



**Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Faculdade de Computação
Curso de Bacharelado em Sistemas de Informação**

**DESENVOLVIMENTO DE UM *WEBAPP* PARA GERENCIAMENTO
E MONITORAMENTO DE TERAPIAS - *Therapy***

**JÚLIO RAMON NOGUEIRA LISBOA
GERSON SOUZA DA SILVA**

Belém – PA

2017

JÚLIO RAMON NOGUEIRA LISBOA
GERSON SOUZA DA SILVA

**DESENVOLVIMENTO DE UM *WEBAPP* PARA GERENCIAMENTO
E MONITORAMENTO DE TERAPIAS - *Therapy***

Trabalho de Conclusão de Curso apresentado como
requisito parcial para a obtenção do grau de Bacharel
em Sistemas de Informação pela Faculdade de
Computação do Instituto de Ciências Exatas e Naturais
da Universidade Federal do Pará.

Orientador: Prof. Dr. Alfredo Braga Furtado

Belém – PA

2017

JÚLIO RAMON NOGUEIRA LISBOA
GERSON SOUZA DA SILVA

**DESENVOLVIMENTO DE UM *WEBAPP* PARA GERENCIAMENTO
E MONITORAMENTO DE TERAPIAS - *Therapy***

Trabalho de Conclusão de Curso
apresentado como requisito para obtenção do
Grau de Bacharel em Sistemas de Informação,
Universidade Federal do Pará.

Aprovado em ____/____/2017.

BANCA EXAMINADORA

Prof. Dr. Alfredo Braga Furtado

Prof. Dr. Raimundo Viégas Júnior

Prof. M. Sc. José Maria Nascimento Bitar

Dedico este trabalho à Deus, sem Ti, Jesus, não seria possível. À minha esposa amada que me apoia e motiva com seu amor, aos meus filhos, minha mãe, minha irmã e meu sobrinho, aos meus familiares e amigos.

**JULIO RAMON NOGUEIRA
LISBOA**

Dedico esse trabalho primeiramente a Deus que me deu a capacitação e a força necessária. Aos meus pais, pelo apoio, incentivo, e que nas horas boas e ruins, sempre estiveram ao meu lado. Dedico também aos meus professores em especial nessa longa jornada.

GERSON SOUSA DA SILVA

AGRADECIMENTOS

Agradeço em especial a meu bom Deus, por toda força, por sempre me mostrar o melhor caminho a ser seguido, e por me lembrar de suas promessas.

Agradeço a minha esposa, por me ajudar em todos os momentos, motivar e me ajudar com soluções, minha auxiliadora e porto seguro.

Agradeço aos meus filhos por serem a motivação que preciso no momento em que as forças acabam, e lembrar que preciso ter sucesso em tudo, por eles.

Agradeço à minha mãe por sempre acreditar, minha irmã por acreditar e ajudar.

Agradeço ao Prof. Dr. Alfredo Braga Furtado por nos orientar, com excelência e acreditar no ensino e na pesquisa, assim como por toda sua contribuição ao meio acadêmico.

Agradeço à terapeuta ocupacional Fabiani Figueira, pedagoga Rose Montelo e à psicóloga Thais, da Casa do Autista, por nos ajudarem sugerindo melhorias no protótipo.

“Não a nós, Senhor, nenhuma glória para nós, mas sim ao teu nome, por teu amor e por tua fidelidade!” (SL 115.1)

JULIO RAMON NOGUEIRA LISBOA

Agradecer é um ato nobre do homem. É o ato de reconhecer o quão válido foi à participação do próximo em algum momento da sua vida.

Agradeço ao Bom Deus, pelo dom da vida e por ter-me conduzido até este momento de enriquecimento profissional.

Agradeço também aos professores que passarão na minha carreira acadêmica em especial ao meu orientador, obrigado pelos conhecimentos transmitidos.

A todos os meus familiares, irmã, primos, tios (In memoriam) e amigos. Não citarei nomes, para não me esquecer de ninguém, partilho a alegria deste momento.

Aos meus pais, vocês que, muitas vezes, renunciaram aos seus sonhos para que eu pudesse realizar o meu, por me aconselharem, por ter confiado em mim.

“Tudo tem o seu tempo determinado, e há tempo para todo propósito debaixo do céu.” (Eclesiastes 3:1.).

GERSON SOUSA DA SILVA

RESUMO

A plataforma *WEB* passou por muitas modificações nos últimos anos, sua arquitetura foi melhorada e devido aos seus avanços, foi possível desenvolver *software* mais robusto, com dimensões de sistemas de grande porte, o que só era possível em sistemas *Desktop*. Um novo nicho surgiu: os *WebApps*. Portanto, este trabalho aborda o desenvolvimento de um sistema, na plataforma, de gerenciamento e monitoramento de terapias (chamado *Therapy*). Também descreve os processos ágeis e metodologias adotadas para o desenvolvimento do sistema, respeitando os padrões estabelecidos pela comunidade de engenharia de software. O *Therapy* tem por objetivo organizar, gerir, montar histórico de resultados das terapias e auxiliar na tomada de decisão por meio dos relatórios emitidos a partir de um gerador de formulários, incorporado ao prontuário do paciente. Tal formulário será criado pelos terapeutas participantes do sistema. As tecnologias escolhidas e a metodologia aplicada tiveram sucesso em seu objetivo, sendo assim os pontos não contemplados e desenvolvidos (emissão de relatórios e testes de usabilidade), não se deram por limitações da metodologia ou recursos da tecnologia utilizada, mas por outros motivos (tempo hábil). Apesar das lacunas abertas, o *software* possui grande potencial para um produto final comercializável e/ou produzir outros trabalhos semelhantes.

Palavras-Chave: Gerenciamento de Terapias. Desenvolvimento de Software. *WebApps*. Sistemas em PHP.

ABSTRACT

The *WEB* platform underwent many modifications in the last years, its architecture was improved and due to its advances, it was possible to develop more robust *softwares*, with dimensions of large systems, which was only possible in *Desktop* systems. Therefore, a new niche emerged: the *WebApps*. In this sense, this work addresses the development of a system, on the *WEB* platform, for management and monitoring of therapies. It also describes the agile processes and methodologies adopted for the development of the system, respecting the standards established by the software engineering community. The purpose of *Therapy* is to organize, manage, record the results of therapies and assist in decision-making through reports generated from a form generator, incorporated into the patient's chart. This form will be created by the system's participating therapists. The chosen technologies and the applied methodology were successful in their objective, so the not contemplated and developed points (reporting and usability tests), were not due to limitations of the methodology or resources of the used technology, but for other reasons (skillful time). Despite the open gaps, the *software* has great potential for a marketable end product and/or producing other similar jobs.

Keywords: Management of Therapies. Software development. *WebApps*. Systems in PHP.

SUMÁRIO

LISTA DE FIGURAS	8	
LISTA DE TABELAS	9	
1	INTRODUÇÃO	10
1.1	Motivação	11
1.2	Justificativa	11
1.3	Objetivos	12
1.3.1	Objetivo Geral	12
1.3.2	Objetivos Específicos	13
1.4	Organização do trabalho	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Engenharia de Software	14
2.2	Sistemas de Informação e <i>WEBAPP</i>	15
2.3	Qualidades <i>WebApps</i>	16
2.4	Arquiteturas	17
2.4.1	Arquiteturas	17
2.5	Usabilidade	17
2.6	Ferramentas e Tecnologias	18
3	REVISÃO BIBLIOGRÁFICA	19
3.1	Trabalhos Relacionados	19
3.1.1	ProDoctor	19
3.1.2	Clinica nas Nuvens	20
4	MATERIAL E MÉTODOS	22
4.1	Modelo de Processo de Software	22
4.2	Material	22
4.2.1	Ferramentas de Desenvolvimento	22
4.2.2	Para a codificação: <i>Sublime Text 3</i>	23
4.2.3	Para o banco: <i>SGBD MySQL</i>	23
4.2.4	Para a modelagem do banco: <i>MySQL Workbench</i>	23
4.2.5	Padrão de Projeto	23
4.2.6	Linguagem de Programação	24

4.3	Descrição do Projeto	25
4.3.1	Requisitos	26
4.3.2	Layout da Aplicação	27
4.4	Modelagem de software	27
4.4.1	Casos de Uso	28
4.4.2	Modelo Relacional	30
5	IMPLEMENTAÇÃO E AVALIAÇÃO DOS RESULTADOS	32
5.1	Telas do <i>Therapy</i>	32
5.2	Testes realizados no desenvolvimento	38
5.3	Resultado da Pesquisa de Aceitabilidade	38
6	CONCLUSÃO	41
6.1	Considerações finais	41
6.2	Trabalhos Futuros	41
	REFERÊNCIAS	42
	APÊNDICE A – Documento de Especificação de Requisitos	45
	Ambiente do Sistema	45
	Objetivo do produto	45
	Benefício do programa	46
	APÊNDICE B – Documento de Especificação de Análise	52
	Diagrama de Pacotes	52
	Diagrama de Classes	52
	Diagrama de Atividades	53
	APÊNDICE C – Dicionário de Dados	56

LISTA DE FIGURAS

Figura 1 - Qualidade em <i>WEBAPPS</i>	17
Figura 2 - <i>Software ProDoctor</i> . Fonte: <i>Prodoctor Software S/A</i> (2016).	20
Figura 3 - Sistema Clínica nas Nuvens Fonte: <i>Camtwo Sistemas</i> (2015).	21
Figura 4 Fluxo de dados e requisições no <i>CodeIgniter</i> [codeigniter.com, 2017]	25
Figura 5 Tela de Prontuário. Principal do <i>Therapy</i> .	27
Figura 6 Diagrama de Casos de Uso do Usuário	29
Figura 7 Diagrama de Casos de Uso do Terapeuta	30
Figura 8 Diagrama de Casos de Uso do Paciente	30
Figura 9 Modelagem do banco de dados, entidade relacional.	31
Figura 10 Tela inicial. Escolhe-se o tipo de <i>login</i>	32
Figura 11 Tela de <i>Login</i> . Nesse caso, <i>login</i> dos terapeutas.	32
Figura 12 Tela de gerenciamento de pacientes.	33
Figura 13 Tela de inserção. Aqui cria-se um novo paciente.	33
Figura 14 Tela de terapias do terapeuta que está logado.	34
Figura 15 Tela dos formulários da terapia escolhida.	35
Figura 16 Tela das questões do formulário.	35
Figura 17 Tela de pesquisa de pacientes no prontuário.	36
Figura 18 Tela de prontuário. Onde estão concentrados todos os dados.	37
Figura 19 Tela da anamnese do paciente.	37
Figura 20 - Resposta da primeira pergunta, com base em 25 respostas	38
Figura 21 - Resposta da segunda pergunta, com base em 25 respostas	39
Figura 22 - Resposta da terceira pergunta, com base em 25 respostas	39
Figura 23 - Resposta da quarta pergunta, com base em 25 respostas	40
Figura 24 Diagrama de Classes	53
Figura 25 Diagrama de Atividades	54
Figura 26 Diagrama de Atividades Criação de Anamnese/Evoluções/Devolutivas	55

LISTA DE TABELAS

Tabela 1 - Profissionais que participaram da pesquisa de aceitabilidade da proposta	12
Tabela 2 - Requisitos funcionais. Grau de prioridade: alta, média, baixa.....	26
Tabela 3 - Entidade Usuário	56
Tabela 4 - Entidade Terapeuta.....	56
Tabela 5 - Entidade Paciente	57
Tabela 6 - Entidade Anexo	57
Tabela 7 - Entidade Terapia	58
Tabela 8 - Entidade Especialidade	58
Tabela 9 - Entidade Paciente_Terapia.....	58
Tabela 10 - Entidade Paciente_Formulário	58
Tabela 11 - Entidade Formulário_Terapia	59
Tabela 12 - Entidade Formulário_Pergunta	59
Tabela 13 - Entidade Formulário_Pergunta_Escala.....	60
Tabela 14 - Entidade Formulário_Pergunta_Opção.....	60
Tabela 15 - Entidade Formulário_Pergunta_RespostaMTE	60
Tabela 16 - Entidade Formulário_Pergunta_RespostaTXT	61

1 INTRODUÇÃO

O Ambiente *web*, como mostrado hoje, era um lugar inanimado em meados dos anos 1990. Quando tudo era baseado apenas em *Hypertext Markup Language* – HTML, sem muitos recursos gráficos e dinamismo, tampouco qualquer animação em suas páginas.

O mercado atual tornou-se cada vez mais exigente e dependente quanto ao software que lhe é disponibilizado, e um dos fatores que promove este aumento de exigência é a evolução das tecnologias. A importância do software, em vários setores do mercado, torna-se fator crítico para que o investimento em desenvolvimento de aplicações tenha adquirido proporções cada vez maiores, principalmente no que diz respeito à qualidade de produto (FUGGETTA, 2000).

Os investimentos proporcionaram outros horizontes para o mundo da *Internet*. Não só computadores com altas capacidades de processamentos, mas infraestrutura na troca de dados de um canto do mundo a outro. Dessa forma, surgiram outros nichos que absorveram essa nova proposta de comércio na comunicação, produção de conteúdo, jogos, etc.

Apesar desta crescente exigência, muito software ainda é produzido com atrasos e consumo de mais recursos do que o planejado inicialmente, o que muitas vezes provoca o cancelamento do projeto durante o seu desenvolvimento ou leva a produção de um produto final com qualidade menor que a esperada (STANDISH GROUP, 2009).

Quanto mais a *Internet* proporcionava recursos e ferramentas, mais os usuários exigiam. Neste sentido necessitou-se de um tipo de ferramenta a qual fosse capaz de armazenar dados e processá-los de forma a ter regras de negócios, e que tivesse dinamismo ao ponto de se adequar a diferentes públicos e situações.

Nesse momento surge o conceito de aplicação *web*, ou *WebApp*, que consiste em um conjunto de páginas Web que interagem com o visitante, provendo, armazenando e processando informações. Exemplos são *websites* informativos, interativos, aplicações, transacionais ou baseadas em fluxos que executam na *web*, ambientes colaborativos de trabalho, comunidades online e portais (GINIGE e MURUGESAN, 2001).

Pensou-se em aplicativos para a *web*, recém-nascida, e que proporcionasse um atrativo a mais: a possibilidade de conectividade mundial. Os *sites* que surgiram após a entrada dos *WebApps*, são verdadeiros aplicativos para gerenciamento de conteúdos, conhecidos como *Content Management System* – CMS. (CONALLEN, 2003)

Logo, este cenário tornou-se muito bem estruturado, e, assim, diversos aplicativos

foram inseridos nesse contexto. Facilidade de Base de Dados estarem em um servidor disponibilizado na *web*, incentivou também o desenvolvimento para esta plataforma. *MySQL*, *Postgres*, *SQLite*, *Mongo* e diversos outros, desde relacionais até *NoSql*'s.

1.1 Motivação

Há muitas clínicas, hospitais e centros terapêuticos espalhados pelo mundo todo. Porém nem todas têm acesso há um sistema especializado, customizado e/ou customizável que os auxilie no monitoramento e acompanhamento do desenvolvimento dos seus pacientes e suas terapias empregadas.

Hoje o que se espera em muitos lugares são automatização de processos, assim como agilidade e eficiência nos serviços prestados. Nesse sentido, surgiu a necessidade de se ter um Software que os auxiliassem no desenvolvimento de suas atividades. Algo que seja maleável e intuitivo de ser manuseado.

1.2 Justificativa

Há muitas clínicas, hospitais e centros terapêuticos espalhados pelo mundo todo. Porém, nem todas têm acesso a um sistema especializado, customizado, que os auxilie no monitoramento e no acompanhamento do desenvolvimento dos seus pacientes e suas terapias empregadas.

Desde então, e durante muito tempo, a automação foi basicamente empregada na substituição de ações repetitivas, puramente mecânicas, mas esse tipo simples de utilização foi superado e, no fim do século passado, os robôs passaram a ser empregados em tarefas mais complexas e/ou sofisticadas, como por exemplo, a realização de delicadas cirurgias no cérebro de pacientes humanos, performances musicais, soldagem de partes e pintura final de carros produzidos pela indústria automobilística (GATES, 2007).

