

UNIVERSIDADE FEDERAL FLUMINENSE
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

MONOGRAFIA DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PROTÓTIPO DE UM SISTEMA DE RECEITUÁRIO *ONLINE*

**JOÃO PAULO LETHIER
MAGNO MATHIAS**

**NITERÓI
2012**

JOÃO PAULO LETHIER
MAGNO MATHIAS

PROTÓTIPO DE UM SISTEMA DE RECEITUÁRIO *ONLINE*

Monografia apresentada no Curso de
Ciência da Computação da Universidade
Federal Fluminense como parte dos
requisitos para obtenção do título de
Bacharel em Ciência da Computação.

Orientador: Prof. Dr. JOSE RAPHAEL BOKEHI

Niterói
2012

JOÃO PAULO LETHIER
MAGNO MATHIAS

PROTÓTIPO DE UM SISTEMA DE RECEITUÁRIO *ONLINE*

Monografia apresentada no Curso de
Ciência da Computação da Universidade
Federal Fluminense como parte dos
requisitos para obtenção do título de
Bacharel em Ciência da Computação.

Aprovado em janeiro de 2012.

BANCA EXAMINADORA

Prof. Dr. JOSE RAPHAEL BOKEHI – Orientador
UFF

Prof.^a Dr. CRISTINA NADER VASCONSELOS
UFF

Prof.^a Dr. SELMA RODRIGUES DE CASTILHO
UFF

Niterói
2012

AGRADECIMENTOS

Gostaríamos de agradecer:

Às nossas famílias, que nos apoiaram e incentivaram durante todo o período da graduação, principalmente nossos pais.

A todos os professores da Universidade Federal Fluminense com os quais tivemos aulas e contribuíram para a nossa formação.

Às nossas namoradas, que nos ajudaram na monografia, incentivando e revisando o texto.

À professora e farmacêutica Selma Rodrigues de Castilho e ao médico Igor Braz, que foram de grande ajuda no levantamento e análise dos requisitos para esse trabalho.

Ao professor e orientador dessa monografia, José Raphael Bokehi, por ter acreditado no nosso potencial e nos ajudado no desenvolvimento desse trabalho.

E aos nossos amigos de faculdade, que nos acompanharam no dia a dia de faculdade, fizeram trabalhos conosco, nos ajudaram a estudar, e a tornar o período da faculdade algo único e memorável.

SUMÁRIO

RESUMO	8
ABSTRACT	9
CAPÍTULO 1 – INTRODUÇÃO	10
1.1. MOTIVAÇÃO	10
1.2. OBJETIVO	13
CAPÍTULO 2 - ANÁLISE DE REQUISITOS	14
CAPÍTULO 3 – MODELAGEM	18
3.1. MODELO	18
3.2. CLASSE	19
CAPÍTULO 4 – IMPLEMENTAÇÃO	20
4.1 METODOLOGIA.....	20
4.2 FERRAMENTAS UTILIZADAS	20
4.3 TESTES	25
CAPÍTULO 5 – RECEITUÁRIO ONLINE.....	27
5.1. ACESSO AO SISTEMA	27
5.2. CADASTRO DE FARMÁCIA	27
5.3. CADASTRO DE CLÍNICA/ HOSPITAL/ CONSULTÓRIO	28
5.4. FARMÁCIA	29
5.6. CLÍNICA	30
5.7. MÉDICO	30
5.8. PACIENTE	31
5.9. FARMACÊUTICO/BALCONISTA	31
5.10. TESTES	31
CAPÍTULO 6 - DISCUSSÃO E CONCLUSÃO.....	35
6.1. TRABALHOS FUTUROS.....	36
CAPÍTULO 7 - REFERÊNCIAS BIBLIOGRÁFICAS	37

LISTA DE FIGURAS

Figura 1 - Diagrama de casos de uso	17
Figura 2 - Diagrama de classes.....	19
Figura 3 – Pesquisa sobre utilização de banco de dados feita pela Gartner [16].....	21
Figura 4 – Exemplos de comandos do Rails	24
Figura 5 – Instalação de uma biblioteca do Rails	24
Figura 6 - Exemplo de utilização retirado da documentação do RSpec [23]	26
Figura 7 - Página de login do sistema	27
Figura 8 - Cadastro de farmácia.....	28
Figura 9 - Cadastro de clínicas.....	29
Figura 10 – Código de alguns testes feitos no protótipo	32
Figura 11 - Testes do protótipo passando.....	32
Figura 12 - Código de alguns testes de interface feitos no protótipo.....	33
Figura 13 - Exemplo de alguns testes de interface rodando	34
Figura 14 - Número de cenários testados	34

LISTA DE TABELAS

Tabela 1- Lista de requisitos funcionais.....	16
Tabela 2- Ranking das linguagens de programação [19]	23

RESUMO

No Brasil, para que um paciente possa comprar um medicamento que necessita de prescrição médica, ainda é necessária a geração de uma receita impressa para que o paciente possa levar à farmácia e efetuar a compra do medicamento. A informatização desse processo, que pode trazer vantagens para médicos, farmacêuticos e pacientes, motivou o desenvolvimento de um protótipo de um sistema que faça o controle de receitas de forma *online*. Esse protótipo permite ao médico cadastrar a receita do paciente no sistema e à farmácia verificar a veracidade da mesma quando o paciente for comprar o medicamento. O desenvolvimento do protótipo foi feito com a utilização do framework de desenvolvimento de sistemas voltados para plataformas *web*, chamado *RubyOnRails*, e ferramentas de testes automatizados, como o *Cucumber* e o *RSpec*.

Palavras chave: *receituário, protótipo, RubyOnRails, Cucumber, RSpec*

ABSTRACT

In Brazil, drugs which requires a prescription can only be purchased with a printed copy of it. The benefits that the informatization of this process could bring to doctors, pharmacists and patients, motivated the development of a system's prototype that manages an online prescription environment. In this prototype doctors create prescriptions, pharmacists confirm the existence of it, and patients have access to his prescriptions's history. Everything in an online environment without any printed piece of paper. The tools and frameworks used in the development of this prototype were RubyOnRails, RSpec and Cucumber.

Keywords: medical prescription, prototype, RubyOnRails, Cucumber, RSpec

CAPÍTULO 1 – INTRODUÇÃO

Para a evolução de uma nação, o desenvolvimento de políticas públicas tem um papel fundamental [1]. Dentre as atualmente implementadas no Brasil, as que notoriamente se destacam são as de Educação, Segurança e Saúde por possuírem um papel fundamental para o crescimento da vida em sociedade.

Em relação a outros países, existem inúmeras diferenças referentes ao controle e monitoramento feito pelo governo nessas áreas, isso pode ser observado através dos resultados referentes aos indicadores sociais de dois países, um deles sendo a Suécia e o outro sendo o Brasil [2].

A Suécia, um exemplo de país desenvolvido, de acordo com a *Wikipédia* [3], em 2010 detinha o quarto maior Índice de Desenvolvimento Humano (IDH) do planeta, assim como a esperança de vida em torno de 80,9 anos, a mortalidade infantil de 3,2 mortes a cada mil nascimentos e a alfabetização de 99% da população. Esses índices mostram que a Suécia é realmente um país de altíssimo nível socioeconômico. Isso mostra que nesse País, os serviços prestados, sejam eles públicos ou privados, para a população e as políticas de controle executadas pelo governo são satisfatórias.

Em contrapartida, de acordo com a *Wikipédia* [4], em 2011, o IDH brasileiro detinha o octogésimo quarto lugar no *ranking* mundial, com uma esperança de vida em torno de 73,5 anos, mortalidade infantil de 19,3 mortes a cada mil nascimentos e uma alfabetização de 90,4% da população, mostrando o quanto o país ainda carece de recursos para elevar seus indicadores sociais.

Todos esses indicadores mostram a importância de se ter um controle eficiente sobre as políticas públicas de gestão governamental. Dentre todas elas, a de saúde será destacada, sendo essa o foco da abordagem deste projeto.

1.1.MOTIVAÇÃO

No Brasil, pode-se considerar que há um baixo controle sobre as prescrições médicas que circulam no território nacional. Em países como a Suécia, por exemplo, há um sistema integrado entre médicos e farmácias para prescrição de receitas

médicas e vendas de medicamentos [5] [6].

