

FACULDADE DE ENGENHARIA DE COMPUTAÇÃO

PROJETO FINAL I e II

PLANO DE TRABALHO

Informatização de Clínica Pediátrica

Tuane Quintella de Andrade

10/03/2012

INTRODUÇÃO

A clínica Crescere é uma clínica pediátrica que funciona com quatro médicos e seus respectivos pacientes, uma secretária e um computador sem um sistema específico de gestão de clínica. A maioria das atividades é feita de forma manual, sendo informatizada apenas a autorização de consultas pelo sistema dos convênios. As atividades mais importantes da clínica, de acordo com a secretária, são: cadastro de pacientes, agendamento de consultas, atendimento dos pacientes e arquivamento de fichas e exames dos mesmos. Além destas, são realizados na clínica processos de autorização de serviços prestados (pelos convênios atendidos) e de administração de finanças.

Os clientes fazem o agendamento de consultas com a secretária, via telefone. Esta organiza os horários de cada médico a lápis, na respectiva agenda e, para cada consulta anota o nome completo do paciente, o convênio (se tiver), um telefone para contato e faz um sinal se for paciente novo (primeira vez na clínica) ou não. Ao início de cada dia de trabalho, copia as consultas do dia em um papel denominado Agenda do Dia e entrega ao médico em sua chegada à clínica.

Para realizar o atendimento, o médico necessita da ficha de acompanhamento médico do paciente. Estas fichas ficam organizadas de três maneiras: em pastas numeradas individualmente e ordenadas em uma estante na recepção (quando o paciente tem um histórico regular de consultas), ou em pastas denominadas de Primeiras Vezes (quando o paciente foi poucas vezes a clínica, não se mostrando muito regular), ou ainda em pastas do Arquivo Morto (quando o paciente não vai à clínica por um grande período de tempo). A secretária deve separar todas as fichas correspondentes aos pacientes da Agenda do Dia do médico e entregá-las no começo do período de atendimento. Para pacientes novos a secretária separa apenas uma ficha em branco para que seja preenchida quando o paciente chegar à clínica. Para pacientes regulares, ela busca o ultimo sobrenome de cada paciente em uma caixa de cartões organizada alfabeticamente, cada cartão contém o nome completo do paciente e o número da ficha correspondente ao mesmo. Caso o paciente não seja encontrado na caixa de cartões, ela então irá procurar a ficha pelo último sobrenome do paciente nas pastas de Primeiras Vezes (ordenadas

alfabeticamente). Caso a ficha ainda não seja encontrada ela irá procurar novamente a ficha pelo último sobrenome do paciente nas pastas do Arquivo Morto. Quando o paciente chega à clínica para ser consultado, a secretária deve realizar o processo de autorização de serviço pelo convênio do mesmo ou, caso seja uma consulta particular, ela recebe o pagamento e emite o recibo para o paciente antes da realização do atendimento.

O médico, com as fichas dos pacientes empilhadas de acordo com a Agenda do Dia na sua mesa, chama pelo próximo a atender na sala de espera. Ele realiza a consulta, documentando-a na ficha do paciente e escreve, na Agenda do Dia, o CID (Código Internacional da Doença) diagnosticado na frente do nome do paciente. Ao final do dia, o médico entrega as fichas dos pacientes atendidos à secretária, para que sejam arquivadas novamente. Neste momento são feitas pequenas modificações: se uma ficha anteriormente nas pastas de Primeiras Vezes tornou-se regular deve passar para uma pasta individual numerada, fichas novas devem ir para as pastas de Primeiras Vezes e fichas que estavam no Arquivo Morto devem voltar à estante das pastas regulares. O médico também entrega a Agenda do Dia, para que a secretária possa preencher os CIDs nos sistemas dos respectivos convênios. Cada convênio tem seu sistema específico, que pode ser acessado através do site do próprio convênio na internet. A secretária deve acessá-los com o *login* da clínica e, para cada consulta autorizada anteriormente, preencher o CID diagnosticado na mesma. Para serviços particulares, após receber o pagamento e emitir o recibo, a secretária apenas guarda o dinheiro ou cheque dentro da agenda do médico correspondente, para que este o pegue na hora que estiver indo embora.

No fim de cada mês, um relatório de serviços prestados é gerado no sistema de cada convênio, a partir dos CIDs preenchidos. Este relatório deve ser checado e enviado ao convênio para que este possa realizar o pagamento dos profissionais de acordo com as atividades exercidas.

Durante a rotina da clínica Crescere, a secretária atende a todos os telefonemas, dando informações, ou redirecionando as ligações necessárias e anotando os recados, que devem ser passados aos médicos assim que possível (quando urgente) ou no final do dia (quando não urgente). Também faz parte do trabalho da secretária gerenciar as correspondências, despesas da empresa e despesas pessoais de cada médico que lhe solicitar, impedindo que compromissos sejam esquecidos e pagamentos sejam atrasados, indo aos bancos, sempre que necessário. A secretária também é responsável por obter a confirmação de presença dos pacientes às consultas e por emitir atestados e recibos quando requisitado.