Hoje, o que se espera em muitos lugares são automatização de processos, assim como agilidade e eficiência nos serviços prestados. Nesse sentido, surgiu a necessidade de se ter *software* que os auxiliasse no desenvolvimento de suas atividades. Algo que seja maleável e intuitivo de ser manuseado.

Observando algumas clínicas aqui em Belém do Pará, pode-se notar que um *software* que servisse de apoio para o desenvolvimento de seus trabalhos iria trazer mais eficiência nas

terapias. Poderá observar quando e qual terapia está dando certo para dado paciente, e gerar devolutivas (relatórios periódicos sobre o desenvolvimento do paciente) mais rápidas, claras e objetivas, com automatização de resultados por meio de quantificadores e qualificadores predeterminados.

A precisão nas informações é um fator fundamental na colocação de sistemas clínicos eficientes. Profissionais da área de saúde têm o acesso a informações de pacientes que, até um tempo atrás, era algo mecânico. Dessa forma, a automatização dos terapeutas equilibra a qualidade do atendimento entre a terapia e o paciente.

Uma pesquisa foi realizada com alguns profissionais no intuito de verificar a aceitabilidade da proposta, bem como o impacto nas atividades profissionais destes (Tabela 1). Os gráficos com resultados dessa pesquisa encontram-se na subseção 5.3.

Tabela 1 - Profissionais que participaram da pesquisa de aceitabilidade da proposta

Profissional	Quantidade
Analista de Sistemas	1
Assistente Social	1
Assistente de RH	1
Nutricionista	4
Fonoaudióloga	1
Terapeuta Ocupacional	2
Médico	1
Psicóloga	2
Técnica de Enfermagem	2
Estudante	8

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um aplicativo WEB que auxilie, monitore e gerencie os dados médicos de uma clínica, hospital e ou centro terapêutico em relação às terapias trabalhadas, de uma forma customizável/ajustável às suas necessidades e dinâmicas, com simplicidade e intuitividade, além de usar do lúdico para deixar ainda mais

simples o manuseio do Software, bem como a satisfação deste como Sistema.

1.3.2 Objetivos Específicos

- Gerenciar os dados pessoais e terapêuticos dos pacientes, terapeutas e envolvidos nas terapias (estagiário, assistente, etc.) - Prontuário;
- Possibilitar a criação de terapias: customização dos formulários de dados a serem coletados, quantificação e qualificação nos seus formulários, e associação aos pacientes, com histórico de terapias usadas e dados de evolução;
- Mostrar o grau de satisfação do corpo clínico em função do uso na aplicação.

1.4 Organização do trabalho

O presente trabalho está organizado em seis capítulos, no primeiro capítulo é apresentado a motivação, bem como justificativa e objetivos, geral e específicos.

O capítulo 2 é a fundamentação teoria a qual serve de base e respaldo para o desenvolvimento do trabalho, e norteia as tecnologias e metodologia aplicada.

No capítulo 3, o qual é uma revisão da bibliografia, preocupa-se em apresentar software semelhante ao proposto nesta monografia.

Podemos definir o capítulo 4 como sendo a metodologia e todos os matérias utilizados para a elaboração do protótipo *Therapy*.

O capítulo 5 é o produto de tudo abordado nos capítulos anteriores, bem como a avaliação dos resultados obtidos.

No capítulo 6 encontra-se as considerações finais das sugestões para trabalhos futuros.

Por fim há três apêndices (A, B e C), os quais são, respectivamente, Documentos de Especificações: de Requisitos e de Análises, e também o Dicionário de Dados.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Engenharia de Software

Engenharia de software é metodologia de desenvolvimento e manutenção de sistemas modulares, com as seguintes características: processo (roteiro) dinâmico, integrado e inteligente de soluções tecnológicas; adequação aos requisitos funcionais do negócio do cliente e seus respectivos procedimentos pertinentes; efetivação de padrões de qualidade, produtividade e efetividade em suas atividades e produtos; fundamentação na Tecnologia da informação disponível, viável, oportuna e personalizada; planejamento e gestão de atividades, recursos, custos e datas (REZENDE, 2005).

Para Pressman (2011, p. 32):

Software consiste em instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados. Estruturas de dados que possibilitam aos programas manipular informações adequadamente. E informação descritivas, tanto na forma impressa como na virtual, descrevendo a operação e o uso dos programas.

Umas das principais características do software em relação a outros produtos, principalmente o hardware, o qual trabalha lado a lado, é o fato de o software não se “desgastar”, não sofrendo ações físicas do tempo e ambiente, diferentemente do hardware. Mas o software se deteriora, pois à medida que o tempo passa, ele sofre mudanças e manutenções, que, conseqüentemente, aumentam as chances de terem erros e, no caso, não há peças de reposição, como em hardware, sendo assim, a complexidade em relação à manutenção do software é superior ao do hardware. (PRESSMAN, 2011).

É possível citar sete categorias de software. São elas (PRESSMANN, 2011):

- Software de sistema: São programas para atender a outros programas, como, por exemplo, compiladores, editores de texto, além de aplicações de sistemas como drivers e softwares rede. Tem uma forte ligação com o hardware;
- Software de aplicação: Programas feitos sob medida, personalizados, usado para uma necessidade específica do negócio.

- Software científico/ de engenharia: Caracterizado pelo uso de algoritmos para lidar com número massivo de dados numéricos.
- Software embutido: Sistema residente de algum produto eletrônico para o usuário final, como, por exemplo, controle de painel de um carro ou micro-ondas.
- Software para linha de produtos: Tem a característica de abranger praticamente todo tipo de usuário, sendo para casos específicos como controle de estoque ou para planilhas eletrônicas do mercado de consumo em massa;
- Aplicações para web: Conhecidos também como webapps, é centralizado em redes, podendo ser apenas conjunto de arquivos textos até aplicações computacionais com recursos especializados para usuários finais;
- Software de inteligência artificial: Usado para solução de problemas complexos não passíveis de análises diretas ou computação, a partir de algoritmos não numéricos.

2.2 Sistemas de Informação e *WEBAPP*

De acordo com Turban *et al.* (2004), sistema de Informação é a expressão utilizada para descrever um sistema que seja automatizado, ou seja manual, abrangendo pessoas, máquinas, métodos organizados para coletar, processar, transmitir e disseminar dados que representam informação para o usuário.

Todo sistema de informação que manipula dados e gera informação, com ou sem o uso de recursos de tecnologia, pode ser genericamente considerado como um sistema de informação. (LAUDON e LAUDON, 2010)

Um sistema de informação, segundo Laudon e Laudon (2010), pode ser definido como um conjunto de componentes inter-relacionados trabalhando juntos para coletar,

recuperar, processar, armazenar e distribuir informações, com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em organizações.

Baseado nisto surge o termo *Progressive Web App (WEBAPP)*, denominado para contextualizar uma nova metodologia de desenvolvimento de software. São diferentes dos softwares tradicionais, pois abrangem uma mistura de desenvolvimento de sistemas, preza a constante evolução e manutenção de sua arquitetura para proporcionar o crescimento consistente.

Com o crescimento da internet e dos computadores na década de 1990, impulsionou-se o desenvolvimento de aplicações web, utilizadas para proporcionar repasse de informações de maneira rápida e segura.

Desta maneira, as aplicações *web* logo evoluíram de simples *sites* para grandes sistemas estruturados e o tempo trouxe a produção dos *WebApps*, que são desenvolvidas a partir de um sistema hipermídia, mas utilizando de recursos que tornem sua estrutura dinâmica e utilizando regras de negócio.

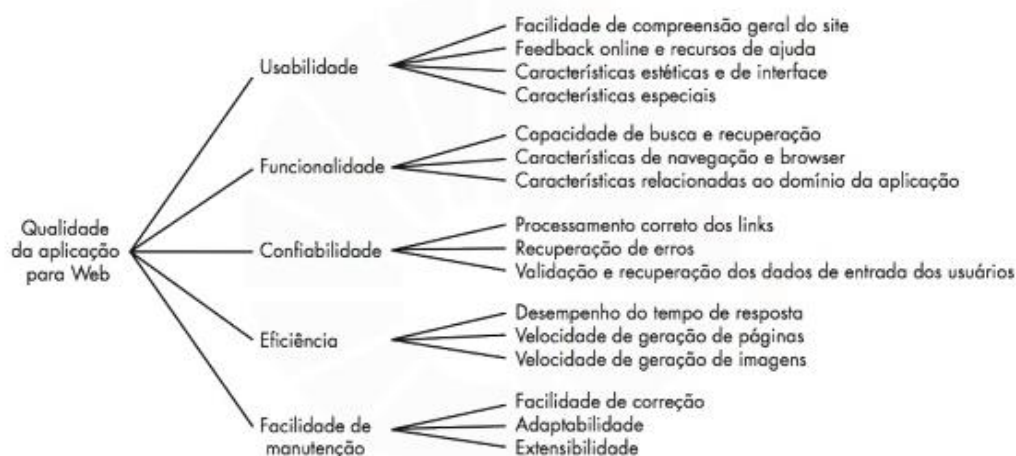
Conallen (2003) afirma que, certamente, se não houver nenhuma regra de negócio no servidor, o sistema não deverá ser visto como uma aplicação Web.

Projeto *WebApps* engloba atividades técnicas. A aparência do conteúdo é desenvolvida como parte do projeto gráfico, o leiaute de estética da interface como usuário é criado como parte do projeto de interface, e a estrutura técnica da *WebApps* é modelada como parte do projeto arquitetural e navegacional. (PRESSMAN, 2006).

2.3 Qualidades *WebApps*

Conforme Pressman (2006, p. 427), “Projeto é uma atividade da engenharia que leva a um produto de alta qualidade”. Nas aplicações os usuários têm uma perspectiva do que torna um *WebApps* bom. Alguns buscam o bom conteúdo, outros a qualidade em leiaute, velocidade na aplicação. Essa percepção está ligada diretamente no sucesso do *WebApp*. Na Figura 3 podemos, claramente, vislumbrar os conceitos em grupos e subgrupos aos quais Presman (2010) denota.

Qualidade



Fonte: Pressman (2010)

Figura 1 - Qualidade em WEBAPPs

2.4 Arquiteturas

2.4.1 Arquiteturas

A arquitetura do *software* possui um papel fundamental durante o seu desenvolvimento. É a partir da arquitetura que todos os envolvidos no desenvolvimento do projeto irão buscar o entendimento do que o *software* irá fazer como irá funcionar e como estará organizado. Portanto, a arquitetura MVC (*model-view-controller*) separa a aplicação em camadas, essas três camadas, conhecidas como modelo-visão-controlador, organizam e separam o sistema em responsabilidades, respectivas às suas camadas.

2.5 Usabilidade

A interação entre todos os agentes que compõem um sistema, juntamente com a interatividade e a compreensão bem-sucedida do usuário formam a usabilidade do *WebAPP*. Para uma boa usabilidade, o *WebApp* deve funcionar com uma velocidade considerável, independente da plataforma operacional que o usuário possua. Nielsen (1996) relata que o tempo de resposta de um sistema para que o usuário não perca o interesse é de no máximo dez segundos, mas em interfaces para Internet esse tempo pode ser estender um pouco mais.

O autor destaca também que a sobrevivência de um sistema depende de outras condições, perderá seu usuário se:

- For difícil de usar;
- Falhar em mostrar claramente o que uma companhia oferece e o que ele pode fazer no seu local;
- Ele se perder durante a navegação;
- A informação for difícil de ler ou não responder às suas perguntas.

2.6 Ferramentas e Tecnologias

Ao longo dos anos no ambiente *web*, algumas tecnologias surgiram e ganham seu lugar no mercado de trabalho. Cada vez mais se fala em eficiência, ganho de tempo, robustez e interatividade com o usuário (TAVARES, 2012) (HACKENHAAR, 2010).

Dentre as linguagens mais usadas no ambiente web estão:

- *Java (JSP, JSF, Primefaces);*
- *PHP (Zend, Laravel, CodeIgniter);*
- *ASP.NET;*
- *Python;*
- *Ruby on rails.*

Para o layout, interação com usuário:

- *HTML / CSS (Bootstrap);*
- *Javascript (jQuery, Bootstrap, jQuery.ui);*

Essas tecnologias são as mais utilizadas, hoje em dia, para desenvolvimento de sistemas e sites. Pois possuem uma infinidade de recursos já implementados que facilitam o desenvolvimento e permitem que o usuário interaja com o aplicativo de maneira mais eficaz. (RODRIGUES, 2016)

3 REVISÃO BIBLIOGRÁFICA

A revisão bibliográfica mostrada neste capítulo consistirá na apresentação de conceitos de diferentes autores e pesquisas sobre o que será utilizado no desenvolvimento do software em questão.

É possível observar que sistemas de gerenciamento médicos já se fazem presentes na maioria das clínicas nacionais, mas ainda existe uma grande parcela delas que realizam seu gerenciamento por meio de anotações em papel e/ou planilhas eletrônicas, fruto de uma sociedade sem automatização. Esses costumes podem funcionar positivamente por um tempo, mas com o aumento de clientes e de atendimentos este processo pode tornar-se um problema generalizado, levando a um grande acúmulo de registros digitais ou físico.

Esses dados poderiam ser mais bem manipulados se estivessem sendo gerenciados em um sistema próprio para essa finalidade, contribuindo para a diminuição digital ou física.

Como empreendimento na forma de clínica de saúde não é algo recente, já existem no mercado vários sistemas que buscam gerenciar várias atividades de uma clínica. A seguir serão apresentados alguns sistemas importantes no segmento de gerenciamento de clínicas de saúde.

3.1 Trabalhos Relacionados

Como o investimento na forma de clínica de saúde não é algo recente, já existem no mercado vários *WebApp* que buscam gerenciar as atividades de uma clínica. Nas subseções a seguir serão apresentados alguns sistemas importantes no segmento de gerenciamento de clínicas de saúde.

3.1.1 ProDoctor

ProDoctor é um software para gestão de consultórios e clínicas médicas desenvolvido pela *ProDoctor Software S/A*, e fornece aos usuários funções como: agendamento de consultas, prontuário eletrônico, faturamento, controle financeiro, controle de estoque, chat, relatórios e outras (PRODOCTOR SOFTWARE S/A, 2016).

O sistema pode realizar o agendamento de consultas e integrar o prontuário dos pacientes, como também gerar um histórico completo dos pacientes para futuras consultas. No que se refere a controle financeiro, possibilita realizar o lançamento de contas e faturamento no padrão TISS (Troca de Informações na Saúde Suplementar) que é estabelecido pela ANS

(Agência Nacional de Saúde Suplementar) para a as trocas de informações, com o objetivo de padronizar o envio eletrônico de informações administrativas (ANS, 2016). O sistema também possui um gerenciamento de estoque e da movimentação financeira.

O sistema não possui versão gratuita e é disponibilizado em duas versões:

- *ProDoctor Corp*: indicado para clínicas e pequenos hospitais;
- *ProDoctor Plus*: indicado para consultórios e pequenas clínicas

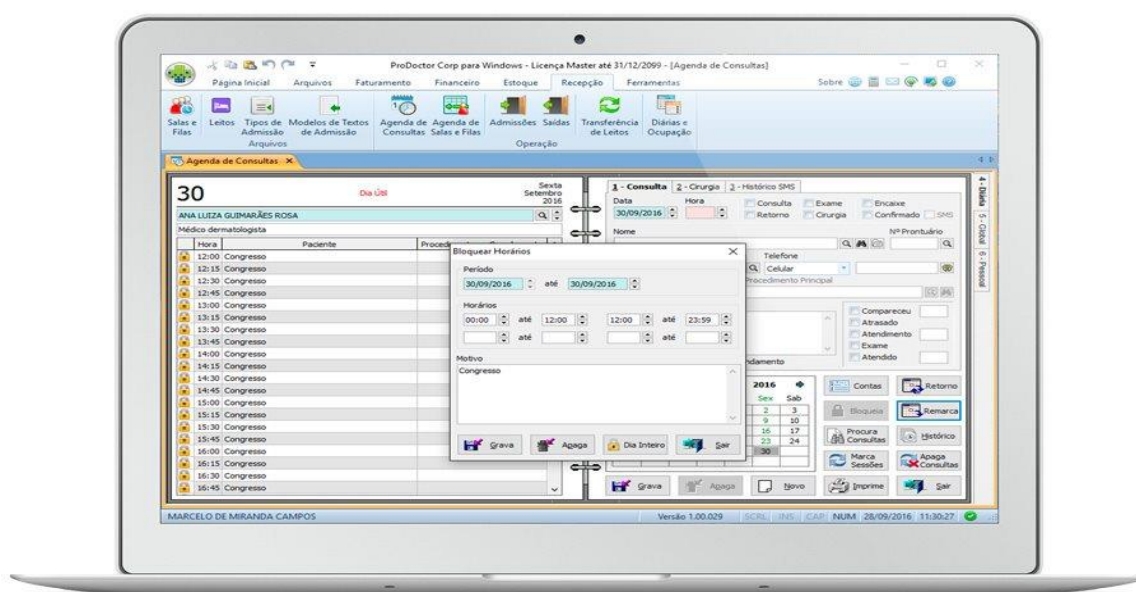


Figura 2 - Software ProDoctor. Fonte: Prodoctor Software S/A (2016).

