**Prontuário Eletrônico de Pacientes através da computação nuvens:**

**Conceitos, gerência de configuração de ambiente computacional e levantamento de requisitos**

Rafael Mota Correia

Centro Universitário Estácio do Ceará (rafaelmotac@live.estacio.br)

Josyane Lannes Florenzano de Souza

Centro Universitário Estácio do Ceará (josyane.souza@estacio.br)

Henrique Mota

Centro Universitário Estácio do Ceará (mota.henrique@gmail.com)

**Resumo:** O presente estudo apresenta um recorte quanto ao estado da arte sobre computação nas nuvens. A pesquisa se insere no âmbito da engenharia de software, primeiro quanto à preparação do ambiente computacional, a gerência de configuração de ambiente para dar início ao levantamento de requisitos funcionais e assim alcançar o desenvolvimento do prontuário eletrônico de pacientes de uma clínica médica, localizada no interior do estado do Pará. Diante da necessidade de controle dos dados serem via web, surge a computação em nuvem que é uma tendência, pois tem por objetivo proporcionar serviços de tecnologia da Informação (TI) sob demanda com pagamento baseado no uso. Computação em nuvem pretende ser global e prover serviços para todos, desde o usuário final que hospeda seus documentos pessoais na Internet até empresas que terceirizarão toda a parte de TI para outras empresas (TAURION, Cezar).

**Palavras-chave**: Gerência de configuração, computação em nuvens, prontuário eletrônico do paciente

# 1. Introdução

O presente estudo apresenta um recorte quanto ao estado da arte sobre computação nas nuvens. A pesquisa se insere no âmbito da engenharia de software, primeiro quanto à preparação do ambiente computacional, a gerência de configuração de ambiente para dar início ao levantamento de requisitos funcionais e assim alcançar o desenvolvimento do prontuário eletrônico de pacientes de uma clínica médica, localizada no interior do estado do Pará. Estudar conceitos de gerência de configuração e levantar seus padrões de soluções existentes no mercado permeiam esta pesquisa, pois acredita-se que o devido estudo e uso dos melhores padrões podem contornar o problema de configuração e versionamento de hardware e software. Diante da necessidade de controle dos dados serem via web, surge a computação em nuvem que é uma tendência, pois tem por objetivo proporcionar serviços de tecnologia da Informação (TI) sob demanda com pagamento baseado no uso. Computação em nuvem pretende ser global e prover serviços para todos, desde o usuário final que hospeda seus documentos pessoais na Internet até empresas que terceirizarão toda a parte de TI para outras empresas (TAURION, Cezar).

Hoje, a responsabilidade pelo cuidado do paciente é transferida para diferentes equipes de profissionais. A associação entre a crescente geração e demanda por informações estruturadas e acessíveis, concomitante ao desenvolvimento da área da informática despertou o interesse para o desenvolvimento do prontuário eletrônico do paciente (PEP). O prontuário, criado inicialmente para documentar informações de saúde e doença, tornou-se mais complexo, passando a ter um papel importante na sociedade moderna, com o subsídio de manutenção da saúde do paciente, no compartilhamento de informações. O prontuário serve como base legal para fonte de pesquisas, educação e reciclagem médica; fornecendo sustentação para administração de serviços de saúde. O PEP permitirá a clínica manter o registro eletrônico das informações de seus pacientes, a partir de qualquer lugar.

Este contexto justifica o estudo dos conceitos de computação nas nuvens através do desenvolvimento do estudo de caso da clínica médica. Foi realizado a Estrutura Analítica de Projetos (EAP) através do processo de subdivisão de entregas e do trabalho do projeto em componentes menores e mais facilmente gerenciáveis. Cuja estrutura de árvore exaustiva, hierárquica (de mais geral para mais específica) orientada às entregas, fases de ciclo de vida que precisam ser feitas para completar o projeto. Em seguida, foi realizado o levantamento de requisitos funcionais através dos diagramas da UML (*Unified Modeling Language*), Diagrama de Caso de Uso e Diagrama de Classes. Para análise e controle dos dados nas nuvens, utilizou-se a técnica de Modelo de Entidade Relacionamento (MER) proposta por Codd em 1970, para definição do banco de dados relacional através da definição dos modelos conceitual, lógico e físico dos dados.

