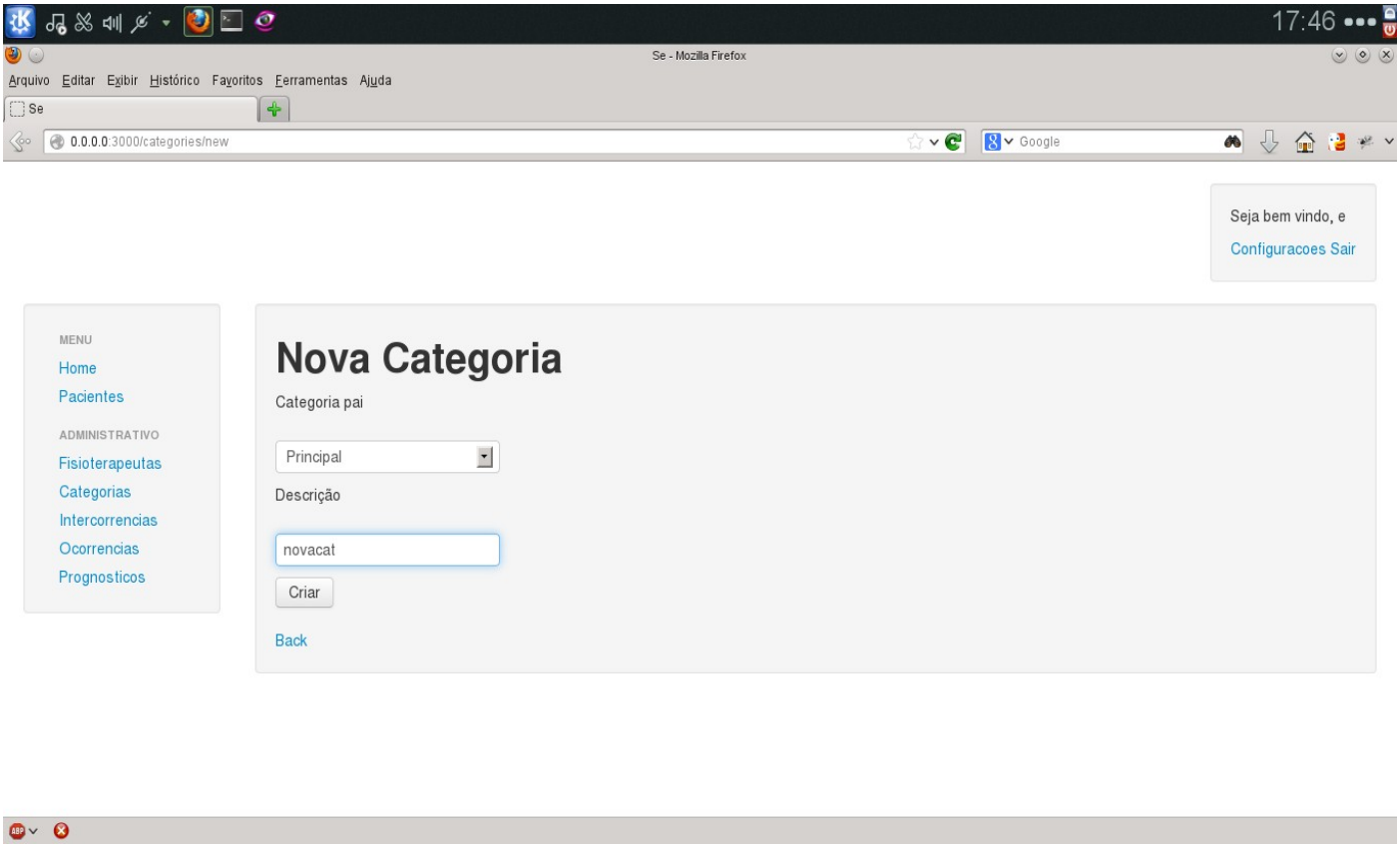
4.2 CASO DE USO 05 – MANTER CATEGORIA

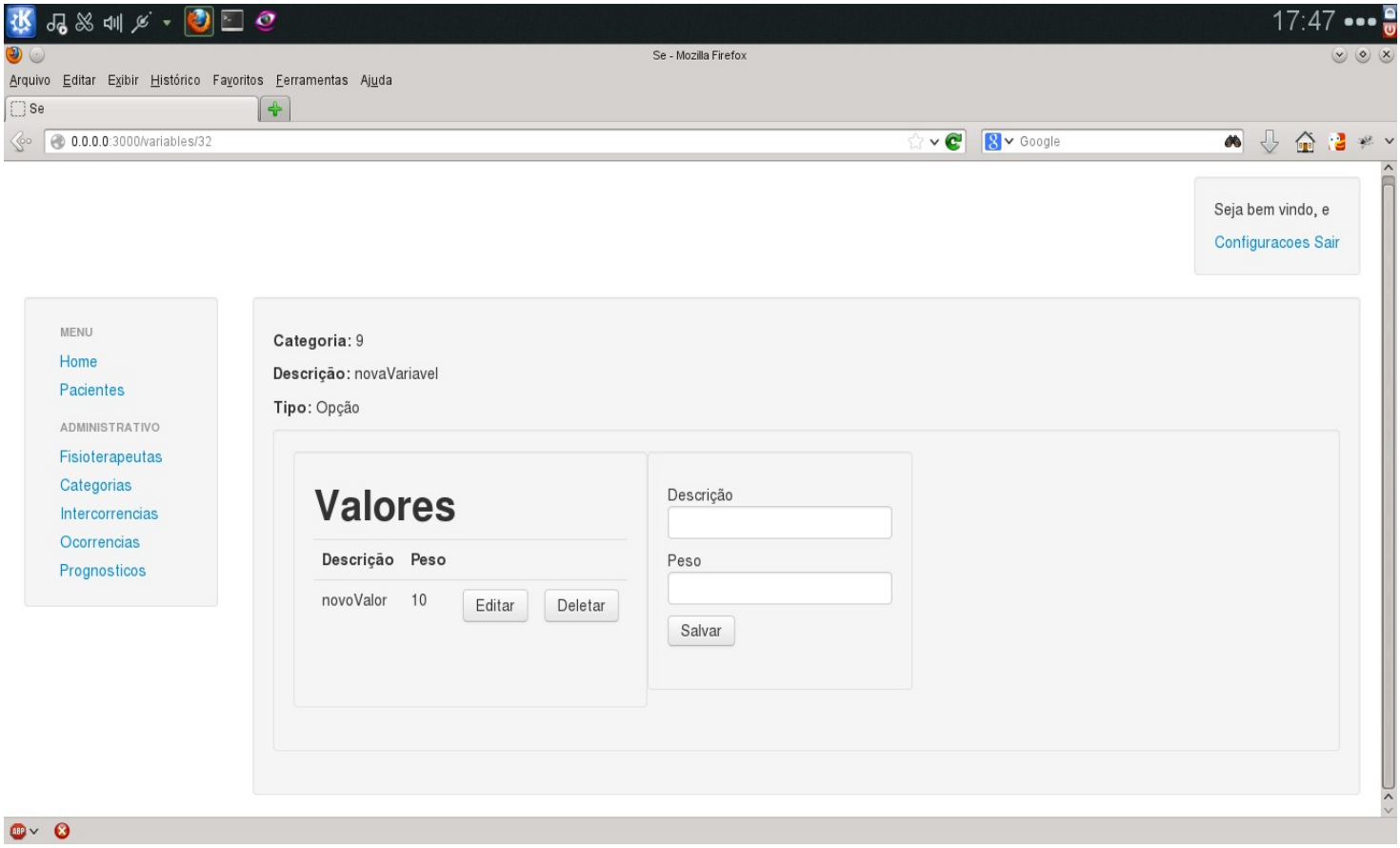
4.2.1 Funcionamento Geral



4.2.2 Especificação das interfaces visuais







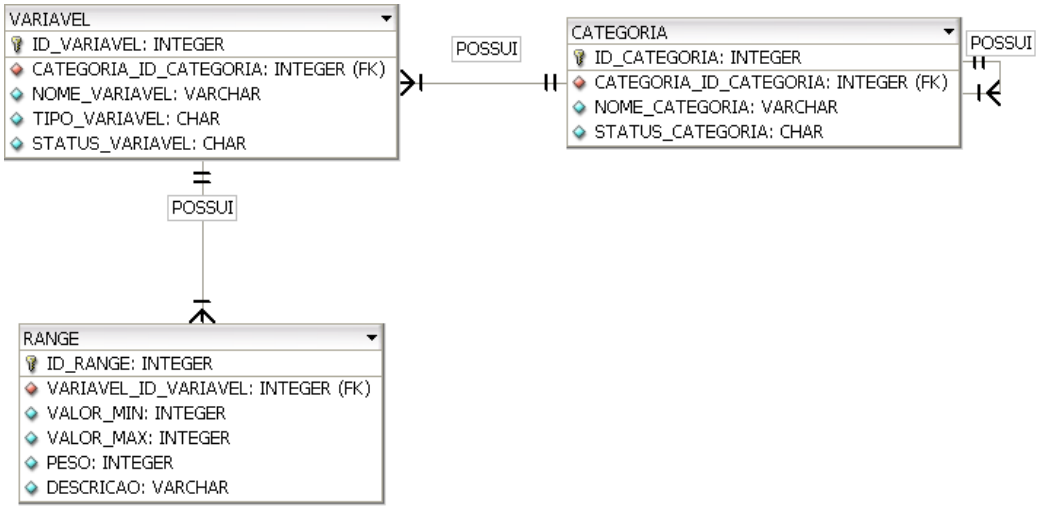
4.2.3 Especificações dos serviços

Descrição	Nome do Serviço	Entrada	Saída	Pré-cond	Pós-cond
Criar categoria nova	criarCategoria	Categoria pai, descrição, tipo,	Categoria	Fisioterapeuta precisa ter perfil de adm	Categoria criada
Editar categoria	editarCategoria	Categoria	Categoria	Fisioterapeuta precisa ter perfil de adm,	Categoria alterada
Excluir categoria	excluirCategoria	Categoria	Mensagem de sucesso	Fisioterapeuta precisa ter perfil de	Categoria excluída

				adm, categoria não pode ter variáveis	
Criar variável	criarVariavel	Variavel	Variavel	Fisioterapeuta precisa ter perfil de adm, tipo da categoria pai deve ser grupo	Variavel cadastrada
Editar variavel	editarVariavel	Variavel	Variavel	Fisioterapeuta precisa ter perfil de adm	Variavel alterada, status dos pacientes alterado
Excluir variavel	excluirVariavel	Variavel	Mensagem de sucesso	Fisioterapeuta precisa ter perfil de adm, variavel não pode ter sido usada em histórico	Variavel excluída