O *ProDoctor* possui mais de 10.000 usuários por todo o Brasil, porém é mais voltado a área médica e totalmente dependente da instalação em sistemas operacionais por não ser baseado em tecnologia web. Na Figura 1 é apresentada uma tela de guia Agendamento de Consultas *ProDoctor* em sua versão desktop.

3.1.2 Clínica nas Nuvens

O sistema Clínica nas Nuvens é um sistema online para o controle de clínicas e consultórios, possuindo as seguintes principais funcionalidades: interação com pacientes, multiclinicas e multiusuários, galeria de imagens e documentos, pré-consulta, prontuário eletrônico, multiagendas, compatibilidade com dispositivos móveis, administração financeira completa, SMS e E-mails, backup automático dos dados, dentre outras funcionalidades

(CAMTWO SISTEMAS, 2015).

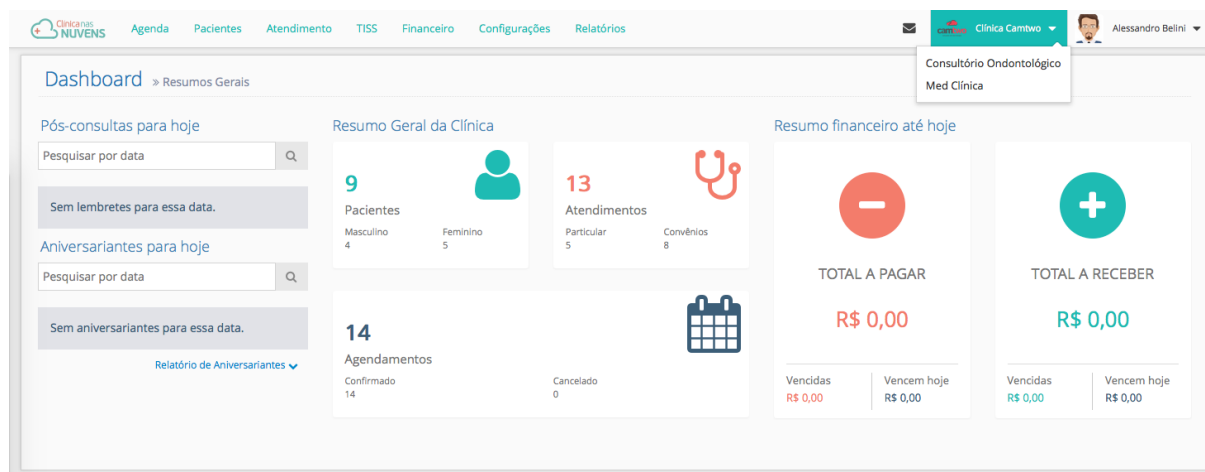


Figura 3 - Sistema Clínica nas Nuvens Fonte: Camtwo Sistemas (2015).

O sistema possui uma versão gratuita para testes por um período de 15 dias. Após este período, o usuário é direcionado para uma página de compra da licença. O sistema se comporta de forma mais simples, mas não possui uma área personalizada para cada tipo de profissional, havendo apenas a disponibilização da função multi clínicas para gerenciamento de várias clínicas.

Na Figura 2, é possível visualizar uma tela de gerenciamento de recursos disponíveis no sistema Clinica nas Nuvens.

Por fim, o Clínicas nas Nuvens é um software que pode executar em qualquer sistema operacional de computadores pessoais ou dispositivos móveis, por ser acessível via navegadores web.

4 MATERIAL E MÉTODOS

Este capítulo aborda todos os materiais utilizados para o desenvolvimento da ferramenta, e também da metodologia utilizada, com base nos padrões recomendados pela W3C, a qual é referência no ambiente web, e também com base em metodologias ágeis. Nos apêndices A e B estão descritos com detalhes os métodos e materiais usados no desenvolvimento, assim como diagramas de atividades, casos de usos, etc. No apêndice C está o dicionário de dados, referente a todas as entidades do sistema.

4.1 Modelo de Processo de Software

O modelo de processo de *software* adotado foi o *Kanban*, pois vislumbramos um meio de ter um fluxo rápido e contínuo de atividades em desenvolvimento e planejamento. Para este definimos um WIP (*Work in Progress*), que é a quantidade de tarefas em desenvolvimento, que atendesse a necessidade de ter o desenvolvimento com um pouco mais de rapidez e também nos permitisse revisar e testar os recursos implementados. (ARRUDA, 2012)

Com este modelo foi possível desenvolver o esperado com rapidez, e assim foi possível implementar o planejado e testar com tempo favorável. O interessante é que no meio do processo, abandonamos um caminho e adotamos outro, para assim atender o esperado pelos terapeutas. Isto só foi possível devido à dinâmica do processo *Kanban*.

Como ferramenta para auxiliar-nos, usamos a plataforma *Trello*, do grupo *Atlassian*. Esta plataforma dispõe de diversos recursos preparados para tornar mais eficiente e dinâmico o processo de desenvolvimento, ou organização das tarefas, assim potencializando o uso do *Kanban*.

4.2 Material

Para este projeto foi selecionadas algumas ferramentas, que foram escolhidas com base em: rapidez no desenvolvimento, fluidez no trabalho, acessibilidade por parte dos usuários, robustez e portabilidade – para este foi escolhido o ambiente *web*.

4.2.1 Ferramentas de Desenvolvimento

Esta seção aborda as ferramentas utilizadas no desenvolvimento e também corrobora

a escolha do padrão de projeto. Para tais, foram levantadas informações que permitissem escolher ferramentas e metodologias que conversassem entre si.

4.2.2 Para a codificação: *Sublime Text 3*

De acordo com o *site* de tecnologias DevMedia, um dos mais renomados para a área de TI, a IDE (*Integrated Development Environment* – ambiente de desenvolvimento integrado) *Sublime Text 3* é uma das ferramentas mais eficazes e eficientes para desenvolvimento de aplicações, em diversos ambientes, incluindo a *web*, e a linguagem PHP – tecnologias adotadas neste trabalho, pois ela possibilita ao desenvolvedor uma rapidez na escrita do código, e também integra vários recursos, entre eles: dicionário de dados da linguagem, ferramenta GIT para versionamento, *preview* de telas, etc.

4.2.3 Para o banco: SGBD *MySQL*

A escolha do Sistema de Gerenciamento de Banco de Dados - SGBD para essa aplicação foi baseada em alguns aspectos, entre eles: consistência, rapidez, robustez e facilidade de disponibilidade do banco na rede, além de ter se tornado a principal opção de banco de dados para aplicativos *WEB*. [ORACLE, 2017] Com base nisso, o *MySQL* mostrou-se mais adequado a este trabalho, e para tanto mostrou-se muito eficiente nos testes realizados.

4.2.4 Para a modelagem do banco: *MySQL Workbench*

A escolha desta ferramenta para a modelagem foi feita considerando vários aspectos do processo de desenvolvimento e também da documentação do sistema. O *MySQL Workbench* é um software de design de banco de dados, o qual integra SQL no desenvolvimento e sincroniza o banco com o modelo. Ele é a ferramenta que dá continuidade a outra ferramenta da mesma empresa, fabFORCE.net, o DBDesigner 4.

O uso deste software nos possibilitou um rápido e eficaz desenvolvimento do banco de dados, assim como sua manutenção. Esta ferramenta enquadrou-se muito bem à metodologia de processos escolhida, por sua maleabilidade e também consistência.

4.2.5 Padrão de Projeto

Padrões de projetos são métodos de se desenvolverem projetos, e estes já foram testados e são relativos a certos tipos de problemas. No caso desta aplicação escolhemos a

arquitetura *Model-view-controller* – MVC, modelo-visualização-controlador, a qual divide a aplicação em camadas e permite uma melhor estruturação, manutenção e organização do código:

- *Model* – camada de modelos é responsável pelos dados e interface do sistema com o banco de dados. Nesta camada também está concentrada a regra de negócios do sistema - RN;
- *View* – na camada de visualização é trabalhado todo o leiaute do sistema. Nela há a interação do usuário com a aplicação;
- *Controller* – o controlador é responsável por receber as requisições vindas da camada de visualização e fazer a interface com a camada de modelo.

No apêndice B são apresentados alguns diagramas de atividades, todos feitos no *software ASTAH*. Assim como os diagramas de caso de uso neste mesmo capítulo.

O *ASTAH* é um programa de modelagem de softwares, disponibiliza várias opções de ferramentas de criação de diagramas: classes, casos de uso, máquina de estado, atividades, pacotes, entre outros.

4.2.6 Linguagem de Programação

Como o ambiente escolhido foi a *web*, decidiu-se então usar a linguagem PHP. Esta linguagem é muito eficiente em sistemas *web*, tem perfeita integração com banco de dados *MySQL*, é escalável e tem alta portabilidade.

Em relação à robustez, o PHP proporciona *software* robusto e simples de ser implementado. A curva de aprendizado também é pequena, logo para implementação de funcionalidades mais avançadas, não há complicações ao programador, e também a manutenção passa a ser rápida.

Para uma maior eficiência no desenvolvimento, utilizaremos um *framework* de PHP chamado *CodeIgniter*, ele é orientado a objetos e também foi implementado utilizando a arquitetura *model-view-controller* – MVC, logo os sistemas desenvolvidos com esta API são mais organizados e mais simples de se dar manutenção, pois a aplicação é dividida em camadas.

Na figura 3, abaixo, está a representação gráfica da arquitetura do *CodeIgniter framework*, a qual consiste em um fluxo simples entre as camadas:

- *index.php* faz parte da camada visão, assim como *Routing*, *Security*, *Caching* e

View, essa camada é a própria apresentação e interação com o usuário, onde este faz requisições à camada *controller*.

- *Application Controller* é a camada de controle, onde recebe as requisições do usuário, feitas pela camada de visão (botões, *link*, pesquisa, etc.) e as processa redirecionando para o lugar correto, onde pode ser a camada de modelo, ou bibliotecas, etc.
- A camada de modelo, *Models*, é onde há a interface com o banco de dados, também onde concentra-se as regras de negócio do sistema. Apesar de haver *Libraries*, *Helper*, *Plugins* e *Scripts* no mesmo nível de *Models*, elas não são outras camadas, apenas interagem com a camada *controller* independentes.

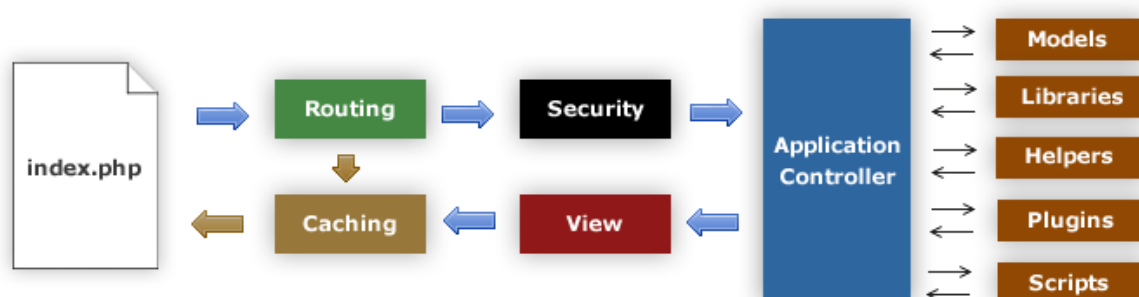


Figura 4 Fluxo de dados e requisições no *CodeIgniter* [codeigniter.com, 2017]

O *CodeIgniter* nos permite abstrair a implementação das operações mais básicas de um sistema, já implementando algumas funcionalidades, e também separando a aplicação em camadas. Mas para deixarmos mais rápida ainda, utilizamos uma API desenvolvida para este *framework* que abstrai todo o CRUD e deixa em sua responsabilidade, o *Grocerycrud*.

Com a utilização do *Grocerycrud* foi possível deixar toda a responsabilidade das operações básicas, envolvendo banco de dados, como cadastros, para que ele as resolva. Além disso, ele já cria as telas para que sejam executadas essas operações, precisando apenas de algumas configurações no *controller* para que faça a interface com as tabelas necessárias.

4.3 Descrição do Projeto

O ambiente da *WEB* tornou-se uma verdadeira plataforma de serviços de softwares ao longo dos anos. Por isso, escolhemos essa plataforma para desenvolvimento do *Therapy*, é

um ambiente de alta portabilidade, acessibilidade e com recursos abundantes a quem deseja criar um sistema interativo.

Para a filosofia do projeto, necessitaríamos de um sistema que pudesse ser acessado de vários lugares, e assim manipulá-lo em qualquer lugar. Como os requisitos para acesso a um sistema *web* são poucos e bem acessíveis, isto encaixou-se perfeitamente bem no que tange às expectativas.

4.3.1 Requisitos

Na tabela abaixo estão especificados os requisitos da aplicação, com suas devidas prioridades. Ela foi criada com a visão de se mapear as funcionalidades completas e complexas de um *software*, tentando abstrair ao máximo, uma vez que estão todas detalhadas no Apêndice A, Documento de Especificação de Requisitos:

Tabela 2 - Requisitos funcionais. Grau de prioridade: alta, média, baixa.

Nº	Requisito	Prioridade
1	O sistema deverá permitir acesso simultâneo	Alta
2	O sistema deverá ter um layout amigável e interativo	Alta
3	Deverá permitir cadastros de: usuários, pacientes, terapeutas e terapias, e fazer o gerenciamento desses dados.	Alta
4	O sistema deverá permitir associar um prontuário ao paciente	Alta
5	O sistema deverá ter um criador de formulários, para servir de apoio aos terapeutas na criação de suas anamneses, devolutivas, evoluções e fichas de avaliações.	Alta
6	Deverá permitir associar um formulário ao paciente	Alta
7	O Prontuário deverá ser uma tela que concentre todas as informações do paciente, referentes aos dados de saúde.	Alta
8	O prontuário deve permitir inserir imagens e/ou arquivos pdf de laudos médicos	Baixa
9	Deverá ter um sistema de <i>login</i> que diferencie: Usuário comum, Terapeuta.	Alta
10	Emitir relatórios automatizados referentes aos formulários criados (Anamnese, Evoluções, Devolutivas, etc.)	Media

4.3.2 Layout da Aplicação

Como o público alvo do *Therapy* são os terapeutas e assistentes responsáveis pelos dados do local em que serão utilizados, pensou-se então em um *design* mais lúdico, intuitivo e que permitisse uma tela bem agradável ao usuário.

Tendo isto em vista, foi escolhido utilizar uma API *WEB* de HTML, CSS e *JavaScript*, o *Bootstrap*. Esta API possui um vasto arsenal de recursos de *layouts* prontos: botões estilizados, janelas interativas, balões, painéis coloridos, notificações, tabelas, etc. O tema dessa biblioteca escolhido foi o *Gentellela Master*, com todos esses recursos estilizados e um *dashboard* todo pronto.

Após isso, integramos o *layout* ao sistema e então passou a dinamizar ainda mais o programa. Como o *Grocerycrud* já usa o *Bootstrap* em sua *API* nativamente, gerando todas as telas com o tema padrão do *Bootstrap*, a integração foi bem simples, com apenas alguns ajustes de *scripts* ele se comportou muito bem, como podemos ver na Figura 5:

Figura 5 Tela de Prontuário. Principal do *Therapy*.