Por fora do funcionamento da clínica em si, os médicos frequentemente sentem a necessidade de fazer pesquisas estatísticas com seus pacientes, para que possam escrever artigos científicos, capítulos de livros ou mesmo montar uma aula (alguns lecionam em universidades). Nestas ocasiões, os médicos separam as fichas dos pacientes com informações que lhes são relevantes, levam para casa e fazem a análise necessária nos dados de cada ficha, montando tabelas, gráficos e indicadores. Ao terminar de usar as fichas, as leva de volta para a clínica.

CARACTERIZAÇÃO DE PROBLEMAS E OBJETIVO(S)

Os principais problemas constatados no contexto descrito na introdução foram:

- Desperdício de tempo na rotina:

Devido ao procedimento manual pouco eficiente há uma sobrecarga da secretária e consequentemente um grande desperdício de tempo para agendamento de pacientes, atendimento, emissão de documentos e principalmente para busca por fichas dos pacientes.

- Perda de informações:

A falta de organização, limpeza e clareza de informações presentes nos processos atuais também geram ocasionalmente perda de informações como a ordem a seguir no atendimento, recados a passar e até mesmo perda de cadastros completos de pacientes, o que prejudica o bom funcionamento da clínica, causando insatisfação nos pacientes.

- Retorno financeiro abaixo do esperado:

Com o desperdício de tempo e sobrecarga da secretária citados no item anterior, se faz necessária a redução do número de pacientes a serem atendidos em um dia, causando assim, uma baixa produtividade para os médicos, e, consequentemente, uma remuneração menor do que a esperada pelos mesmos, levando-se em conta que recebem por consulta prestada.

A falta de organização também citada no item anterior pode causar a perda dos CIDs que deveriam ser preenchidos nos relatórios mensais, anulando assim algumas consultas, que deixarão de ser pagas aos profissionais.

- Grande custo para extração de dados estatísticos:

Ao precisar fazer uma pesquisa e/ou extração de dados com as fichas dos pacientes, os médicos levam muito tempo para selecionar os pacientes específicos e extrair os dados que necessitam um por um. Além de gastarem muito tempo para gerar dados estatísticos consistentes, o médico ainda pode ficar muito tempo com uma grande quantidade de fichas de pacientes inacessíveis, possibilitando a perda destas ou confusão nos processos da clínica, caso o paciente compareça para consulta.

Considerando os problemas descritos nesta seção, o trabalho de conclusão de curso terá como objetivo reduzir o tempo de busca por fichas de pacientes.

PLANO DE AVALIAÇÃO DO TRABALHO

Para certificação do objetivo descrito neste plano de trabalho será realizada uma avaliação composta por dois cenários: uma utilizando os processos atuais da clínica (manuais) e outra baseada no funcionamento da solução a ser desenvolvida (software).

Ambos os cenários contarão com uma base contendo os mesmos 2 médicos e 100 pacientes (50 de cada médico). Para obter uma base balanceada, levando-se em conta a primeira letra do último sobrenome, serão utilizados 78 pacientes divididos em 3 por letra do alfabeto, e mais 22 com sobrenomes iniciados pelas letras mais comuns (A, B, C, F, M, R e S). Ainda serão divididos entre pacientes com ficha numerada existente, pacientes com ficha nas pastas de Primeiras Vezes e pacientes com ficha no arquivo morto.

Na segunda semana do mês de Agosto o desenvolvedor, juntamente com a secretária da clínica que tem acesso ao consultório, vão escolher e separar as 100 fichas com os critérios citados e na mesma data o desenvolvedor vai escolher 40 nomes (existentes ou não dentre estes 100) para a futura busca.

Na terceira semana de Agosto o desenvolvedor, com a ajuda da secretária, vai realizar uma medição cronometrada do tempo de busca por 40 fichas segundo os 40 nomes escolhidos previamente pelo desenvolvedor. O desenvolvedor irá mostrar o nome a buscar, e cronometrar, enquanto a secretária irá buscar da forma convencional (manual) pelo nome mostrado.

Na segunda semana do mês de Setembro o desenvolvedor alimentará a base do software desenvolvido com os mesmos dados do cenário anterior (mesmos 2 médicos e mesmos 100 pacientes).

Na terceira semana de Setembro haverá um treinamento básico da secretária da clínica para utilização do software desenvolvido. Ela irá aprender a utilizar o módulo de busca de fichas de pacientes e interpretar seus resultados.

Na mesma semana o desenvolvedor juntamente com a secretária da clínica irão realizar uma medição cronometrada da busca pelos mesmos 40 nomes citados no primeiro cenário. O desenvolvedor irá mostrar o nome e cronometrar enquanto a secretária realizará a busca pelo software.

Será feita uma comparação direta dos dados coletados para verificação do cumprimento do objetivo inicial.

PROPOSTA DE SOLUÇÃO

Para cumprir o objetivo proposto, pensou-se na confecção de um software, para informatizar os processos da clínica, contando com o desenvolvimento de um sistema web em *Ruby on Rails* para gerenciamento e organização de dados armazenados em um banco *SQLite*. Os registros e interface com banco de dados serão todos utilizando a Gem *ActiveRecord*.