O processo de atendimento médico e compra de medicamentos por um paciente continua o mesmo de décadas atrás, com algumas alterações com relação aos medicamentos controlados. Atualmente este processo pode ser descrito como se segue:

1. O paciente vai ao médico;
2. O médico atende o paciente e entrega a ele uma receita preparada manualmente;
3. O paciente vai à farmácia e compra o medicamento.

Neste processo existem diversos problemas no controle e acesso às informações que poderiam servir em um futuro atendimento médico e/ou em uma próxima compra de medicamentos em uma farmácia. Isto se afirma na medida em que todas as informações desse processo ficam registradas ou na farmácia através da retenção das receitas, ou na clínica para controle dos medicamentos receitados pelos médicos ou simplesmente no papel da receita médica, quando após o paciente sair de uma consulta, em uma clínica que não possui nenhum controle sobre as receitas, não efetuar a compra do medicamento.

O sistema implantado na Suécia, conforme anteriormente citado, faz com que esse mesmo processo seja executado de maneira bem diferente, solucionando naquele país alguns problemas que são encontrados no Brasil.

Nesse sentido, o acesso ao sistema desenvolvido na Suécia é disponível aos médicos, que podem consultá-lo para ver históricos de pacientes. Essa consulta mostra-se útil no atendimento do médico, pois permite o cadastro de receitas médicas no sistema, de forma que um funcionário de farmácia também possa atender esse paciente através do acesso a essa mesma base de dados.

Essa abordagem faz com que as informações referentes ao processo citado não sejam perdidas, além de facilitar e tornar mais rápida a interação e comunicação entre os agentes desse processo.

Um ponto crítico nesse processo é a segurança, tanto do lado da verificação da veracidade e validade do documento pela farmácia, como na garantia de que os medicamentos só sejam vendidos para quem tem realmente permissão para

comprá-los. A verificação dessas garantias se mostra impossível no processo atual, pois é necessária a verificação da assinatura do médico, junto com a conferência de seu número de cadastro no Conselho Regional de Medicina (CRM) pela farmácia. Dessa maneira uma farmácia deveria conhecer todas as assinaturas e números de cadastro no CRM existentes e ativos para viabilizar o controle da autenticidade de uma prescrição médica.

Um problema de importância crescente é a necessidade de se entregar uma receita médica para efetuar a compra do medicamento. Mesmo que hoje seja possível efetuar a compra sem sair de casa, através do serviço de entrega, ainda possui a necessidade de entregar a receita ao entregador.

Com a evolução da internet, os consumidores vêm direcionando todas suas compras, como televisões, livros e DVDs, para os comércios eletrônicos ou *e-commerces*. Este fato se deve ao grande crescimento da necessidade de se ter conforto e comodidade por parte das pessoas[7]. Com a fabricação de uma ferramenta, que pudesse liberar todas as informações presentes na receita para as farmácias, isto se tornaria viável. Entretanto, diante do cenário atual, tal processo de compra não se mostra viável.

Outro problema do processo que envolve desde a emissão da prescrição médica até a compra do medicamento, é referente à legibilidade das receitas médicas. Em um levantamento de 2006, feito pela *Institute of Medicine of The National Academies* (IOM), nos Estados Unidos, foi mostrado que o fato da ilegibilidade de uma receita mata mais de 7000 pessoas e causa danos em mais de 1,5 milhões de pacientes por ano [8]. Para diminuir esse problema no Brasil, o Conselho Federal de Medicina (CFM) publicou uma resolução que determina que as receitas médicas devem ser necessariamente escritas por extenso e de forma legível. Assim como essa resolução, existe a lei 5991 de 1973 que possui um poder maior que essa resolução. Após essas medidas, diversos médicos têm sido multados pelo Conselho [9].

A falta de um histórico do paciente - tanto um histórico de doenças, como de medicamentos receitados - é mais um problema que persiste no Brasil. Esse serviço quando existe, fica distribuído separadamente entre as clínicas, hospitais e consultórios. Caso o paciente seja atendido em um segundo local é necessária a construção de um novo histórico, tanto .

Um problema que persiste no Brasil é a falta de um histórico para o controle de vendas de medicamentos e de prescrições médicas receitadas, este seria responsável pelo controle dos medicamentos vendidos por cada farmácia e dos receitados por cada médico. Para solucionar os casos citados, mostra-se necessária a construção de um protótipo onde seja possível guardar todas as informações pertinentes ao processo de fabricação das receitas, até a sua utilização nas farmácias.

Todos esses fatores apresentados justificam a geração do produto desenvolvido neste trabalho monográfico.

1.2.OBJETIVO

Este trabalho tem como objetivo desenvolver um protótipo de um sistema que integre todos os passos do processo de emissão de receitas até a venda de medicamentos e facilite a comunicação entre todos os seus agentes, como médicos, clínicas, farmácias e pacientes.

CAPÍTULO 2 - ANÁLISE DE REQUISITOS

A análise de requisitos é uma das etapas iniciais da engenharia de requisitos, que é composta por análise, documentação, validação e gerenciamento de requisitos de software. Esta fase tem como propósito a compreensão e o entendimento das necessidades e condições necessárias para se desenvolver um software.

A necessidade de uma análise de requisitos se dá devido à grande importância desta para o desenvolvimento. Para efeito de demonstração, um software que não tenha passado pela análise de requisito, mesmo que seja muito bem desenvolvido e planejado, pode frustrar seu(s) cliente(s) [10].

A concretização dessa etapa tem como resultado final o escopo do software, diagramas e fluxos que servem como auxílio ao entendimento do problema. O escopo do software contém os requisitos para o seu funcionamento. Esses estabelecem o que o sistema deve fazer e definem restrições sobre sua operação e sua implementação [11].

Para que fosse realizada a definição do escopo final do sistema Receituário Online, foram feitas algumas entrevistas com profissionais na área de farmácia e medicina para saber quais requisitos o sistema deveria ter para que realmente suprisse todas as funções propostas.

O primeiro profissional pesquisado foi um médico que tem conhecimento de um sistema usado na Suécia que informatizou o processo do médico prescrever uma receita até a compra de medicamentos por parte dos pacientes.

Inicialmente foi mostrada ao médico a proposta do projeto. Logo depois o médico esclarece que existe um sistema bem parecido com o proposto na Suécia. Foram discutidos alguns outros detalhes do software, como a informação de que a Suécia tem um sistema de saúde totalmente integrado e em relação às farmácias, tem um sistema específico, chamado de Apotekens Service AB [5].

O Apotekens Service AB é responsável por, dentre outras funcionalidades, registrar uma receita e principalmente pela revisão e autorização dos sistemas informatizados utilizados pelas farmácias para a expedição de prescrições. Portanto, esse sistema já possui uma série de requisitos funcionais implementados e em funcionamento e atende uma grande massa de pessoas, no caso toda a população da Suécia.

Devido a isso, esta ferramenta desenvolvida na Suécia foi de suma importância para o levantamento dos requisitos, esclarecimento do problema e solução como um todo.

Por uma questão de preocupação por parte das legislações correntes no Brasil, foi necessário que fossem feitas algumas perguntas a uma professora do curso de Farmácia da UFF e farmacêutica. Através delas foi possível esclarecer pontos importantes relacionados aos processos presentes nas farmácias e os que deveriam estar presentes. Estes temas definem pontos importantes para o desenvolvimento do protótipo sem que tenha nada contra as leis federais, estaduais e municipais.

Com relação à validação ao cadastramento de médicos no sistema, a única opção seria ter acesso a uma listagem de CRM de cada estado.

Em se tratando da existência de um padrão a ser seguido ao prescrever uma receita médica. Foi dito que para a maioria dos medicamentos não existe um padrão, mas para os medicamentos sob controle especial, anteriormente chamados de controlados, existem três padrões [12].

Uma dúvida pertinente é se todos os funcionários da farmácia, dentre eles os farmacêuticos e os balconistas, podem atender qualquer receita médica. Como resposta, foi esclarecido que teoricamente os funcionários só não podem fazer a substituição de medicamentos genéricos e a venda de controlados, mas que isso não condiz com a atual realidade. Uma operação que apenas os farmacêuticos podem realizar é o registro de controlados no SNGPC (Sistema Nacional de Gerenciamento de Produtos Controlados).

A partir das duas entrevistas citadas, foram definidos os requisitos de sistema apresentados na tabela 1.