4.4 Modelagem de software

A modelagem de software implica a formação de modelos gráficos que simbolizam os componentes de software utilizados e os seus inter-relacionamentos. Uma forma comum de

modelagem de programas orientados a objeto é por meio da linguagem unificada UML.

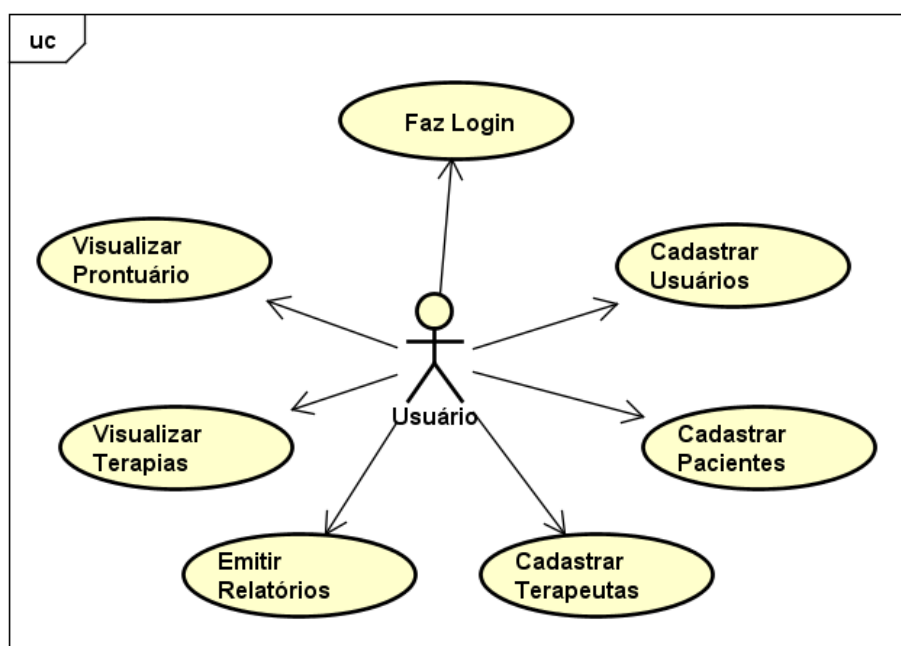
A UML (*Unified Modeling Language*) é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objetos. Sintetiza os principais métodos existentes, sendo considerada uma das linguagens mais expressivas para modelagem de sistemas orientados a objetos. Por meio de seus diagramas é possível representar sistemas de *software* em diversas perspectivas de visualização. Facilita a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema - gerentes, coordenadores, analistas, desenvolvedores - por apresentar um vocabulário de fácil entendimento (OMG, 2005a) (OMG, 2005a) (OMG, 2005b) (OMG, 2005c) (OMG, 2006).

A UML é dividida em dois tipos de diagramas, os diagramas estruturais e de comportamento. Dentre os diagramas do tipo estrutural, podemos citar os diagramas de classe, diagramas de pacotes, de componentes, de objetos, de utilização e de estrutura composta. Esses diagramas servem para visualização, especificação, construção e documentação dos aspectos estáticos do sistema, como classes, interfaces e componentes, ditos como estáveis. (OMG, 2006 apud DE SOUZA VARGAS, 2007).

4.4.1 Casos de Uso

Nesta seção apresentaremos os diagramas de caso de uso do usuário, terapeuta e paciente, os únicos agentes envolvidos no sistema, sendo assim as ações ficaram bem diretas.

Abaixo a Figura 5 apresenta o diagrama de caso de uso do agente usuário, o qual é um usuário geral do sistema: administrativo, RH, responsável do paciente. Este agente pode fazer diversas ações, dependendo do seu tipo de usuário. Em nível de banco, todos são usuários, e todos têm um registro na tabela de usuários, o que os diferem são: se for terapeuta, tem um registro relacionado ao identificador do terapeuta, se for paciente, tem um registro relacionado ao identificador do paciente. Assim, o usuário mostrado abaixo, é um usuário que não é Terapeuta e nem Paciente. Portanto pode: fazer *login*, emitir relatórios gerais, cadastros e visualizar os dados – pessoais e prontuário. (FURTADO & COSTA JÚNIOR, 2010)



powered by Astah

Figura 6 Diagrama de Casos de Uso do Usuário

No diagrama da Figura 6, apresenta as ações possíveis do agente Terapeuta. Diferente do agente Usuário, o Terapeuta pode criar terapias/anamnese/evoluções/devolutivas, formulários para essas terapias/anamnese/evoluções/devolutivas além de ser o agente responsável por preencher e responder esses formulários, montando então os resultados que serão visualizados por todos do sistema.

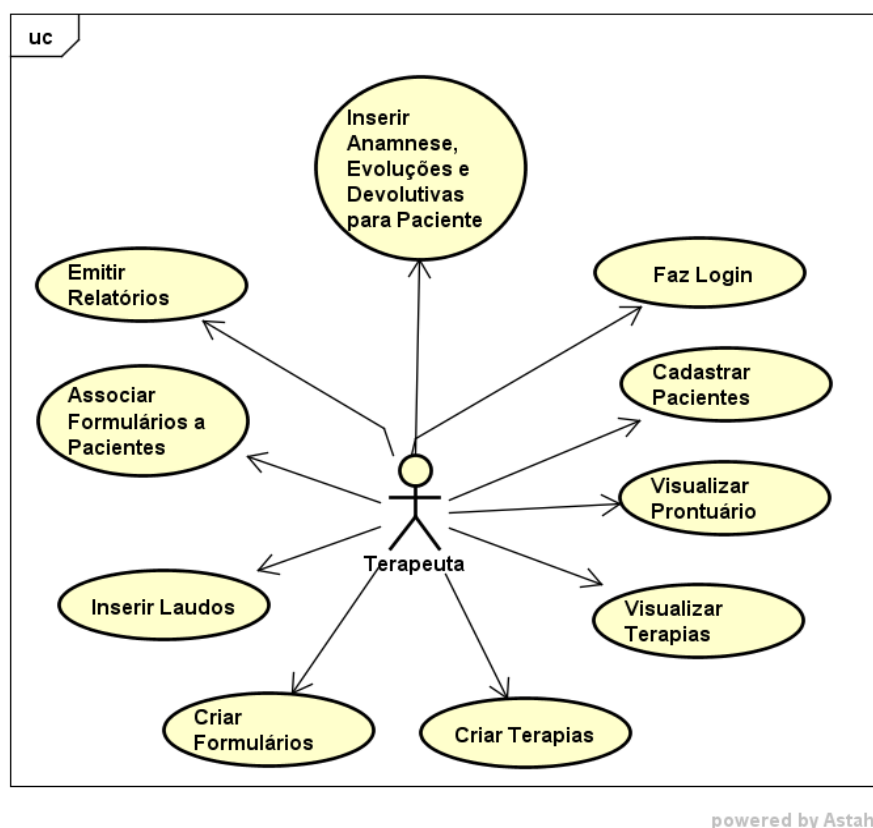


Figura 7 Diagrama de Casos de Uso do Terapeuta

O agente Paciente é um dos principais do sistema, porém ele é um agente que possui poucas ações, já que ele é o alvo dos dados e também o concentrador das informações, elas circulam ao redor do paciente. Na Figura 8 são representadas as ações do Paciente:

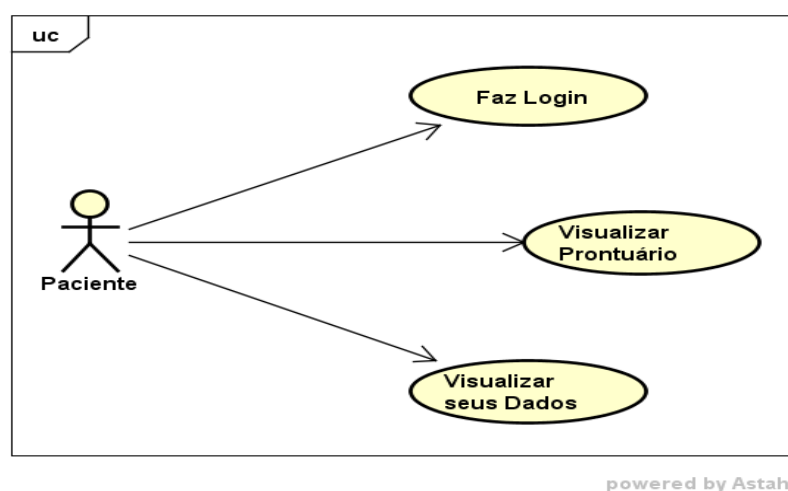


Figura 8 Diagrama de Casos de Uso do Paciente

4.4.2 Modelo Relacional

Como explicado, o modelo relacional foi desenvolvido utilizando a ferramenta

MySQL Workbench. Obedecendo o *SQL Pattern*, o banco foi modelado com boas práticas, tornando-o simples de ser manipulado. Abaixo segue a modelagem do banco:

A modelagem do banco foi pensada modularmente, separando os agentes do gerador de formulários e do prontuário. No entanto, como podemos visualizar na Figura 8, as entidades são todas relacionadas. E o paciente concentra todos os relacionamentos, pois ele se relaciona com os formulários, prontuário e todo paciente pode ser um usuário.

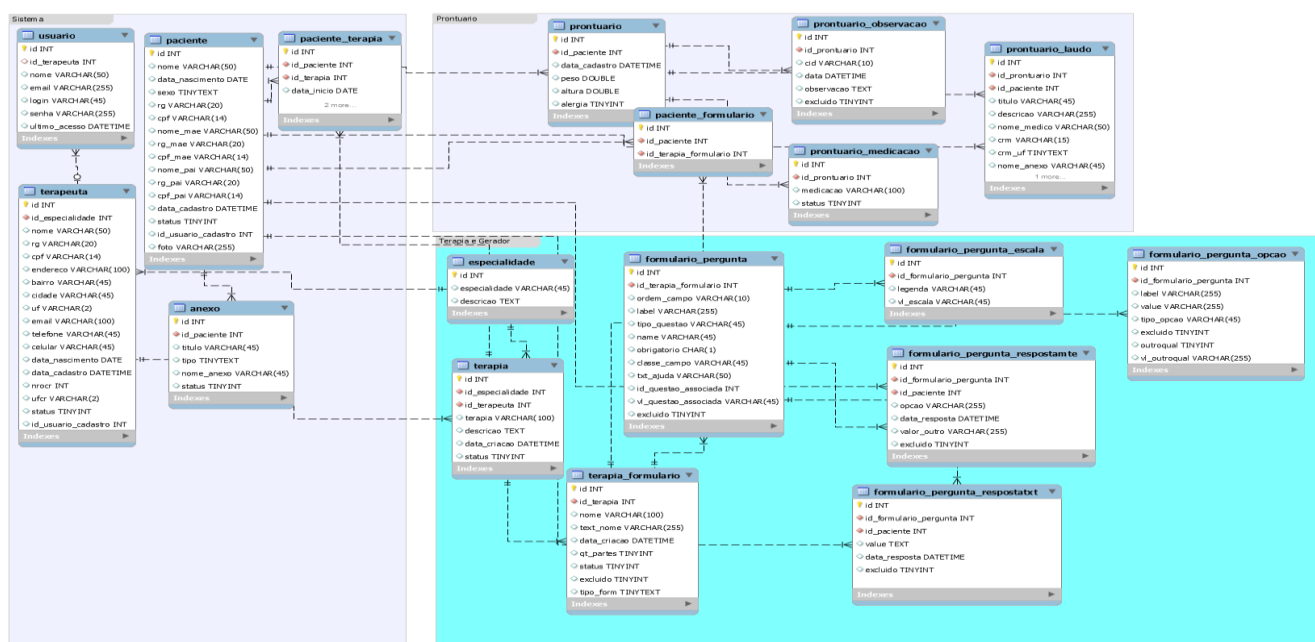


Figura 9 Modelagem do banco de dados, entidade relacional.

5 IMPLEMENTAÇÃO E AVALIAÇÃO DOS RESULTADOS

A implementação deste aplicativo *WEB* seguiu um roteiro simples: instalação do serviço *Wampserver*, o qual já vem com o servidor *Apache* embutido, mais o interpretador PHP e o banco de dados *MySQL*. Depois iniciou-se um projeto PHP *CodeIgniter*, com a API *Grocerycrud*, os configurando para os padrões do Brasil e integrando o sistema ao *layout* do *bootstrap*.

5.1 Telas do *Therapy*

A primeira tela do sistema (Figura 9) dá a opção de o usuário escolher seu tipo de *login*. Então é direcionado à tela de *login* (Figura 10).



Figura 10 Tela inicial. Escolhe-se o tipo de *login*

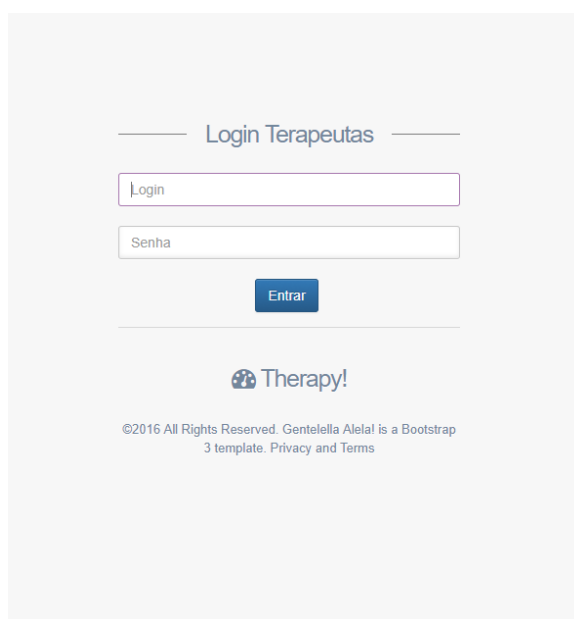
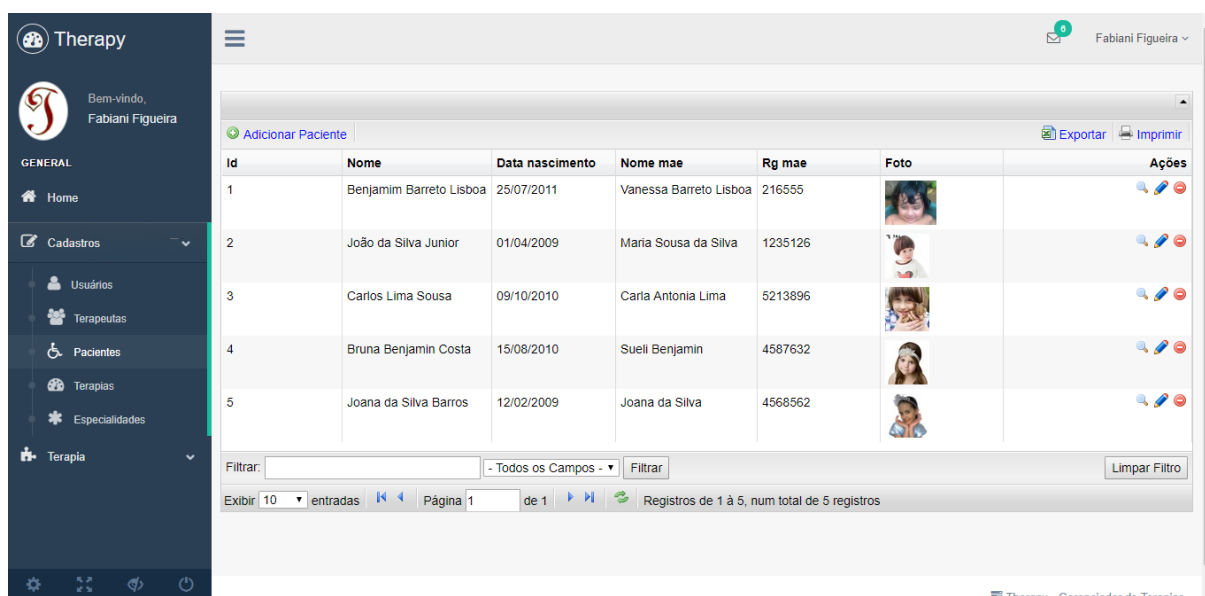


Figura 11 Tela de *Login*. Nesse caso, *login* dos terapeutas.