O software contará com fichas eletrônicas de paciente e de médico. Essas fichas eletrônicas serão controladas pelos módulos:

- Cadastro, atualização e exclusão de paciente – o cadastro do paciente irá agregar dados pessoais e de contato.
- Busca inteligente de pacientes com a utilização de redes complexas utilizando grafos de derivação de palavras (ANTIQUEIRA, 2005). Essa técnica será utilizada para achar nomes semelhantes ou erros de digitação.
- Cadastro, atualização e exclusão de médico – o cadastro do médico irá agregar dados pessoais e de contato também.

Haverá um módulo de agendas de consultas, que estabelecerá a conexão entre um médico e N registros de consultas, relacionados a um paciente cada. Esta agenda será implementada com *jQuery DatePicker* e *Calendar DateSelect*, além de outras Gems de interface. Esse registro de consulta será gerenciado pelos módulos:

- Agendamento e cancelamento de consulta – o agendamento de pacientes cadastrados irá recuperar as informações do mesmo para exibí-las, já o agendamento de um paciente novo irá gerar um cadastro temporário e incompleto, para ser completo quando ele comparecer à clínica. A consulta agendada já mostrará informações convenientes como: data da última consulta (retorno ou não), arquivo aonde a ficha do paciente se encontra (pastas numeradas, primeiras vezes ou arquivo morto) e telefone de contato.
- Envio de confirmação – módulo responsável por oferecer o envio de confirmações de consulta por e-mail para os pacientes. Esta funcionalidade será implementada com *Action Mailer*.
- Execução de consulta – o registro de consulta contará com um status sobre seu paciente que poderá ser: agendado, em espera, em atendimento, atendido e faltou. O status da consulta será atualizado em tempo real utilizando *Ajax*.

Haverá um registro de prontuário, que relacionará um paciente a N consultas. Este registro será utilizado pelos módulos:

- Execução de consulta – módulo utilizado pela interface do médico para documentar as consultas e registrá-las nos prontuários dos pacientes.
- Emissão de receitas e pedidos de exames – módulo utilizado para gerar receitas e pedidos de exames de acordo com o layout pré-definido da folha de impressão. Para exportar arquivos para impressão será utilizada a Gem *Prawn*.
- Recuperação de exames virtualizados a partir de sites dos laboratórios e ferramenta para anexar tais exames ao prontuário do paciente. Para essa funcionalidade será utilizada a Gem *Rails Uploader* com *jQuery File Upload*.

O software irá agregar um registro de bloco de notas referente a cada usuário (médico ou secretária). Esses blocos de notas irão conter anotações de tarefas e recados pendentes, que serão administradas pelos módulos:

- Inserção, alteração e exclusão de anotação no bloco de notas de um usuário.

Haverá também um registro de agenda financeira associado a cada médico e à clínica. Essas agendas serão implementadas utilizando os mesmos recursos da agenda de consultas (*jQuery DatePicker* e *Calendar DateSelect*), e agregarão registros de pagamentos que serão administrado pelos módulos:

- Cadastro, atualização e exclusão de pagamentos – o registro contará com os status: *a efetuar, efetuado, a receber e recebido*.

O software também agregará uma agenda de contatos. Esses registros de contatos serão controlados pelos módulos:

- Cadastro, atualização e exclusão de contato – o cadastro de contato irá conter telefones, fax, endereços, e-mails e sites.
- Busca inteligente de contatos conforme o módulo de busca de pacientes, utilizando grafos de derivação de nomes como redes complexas.

O software contará com um módulo de relatório mensal dos serviços prestados. O módulo gerará um relatório de todas as informações relevantes das consultas (data, paciente, CID, retorno ou não) realizadas em um período de tempo selecionado, para um ou todos os médicos, para um ou todos os convênios. O relatório poderá ser gravado em arquivo PDF ou impresso, utilizando a *Gem Prawn* como na emissão de receitas e pedidos de exame. Ainda contará com linkn para o preenchimento desses dados nos sistemas de cada convênio.

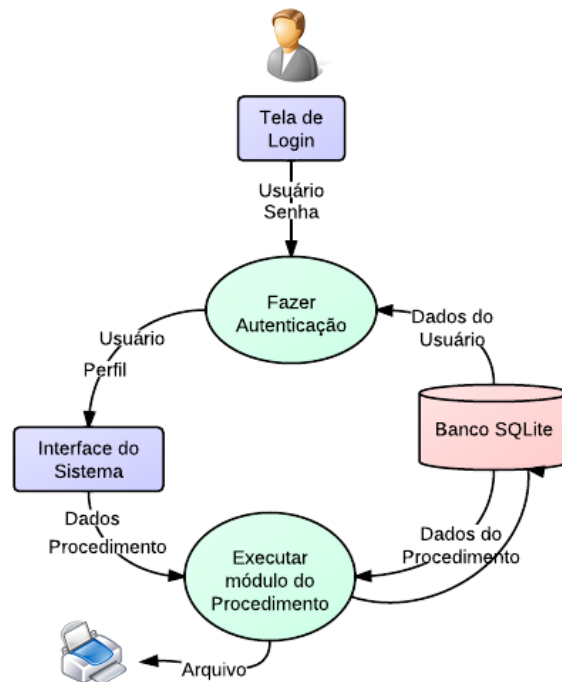
Haverá um módulo para geração de estatísticas, que oferecerá algumas opções de filtros de dados a serem aplicados na base de pacientes. Esses filtros devem ser definidos na especificação de requisitos. O módulo oferecerá a aplicação de algumas fórmulas estatísticas sobre os dados selecionados, e gerará tabelas e gráficos exportáveis. Os gráficos serão gerados utilizando *Google Charts*, através da API para *Ruby on Rails Mattetti-GoogleCharts*. As tabelas serão exportáveis para arquivos CSV e gráficos (com as tabelas) serão exportáveis para arquivos XLS (Planilha do *Microsoft Office Excel* ou *OpenOffice Calc*).