Tabela 1- Lista de requisitos funcionais

Nome	Descrição
Cadastro de farmácia	Uma farmácia para ter acesso ao sistema deve efetuar seu cadastro na tela de cadastro de farmácia, preenchendo todos os campos necessários corretamente e clicar em confirmar.
Login	Um usuário cadastrado, para efetuar o login no sistema, deve acessar a tela de login, preencher os campos de CPF/CNPJ e senha e clicar em confirmar, sendo assim autenticado no sistema e redirecionado para a página inicial.
Cadastro de Clínica	Para uma clínica ter acesso ao sistema, é necessário que a mesma faça seu cadastro através da tela de cadastro de clínicas preenchendo todos os campos e clicando em confirmar.
Cadastro de médico	Uma clínica cadastrada no sistema, e devidamente autenticada, pode cadastrar médicos que trabalhem nela para que os mesmos possam cadastrar receitas médicas para os pacientes no sistema.
Cadastro de funcionário da farmácia	Uma farmácia cadastrada no sistema, e devidamente autenticada, pode cadastrar um funcionário para que o mesmo passe a ter acesso ao sistema e possa atender pedidos de receitas prescritas por médicos.
Cadastro de receita médica	Ao final do atendimento a um paciente, o médico deve ser capaz de cadastrar no sistema uma receita médica com todos os remédios que o paciente deve tomar, assim como as recomendações de uso de cada remédio.
Cadastro de paciente	Um médico antes de cadastrar uma receita para um paciente que não tenha cadastro no sistema deve cadastrá-lo através do link 'Cadastro de Paciente', preencher o formulário de cadastro e clicar em confirmar.
Atendimento de receita médica	Um balconista/farmacêutico, ao efetuar a venda dos remédios prescritos na receita médica para um paciente, deve marcar a receita médica no sistema como atendida, sendo salva no banco a data desse atendimento, assim como o funcionário que fez essa venda e em que farmácia a receita foi atendida.

A lista de requisitos funcionais, representados pela tabela 1 e o diagrama de casos de uso representado pela figura 1, que é uma representação do ponto de vista do usuário [13], possuem objetivos semelhantes. Tanto um como o outro facilitam a comunicação entre os analistas e o cliente, mostrando de uma forma mais clara o que o software será capaz de fazer.

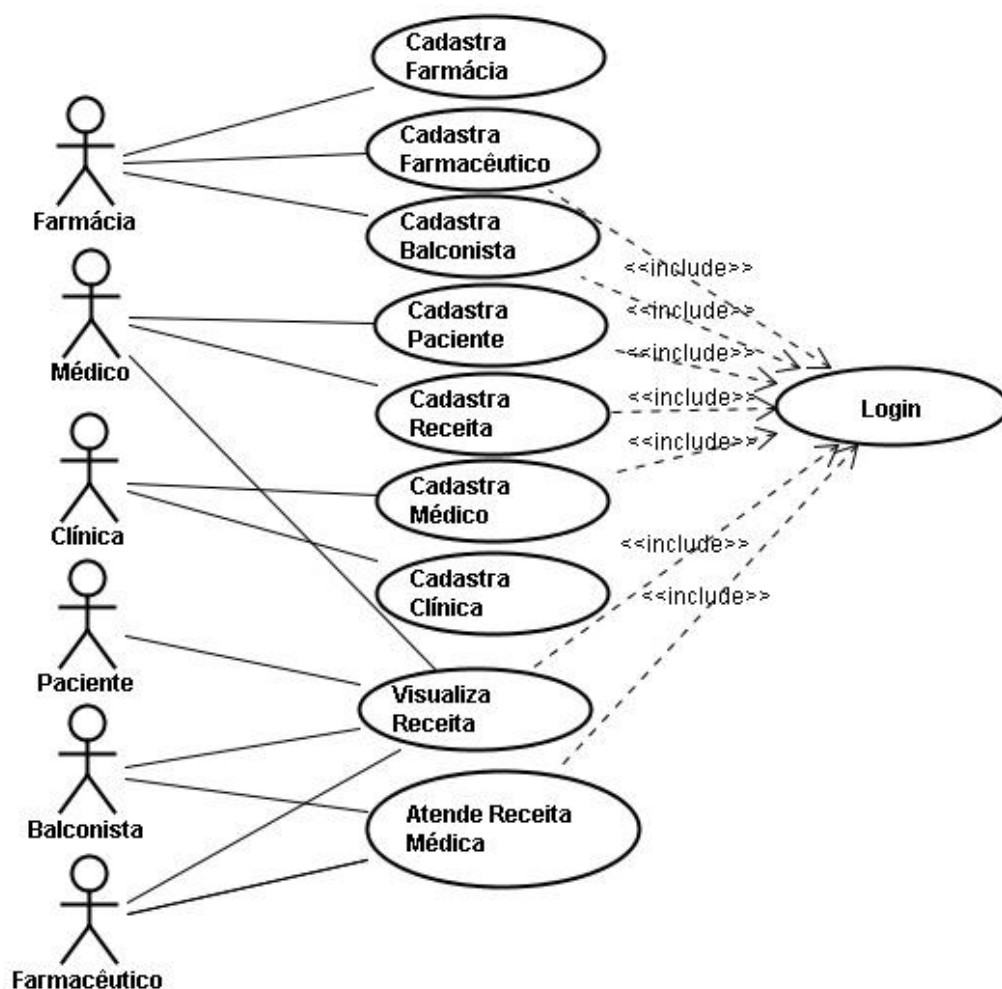


Figura 1 - Diagrama de casos de uso

CAPÍTULO 3 – MODELAGEM

Este capítulo aborda a modelagem do protótipo proposto diante deste projeto. A modelagem de dados é outro processo da engenharia de software, assim como a análise de requisitos, e tem como papel fundamental a criação de um modelo de dados para o protótipo em questão.

3.1.MODELO

O modelo desse protótipo foi pensado de forma que atendesse a todos os requisitos definidos a partir das entrevistas e reuniões feitas na análise de requisitos. Sendo assim, o modelo deve conter todas as classes e campos necessários para permitir o acesso por qualquer um dos agentes integrantes do processo de prescrição e atendimento de receitas.

Para isso, foram criadas as tabelas farmácia, farmacêutico, balconista, clínica, médico, paciente e receita. Além dessas tabelas, também foi modelada uma tabela chamada usuário, para centralizar e evitar repetição dos dados de cadastro e acesso ao sistema.

Para que seja efetuado o cadastro de um novo usuário, seja ele farmacêutico ou médico ou qualquer outro, é necessário que todos os campos sejam preenchidos, visto que, todos eles são obrigatórios. Os campos a serem preenchidos por obrigatoriedade são: nome, email, CPF ou CNPJ e senha.

As clínicas e as farmácias, além dos campos herdados de usuário, possuem também um campo de endereço, para que seja possível localizar as mesmas geograficamente.

Em se tratando da tabela de médicos, esta possui um campo para o número do CRM (Conselho Regional de Medicina) de cada médico, enquanto as tabelas de balconistas, farmacêuticos e pacientes não possuem nenhum campo além dos herdados da classe usuário.

O modelo de receita implementado foi o mais simples, levando em conta que a idéia é desenvolver somente um protótipo, com as informações sobre o paciente para o qual é a receita, que médico a receitou, em que clínica, e, após ser atendida, quando foi atendida, por que funcionário, farmacêutico ou balconista, e em que farmácia.

Os relacionamentos entre médicos e clínicas e farmacêuticos e balconistas com farmácia, foram definidos como muitos para muitos, ou seja, um médico pode trabalhar em mais de uma clínica, e a mesma pode ter mais de um médico, seguindo o mesmo pensamento para a farmácia e seus funcionários. Por isso nessas relações foram criadas tabelas intermediárias que ligam cada par dessas relações.

Além de tudo anteriormente citado, todas as tabelas possuem campos com a informação de quando os registros foram criados e quando foram atualizados pela última vez.

3.2. CLASSE

O diagrama de classes referente ao protótipo do sistema de receituário online é apresentado na figura 2. Como é possível visualizar, todas as classes, excetuando-se uma que é a Receita, herdam de usuário, pois todos eles terão acesso ao sistema, mas com papéis diferentes, ou seja, funcionalidades diferentes aparecerão para cada um dependendo de seu papel.