Após o *login*, o usuário é direcionado para a tela de apresentação do sistema, a qual será customizada para cada cliente. Nela há os menus de navegação para as outras partes do software. Entre elas, estão as telas de gerenciamento, a qual são *CRUD's* gerados pelo *API Grocerycrud*.

Pelo padrão da *API*, é criada uma tela de lista com os registros que foram inseridos no banco de dados, e também possui *link's* para as operações básicas (criação, edição, visualização e exclusão – Figura 12). A figura 11 mostra a tela de gerenciamento de pacientes:

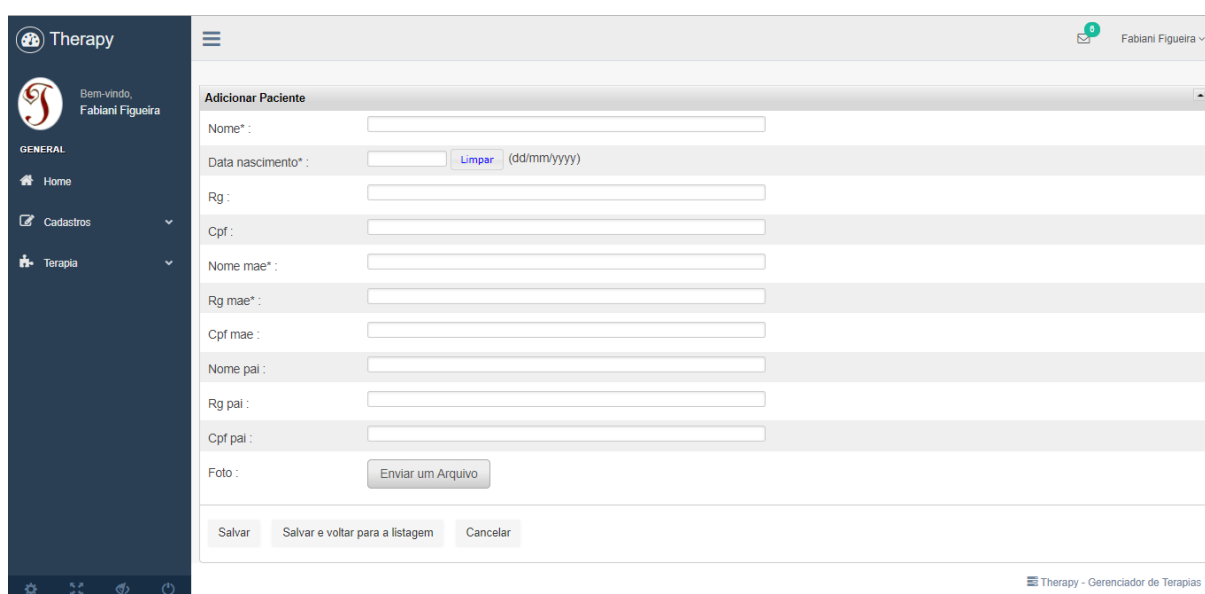


Id	Nome	Data nascimento	Nome mae	Rg mae	Foto	Ações
1	Benjamim Barreto Lisboa	25/07/2011	Vanessa Barreto Lisboa	216555		 
2	João da Silva Junior	01/04/2009	Maria Sousa da Silva	1235126		 
3	Carlos Lima Sousa	09/10/2010	Carla Antonia Lima	5213896		 
4	Bruna Benjamin Costa	15/08/2010	Suell Benjamin	4587632		 
5	Joana da Silva Barros	12/02/2009	Joana da Silva	4568562		 

Filtrar: - Todos os Campos -

Exibir 10 entradas   Página 1 de 1  Registros de 1 à 5, num total de 5 registros

Figura 12 Tela de gerenciamento de pacientes.



Adicionar Paciente

Nome* :

Data nascimento* : (dd/mm/yyyy)

Rg :

Cpf :

Nome mae* :

Rg mae* :

Cpf mae :

Nome pai :

Rg pai :

Cpf pai :

Foto :

Figura 13 Tela de inserção. Aqui cria-se um novo paciente.

Obedecendo aos requisitos, o sistema também tem um gerador de formulários, a qual serve como ferramenta de avaliação das terapias, evoluções e resultados dos pacientes. A vantagem é que o terapeuta pode criar diferentes formulários para uma mesma terapia, ou paciente e associá-lo aos pacientes. Abaixo seguem as telas de Listagem dos formulários criados, o qual tem um menu de ações que permite criar questões para o formulário. Também há uma tela de administração das questões do formulário, que permite uma prévia das questões já criadas, com suas devidas opções, se forem questões objetivas, matriz, etc.

The screenshot shows the 'Therapia' management interface. The sidebar on the left contains the following menu items: 'Home', 'Cadastros', 'Terapia' (selected), 'Prontuário', and 'Evolução'. The main content area is titled 'Terapia' and displays a table titled 'Listagem de Terapias'.

#	Terapia	Especialidade	Data	Ações
1	Aba	Terapia Ocupacional	28/08/17	[Grid Icon] [Document Icon] [Delete Icon]
2	Sensorial	Terapia Ocupacional	29/08/17	[Grid Icon] [Document Icon] [Delete Icon]
3	Anamnese	Terapia Ocupacional	30/08/17	[Grid Icon] [Document Icon] [Delete Icon]

Figura 14 Tela de terapias do terapeuta que está logado.

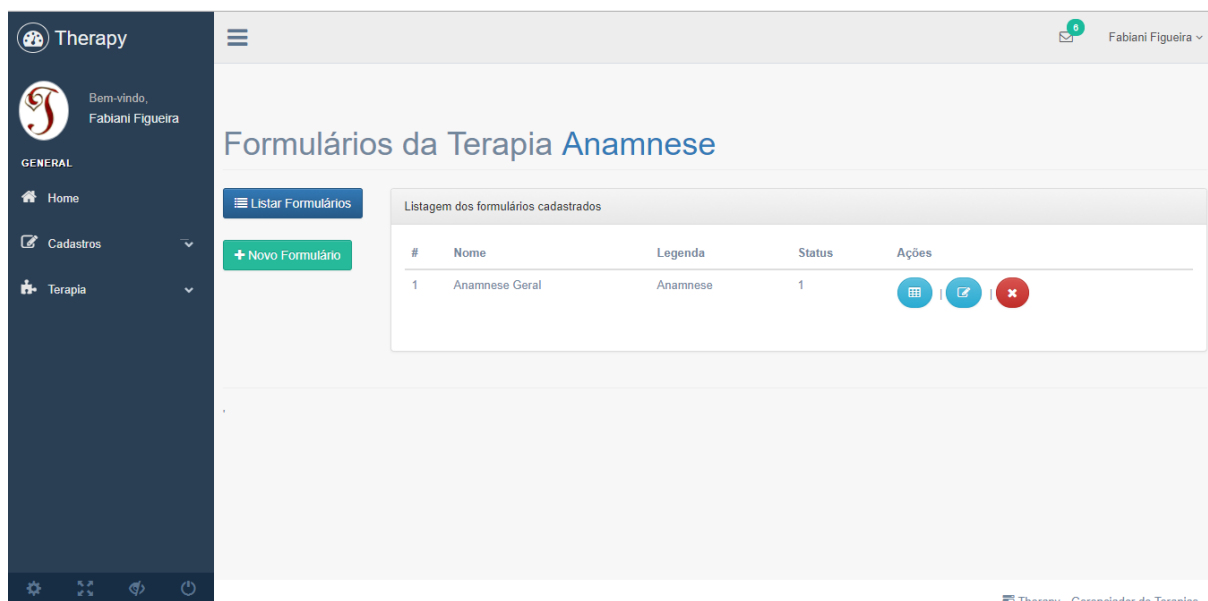


Figura 15 Tela dos formulários da terapia escolhida.

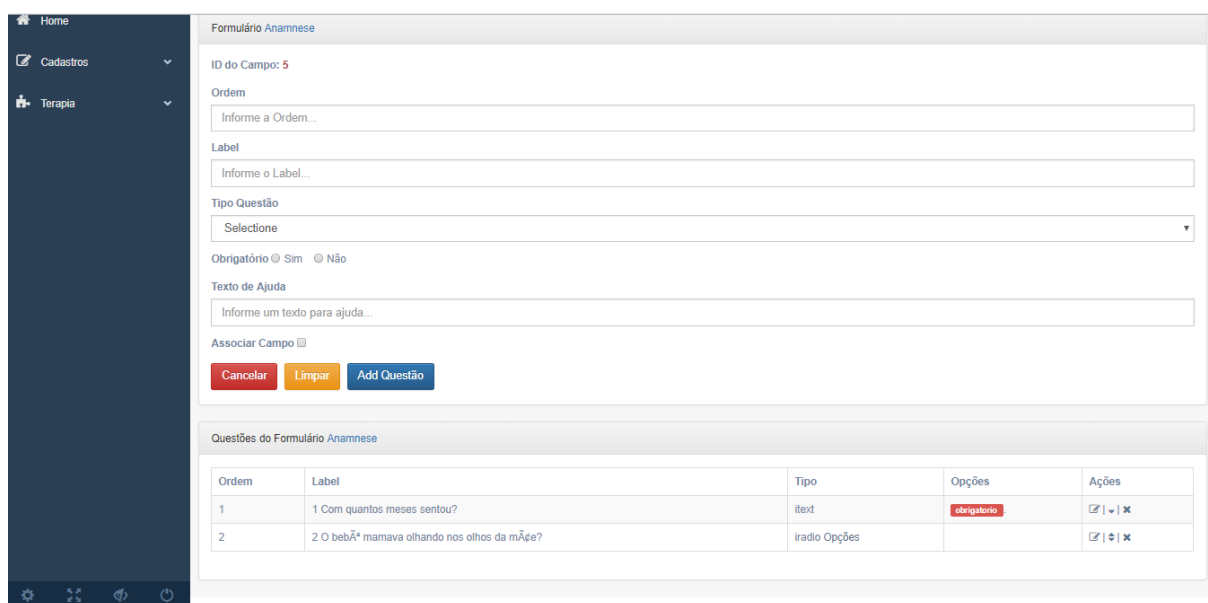


Figura 16 Tela das questões do formulário.

Na tela mostrada na Figura 15, pode-se ver que o terapeuta tem a possibilidade de criar diversos tipos de questões, além de poder visualizá-las após criadas, na ordem a qual foi definida em sua criação. Além disso, pode manipulá-las após sua criação: alterar ordem, nome, texto, etc.

Esses formulários terão impacto na tela de prontuário. Quando se quer criar uma anamnese, para que ela apareça na aba de anamneses, então deverá criar uma terapia do tipo

anamnese. Note que o nome terapia, neste caso, é apenas para concentrar em um ponto a criação dos formulários, mas na verdade, pode-se criar qualquer tipo de formulários para diversas situações.

As devolutivas e as evoluções serão geradas automaticamente e aparecerão nas abas respectivas. Para isso, o sistema deverá ser alimentado com frequência, para que sejam gerados os dados que expressem a realidade.

Nas Figuras 16 e 17, é possível pesquisar os pacientes por nome, os quais são mostrados em uma lista por meio de um *modal* com o resultado da pesquisa. Para acessar essa pesquisa, basta apenas pressionar a tecla F10, informando parte do nome, para um filtro melhor, ou apenas pressionando a tecla para todos os paciente cadastrados serem mostrados. Após selecionar o paciente desejado, este é apresentado no prontuário, com todos os seus dados informados nos campos.

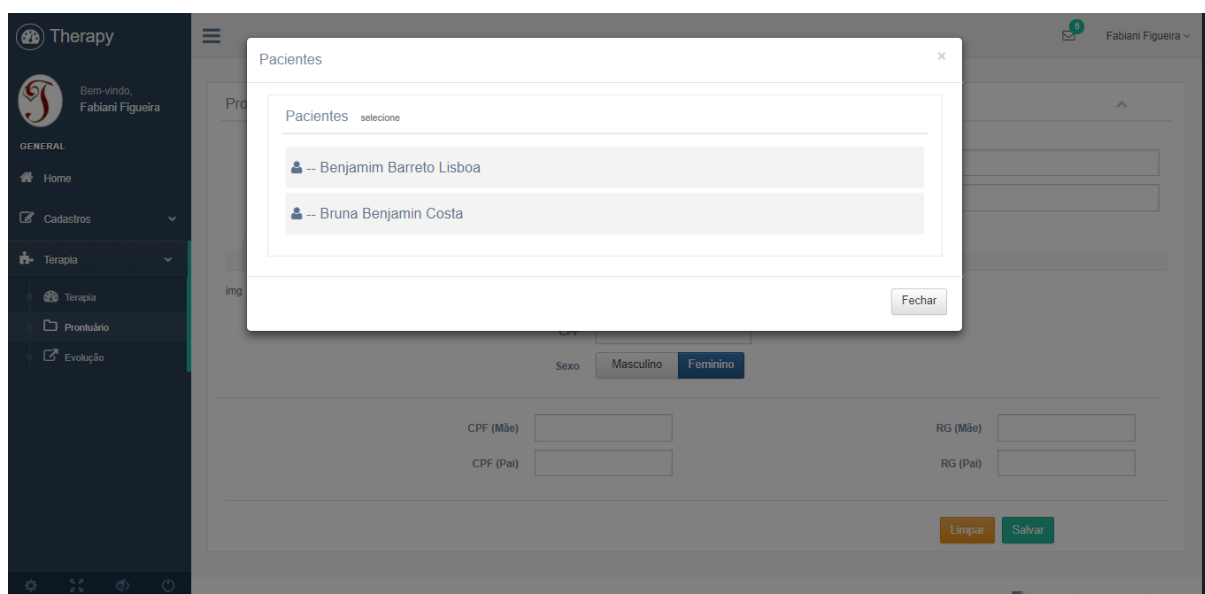


Figura 17 Tela de pesquisa de pacientes no prontuário.

Therapy
Bem-vindo,
Fabiani Figueira

GENERAL
Home
Cadastros
Terapia
Prontuário
Evolução

Prontuário todos os dados do paciente*

Nome: Benjamin Barreto Lisboa
Nascimento: 25/07/2011
Mãe: Vanessa Barreto Lisboa
Pai: Ramon Lisboa

Dados Pessoais | Laudo | Anamnese | Evoluções | Devolutivas

RG: 1245
CPF: 12345678910
Sexo: Masculino / Feminino

CPF (Mãe): 33319707272
RG (Mãe): 216555
CPF (Pai): 92592821287
RG (Pai): 5338063

Limpar Salvar

Figura 18 Tela de prontuário. Onde estão concentrados todos os dados.

Na tela de anamnese, mostrada na Figura 18, ele traz todas as anamneses do paciente, mesmo que seja de outros profissionais, e também traz todos os formulários de anamneses criados pelo terapeuta. Há também as aba de laudo, que é inserido o arquivo de laudo emitido pelo profissional qualificado. Também nas abas de devolutivas e evoluções, traz os dados nesta mesma dinâmica.

Therapy
Bem-vindo,
Fabiani Figueira

GENERAL
Home
Cadastros
Terapia
Prontuário
Evolução

Prontuário todos os dados do paciente*

Nome: Benjamin Barreto Lisboa
Nascimento: 25/07/2011
Mãe: Vanessa Barreto Lisboa
Pai: Ramon Lisboa

Dados Pessoais | Laudo | Anamnese | Evoluções | Devolutivas

Anamneses Criadas

Anamneses de Fabiani Figueira

#	Legenda	Criação
1	Anamnese	01/09/2017

Anamneses de Benjamin Barreto Lisboa

Anamnese Perguntas

1 Com quantos meses sentou?
5

2 O bebê mamava olhando nos olhos da mãe?
☒ Sim
☐ Não

Enviar

Figura 19 Tela da anamnese do paciente.

5.2 Testes realizados no desenvolvimento

Durante o desenvolvimento do sistema, foram realizados alguns testes de usuário: uma terapeuta ocupacional e uma pedagoga, o que nos norteou sobre criar uma tela única de prontuário, onde concentrasse as informações do paciente e fosse possível manipular os dados a partir desta tela.

Os ajustes foram realizados com base nas considerações desses dois usuários, e seguindo os padrões de projeto de sistemas *web*: *Design* e interface visual e navegação.