O sistema será multiusuário (para acesso dos médicos e da secretária ao mesmo tempo), e multiperfil (por conta das atividades diferenciadas entre médicos e secretária). Os perfis existentes serão Médico, Secretária e Administrador.

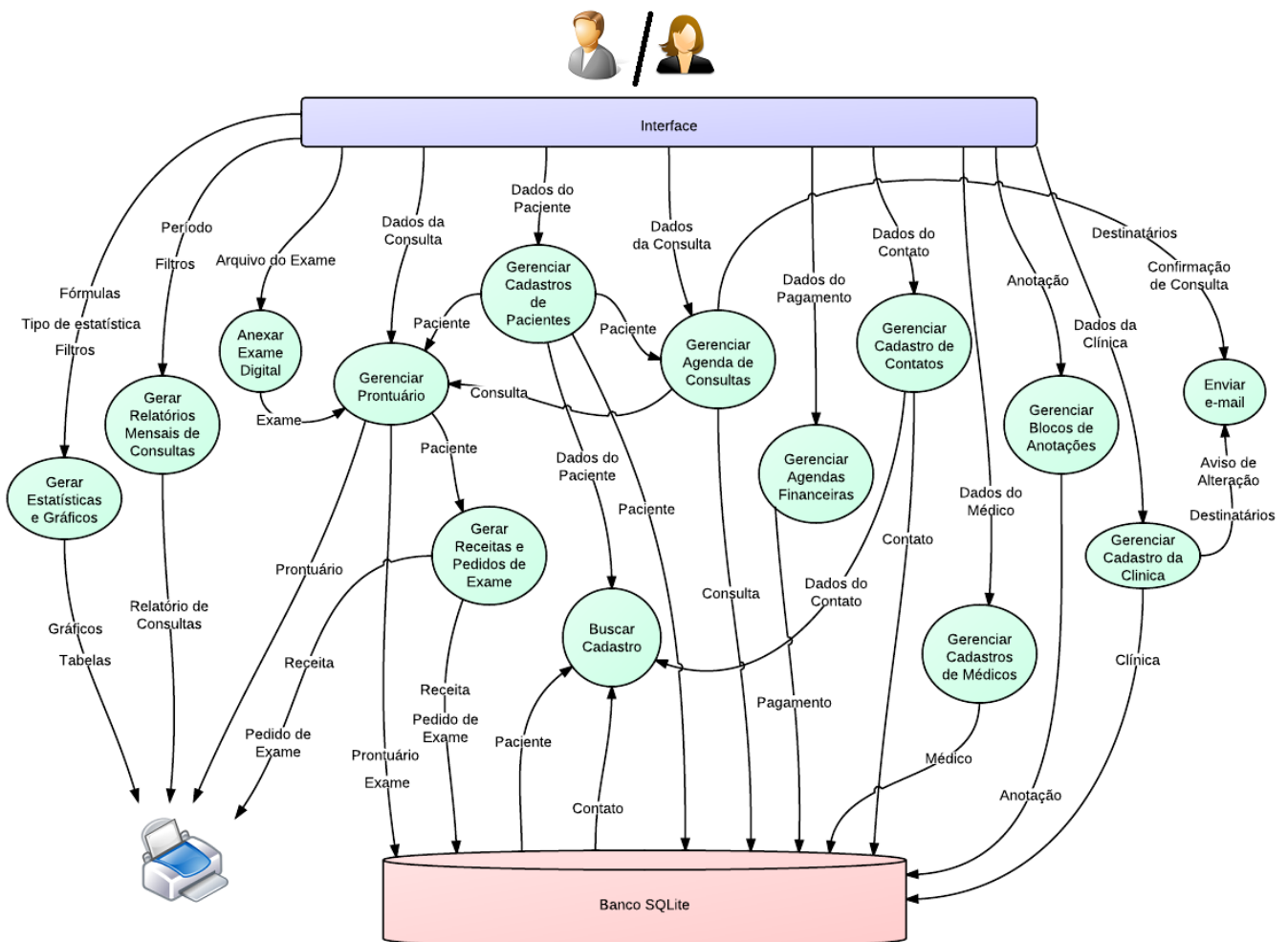
DIAGRAMA DE ARQUITETURA

Os diagramas estão divididos conforme o perfil do usuário que acessa o sistema. Por questão de organização nos diagramas não estão sendo representados os fluxos de dados de **volta** para a interface do usuário, podendo-se admitir que qualquer interação a partir da interface do usuário, gera um fluxo de ida, uma sequência de ações de módulos, e uma resposta à interface do usuário (confirmações, resultados de buscas, etc..).