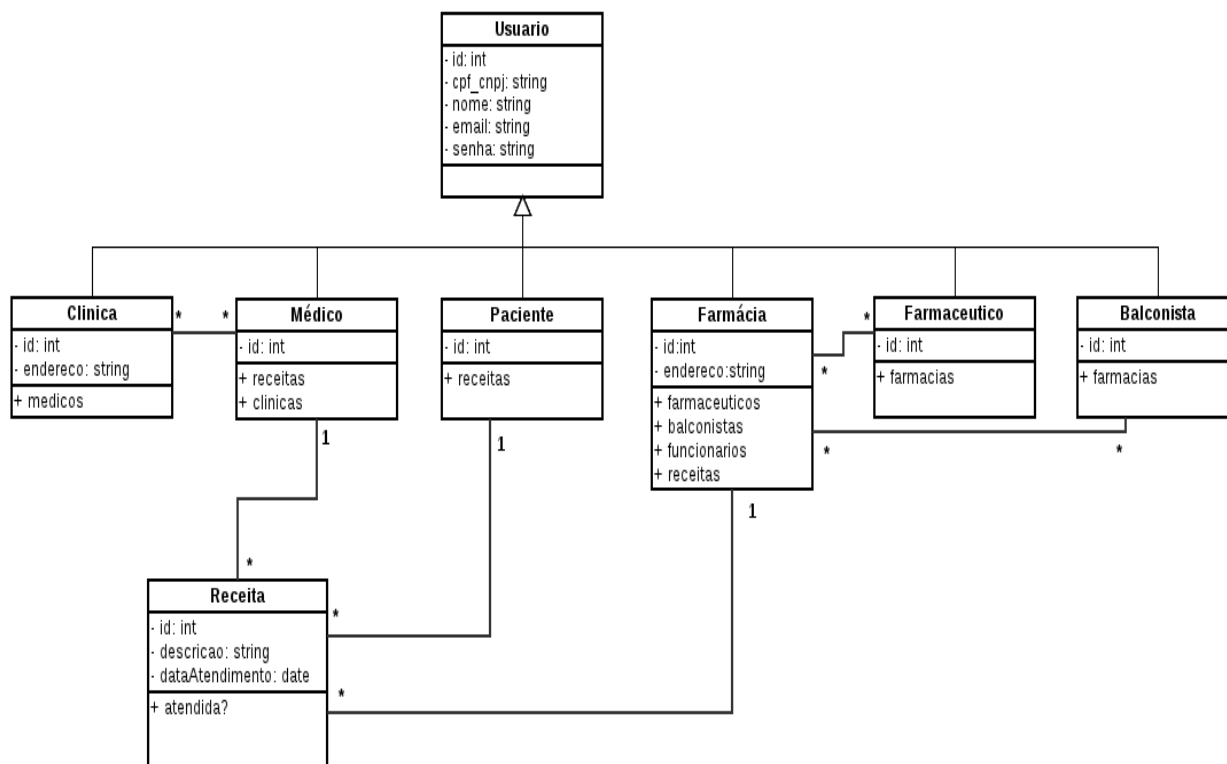


Figura 2 - Diagrama de classes

CAPÍTULO 4 – IMPLEMENTAÇÃO

Neste capítulo será abordado o processo de implementação presente na engenharia de software. Este processo se baseia em tornar o que foi planejado em algo concreto, como um sistema ou um protótipo, através da programação.

Como a programação é uma das etapas mais complexas, devido à grande exigência de experiência, metodologias, técnicas, ferramentas para ser efetuada e testes, foi necessária a divisão desta seção em subseções.

4.1 METODOLOGIA

Hoje em dia existe uma série de metodologias que ajudam a planejar e de fato executar o projeto, dentre elas existem o *Agile*, *Waterfall*, *Lean*, SCRUM, XP (*Xtreme Programming*) e RUP (*Rational Unified Process*). Nesse trabalho foi utilizada a metodologia XP, com algumas pequenas modificações [14].

Um dos métodos para desenvolvimento pertencente à metodologia XP utilizado foi o *Pair Programming*, onde o desenvolvimento é feito por duas pessoas, sendo uma o piloto, que controla o computador, e outra o co-piloto, que fica ao lado do piloto observando e ajudando no desenvolvimento, durante um intervalo de tempo previamente definido, que no caso deste projeto foram cinco minutos. Após este tempo, o piloto e o co-piloto trocam de função [14].

Para que a garantia de qualidade não fosse comprometida, foi utilizada durante o desenvolvimento, o TDD (*Test Driven Development*), ou seja, um desenvolvimento orientado a testes, onde o próprio teste é escrito antes de ser feita a funcionalidade. Essa prática garante que todo código escrito está sendo testado, dado que o código só é escrito após o teste dele ser feito, garantindo assim uma cobertura de testes muito grande [14].

4.2 FERRAMENTAS UTILIZADAS

4.2.1 MySQL

O MySQL é um Sistema Gerenciador de Banco de Dados (SGBD) que utiliza uma linguagem chamada SQL (*Structured Query Language*) como interface.

O MySQL é o sistema gerenciador de banco de dados *open source* mais popular do mundo, chegando a uma marca de mais de 65 mil *downloads* por dia e

de acordo com o site da empresa Forrester, citado no próprio site do MySQL, tem o maior crescimento e adoção e continua tendo o maior pedaço do mercado (*marketshare*) em *open source* na área de bancos de dados e possui o maior número de clientes pagantes pelo suporte do produto, estimando-se em torno de 16 mil [15].

De acordo com a pesquisa da Gartner, ilustrada pela figura 3, feita em 2008, com os índices de instalações e planos de implantações do banco de dados, o MySQL segue como líder *open source* [16].

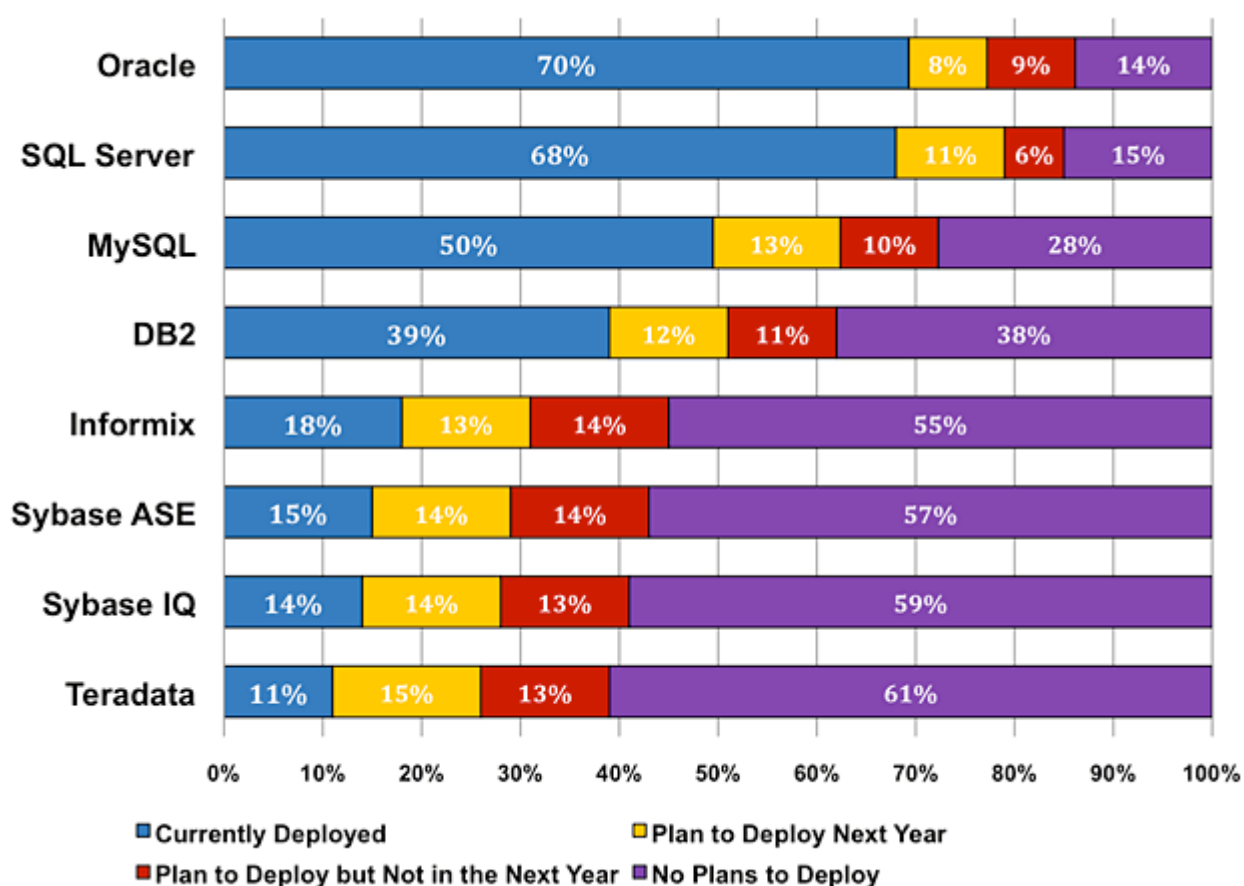


Figura 3 – Pesquisa sobre utilização de banco de dados feita pela Gartner [16]