5.3 Resultado da Pesquisa de Aceitabilidade

1. Já usou alguma ferramenta virtual de gerenciamento de terapias?

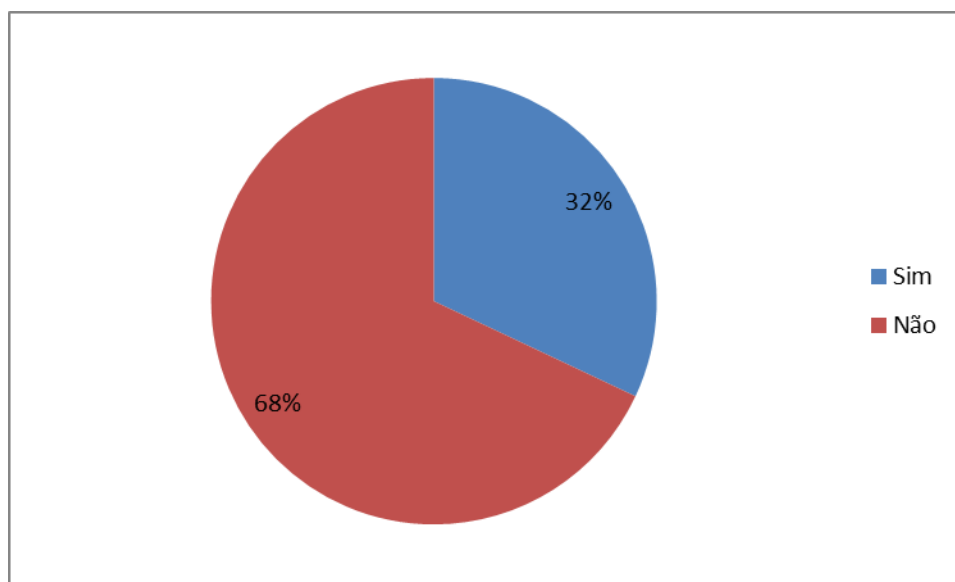


Figura 20 - Resposta da primeira pergunta, com base em 25 respostas

2. Qual sua opinião a respeito de ter uma ferramenta que faça gerenciamento de terapias, onde você possa criar seus métodos de avaliação e ela lhe dê os resultados e relatórios automaticamente?

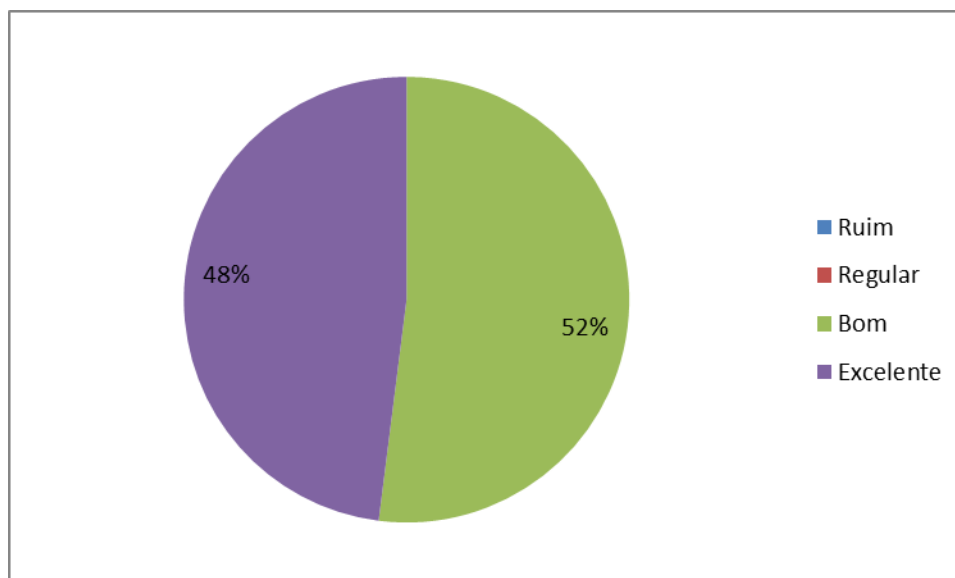


Figura 21 - Resposta da segunda pergunta, com base em 25 respostas

3. Qual sua opinião sobre a ferramenta acima, se ele tiver também um prontuário do paciente, associado às terapias, e onde possa registrar anamneses, evoluções, devolutivas, etc. ?

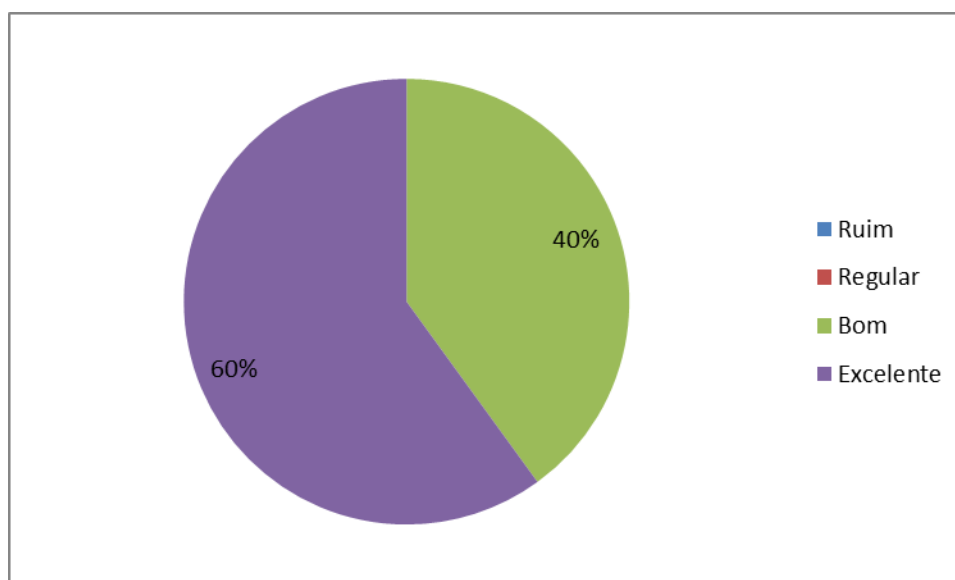


Figura 22 - Resposta da terceira pergunta, com base em 25 respostas

4. Em sua opinião, as clínicas e profissionais se beneficiariam de um software que tivesse essas características, ou as ferramentas que já existem atendem às necessidades?

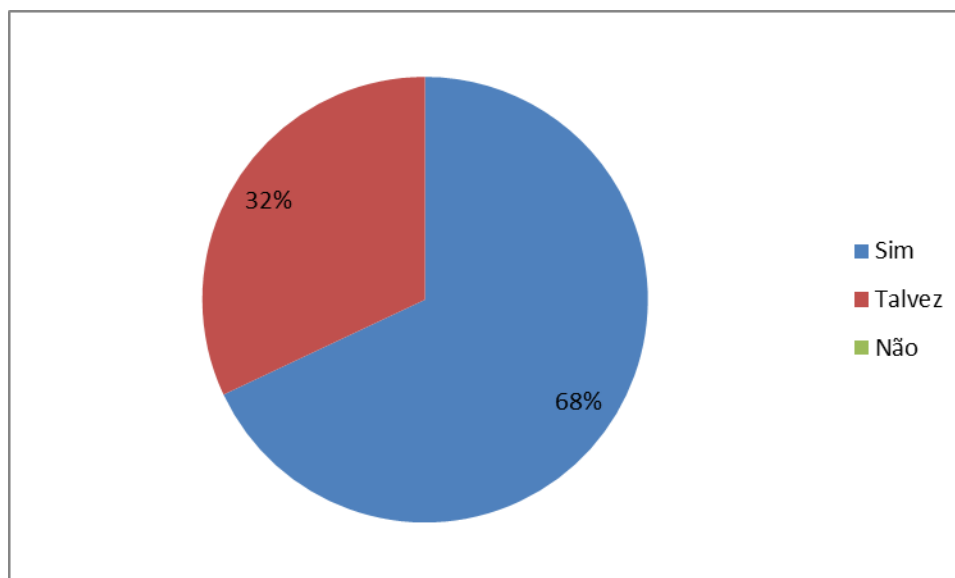


Figura 23 - Resposta da quarta pergunta, com base em 25 respostas

6 CONCLUSÃO

6.1 Considerações finais

Um dos objetivos deste trabalho é disponibilizar o *software* para testes de usabilidade para os usuários alvo: os terapeutas, mas não foi possível realizar tais testes.

Percebemos que um trabalho envolvendo desenvolvimento de *software* deve ter um tempo mais generoso, pois os testes de usabilidades, que são muito essenciais para o direcionamento do sistema, necessitam de um espaço maior para testes, e análise dos mesmos, então compreender qual o real impacto que o produto causou no usuário.

Contudo, apesar do software não ter sido concluído, em dado momento algumas terapeutas puderam acessar o protótipo por um determinado tempo. Isso nos possibilitou vislumbrar quais eram as expectativas por parte destas. Foi unanimidade na avaliação das profissionais, que o sistema poderá se tornar uma ferramenta que ajudará na tomada de decisão, bem como montar histórico do paciente, e/ou armazenar seus registros.

As tecnologias escolhidas e a metodologia aplicada tiveram sucesso em seu objetivo, sendo assim os pontos não contemplados e desenvolvidos, não se deram por limitações da metodologia ou recursos da tecnologia utilizada, mas por outros motivos (tempo hábil). Entretanto, ele já ajudaria uma clínica ou profissional da área – ver Apêndice D, pois já cria o histórico, as terapias, os formulários, as anamneses, etc. Contudo ele ainda não contempla a geração de relatórios.

6.2 Trabalhos Futuros

Para possíveis trabalhos futuros relacionados ao *Therapy*, sugerimos que alguns pontos precisam ser testados e implementados:

- Teste de usabilidade com o usuário;
- Implementação dos relatórios: anamnese, evoluções e devolutivas;

REFERÊNCIAS

AGILE MANIFESTO (2015). **Manifesto for Agile Software Development**. Disponível em: <<http://agilemanifesto.org>>. Acesso em 03 de abril de 2017.

ALMEIDA, R. I. M. **Utilização de padrões de projeto no desenvolvimento de aplicações WEB com PHP 5**. 2004. 41 f. Monografia (Bach. Em Sistemas das Informação) – Centro Universitário Luterano – Palmas, Tocantins, 2004.

AMBLER, S. W. **An Introduction to Process Patterns**. 1998. Disponível em: <<http://www.ambysoft.com/downloads/processPatterns.pdf>> Acesso em 07/08/2017

ARRUDA, L. V. (2012) **Desenvolvimento Ágil de Software: uma análise sintética a partir da metodologia Kanban**. Anais do VII CONNEPI – Congresso Norte Nordeste de Pesquisa e Inovação. Palmas, Tocantins.

As 10 linguagens de programação mais requisitadas pelo mercado. Disponível em: <<http://www.dicasdeprogramacao.com.br/as-10-linguagens-de-programacao-mais-requisitadas-pelo-mercado/>> Acesso em 25 ago. 2017

As 15 principais linguagens de programação do mundo! Disponível em <<https://becode.com.br/principais-linguagens-de-programacao/>> Acesso em 26 ago. 2017

CAMTWO SISTEMAS. Clínica nas Nuvens. Disponível em: <<http://suporte.clinicanasnuvens.com.br/>> Acesso em: 07 abr. 2017

CHAOS. **The Standish Group Report**, 2009.

CONALLEN, J. **Desenvolvendo aplicações Web com UML**. 2. Ed. Tradução de Altair Dias Caldas de Moraes, Cláudio Belleza Dias. Rio de Janeiro: Campus, 2003. 476p

CONFERENCE, 1990, **Seattle. Proceedings...** Seattle, 1990. p. 249-256.

DE SOUZA VARGAS, T. C. (2007). **A História de UML e seus Diagramas**. Disponível em: <https://projetos.inf.ufsc.br/arquivos_projetos/projeto_721/artigo.tcc.pdf>. Acesso em: 20 mar. 2017>/> Acesso em 22 ago. 2017

FUGGETTA, A. **Software Process: A Roadmap**. In: PROC. OF THE FUTURE OF SOFTWARE ENGINEERING, ICSE'2000, Limerick, Ireland, 2000.

FURTADO, A B. **Como escrever trabalhos de conclusão de curso (Graduação)**. 1ª edição. Belém, 2017, 105p.

FURTADO, A B; COSTA JR, J V. **Prática de Análise de Projeto de Sistemas**. 1ª edição. Belém, 2010, 488p.

GATES, B. **Humans and Robots – Future Perspectives**. Scientific American Magazine, n°.25, PP. 3-13, USA, 2007.

GINIGE, A.; MURUGESAN, S. **Web Engineering: An Introduction**. IEEE Multimedia, v.8, n. 1 (Janeiro-Março), p. 14-18, IEEE, 2001.

HACKENHAAR, J; ZANELLA, R; CARDOSO, T. "Um comparativo entre PHP e JSP: definindo a melhor aplicação para o desenvolvimento de projetos web." Revista iTEC 1 2010: p 32-36 .

IDE: **Introdução a melhor IDE para desenvolvimento**. Disponível em: <<http://www.devmedia.com.br/sublime-text-ide-introducao-a-melhor-ide-para-desenvolvimento/34117>> Acesso em 31 ago. 2017

LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação Gerenciais: Administrando a Empresa Digital**. 5 ed. São Paulo: Prentice Hall, 2004.562p.

MARTINS, C. **Padrões de Projeto em Aplicações Web: Desenvolvendo projetos web consistentes baseados em reuso de soluções**. Java Magazine. 2012.

NIELSEN, J. **Designing web usability**. [S.1] New Riders Publishing, 2000.432p

NIELSEN, J.; MOLICH, R. **Heuristic evaluation of user interfaces**. In: ACM CHI'90

OMG. OCL 2.0 **Specification**. 2005. Disponível em: < <http://www.omg.org/spec/>> Acesso em: 10 ago. 2017

OMG. **Unified Modeling Language: diagram interchange**. 2005. . Disponível em: < <http://www.omg.org/spec/>> Acesso em: 10 ago. 2017

OMG. **Unified Modeling Language: infrastructure**. 2006. . Disponível em: < <http://www.omg.org/spec/>> Acesso em: 10 ago. 2017

OMG. **Unified Modeling Language: superstructure**. 2005. . Disponível em: < <http://www.omg.org/spec/>> Acesso em: 10 ago. 2017

ORACLE. Disponível em: <<https://www.oracle.com/br/mysql/index.html>> Acesso em 31 ago. 2017

PRESSMAN, R. S **Engenharia de Software**. 7. ed. Porto Alegre: Pearson Makron Books. 2011, 771p.

PRODOCTOR SOFTWARE S/A. **ProDoctor**. Disponível em: <<https://prodoctor.net/>>. Acesso em: 07 abr. 2017> Acesso em 10 set. 2017

REZENDE, D.A. **Engenharia de software e sistemas de informação**. BRASPORT LIVROS E Multimídia Ltda., 2005. 313p.

RODRIGUES, E C; SIQUEIRA , A dos S. **"SISWEB SMS: uma ferramenta para melhoria da comunicação com grupos utilizando as tecnologias sms e e-mail- SISWEB SMS: a tool for improving communication with groups using sms and email technologies."** Multiverso: Revista Eletrônica do Campus Juiz de Fora-IF Sudeste MG 1.2 2016: p 208-218.

TAVARES, F. **"Desenvolvimento de Aplicações em PHP."** Biblioteca Software Livre. 1ª Ed. FCA. 2012. 438p.

TURBAN, E.; MCLEAN, E.; WETHERBE, J. **Tecnologia da informação para gestão.** Tradução de Renate Schinke. 3.ed. Porto Alegre: Bookman, 2004.

APÊNDICE A – Documento de Especificação de Requisitos

Esse primeiro documento mostrará os requisitos do sistema em questão. Sendo de suma importância, apresentará as funcionalidades da aplicação, extraídos após um processo de análise. Descritos em forma de casos de uso, refletirá em uma visão do problema a ser resolvido, os atores envolvidos com o desenvolvimento do sistema.

Ambiente do Sistema

O sistema permitirá ao terapeuta cadastrar as terapias, evoluções, formulários anamneses e terapeutas onde cada profissional pertencerá a uma determinada terapia cadastrada. O acesso do terapeuta será feito a partir de um nome de usuário e senha cadastrados pelo administrador.