Visão macro:



Visão expandida:



ESTADO DA EVOLUÇÃO / ARTE

Ambiente Computacional de Apoio à Prática Clínica

Disponível em: <<http://www.saudetotal.com.br/tesechao/index.htm>>

Acessado em: 01 mai. 2012.

Autor: WEN, C. L.

Faculdade de Medicina da Universidade de São Paulo

São Paulo, SP - 2000

Managing Medical Images and Clinical Information: InCor's Experience

Disponível em: <<http://ieeexplore.ieee.org/Xplore>>

Acessado em: 01 mai. 2012.

Autor: FURUIE, S. S.

Faculdade de Medicina da Universidade de São Paulo

São Paulo, SP - 2007

Towards scalable authentication in health services

Disponível em: <<http://ieeexplore.ieee.org/Xplore>>

Acessado em: 01 mai. 2012.

Autor: AHN, G. J.

Faculdade de Tecnologia da Informação da Universidade da Carolina do Norte

Charlotte, NC - 2002

Ambiente Computacional de Gestão em uma Clínica Médica

Disponível em: <<http://www.flf.edu.br/monografias.php>>

Acessado em: 01 mai. 2012.

Autor: Cláudio Pinto de Aragão

Bacharelado em Ciência da Computação da Faculdade Lourenço Filho

Fortaleza, CE - 2010

MÉTODO DE DESENVOLVIMENTO

Neste projeto será utilizada a metodologia de desenvolvimento chamada *Scrum Solo*.

Scrum Solo é uma alternativa para a metodologia ágil de gestão e planejamento de projetos de software chamada *Scrum*, com o diferencial de que trabalha-se individualmente.

No *Scrum Solo*, os projetos são divididos em ciclos (tipicamente mensais) chamados de *Sprints*. O *Sprint* é uma iteração que representa um espaço de tempo, dentro do qual um conjunto de atividades deve ser executado (BELL, 2007).

As tarefas a serem executadas em um projeto são mantidas em uma lista que é conhecida como *Product Backlog*. No início de cada *Sprint*, são selecionadas atividades que serão realizadas. As tarefas alocadas em um *Sprint* são transferidas do *Product Backlog* para o *Sprint Backlog*.

Ao final de um *Sprint* faz-se uma retrospectiva para rever o que foi feito e planejar o próximo *Sprint*.

No *Scrum Solo* é possível começar o primeiro *Sprint* sem ter completado o *Product Backlog*, permitindo a inclusão de novas tarefas durante o andamento de outras.

No trabalho de conclusão de curso ao qual esse plano se refere, o *Product Backlog* será a lista de histórias geradas a partir da análise de requisitos. Os *Sprints* terão duração de 20 dias totalizando 8 *Sprints*, que são as atividades de A4 a A11 no cronograma a seguir.

CRONOGRAMA

Identificação da Atividade	Descrição	Duração	
		Início	Fim
A01	Gerenciar o TCC	23/02/12	18/10/12
A02	Projeto	31/03/12	30/06/12
A03	Criação do backlog do produto	25/04/12	30/04/12
A04	Sprint 1	01/05/12	19/05/12
A05	Sprint 2	20/05/12	09/06/12
A06	Sprint 3	10/06/12	29/06/12
A07	Sprint 4	30/06/12	19/07/12
A08	Sprint 5	20/07/12	08/08/12
A09	Sprint 6	09/08/12	28/08/12
A10	Sprint 7	29/08/12	17/09/12
A11	Sprint 8	18/09/12	07/10/12
A12	Avaliar e validar trabalho	16/08/12	13/10/12
A13	Documentação do trabalho	01/05/12	21/10/12
A14	Preparar defesa do TCC	21/10/12	27/10/12

DISTRIBUIÇÃO DE ATIVIDADES NO PRIMEIRO SEMESTRE

Identificação da Atividade	Primeiro Semestre Mês/Semana																			
	Fev				Mar				Abr				Mai				Jun			
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A01				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
A02									X	X	X	X	X	X	X	X	X	X	X	X
A03												X								
A04													X	X	X					
A05															X	X	X			
A06																		X	X	X
A07																				X
A13													X	X	X	X	X	X	X	X

DISTRIBUIÇÃO DE ATIVIDADES NO SEGUNDO SEMESTRE

Identificação da Atividade	Segundo Semestre Mês/Semana																		
	Jul				Ago				Set				Out				Nov		
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36			
A01	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				
A07	X	X																	
A08			X	X	X														
A09						X	X	X											
A10								X	X	X									
A11											X	X	X						
A12						X	X	X	X	X	X	X	X	X					
A13	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				
A14															X	X			