[Fonte: Gartner Group, 2008]

O crescimento dessa ferramenta, que leva a uma utilização cada vez maior da mesma, se deve a uma série de características, listadas abaixo:

- ❖ Grande facilidade de integração entre os sistemas
- ❖ Compatibilidade com uma série de linguagens, como Java, Perl, ASP e

PHP

- ❖ Possui pouca exigência quanto a *hardware*
- ❖ É um *software* livre com base na GPL
- ❖ Tem um excelente desempenho e estabilidade.

Existe uma quantidade enorme de usuários de grande porte, com dados que exigem um nível alto de segurança e confiabilidade como NASA, Banco Bradesco, Nokia, Dataprev, Sony e etc.

Por esses motivos, ele foi o escolhido para o desenvolvimento do projeto, por ser fácil de usar, ser de graça e ainda por cima garante estabilidade e confiabilidade.

4.2.2 RUBY

Ruby é uma linguagem de programação interessante, que foi inventada por um Japonês chamado Yukihiro Matsumoto, mais conhecido como “Matz”. Ele simplesmente queria juntar todas as linguagens de que gostava em uma só. “Matz” diz sempre que está tentando tornar o Ruby natural, mas não simples [17].

Ele citou a seguinte frase:

“O Ruby é simples na aparência, mas é muito complexo por dentro, tal como o corpo humano” [17]

O Ruby vem sendo alavancado por uma grande quantidade de fãs e seguidores, desenvolvedores que realmente acreditam no propósito da linguagem e ajudam no desenvolvimento de novas bibliotecas e atualizações da linguagem. Esse enorme crescimento foi mostrado pelo índice TIOBE, que mede o crescimento das linguagens de programação, que deu ao *Ruby* em 2006 um prêmio de maior crescimento e hoje em dia ele se encontra em décimo primeiro lugar no ranking de linguagens mais populares como pode ser visto na tabela 2 [18].

A linguagem tem algumas características interessantes, como:

- Enxergar tudo como um objeto
- Flexibilidade na hora de criar ou alterar métodos
- Aparência simples e descritiva
- Tratamento de exceções

Por esses motivos a linguagem foi escolhida para auxiliar o desenvolvimento do projeto.

Tabela 2- Ranking das linguagens de programação [19]

Posição Dez 2011	Posição Dez 2010	Mudança Posição	Linguagem de Programação	Ratings Dec 2011	Delta Dec 2010	Status
1	1	=	Java	17.561%	-0.44%	A
2	2	=	C	17.057%	+0.98%	A
3	3	=	C++	8.252%	-0.76%	A
4	5	↑	C#	8.205%	+1.52%	A
5	8	↑↑↑	Objective-C	6.805%	+3.56%	A
6	4	↓↓	PHP	6.001%	-1.51%	A
7	7	=	(Visual) Basic	4.757%	-0.36%	A
8	6	↓↓	Python	3.492%	-2.99%	A
9	9	=	Perl	2.472%	+0.14%	A
10	12	↑↑	JavaScript	2.199%	+0.69%	A
11	11	=	Ruby	1.494%	-0.29%	A
12	10	↓↓	Delphi/Object Pascal	1.245%	-0.93%	A
13	13	=	Lisp	1.175%	+0.11%	A
14	23	↑↑↑↑↑↑↑↑ ↑↑	PL/SQL	0.803%	+0.24%	A
15	14	↓	Transact-SQL	0.746%	-0.03%	A
16	16	=	Pascal	0.734%	-0.03%	A
17	18	↑	Ada	0.632%	-0.02%	B
18	35	↑↑↑↑↑↑↑↑ ↑↑↑	Logo	0.619%	+0.26%	B
19	17	↓↓	Assembly	0.563%	-0.10%	B
20	25	↑↑↑↑↑	ABAP	0.560%	+0.01%	B

[Fonte: Tiobe Software, 2012]

4.2.3 RAILS

Rails é um *framework web*, desenvolvido em *Ruby*, e foi criado para tornar o desenvolvimento *web* muito mais rápido e fácil, fazendo com que o desenvolvedor precise focar somente no desenvolvimento [20] [21].

Para isso o *rails* cria um grau de abstração maior que no desenvolvimento em algumas outras linguagens como *Java*. Por exemplo, para criar um modelo e uma tabela utilizando esse *framework*, basta rodar dois comandos no terminal, como visto na figura 4.

```
rails g model Usuario nome:string cpf_cnpj:string senha:string  
rake db:migrate
```

Figura 4 – Exemplos de comandos do Rails

No primeiro comando, o *rails* cria um arquivo chamado *usuario.rb* e um outro arquivo contendo as regras de criação da tabela no banco, chamado *[timestamp]_create_usuarios.rb*, que é utilizado no segundo comando para a criação da tabela no banco.

Esse exemplo demonstra a preocupação do *framework* em facilitar o desenvolvimento *web* abstraindo as partes de configurações, ou seja, fazendo com que o desenvolvedor, não precise gastar muito tempo com as configurações que são similares em todos os projetos *web*.

Essa preocupação com a facilidade é vista também na criação dos diretórios, que o *rails* separa os arquivos da aplicação, dos arquivos de configuração, por exemplo, e ao trabalhar com a arquitetura MVC (*Model/View/Controller*).

Outra grande vantagem desse *framework*, é que existem diversas ferramentas (*gems*) desenvolvidas que se encontram disponíveis na internet para uso no projeto, sendo necessário, na maioria dos casos, somente um comando para a instalação da *gem*, como exibido na figura 5:

```
gem install heroku
```

Figura 5 – Instalação de uma biblioteca do Rails

Todas essas facilidades, além do vasto conteúdo de ajuda encontrado na internet sobre o *framework*, foram decisivas para a escolha de se trabalhar com *rails* nesse projeto final.

4.3 TESTES

Os testes foram abordados de duas maneiras: testes unitários para cada modelo, validando suas regras, como campos obrigatórios, campos que possuem formato específico, entre outras regras e testes de interface, validando o caminho percorrido pelo usuário, os *links*, e todos os outros componentes da interface.

Para isso, foram utilizadas duas ferramentas, uma para cada tipo de abordagem. Para os testes unitários, a ferramenta escolhida foi o RSpec, que facilita a criação de modelos e a validação das suas regras. Já para os testes de integração, foi utilizada a ferramenta chamada Cucumber, que permite simular a navegação do usuário pela interface. Além disso, também foi utilizada uma ferramenta chamada FactoryGirl, para permitir a criação de objetos dos modelos de maneira mais fácil no ambiente de testes.

4.3.1 RSPEC

RSpec é uma ferramenta de *Behaviour-Driven Development* (BDD), desenvolvida para ajudar na parte de *Tests Driven Development* (TDD) do projeto, focando assim na documentação e design dos testes [22].

A utilização dessa ferramenta em um projeto desenvolvido em *RubyOnRails* é muito simples, pois a mesma é disponibilizada como uma gem, bastando, como citado no tópico sobre o framework rails, utilizar o comando `gem install rspec` a partir do terminal para a sua instalação.