O Terapeuta será responsável por cadastrar novos pacientes o cadastro do paciente terá os seguintes dados: nome, data de nascimento, identidade, cpf, nome da mãe, cpf da mãe, nome do pai, cpf pai, foto.

O cadastro da terapia pelo terapeuta deverá ter os seguintes dados: data, hora, e terapeuta que está realizando a terapia. O terapeuta ao atender o paciente deverá utilizar uma anamnese correspondente à sua área para auxiliar no atendimento. Na terapia o terapeuta deverá cadastrar também a evolução do paciente.

O Therapy é facilitara o gerenciamento de multiterapias, sendo uma base de informações que poderá ser compartilhada entre os diversos elementos envolvidos, terão acesso a dados estatísticos sobre as terapias e gerenciamento de cadastro de pacientes e consultas na clínica.

Todos os Profissionais terá acesso ao histórico de todos os pacientes cadastrados no sistema. Esse histórico terá os dados cadastrados pelo Terapeuta e as terapias realizadas com diagnósticos, orientações e sintomas do paciente.

O Therapy irá gerar dados estatísticos e gráficos relevantes aos terapeutas, como número de terapias realizadas de determinadas pacientes em um intervalo de tempo, por exemplo.

Objetivo do produto

O Therapy deverá auxiliar o atendimento buscando observar quando e qual terapia

está dando certo para certo paciente, e gerar devolutivas (relatórios periódicos sobre o desenvolvimento do paciente) mais rápidas, claras e objetivas, com automatização de resultados através de quantificadores e qualificadores pré-determinados.

Benefício do programa

O sistema contará com as seguintes funcionalidades presentes em sua composição.

- Cadastro de pessoa física (inclusão, alteração, consulta).
- Atualização nas terapias cadastradas (inclusão, alteração, consulta).
- Registro de atendimentos terapêuticos.
- Relatórios com históricos de terapias.
- Consulta ao histórico de terapias, com base em uma palavra – chave, nome do paciente, nome da terapia.
- Diagnostico para determinada terapia.

Modelos de casos de uso

Os atores principais utilizados nesse sistema.

- **Ator Usuário:** Usuário do sistema que utiliza a aplicação para realizar cadastro de terapias, usuários, pacientes, emitir relatórios, visualizar portuários.
- **Ator Terapeuta:** O terapeuta responsável cadastrar o paciente, visualizando seus prontuários e terapias correspondentes, criar suas terapias e formulários necessários, inserir laudos, associando formulários ao paciente.
- **Ator paciente:** Usuário do sistema que possui dados clínicos no sistema.

Caso de uso 01: Cadastro do paciente

Ator primário: Usuário

Precondições: o Usuário tem quer estar conectado no sistema.

Fluxo Principal:

1. O usuário faz o login no sistema.
2. O sistema exibe as operações que podem ser realizadas: cadastro, alteração e consulta de um paciente.
3. O usuário indica a opção desejada ou finaliza o caso de uso.

4. Se o usuário deseja continuar com a inclusão de cadastro de pacientes retorna ao passo 2, caso contrario o caso de uso termina.

Fluxo alternativo : Cadastro paciente.

- a. O usuário cadastra o paciente no sistema.
- b. O sistema exibe um formulário para ser preenchido com os dados pessoais do paciente (nome, data de nascimento, RG, CPF, nome da mãe , RG da mãe, CPF da mãe, nome do pai, CPF do pai, RG do pai, foto).
- c. O usuário digita os dados do paciente.
- d. O sistema verifica a validade das informações digitados, se os dados forem válidos, inclui os dados do paciente, caso contrario, o sistema exibe mensagem indicativa de erro, solicita a correção dos dados, e volta para o passo b).

Fluxo alternativo : Alteração dos dados dos Pacientes

- a. O usuário informa o nome do paciente que deseja alterar.
- b. O sistema exibe os dados pessoais cadastrados no sistema do paciente e solicita confirmação de alteração.
- c. Se usuário confirmar a alteração,
 - c.1. O sistema pede redigitação do(s) campo(s) que deseja alterar;
 - c.2. O usuário redigita os campos que deseja alterar;
 - c.3. Se os dados digitados são validos
 altera o registro do paciente
 senão
 o sistema exibe mensagem indicativa de erro, e retorna ao passo c.1. fimse.

Fluxo alternativo (4): Consulta dos dados de um Paciente

- a. O usuário informa o nome do paciente a ser consultado.
- b. O sistema exibe os dados pessoais cadastrados no sistema do paciente.
- c. Se o usuário confirma a visualização, o caso de uso e terminado, caso contrario retorna ao passo a).

Pós-condições: A operação solicitada foi executada no cadastro de pacientes.

Frequência de uso: Diário.

Caso de uso 02: Inclusão das terapias

Descrição: O terapeuta cadastra no sistema as terapias que serão usadas, conforme análises, observações e comportamentos para um determinado perfil de paciente.

Ator Primário: Terapeuta.

Precondições: O terapeuta foi identificado no sistema.

Fluxo Principal:

1. O terapeuta faz o login no sistema.
2. O sistema exibe as operações que podem ser realizadas: cadastro paciente, visualizar prontuário.
3. O terapeuta indica a opção desejada ou finaliza o caso de uso.
4. O farmacêutico informa a opção desejada: cadastro paciente, visualizar prontuário.
5. Se o terapeuta deseja continuar com a inclusão de cadastro retorna ao passo 2, caso contrario o caso de uso termina.

Fluxo alternativo (4): Cadastro das Terapias

- a. O terapeuta cadastra a terapia no sistema.
- b. O sistema exibe um formulário para ser preenchido com os dados do terapia (Ordem, label , tipo de questão, texto de ajuda).
- c. O terapeuta digita os dados da terapia.
- d. O sistema verifica a validade dos dados digitados, se os dados forem válidos, inclui os dados da terapia, caso contrario, o sistema exibe mensagem indicativa de erro, solicita a correção dos dados, e volta para o passo b).

Fluxo alternativo (4): Alteração dos dados da Terapia

- a. O terapeuta informa o nome da terapiaque deseja alterar.
- b. O sistema exibe os dados das terapias cadastradas no sistema e solicita confirmação de alteração.
- c. Se o terapeuta confirmar a alteração,
 - c.1. O sistema pede redigitação do(s) campo(s) que deseja alterar;

c.2. O terapeuta redigita os campos que deseja alterar;

c.3. Se os dados digitados são válidos

altera os dados dos medicamentos

senão

o sistema exibe mensagem indicativa de erro, e retorna ao passo c.1.

fimse

Fluxo alternativo : Consulta das terapias

a. O terapeuta informa o nome da terapia a ser consultado.

b. O sistema exibe os dados da terapia cadastrada no sistema.

c. Se o terapeuta confirma a visualização, o caso de uso é terminado, caso contrário retorna ao passo a).

Pós-condições: A operação solicitada foi executada no cadastro de terapias.

Prioridade de desenvolvimento: 1

Frequência de uso: Diário.

Caso de uso 03: Preenchimento do Prontuário Eletrônico

Descrição: O terapeuta realiza o atendimento do paciente, informando no prontuário eletrônico, as observações percebidas e seus possíveis diagnósticos, e a terapia que ele irá utilizar para o tratamento do paciente.

Ator Primário: Terapeuta.

Precondições: O Médico foi identificado no sistema.

Fluxo Principal:

1. O terapeuta faz o login no sistema.

2. O sistema exibe as operações que podem ser realizadas: tipo de atendimento, observações, prescrição, editar, salvar.

3. O terapeuta indica a opção desejada ou finaliza o caso de uso.

4. O terapeuta informa a opção desejada: tipo de atendimento, observações, prescrição, editar ou salvar.

5. Se o médico deseja continuar com a inclusão de cadastro de medicamentos retorna ao passo 2, caso contrário o caso de uso termina.

Fluxo alternativo (4): Atendimento do Paciente

- a. O terapeuta realiza o atendimento do paciente.
- b. O sistema exibe um formulário para ser preenchido com os sintomas do paciente (Tipo de atendimento, histórico de consultas, observações).
- c. O terapeuta faz o devido preenchimento, finaliza e é redirecionado para a tela de prontuário.
- d. O sistema verifica a validade dos dados digitados, se os dados forem válidos, abre a tela de prontuário, onde serão informados os seguintes dados: laudo, anamnese, evoluções, devolutivas
- e. O terapeuta terá a liberdade de escolher se irá continuar com a terapia ou realizar a troca, para logo em seguida salvar os dados digitados no prontuário eletrônico e finalizar a consulta.

Fluxo alternativo : Alteração do Prontuário do Paciente

- a. O terapeuta informa o nome do paciente que deseja alterar.
- b. O sistema exibe o prontuário do paciente cadastrado no sistema e solicita confirmação de alteração.
- c. Se o terapeuta confirmar a alteração,
 - c.1. O sistema pede redigitação do(s) campo(s) que deseja alterar;
 - c.2. O Terapeuta redigita os campos que deseja alterar;
 - c.3. Se os dados digitados são validos
 altera o prontuário do paciente senão
 o sistema exibe mensagem indicativa de erro, e retorna ao passo c.1.
 fimse.

Fluxo alternativo: Consulta do Prontuário do Paciente

- a. O terapeuta informa o nome do paciente a ser consultado.
- b. O sistema exibe os dados do paciente cadastrado no sistema.
- c. Se o terapeuta confirma a visualização, o caso de uso e terminado, caso contrario

retorna ao passo a)

Pós-condições: A operação solicitada foi executada no preenchimento do prontuário eletrônico.

Prioridade de desenvolvimento: 1

Frequência de uso: Diário.

Requisitos não funcionais

Portabilidade: O sistema deve favorecer facilidades para os usuários que o manipulam, sempre frisando as senhas de acesso. Todas as operações devem ser geradas logs de controle.

Desempenho: Todas as navegações no software devem ter tempo considerável para uma navegação rápida.

Interface: É de suma importância o sistema apresenta-se como uma ferramenta didática, pois será executado por usuários poucos especializado.

Segurança: O sistema deve favorecer total facilidade para a autenticação dos atores do sistema, com a digitação para a login e senha de acesso. A operação realizada no sistema deverá registrar log de quem executou a operação.

Restrições do software

O software deve roda em plataformas comuns no mercado, sendo necessária a certificação que a produção do software foi com códigos eficientes. O sistema deverá rodar em equipamentos a partir da família Pentium 3.

APÊNDICE B – Documento de Especificação de Análise

No documento de especificação de análise está toda a especificação resultante da análise de requisitos, o qual serviu de base para elaboração da análise em si. Alguns aspectos foram levados em consideração na elaboração da especificação de análise, identificados nos requisitos.

Através do devido apoio, a análise teve preocupações, principalmente, quanto ao correto desenvolvimento do programa, seguindo os padrões de projetos para sistemas *WEB*. Portanto, uma vez identificado o contexto, descobriu-se o problema referente à este contexto, no caso deste projeto: falta de uma eficiente organização das informações dos pacientes, não informatização das clínicas e terapeutas, dificuldade de se visualizar grande volume de dados. Para estes a solução apresentada: informatizar os atendimentos, tornando-os mais visualizáveis e ajudando na tomada de decisão, por parte dos terapeutas, com os dados resultantes do sistema. (MARTINS, 2012)

Também, seguindo o *Design Pattern*, o *software* foi desenvolvido obedecendo as três dimensões:

- *Design* e interface visual;
- Interação e navegação;
- Arquitetura de construção.

Diagrama de Pacotes

Como o diagrama de pacotes é geralmente utilizado em sistemas grandes, não foi necessário criá-lo neste projeto. Suas relações estão bem concentradas, de modo que poucos diagramas contemplam todo o funcionamento do sistema. (FURTADO, 2010)

Diagrama de Classes

Este modelo apresenta as classes utilizadas no *Therapy*, bem como seus atributos e relacionamentos. (FURTADO, 2010) Como o *Grocerycrud* já faz o mapeamento através de uma configuração simples na *Action* do *controller* que gera as telas de gerência, então ele mesmo mapeia os atributos e gera para cada um três métodos básicos: *add*, *edit*, *delete*, e o método de visualização o qual é apresentado em uma tabela de dados com acesso a todas as operações de manipulação dos dados. Abaixo o diagrama de classe:

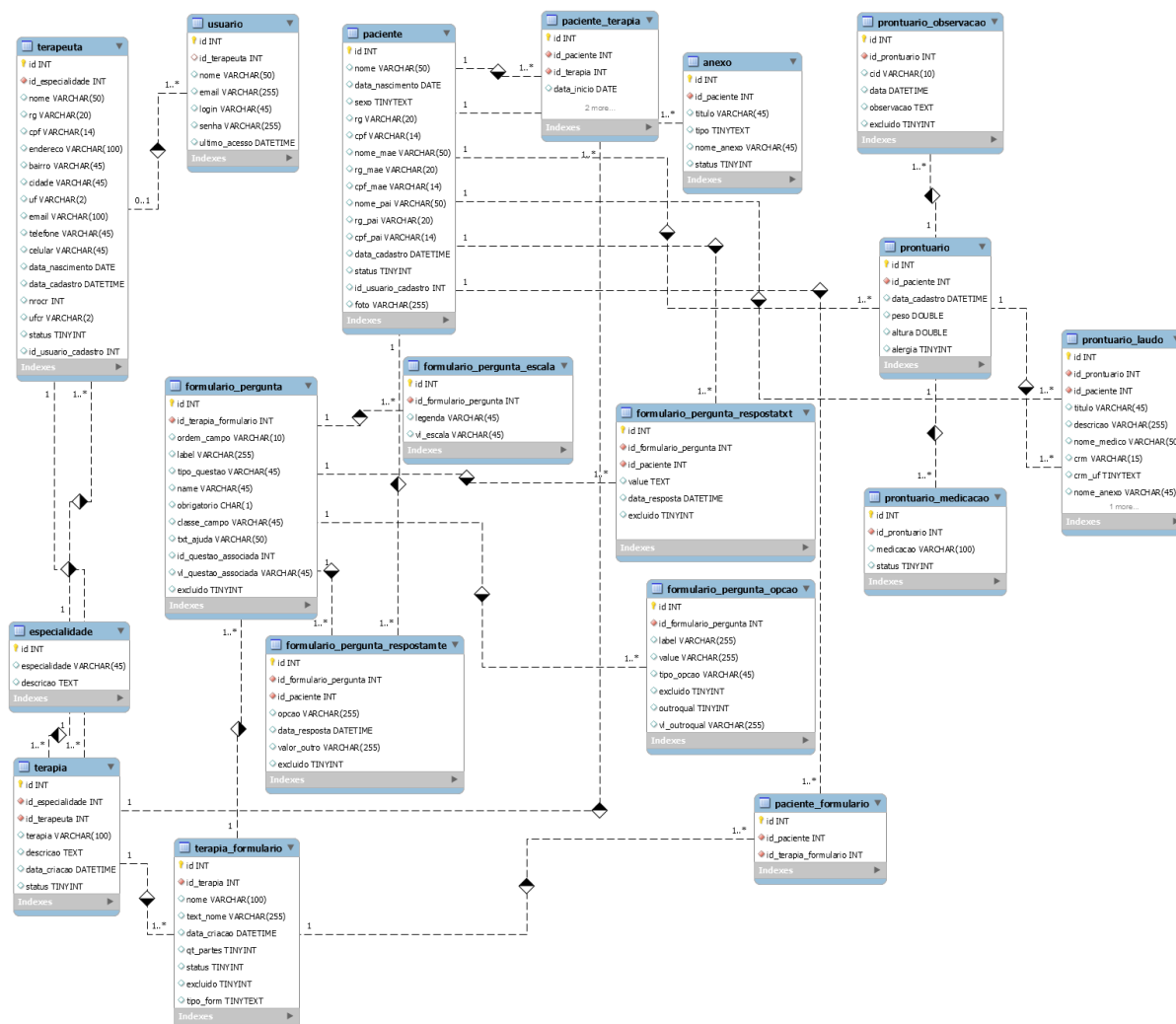
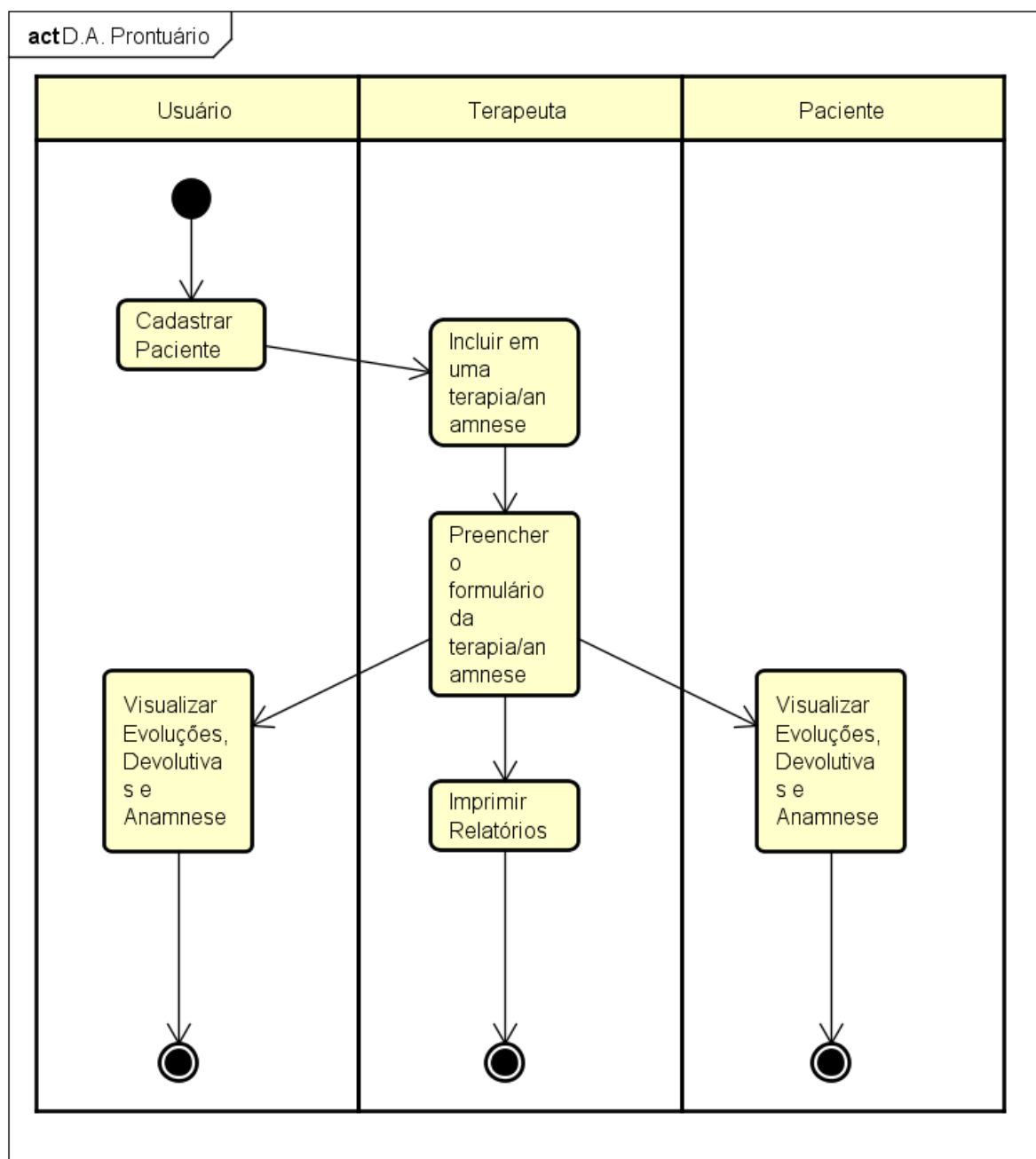