4.2.4 Expecificacao da camada de persistência

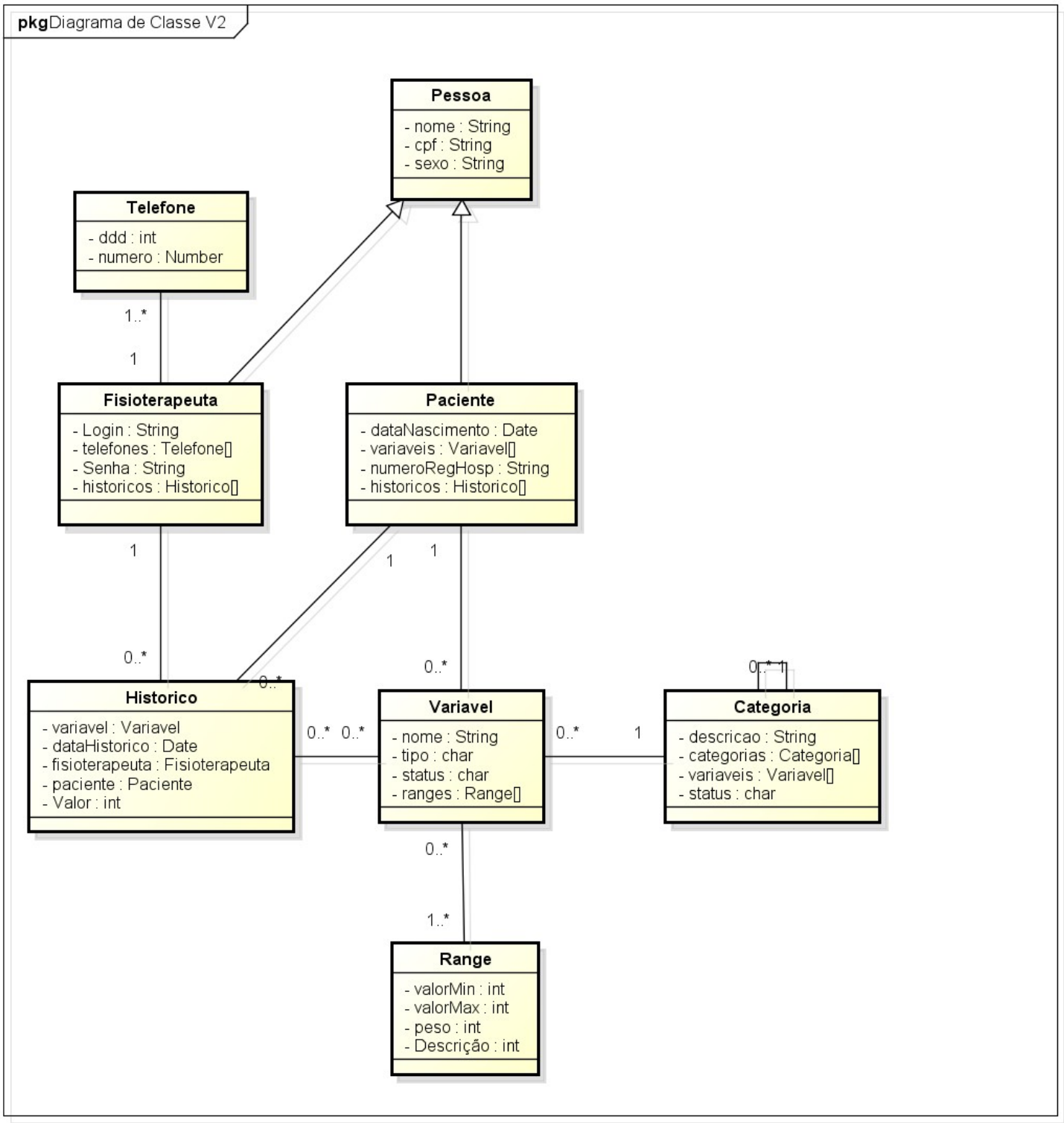
Na camada de persistência de categoria utilizaremos as tabelas mostradas na imagem abaixo.