Um exemplo de como essa ferramenta pode ser utilizada nos testes é descrito na documentação da mesma, como ilustra a figura 6:

```
# game_spec.rb

describe Game do
  describe "#score" do
    it "returns 0 for all gutter game" do
      game = Game.new
      20.times { game.roll(0) }
      game.score.should == 0
    end
  end
end
```

Figura 6 - Exemplo de utilização retirado da documentação do RSpec [23]

Após a instalação dessa ferramenta, uma pasta chamada 'spec' é criada no diretório raiz do projeto, e dentro dessa pasta os testes devem ser organizados, separando-os em testes de controladores, integração, modelos e interface. Essa ferramenta pode ser utilizada para testes de todas as camadas do sistema, desde a interface, passando pelos controladores, até os modelos, podendo também ser utilizada para fazer testes de integração.

Nesse trabalho a ferramenta foi utilizada para teste unitários dos modelos somente, e outra ferramenta foi utilizada para os testes da interface.

4.3.2 CUCUMBER

Cucumber é uma ferramenta criada para ajudar o trabalho com BDD (*Behavior Driven Development* – Desenvolvimento Guiado por Comportamento), uma técnica de desenvolvimento ágil para testes focados no comportamento que o sistema deve ter, sendo usado amplamente como uma ferramenta de testes [24].

Essa ferramenta pode ser utilizada em qualquer linguagem, e foi escrita em Ruby. A sua utilização em desenvolvimento Ruby é feita através da instalação de uma gem chamada cucumber-rails.

CAPÍTULO 5 – RECEITUÁRIO ONLINE

Nesta seção, serão apresentadas as funcionalidades referentes ao protótipo que foram implementadas, implantadas e testadas. Dentre todas as funcionalidades do sistema, é possível destacar as que detêm um papel essencial para alcançar o objetivo principal deste trabalho, como o controle de acesso de todos os agentes envolvidos, cadastramento de receitas por parte do médico, atendimento de receitas pelo farmacêutico ou balconista e a busca de receita pelos atores do sistema.

5.1.ACESSO AO SISTEMA

Para ter acesso ao sistema e poder efetuar suas funções em si é necessário ter um *login* e senha cadastrados no sistema.

Ao entrar no sistema e clicar no item do menu chamado “Login”, como mostrado na figura 7, será possível visualizar a tela de login no sistema, onde é necessário inserir seu CPF se for pessoa física ou CNPJ se for pessoa jurídica, depois inserir a senha e clicar em “Login”.

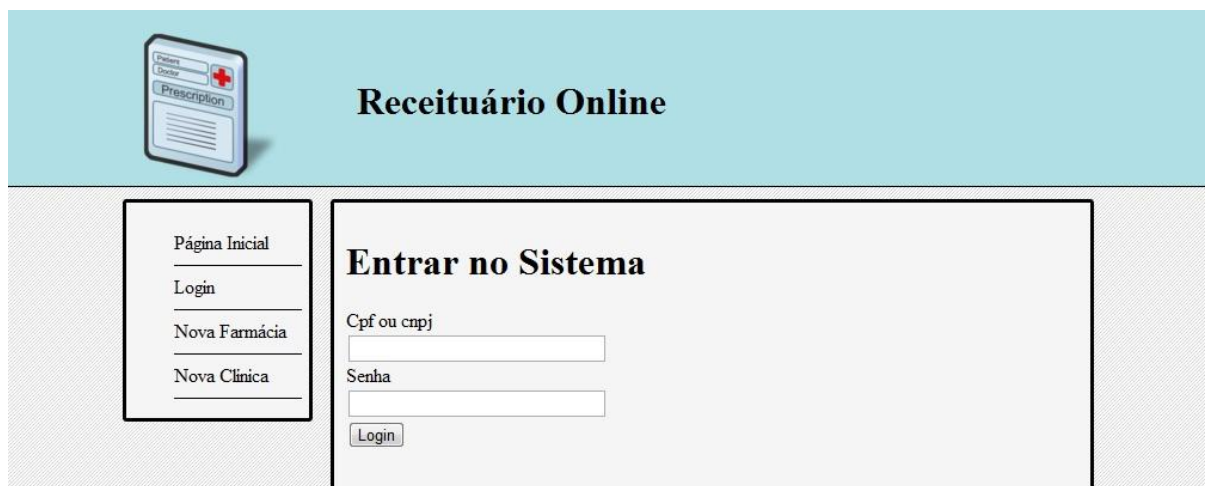


Figura 7 - Página de login do sistema

5.2.CADASTRO DE FARMÁCIA

Assim que o sistema é aberto existe um item no menu lateral esquerdo, chamado de “Nova farmácia”, como mostra a figura 8, que quando é clicado mostra uma tela de cadastro com as principais informações necessárias para se cadastrar

uma farmácia, como:

- CNPJ
- Nome
- Endereço
- Telefone



The image shows a screenshot of a web application titled "Receituário Online". At the top left, there is a small icon of a prescription slip with a red cross and the word "Prescription". The main header is a light blue bar with the text "Receituário Online" in bold. Below the header, on the left side, there is a vertical menu with four items: "Página Inicial", "Login", "Nova Farmácia", and "Nova Clínica". The main content area is titled "Cadastro de Farmácia" and contains a form with the following fields: "Nome", "Cnpj", "Email", "Senha", and "Endereço". Each field has a corresponding text input box. At the bottom of the form is a "Confirmar" button.

Figura 8 - Cadastro de farmácia

5.3.CADASTRO DE CLÍNICA/ HOSPITAL/ CONSULTÓRIO

Assim que o sistema é aberto existe um item no menu lateral esquerdo, chamado de “Nova clínica”, como é possível ver na figura 9, que quando é clicado mostra uma tela de cadastro com as principais informações necessárias para se cadastrar uma clínica, como:

- CNPJ
- Nome
- Endereço
- Telefone



The image shows a web interface for 'Receituário Online'. At the top left, there is a logo depicting a prescription slip with fields for 'Patient', 'Doctor', and 'Prescription'. The main header is 'Receituário Online'. On the left side, there is a navigation menu with the following items: 'Página Inicial', 'Login', 'Nova Farmácia', and 'Nova Clínica'. The main content area is titled 'Cadastro de Clínica' and contains a registration form with the following fields: 'Nome', 'Cnpj', 'Email', 'Senha', and 'Endereco'. A 'Confirmar' button is located at the bottom of the form.

Figura 9 - Cadastro de clínicas

5.4.FARMÁCIA

Quando os balconistas e/ou farmacêuticos ainda não possuem cadastro no sistema, é necessário que a farmácia faça esse cadastro, sendo o balconista/farmacêutico automaticamente vinculado à farmácia que o cadastrou. Caso o balconista ou o farmacêutico possua cadastro no sistema, a alocação dele em uma farmácia é feita de uma maneira muito mais rápida, através do link “Alocar balconista à farmácia” e “Alocar farmacêutico à farmácia”, que se encontra na página de perfil de balconista e farmacêutico, respectivamente.

É permitido também à farmácia acessar a listagens de todos os balconistas e farmacêuticos que nela trabalham. Isso pode ser necessário, por exemplo, para desalocar um funcionário da farmácia, e também, para visualizar as receitas atendidas por esses funcionários.

Assim que o paciente for à farmácia será possível procurar por ele através do link “Procurar Pacientes” e caso encontre o paciente, será possível visualizar as receitas referentes ao paciente, assim como as que ainda estão pendentes, ou seja, que podem ser atendidas, caso contrário, isso quer dizer que a receita dele não foi cadastrada no sistema, tendo que executar seu processo através da receita

impressa.

A farmácia pode também listar todas as receitas.

5.6.CLÍNICA

Quando o médico ainda não possui cadastro no sistema, é necessário que a clínica faça esse cadastro, sendo o médico automaticamente vinculado à clínica que o cadastrou. Caso o médico possua cadastro no sistema, a alocação dele em uma clínica é feita de uma maneira muito mais rápida, através do link “Alocar médico à clínica”, que se encontra na página de perfil do médico no sistema.

É permitido também à clínica acessar a listagens de todos os médicos que nela trabalham. Isso pode ser necessário, por exemplo, para desalocar um médico da clínica, dado que ele não tenha deixado de trabalhar para a mesma, e também, para visualizar as receitas geradas por esses médicos.

É possível verificar se um paciente tem ou não cadastro no sistema através do link “Procurar Pacientes”. Caso o paciente não esteja cadastrado no sistema, é feito um cadastro dele, através do link “Cadastrar Paciente”, para que quando o médico for atender o paciente, não seja necessário que se faça isso, facilitando o processo do médico e deixando para ele apenas a seleção do paciente na hora de criar a receita.