Figura 24 Diagrama de Classes

Diagrama de Atividades

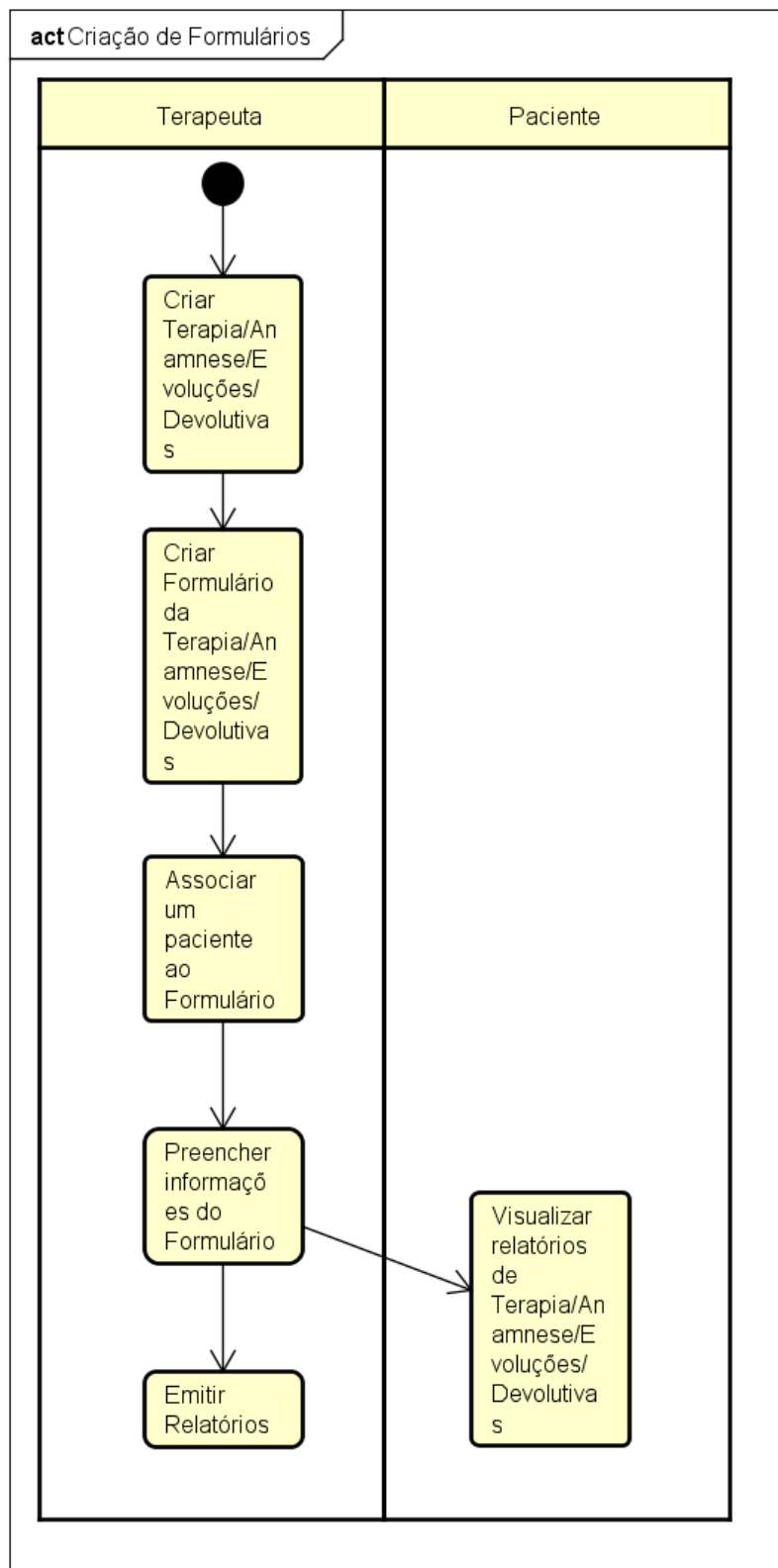
O diagrama abaixo mostra o fluxo principal do sistema. Esta representação é o sistema, o coração do sistema. Para um melhor esclarecimento, este documento define o principal propósito do *Therapy*: manter os dados do paciente concentrados, montar histórico dos dados médicos/terapêuticos, e ver a resposta do paciente em determinada terapia.



powered by Astah

Figura 25 Diagrama de Atividades

Para um melhor entendimento do propósito de dinamização da criação dos formulários associados às terapias e aos pacientes, abaixo está o diagrama de atividades relativo a esse recurso do *Therapy*, o qual possibilita uma rápida visualização dos dados das terapias, auxiliando na tomada de decisões referente à terapia empregada no paciente.



powered by Astah

Figura 26 Diagrama de Atividades Criação de Anamnese/Evolução/Devolutivas

APÊNDICE C – Dicionário de Dados

Tabela 3 - Entidade Usuário

USUARIO				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador do Terapeuta	id_terapeuta	int	Não	Sim
AutoExplicativo	nome	varchar(50)	Não	Não
AutoExplicativo	email	varchar(255)	Não	Não
AutoExplicativo	login	varchar(45)	Não	Não
AutoExplicativo	senha	varchar(255)	Não	Não
AutoExplicativo	ultimo_acesso	datetime	Não	Não

Tabela 4 - Entidade Terapeuta

TERAPEUTA				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador da Especialidade	id_especialidade	int	Não	Sim
Autoexplicativo	nome	varchar(100)	Não	Não
Autoexplicativo	rg	varchar(20)	Não	Não
Autoexplicativo	cpf	varchar(14)	Não	Não
Autoexplicativo	endereco	varchar(100)	Não	Não
Autoexplicativo	bairro	varchar(45)	Não	Não
Autoexplicativo	cidade	varchar(45)	Não	Não
Autoexplicativo	uf	varchar(2)	Não	Não
Autoexplicativo	email	varchar(100)	Não	Não
Autoexplicativo	telefone	varchar(45)	Não	Não
Autoexplicativo	celular	varchar(45)	Não	Não
Autoexplicativo	data_nascimento	date		
Autoexplicativo	data_cadastro	datetime		
Número de Registro no Conselho	nrocr	int(11)		
Estado do Registro do Conselho	ufcr	varchar()		
Autoexplicativo	status	tinyint(4)	Não	Não

Tabela 5 - Entidade Paciente

PACIENTE				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Autoexplicativo	nome	varchar(50)	Não	Não
Autoexplicativo	data_nascimento	date	Não	Não
Autoexplicativo	sexo	tinytext	Não	Não
Autoexplicativo	rg	varchar(20)	Não	Não
Autoexplicativo	cpf	varchar(14)	Não	Não
Autoexplicativo	nome_mae	varchar(50)	Não	Não
Autoexplicativo	rg_mae	varchar(20)	Não	Não
Autoexplicativo	cpf_mae	varchar(14)	Não	Não
Autoexplicativo	nome_pai	varchar(50)	Não	Não
Autoexplicativo	rg_pai	varchar(20)	Não	Não
Autoexplicativo	cpf_pai	varchar(14)	Não	Não
Autoexplicativo	data_cadastro	datetime	Não	Não
Autoexplicativo	status	tinyint(4)	Não	Não
Identificador do Usuário	id_usuario_cadastro	int(11)	Não	Sim
Nome do arquivo de foto do paciente	foto	varchar(255)	Não	Não

Tabela 6 - Entidade Anexo

ANEXO				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador do paciente	id_paciente	int	Não	Sim
Título nas telas	titulo	varchar(45)	Não	Não
Autoexplicativo	tipo	tinytext	Não	Não
Autoexplicativo	nome_anexo	varchar(45)	Não	Não
Autoexplicativo	status	tinyint(4)	Não	Não

Tabela 7 - Entidade Terapia

TERAPIA				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador da Especialidade	id_especialidade	int	Não	Sim
Identificador do Terapeuta	id_terapeuta	int	Não	Sim
Nome da terapia	terapia	varchar(100)	Não	Não
Autoexplicativo	descrição	text	Não	Não
Autoexplicativo	data_criacao	datetime	Não	Não
Autoexplicativo	status	tinyint(4)	Não	Não

Tabela 8 - Entidade Especialidade

ESPECIALIDADE				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Nome da especialidade	especialidade	varchar(45)	Não	Não
Autoexplicativo	descrição	text	Não	Não

Tabela 9 - Entidade Paciente_Terapia

PACIENTE_TERAPIA				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador do paciente	id_paciente	int	Não	Sim
Identificador da Terapia	id_terapia	int	Não	Sim
Autoexplicativo	data_criacao	datetime	Não	Não
Autoexplicativo	data_criacao	datetime	Não	Não
Autoexplicativo	status	tinyint(4)	Não	Não

Tabela 10 - Entidade Paciente_Formulário

PACIENTE_FORMULARIO				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador do paciente	id_paciente	int	Não	Sim
Identificador do formulário	id_terapia_formulario	int	Não	Sim

Tabela 11 - Entidade Formulário_Terapia

FORMULARIO_TERAPIA				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador da Terapia	id_terapia	int	Não	Sim
Nome Interno do Formulário	nome	varchar(100)	Não	Não
Nome Visível pro usuário	text_nome	varchar(255)	Não	Não
Autoexplicativo	data_criacao	datetime	Não	Não
Autoexplicativo	qt_partes	tinyint(4)	Não	Não
Autoexplicativo	status	tinyint(4)	Não	Não
Autoexplicativo	excluido	tinyint(4)	Não	Não
Tipo de Formulário ["Terapia", "Anamnese", "Evolução", "Devolutiva"]	tipo_form	tinytext(4)	Não	Não

Tabela 12 - Entidade Formulário_Pergunta

FORMULARIO_PERGUNTA				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador do Formulário	id_terapia_for mulario	int	Nã o	Sim
AutoExplicativo	ordem-campo	varchar(10)	Nã o	Não
Label da pergunta	label	varchar(25 5)	Nã o	Não
Tipo de questão ["text", "checkbox", "radio", "textarea", "password", "m atriz"]	tipo_questao	varchar(45)	Nã o	Não
Atributo name que fica no campo do formulário	name	varchar(45)	Nã o	Não
AutoExplicativo	obrigatorio	char(1)	Nã o	Não
Atributo class do html, caso haja uma classe estilizada diferente para a pergunta	classe_campo	varchar(45)	Nã o	Não
AutoExplicativo	txt_ajuda	varchar(50)	Nã o	Não
Identificador da questão associada	id_questao_as sociada	int	Nã o	Sim
Valor da questão associada	vl_questao_as sociada	varchar(45)	Nã o	Não
AutoExplicativo	excluido	tinyint(4)	Nã o	Não

Tabela 13 - Entidade Formulario_Pergunta_Escala

FORMULARIO_PERGUNTA_ESCALA				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador da Pergunta	id_formulario_pergunta	int	Não	Sim
AutoExplicativo	legenda	varchar(45)	Não	Não
Valor da Escala	vl_escala	varchar(45)	Não	Não

Tabela 14 - Entidade Formulario_Pergunta_Opcão

FORMULARIO_PERGUNTA_OPCAO				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador da Pergunta	id_formulario_pergunta	int	Não	Sim
Label da Opção	label	varchar(255)	Não	Não
Valor da Opção	value	varchar(255)	Não	Não
AutoExplicativo	tipo_opcao	varchar(45)	Não	Não
AutoExplicativo	excluido	tinyint(4)	Não	Não
Se tem outra opção informável por texto	outroqual	tinyint(4)	Não	Não
Valor da Outra opção	vl_outroqual	varchar(255)	Não	Não

Tabela 15 - Entidade Formulario_Pergunta_RespostaMTE

FORMULARIO_PERGUNTA_RESPOSTAMTE				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador da Pergunta	id_formulario_pergunta	int	Não	Sim
Identificador do Paciente	id_paciente	int	Não	Sim
Valor da Opção	opcao	varchar(255)	Não	Não
AutoExplicativo	data_resposta	datetime	Não	Não
valor do Outro informado da Opção	valor_outro	varchar(255)	Não	Não
AutoExplicativo	excluido	tinyint(4)	Não	Não

Tabela 16 - Entidade Formulario_Pergunta_RespostaTXT

FORMULARIO PERGUNTA_RESPOSTATXT				
Descrição	Cód. Atributo	Tipo	Pk	Fk
Identificador do Registro	id	int	Sim	Não
Identificador da Pergunta	id_formulario_pergunta	int	Não	Sim
Identificador do Paciente	id_paciente	int	Não	Sim
Valor do Texto	value	text	Não	Não
AutoExplicativo	data_resposta	datetime	Não	Não
AutoExplicativo	excluido	tinyint(4)	Não	Não