RESULTADOS ESPERADOS

Identificação do Resultado	Descrição	Identificação da Atividade
R1	Plano de Trabalho	A01
R2	Relatório de Atividades	A01
R3	Análise de Requisitos	A02
R4	Product Backlog	A03
R5	DFD	A02
R6	Diagrama de Classes	A02
R7	MER	A02
R8	Tarefas do backlog do Sprint 1	A04
R9	Tarefas do backlog do Sprint 2	A05
R10	Tarefas do backlog do Sprint 3	A06
R11	Tarefas do backlog do Sprint 4	A07
R12	Tarefas do backlog do Sprint 5	A08
R13	Tarefas do backlog do Sprint 6	A09
R14	Tarefas do backlog do Sprint 7	A10
R15	Tarefas do backlog do Sprint 8	A11
R16	Dados para comprovação do objetivo	A13
R17	Monografia do TCC	A14
R18	Defesa do TCC	A15

RECURSOS MATERIAIS

- GitHub, Inc. [US], para backup com gerenciamento do controle de versões
- Xmind v3.2.1, para criação de diagramas e *mind-maps*
- ScrumDo, LLC., para gerenciamento de projeto
- LucidChart 2012, para criação de diagramas
- JQuery 1.7.2 com Ajax, para elementos de interface

- Ruby v1.9.3, como linguagem de programação
- Ruby on Rails v3.2.0, como framework para programação web
- WebbyNode, Inc. 2012, como máquina virtual para hospedar o servidor
- Dell XPS 14z, como computador para desenvolvimento

UTILIZAÇÃO DOS RECURSOS MATERIAIS

Todos os recursos materiais serão providenciados pela aluna. Os recursos de hardware necessários para o desenvolvimento já foram adquiridos pela mesma. Os recursos de software serão providenciados pela aluna e podem ser encontrados gratuitamente na internet.

GRAU DE DIFICULDADE

Controle de versões – será utilizado controle de versões para todos os arquivos que forem produzidos (executáveis, fontes, documentos). Este item é considerado uma complexidade uma vez que a aluna utilizou poucas vezes um sistema de controle de versões em trabalhos acadêmicos. O controle de versões escolhido foi o Git, que a aluna não teve contato além do básico. Foi escolhido por ser gratuito e ter fácil integração com o sistema de backup apresentado no item a seguir.

Backup – será utilizado um utilitário de backup para salvar todos os arquivos deste trabalho, desde este plano de trabalho, arquivos de projeto e arquivos do software desenvolvido, até a documentação completa e materiais de apoio. Cópias completas serão geradas periodicamente, bem como cópias parciais (que contemplam arquivos modificados ou criados desde a última cópia). A aluna nunca utilizou uma estratégia específica para backup em trabalhos acadêmicos, e por isso este item pode ser considerado uma complexidade. O utilitário de backup escolhido foi o GitHub (<https://github.com/tuanequintella/TCC>), por ser baseado no armazenamento em nuvem, gratuito e por se integrar facilmente com o controle de versões Git.

Plataforma Web – o artefato desenvolvido será um sistema Web por ser mais adequado para um sistema multiusuário e pela eventual necessidade do cliente acessá-lo de outros locais que não sejam a clínica. Como a aluna nunca teve experiência significativa com desenvolvimento Web, sua estrutura, restrições e padrões, este item é uma complexidade relevante.

Ruby on Rails - A linguagem de programação a utilizar será Ruby, juntamente com o framework Ruby on Rails. A linguagem foi escolhida por ser de alto nível e fácil aprendizagem, contando com a aplicação do framework Ruby on Rails que facilitará o desenvolvimento Web. Aluna nunca utilizou a linguagem, tampouco o framework, então será necessário um amplo estudo das características da linguagem, estrutura de projeto, configurações e bibliotecas existentes (Gems).

Busca inteligente – como o ponto principal do trabalho é a melhoria no processo de busca de dados, será estudada e implementada uma rede complexa com grafos de derivação dos nomes cadastrados no sistema, a fim de utilizar uma busca inteligente para encontrar estes cadastros. Esta implementação irá contemplar busca com resultados aproximados como nomes com grafias diferentes e possíveis erros de digitação.

Envio de e-mails – em alguns módulos do software será necessário enviar e-mails aos pacientes da clínica. Assim será necessário o estudo de um protocolo de e-mail e sua utilização em Rails, tópicos estes, com os quais a aluna nunca teve contato antes, sendo então considerada uma complexidade deste trabalho.

Layout de impressão – alguns módulos oferecerão dados para serem impressos, inclusive alguns com layout de impressão pré-definido para folha com campos delimitados (guias de exames e receitas). Esta complexidade é significativa pois a aluna não teve experiências com funcionalidades parecidas, então deverá estudar a formatação de layouts de impressão e como implementá-los.

Geração de estatísticas / Integração com Google Charts – o módulo de estatísticas deverá trabalhar com a criação de tabelas de dados a partir de critérios definidos pelo usuário, aplicação de fórmulas estatísticas e geração de gráficos visuais. A aluna nunca trabalhou com bibliotecas gráficas e tratamento de dados para estes fins, e este módulo também será integrado com a ferramenta Google Charts, que irá requerir um estudo de suas funcionalidades e de uma API específica.