5.7.MÉDICO

Quando um médico necessitar criar uma receita, durante a consulta com um paciente, isso será feito através do link “Criar Receita Médica”, onde será possível associar a receita ao paciente. Caso o paciente ainda não possua um cadastro no sistema, será necessário cadastrá-lo através do link “Cadastrar Paciente”.

O médico pode visualizar o histórico dos pacientes procurando e selecionando o paciente através do link “Procurar Pacientes”, acessando assim o histórico de receitas e consultas do mesmo.

Outra opção disponível para o médico é poder listar todas as receitas que foram criadas por ele anteriormente, para visualizar a quantidade de receitas, visualizar a frequência dos pacientes e assim por diante.

5.8.PACIENTE

Caso o paciente queira visualizar todas as suas receitas ou tiver o interesse de salvar em versão PDF ou até mesmo imprimir, ele será capaz de fazer isso, sem poder de edição, de qualquer lugar do mundo que tenha acesso à internet.

5.9.FARMACÊUTICO/BALCONISTA

Os farmacêuticos e balconistas possuem acesso às mesmas funcionalidades do protótipo.

Assim que o paciente entra na farmácia é necessário procurar por ele no sistema, através do link “Procurar Pacientes”, podendo assim visualizar todas as receitas a ele vinculadas. Dentre essas, poderão ser vistas as receitas que ainda não foram atendidas, que no caso serão as escolhidas para serem atendidas por parte do farmacêutico ou balconista se houver estoque suficiente para suprir a quantidade necessária de medicamentos.

5.10.TESTES

Para que o protótipo obtivesse uma garantia de qualidade adequada, foi necessária a realização de uma grande variedade de testes. Os testes que foram realizados, não são testes manuais, onde é executado o programa a ser testado e a conferência do resultado é feita visualmente. Todos os testes aqui apresentados são testes automatizados, onde existe um programa que executa o que vai ser testado e verifica automaticamente os resultados obtidos com os resultados esperados.

Neste protótipo foram testadas todas as regras de cada modelo, como a acessibilidade dos seus atributos e a obrigatoriedade de alguns campos, por exemplo.

Na figura 10 é possível visualizar o código de alguns testes como exemplo.

```

it "cpf ou cnpj deve ser obrigatório" do
  usuario.cpf_cnpj = ""
  usuario.should_not be_valid
end

it "cpf ou cnpj devem ser numéricos" do
  usuario.cpf_cnpj = "a"
  usuario.should_not be_valid
end

it "nome deve ser obrigatório" do
  usuario.nome = ""
  usuario.should_not be_valid
end

it "senha deve ser obrigatória" do
  usuario.senha = ""
  usuario.should_not be_valid
end

describe "email" do

  it "email deve ser obrigatório" do
    usuario.email = ""
    usuario.should_not be_valid
  end

  it "o email joaozinho.com deve ser inválido" do
    usuario.email = "joaozinho.com"
    usuario.should_not be_valid
  end

  it "o email a@.com deve ser inválido" do
    usuario.email = "a@.com"
    usuario.should_not be_valid
  end

  it "o email a@.com deve ser inválido" do
    usuario.email = "@com"
    usuario.should_not be_valid
  end
end

```

Figura 10 – Código de alguns testes feitos no protótipo

Na figura 11 é possível ver a quantidade de testes que foram feitos no desenvolvimento desse protótipo, e constatar que o desempenho do sistema foi satisfatório em todos eles.

```

Finished in 20.65 seconds
67 examples, 0 failures

```

Figura 11 - Testes do protótipo passando

Neste protótipo foram testadas também as funcionalidades através da interface, utilizando o Cucumber.

Nas figuras 12 e 13, é possível visualizar o código de alguns testes e alguns testes passando, como exemplo.


```
Feature: Cadastrar Medico
  Uma clinica cadastrada e autenticada no sistema deve poder cadastrar
  Um médico para que ele possa acessar o sistema para atender os pacientes

  Scenario: Clinica cadastra medico com dados validos
    Given I am logged in as a clinica with [cnpj] "123" and senha "senha"
    And I follow "Cadastrar Médico"
    And I fill in medico's nome with "nome"
    And I fill in medico's email with "user@test.com"
    And I fill in medico's cpf with "987"
    And I fill in medico's senha with "senha"
    And I fill in medico's crm with "crm"
    And I press "Confirmar"
    Then I should see "Conta criada com sucesso"

  Scenario: Clinica tenta cadastrar medico com um email inválido
    Given I am logged in as a clinica with [cnpj] "123" and senha "senha"
    And I follow "Cadastrar Médico"
    And I fill in medico's nome with "nome"
    And I fill in medico's email with "mailinvalido"
    And I fill in medico's cpf with "1234"
    And I fill in medico's senha with "senha"
    And I fill in medico's crm with "crm"
    And I press "Confirmar"
    Then I should see "Erro ao criar a conta"

  Scenario: Clinica tenta cadastrar medico sem email
    Given I am logged in as a clinica with [cnpj] "123" and senha "senha"
    And I follow "Cadastrar Médico"
    And I fill in medico's nome with "nome"
    And I fill in medico's email with ""
    And I fill in medico's cpf with "1234"
    And I fill in medico's senha with "senha"
    And I fill in medico's crm with "crm"
    And I press "Confirmar"
    Then I should see "Erro ao criar a conta"
```

Figura 12 - Código de alguns testes de interface feitos no protótipo

```

Scenário: Farmacia cadastra paciente com dados validos # features/pacientes/cadastro.feature:5
  Given I am logged in as a medico with cpf "123" and senha "senha" # features/step_definitions/medico_steps.rb:1
  And I follow "Cadastrar Paciente" # features/step_definitions/web_steps.rb:33
  And I fill in paciente's nome with "nome" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's email with "user@test.com" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's cpf with "987" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's senha with "senha" # features/step_definitions/helper_steps.rb:1
  And I press "Confirmar" # features/step_definitions/web_steps.rb:27
  Then I should see "Conta criada com sucesso" # features/step_definitions/web_steps.rb:107

Scenário: Farmacia tenta cadastrar paciente com um email inválido # features/pacientes/cadastro.feature:15
  Given I am logged in as a medico with cpf "123" and senha "senha" # features/step_definitions/medico_steps.rb:1
  And I follow "Cadastrar Paciente" # features/step_definitions/web_steps.rb:33
  And I fill in paciente's nome with "nome" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's email with "emailinvalido" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's cpf with "1234" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's senha with "senha" # features/step_definitions/helper_steps.rb:1
  And I press "Confirmar" # features/step_definitions/web_steps.rb:27
  Then I should see "Erro ao criar a conta" # features/step_definitions/web_steps.rb:107

Scenário: Farmacia tenta cadastrar paciente sem email # features/pacientes/cadastro.feature:25
  Given I am logged in as a medico with cpf "123" and senha "senha" # features/step_definitions/medico_steps.rb:1
  And I follow "Cadastrar Paciente" # features/step_definitions/web_steps.rb:33
  And I fill in paciente's nome with "nome" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's email with "" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's cpf with "1234" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's senha with "senha" # features/step_definitions/helper_steps.rb:1
  And I press "Confirmar" # features/step_definitions/web_steps.rb:27
  Then I should see "Erro ao criar a conta" # features/step_definitions/web_steps.rb:107

Scenário: Farmacia tenta cadastrar paciente sem senha # features/pacientes/cadastro.feature:35
  Given I am logged in as a medico with cpf "123" and senha "senha" # features/step_definitions/medico_steps.rb:1
  And I follow "Cadastrar Paciente" # features/step_definitions/web_steps.rb:33
  And I fill in paciente's nome with "nome" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's email with "a@a.com" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's cpf with "1234" # features/step_definitions/helper_steps.rb:1
  And I fill in paciente's senha with "" # features/step_definitions/helper_steps.rb:1
  And I press "Confirmar" # features/step_definitions/web_steps.rb:27
  Then I should see "Erro ao criar a conta" # features/step_definitions/web_steps.rb:107

```

Figura 13 - Exemplo de alguns testes de interface rodando

Na figura 14 é possível ver a quantidade de testes de interface que foram feitos no desenvolvimento desse protótipo, e constatar que o desempenho do sistema foi satisfatório em todos eles.

```

38 cenários (38 passed)
306 passos (306 passed)

```

Figura 14 - Número de cenários testados

CAPÍTULO 6 - DISCUSSÃO E CONCLUSÃO

Conclui-se que o objeto de estudo e desenvolvimento provêm melhorias no controle de geração de receitas por parte dos médicos e atendimento das receitas por parte dos farmacêuticos e balconistas. Cumpre-se o objetivo principal do estudo e do desenvolvimento que teve como finalidade a criação de um protótipo, capaz de envolver o processo que o paciente passa quando possui uma enfermidade e recorre ao médico, tendo como resultado na saída do consultório, clínica ou hospital uma prescrição médica e esta sendo levada à farmácia para que seja efetuada a compra dos medicamentos receitados.