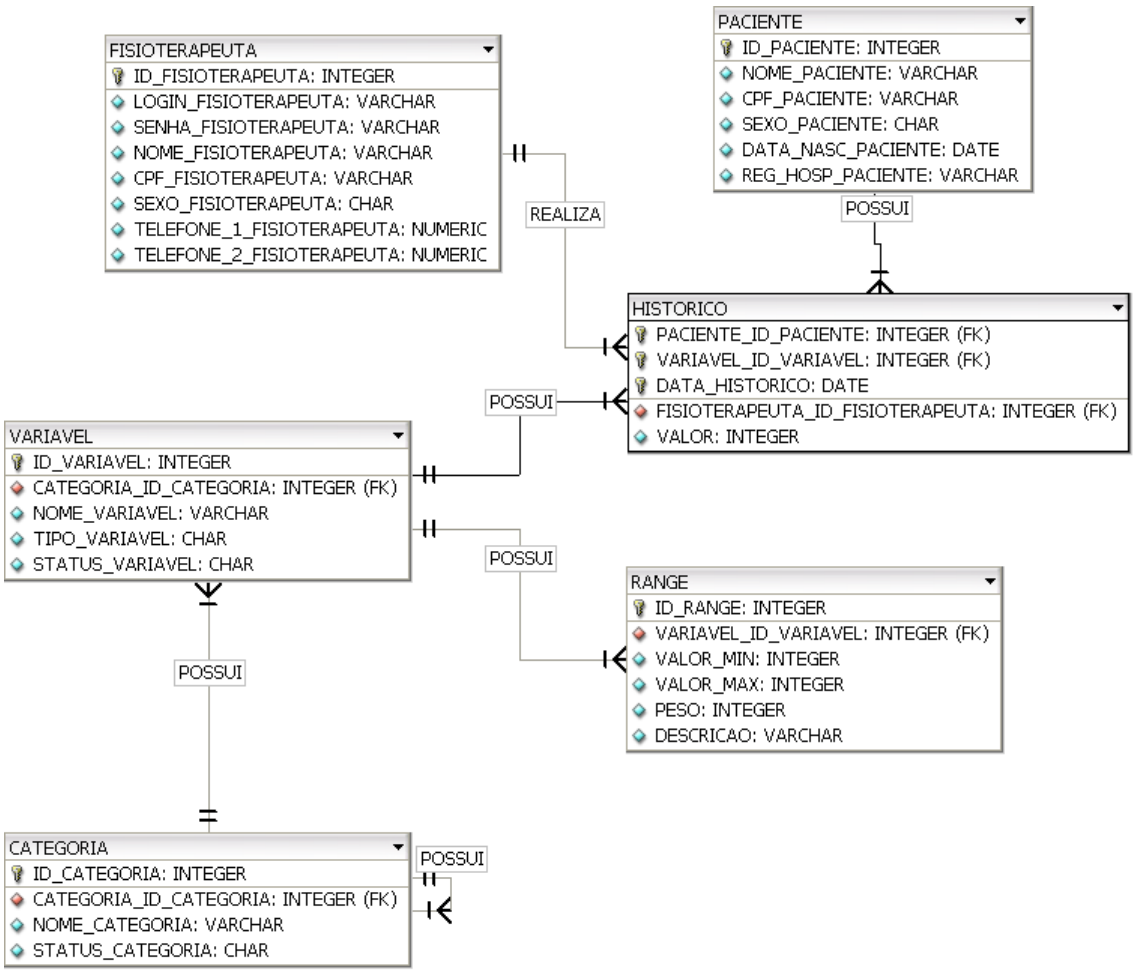
5 COMPONENTES COMUNS

- Formulário para cadastro de variáveis
- Master Page
- Componentes CSS do Bootstrap

6 DIAGRAMA DE CLASSES GERAL DA APLICAÇÃO



7 MODELO FÍSICO DE DADOS



8 PROTÓTIPO DAS INTERFACES

Sera apresentado em sala.

9 CONSIDERAÇÕES FINAIS

Neste documento conseguimos descrever como imaginamos o funcionamento do sistema, decidimos quais linguagens utilizaremos, assim como frameworks, serviços e bibliotecas, descrevemos detalhadamente dois dos casos de uso utilizando de diagramas de sequencia e implementação para ilustrar o funcionamento planejado, o fato de decidirmos realizar o protótipo em uma

linguagem de programação permitiu maior base sobre os possíveis desafios que iremos enfrentar na fase de desenvolvimento, assim como nos disponibilizou trechos de código que poderemos reutilizar.

Estamos confiantes no avanço do sistema e de sua entrega no prazo com 100% das funcionalidades prometidas.

REFERÊNCIAS

PERRONE, Paul J.; Chaganti, Krishna. **J2EE Developer's Handbook**. Indianapolis, Indiana: Sam's Publishing, 2003.

BODOFF, Stephanie; **The J2EE Tutorial**. Boston: Addison-Wesley, 2004.

KUMARASWAMIPILLAI, Arulkumaran; **Java/J2EE Job Interview Companion**. [S.l.: s.n.], 2007.

"Market Share". Why MySQL?. Oracle. Retrieved 17 September 2012.

"DB-Engines Ranking". Retrieved 26 February 2013.

SCHUMACHER, Robin; Lentz, Arjen. "Dispelling the Myths".**MySQL AB**. Archived fromthe original on 6 June 2011. Retrieved 17 September 2012.