Exportação de arquivos – Tanto no módulo de estatísticas como no módulo de emissão de receitas e exames e emissão de relatório mensal de consultas haverá a opção de exportação dos dados e relatórios em alguns tipos de arquivos (planilha XLS, CSV) com os quais a aluna nunca implementou nenhum tipo de aplicação.

Upload de arquivos – No módulo de anexar exames no formato digital ao prontuário do paciente, será necessária a implementação de upload de arquivos. A aluna nunca teve experiência significativa com a implementação de transmissão de arquivos via rede, e por isso será necessário um estudo a respeito deste assunto e de possíveis bibliotecas auxiliares nessa tarefa, tornando este item uma complexidade significativa.

Multiusuário – vários usuários utilizam o sistema ao mesmo tempo. Este item é considerado uma complexidade pois será necessária uma avaliação de recursos críticos que não podem ser acessados simultaneamente e a implementação de uma estratégia de exclusão mútua para estes recursos.

Multiperfil – pelas funcionalidades serem diferentes para cada tipo de usuário, o sistema será multiperfil. Essa complexidade pode ser considerada pois o sistema terá que mudar sua interface e funcionalidades disponíveis de

acordo com a autenticação inicial (login). Isto também implica na definição e implementação de regras de permissões.

ANÁLISE DE RISCOS

Indisponibilidade do WebbyNode, onde estará executando o servidor do sistema, e estará armazenado o banco de dados. Risco **médio**, pois apesar da consequência de indisponibilidade total do sistema, a chance deste problema ocorrer é baixa.

Complexidade do projeto x Tempo disponível da aluna para a quantidade de tarefas a serem efetuadas até a data estipulada (Outubro/2012), desde todo o estudo que será necessário de todas as tecnologias não utilizadas até a implementação, testes e correções. Risco **médio**, pois apesar da chance alta de ocorrência de falta de tempo, este risco pode ser contornado simplificando a implementação do sistema caso necessário, deixando pontos em aberto para futuras implementações.

Indisponibilidade das ferramentas online como o Lucidchart e ScrumDO, utilizados para desenvolvimento do projeto do sistema. Risco **baixo**, pois a aluna mantém cópias relativamente recentes dos projetos realizados nas ferramentas em disco rígido local e na ferramenta de backup utilizada GitHub.

OUTRAS OBSERVAÇÕES

O produto de software citado na Proposta de Solução deste plano de trabalho ainda poderá sofrer futuras implementações como:

- Funcionalidade de sugestão de CIDs para cada consulta, baseada em uma inteligência artificial de análise dos diagnósticos.
- Funcionalidade de sincronização das agendas com uma Google Agenda, possibilitando acesso aos compromissos via dispositivos móveis.

REFERÊNCIAS

RUBY. Disponível em: <<http://www.ruby-lang.org>>. Acesso em: 30 mar. 2012.

RUBY on Rails. Disponível em: <<http://rubyonrails.org>>. Acesso em: 30 mar. 2012.

SQLITE. Disponível em: <<http://www.sqlite.org>>. Acesso em: 26 abr. 2012.

BELL, Peter. *Solo Scrums*. 2007. Disponível em: <<http://www.pbell.com/index.cfm/2007/6/17/Solo-Scrums>>. Acesso em: 24 mar. 2012.

JQUERY. Disponível em: <<http://jquery.com>>. Acessado em: 30 abr. 2012.

GIT. Disponível em: <<http://git-scm.com>>. Acesso em: 30 mar. 2012.

GITHUB. Disponível em: <<https://github.com>>. Acesso em: 23 abr. 2012

ACTIVE Record. Disponível em: <<https://github.com/rails/rails/tree/master/activerecord>>. Acessado em: 30 abr. 2012.

ANTIQUEIRA, L.; NUNES, M.; OLIVEIRA, O.; COSTA, L. *Modelando Textos como Redes Complexas*. Universidade de São Paulo, 2005.

JQUERY DatePicker. Disponível em: <<http://docs.jquery.com/UI/Datepicker>>. Acessado em: 30 abr. 2012.

CALENDAR DateSelect. Disponível em: <https://github.com/timcharper/calendar_date_select>. Acessado em: 30 abr. 2012.

ACTION Mailer. Disponível em: <<https://github.com/rails/rails/tree/master/actionmailer>>. Acessado em: 30 abr. 2012.

JQUERY Ajax. Disponível em: <<http://api.jquery.com/category/ajax>>. Acessado em: 30 abr. 2012.

PRAWN PDF. Disponível em: <<http://prawn.majesticseacreature.com/>> e <<https://github.com/prawnpdf/prawn>>. Acessado em: 30 abr. 2012.

RAILS Uploader. Disponível em: <<https://github.com/superp/rails-uploader>>. Acessado em: 30 abr. 2012.

JQUERY File Upload. Disponível em: <<https://github.com/blueimp/jQuery-File-Upload>>. Acessado em: 30 abr. 2012.