O protótipo desenvolvido tem uma potencial contribuição para a economia de papel com as prescrições médicas, afastando a necessidade da impressão das mesmas. Essa economia é de suma importância para o meio ambiente, evitando a degradação e gastando cada vez menos os recursos que são finitos e escassos.

Além disso, o transtorno com relação à legibilidade das receitas foi uma questão em que houve uma grande contribuição, pois as receitas registradas no sistema terão a mesma fonte, cor e estilo. Isso proporciona uma leitura mais fácil, podendo ser lida por todos.

O objeto de estudo e implementação tem potencial de contribuição para o problema de informações que se encontravam espalhadas, ou muitas vezes desperdiçadas. Isso foi feito através da concentração das mesmas em um único sistema, e garantindo que nada seja apagado da base de dados, garantindo também a segurança dos dados.

A solução proposta para o problema de dissipação das informações agregou valor aos médicos, às clínicas, aos hospitais e aos consultórios, principalmente. Isso ocorre devido à obtenção do acesso ao histórico de receitas que foram prescritas ao paciente, mesmo sendo a primeira visita do paciente a essa clínica, hospital ou consultório, pois o sistema terá esse histórico em atendimentos feitos em outros estabelecimentos. Com isso as clínicas também não precisarão mais se preocupar em guardar os dados e o próprio histórico de receitas prescritas do cliente em uma base de dados própria.

Em se tratando das farmácias e seus funcionários, seus principais benefícios ao utilizarem o protótipo é poder validar a veracidade da receita de forma muito mais fácil e com muito mais segurança, dificultando as vendas de medicamentos de forma

ilegal, levando em conta que todas as receitas atendidas estarão registradas no sistema.

Com relação ao paciente, este terá todas as suas informações com relação às receitas anteriormente prescritas acessíveis em um único sistema, que ele poderá consultar quando achar necessário. Além disso, com a receita cadastrada online, surge a oportunidade de novos negócios, com as vendas de medicamentos online, inclusive controlados, permitindo assim que os pacientes não precisem mais sair de casa para comprar remédios.

Com um sistema centralizado, e a partir do momento em que todas as prescrições e vendas de remédios forem registradas nesse sistema, se torna muito mais fácil controlar esse processo, sendo uma grande vantagem para os órgãos reguladores.

5.11. TRABALHOS FUTUROS

Como trabalhos futuros, um aprimoramento da interface, melhorando a usabilidade e comunicabilidade da mesma é essencial.

Uma melhoria necessária também para aumento da segurança é a implementação de uma verificação dos médicos e farmacêuticos junto aos devidos órgãos reguladores, garantindo assim acesso ao sistema somente por médicos e farmacêuticos que estão devidamente regularizados.

A geração automática de relatórios e históricos, além de gráficos estatísticos, é uma funcionalidade que pode ser desenvolvida para tornar ainda mais fácil o acesso aos dados encontrados centralizados na base de dados do sistema. Esses gráficos poderiam ser utilizados também para planejamento de fabricação de remédios, melhorias na saúde do país, entre outras possibilidades.

CAPÍTULO 7 - REFERÊNCIAS BIBLIOGRÁFICAS

1. FRANCISCO QUEIROZ, **As políticas públicas no Brasil: heranças, tendências e desafios.** Disponível em <http://franciscoqueiroz.com.br/portal/phocadownload/gestao/taniabacelar.pdf>. Acesso em: Novembro de 2011.
2. GOOGLE, **PIB do Brasil.** Disponível em: http://www.google.com.br/publicdata/explore?ds=d5bncppjof8f9 &met_y=ny_gdp_mktp_cd&idim=country:BRA&dl=pt-BR&hl=pt-BR&q=pib+brasil. Acesso em: Dezembro de 2011.
3. WIKIPEDIA, **Suécia,** Disponível em: <http://pt.wikipedia.org/wiki/Su%C3%A9cia>. Acesso em: Dezembro de 2011.
4. WIKIPEDIA, **Brasil.** Disponível em: <http://pt.wikipedia.org/wiki/Brasil>. Acesso em: Novembro de 2011.
5. *APOTEKENS SERVICE.* Disponível em: <http://www.apotekensservice.se/>. Acesso em: Agosto de 2011.
6. WIKIPEDIA, **Healthcare in Sweden.** Disponível em: http://en.wikipedia.org/wiki/Healthcare_in_Sweden#Details_and_patient_costs. Acesso em: Novembro de 2011.
7. UOL, **Preguiça de sair de casa é reflexo de mudança de prioridades.** Disponível em: <http://www1.folha.uol.com.br/folha/equilibrio/noticias/ult263u731140.shtml>. Acesso em: Outubro de 2011.
8. WIKIPEDIA, **Prescrição ilegível.** Disponível em http://pt.wikipedia.org/wiki/Prescri%C3%A7%C3%A3o_m%C3%A9dica_ileg%C3%ADvel. Acesso em: Setembro de 2011.
9. UOL, **Letra ilegível rende multa a médicos de Londrina (PR).** Disponível em: <http://www1.folha.uol.com.br/folha/cotidiano/ult95u562935.shtml>. Acesso em: Agosto de 2011.
10. SLIDESHARE, **Análise de Requisitos de Software.** Disponível em: <http://www.slideshare.net/Ridlo/analise-de-requisitos-software>. Acesso em: Dezembro de 2011.

11. IAN SOMMERVILLE, **Engenharia de Software**. 9 edição. Editora Pearson. 2011. ISBN: 8579361087.
12. CREMERJ, **Manual de Orientação sobre Prescrição de Medicamentos Sujeitos a Controle Especial: informações para médicos e profissionais de saúde**. Disponível em: <<http://www.cremerj.org.br/publicacoes/108.PDF>>. Acesso em: Novembro de 2011.
13. MARCUS SAMPAIO, **Casos de Uso**. Disponível em: <<http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/diagramas/usecases/usecases.htm>>. Acesso em: Novembro de 2011.
14. VINICIUS TELES, **Extreme Programming**. 1 edição. Editora Novatec. 2011. ISBN: 8575220470.
15. MYSQL, **Market share**. Disponível em: <<http://www.mysql.com/why-mysql/marketshare/>>. Acesso em: Novembro de 2011.
16. GARTNER. **Gartner**. Disponível em: <<http://www.gartner.com/technology/about.jsp>>. Acesso em: Setembro de 2011.
17. RUBY. Disponível em: < <http://www.ruby-lang.org/en/about/> >. Acesso em: Agosto de 2011.
18. TIOBE. Disponível em: <http://www.tiobe.com/index.php/content/paperinfo/tpci/tpci_definition.htm>. Acesso em: Dezembro de 2011.
19. TIOBE. Disponível em: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>. Acesso em: Dezembro de 2011.
20. RUBY ON RAILS, **API Ruby On Rails**. Disponível em: <<http://api.rubyonrails.org/>>. Acesso em: Agosto de 2011.
21. RUBY ON RAILS, **Guias Ruby On Rails**. Disponível em: <<http://guias.rubyonrails.com.br/>>. Acesso em: Agosto de 2011.
22. RELISHAPP, **Rspec**. Disponível em <<https://www.relishapp.com/rspec>>. Acesso em Setembro de 2011.
23. RELISHAPP, **Rspec**. Disponível em:

<<https://www.relishapp.com/rspec/docs/gettingstarted>>. Acesso em: Agosto de 2011.

24. CUKES, Cucumber. Disponível em: <<http://cukes.info/>>. Acesso em: Agosto de 2011.