GOOGLE Charts. Disponível em: <<https://developers.google.com/chart/>>. Acessado em: 30 abr. 2012.

MATTETTI GoogleCharts. Disponível em: <<https://github.com/mattetti/googlecharts>>. Acessado em: 01 mai. 2012.

LUCIDCHART. Disponível em: <<http://www.lucidchart.com>>. Acesso em: 21 mar. 2012.

XMIND. Disponível em: <<http://www.xmind.net>>. Acesso em: 21 abr. 2012.

SCRUMDO. Disponível em: <<http://www.scrumdo.com>>. Acesso em: 23 abr. 2012.

WEBBYNODE. Disponível em: <<http://webbynode.com>>. Acesso em: 30 mar. 2012.

DEFINIÇÕES E ABREVIATURAS

Artefato Computacional – sistema de software ou de hardware, ou ainda uma combinação dos dois, que será desenvolvido com vistas à solução de um ou mais problemas identificados em um ambiente de interesse.

Diagrama de Arquitetura – é uma representação gráfica onde aparecem os módulos do artefato computacional a ser desenvolvido e as entidades externas, com seus fluxos de dados. Os módulos podem ser de dois tipos: prontos e que serão integrados, ou a serem desenvolvidos. Podem existir também módulos que existem e serão modificados. Os fluxos têm formatos que devem ser apresentados. Deve-se usar as regras gerais para a definição de diagramas funcionais.

Layout de Impressão – É composto da área de design ou formato de página e as margens para serem impressos.

Backup – É uma cópia de segurança, ou seja, a cópia dos dados de um local de armazenamento a outro para que possam ser restaurados em caso da perda dos dados originais, o que pode envolver apagamentos acidentais ou corrupção de dados.

Controle de Versões – É a atividade de gerenciar diferentes versões no desenvolvimento de um documento qualquer. É comumente utilizado no desenvolvimento de *software* para controlar as diferentes versões (histórico e desenvolvimento) dos códigos-fontes e também da documentação.

Sistema Web / Aplicação Web - É o termo utilizado para designar, de forma geral, softwares projetados para utilização através de um navegador,

na internet ou em redes privadas (intranet). Também é dito de qualquer programa que seja processado em um servidor.

Biblioteca / Gem – Refere-se uma coleção de subprogramas utilizados no desenvolvimento de software. Bibliotecas contém código e dados auxiliares, que provém serviços a programas independentes, o que permite o compartilhamento e a alteração de código e dados de forma modular. Gem é uma biblioteca escrita em linguagem de programação Ruby. Cada Gem tem suas dependências específicas, as quais o próprio gerenciador Ruby já providencia ao instalar a Gem.

Máquina Virtual – É o nome dado a uma máquina, implementada através de *software*, que executa programas como um computador real.

Servidor – É um software que fornece serviços a uma rede de computadores. Comumente esse software hospeda aplicações e serviços Web que são acessados através de clientes (navegadores, etc).

Cliente – Aplicação ou computador que envia requisições de dado para algum dos servidores conectados e recebe ou não uma resposta.

Interface – É a fronteira que define a forma de comunicação entre duas entidades. No caso da Interface de usuário é o conjunto de elementos com o qual os utilizadores interagem com as máquinas, dispositivos e programas de computador (software).

Base de dados / Banco – São coleções de informações armazenadas que se relacionam da forma necessária para terem sentido. Fazem parte importante dos sistemas de informação atuais.

PDF – *Portable Document Format*. Formato de arquivo para representar documentos de maneira independente do aplicativo usado para criá-los. Um arquivo PDF pode descrever documentos que contenham texto, gráficos e imagens num formato independente de dispositivo e resolução.

XLS – Formato de arquivo baseado em XML (*Extensible Markup Language*) utilizado pelos softwares de criação e edição de planilhas mais conhecidos (*Microsoft Office Excel*, *OpenOffice Calc* e *Apple iWork Numbers*).

CSV – *Comma Separated Values*. Formato de arquivo de texto não formatado, com sequências de valores separados por vírgula, normalmente representando uma tabela ou matriz de dados.

API – *Application Programming Interface*. Conjunto de funções e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços.

AJAX – *Asynchronous Javascript and XML*. É o uso de tecnologias como Javascript e XML, providas por navegadores, para tornar páginas Web mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações.

Ficha de Paciente / Prontuário – Formulário com informações cadastrais do paciente da clínica, seguido de um histórico de consultas com seus respectivos resumos contendo sintomas apresentados, diagnósticos, orientações medicamentosas (receitas), orientações não medicamentosas e pedidos de exames.

Arquivo Morto – Condição e local onde são arquivadas fichas de pacientes que não vão à clínica por um longo período de tempo. Esse período varia de acordo com as condições do paciente (idade, motivo das consultas, etc).

CID – Classificação Internacional de Doenças. Fornece códigos relativos à classificação de doenças e de uma grande variedade de sinais, sintomas, aspectos anormais, queixas, circunstâncias sociais e causas externas para ferimentos ou doenças. A cada estado de saúde é atribuída uma categoria única à qual corresponde um código, que contém até 6 caracteres.