# 2. Estado da Arte

## 2.1 Gerência de Configuração

Gerência de Configuração de Software, Gerência de Configuração ou ainda Gestão de Configuração de Software é uma área da engenharia de software cuja equipe é responsável por fornecer o apoio para o desenvolvimento de software. Suas principais atribuições são o controle de versão, o controle de mudança e a auditoria das configurações.

Roger Pressman, em seu livro Software Engineering: A Practitioner's Approach, afirma que a gerência de configuração de software (GCS) é o: “Conjunto de atividades projetadas para controlar as mudanças pela identificação dos produtos do trabalho que serão alterados, estabelecendo um relacionamento entre eles, definindo o mecanismo para o gerenciamento de diferentes versões destes produtos, controlando as mudanças impostas, e auditando e relatando as mudanças realizadas” [PRESSMAN, 2006].

Em outras palavras, a Gerência de Configuração de Software tem como objetivo responder as seguintes perguntas: O que mudou e quando? Por que mudou? Quem fez a mudança? Podemos reproduzir esta mudança?

Cada uma dessas perguntas corresponde a uma das atividades realizadas pela Gerência de Configuração de Software. O controle de versão é capaz de dizer o que mudou e quando mudou. O controle de mudanças é capaz de atribuir os motivos a cada uma das mudanças. A Auditoria por sua vez responde as duas últimas perguntas: Quem fez a mudança e podemos reproduzir a mudança?

A história da Gerência de Configuração de Software surge em meados da década de 1970, quando os microprocessadores se tornaram populares e o software deixou de ser considerada parte integrante do hardware para se tornar um produto independente. Nesta época, os sistemas se tornaram cada vez maiores e sofisticados, ficando claro que seriam necessárias metodologias próprias, diferentes das usadas no desenvolvimento de hardware, para controlar o desenvolvimento desses sistemas.

O Departamento de Defesa (DoD) dos EUA foi um dos pioneiros nessa área, criando o padrão DoDSTD-2167 que abordava o desenvolvimento de software. A abordagem do DoD para controlar isto consistia da adoção de uma linguagem padronizada -- Ada -- e de práticas padrão para desenvolvimento de software e Gerência de Configuração. Outras organizações estavam por sua vez adotando práticas e métodos de Gerência de Configuração de Software, algumas das quais envolvidas também com o desenvolvimento de sistemas para o DoD e outros órgãos do governo Americano. Isto levou o próprio DoD a adotar algumas práticas comerciais e eventualmente uni-las com seus padrões, gerando assim o padrão IEEE/IEA 12207. Um outro padrão a respeito é o IEEE 828-1983.

A Gerência de Configuração e Software é definida por quatro funções básicas: Identificação, Documentação, Controle, Auditoria.

No início do desenvolvimento, a GCS permite à equipe de desenvolvimento identificar as unidades que compõem o sistema de acordo com as funcionalidades que elas deverão desempenhar, e as interfaces entre estas unidades, documentando assim a interação entre elas. O controle contínuo da evolução destas funcionalidades e interfaces permite que a integração entre estas unidades tenha sucesso continuado, com as mudanças devidamente gerenciadas e documentadas. Por fim, a auditoria das funcionalidades identificadas, documentadas e controladas garante a confiabilidade do sistema.

A terminologia especifica da GCS, como também sua história, tem dado origem a controvérsias, de freqüentes variações. Ferramentas vendidas como também acadêmicas tiraram vantagem disto para deliberadamente mudar a terminologia ou procedimentos para reduzir a possibilidade dos clientes para mudanças, algumas vezes tentando desta maneira redefinir o estabelecimento de acrônimos.

Em particular, o vendedor conhecido como Atria (depois Rational Software), agora uma parte da IBM, usava SCM como padrão para Software Configuration Management (em portugês: "Gerência de Configuração de Software") enquanto o Gartner Group usa o termo SCCM ou Software Change and Configuration Management (em português: "Gerência de Mudanças e Configuração de Software").

Entretanto, os conceitos básicos da GCS descritos abaixo são bem aceitos, divergindo de um autor ou fornecedor para o outro meramente nos termos utilizados.

Configuração é o estado em que um sistema se encontra em um determinado momento. Este sistema pode ser composto de todo tipo de elementos, como peças de hardware, artefatos eletrônicos ou não (i.e. documentos em papel), etc. A Configuração de Software trata apenas dos elementos que se encontram em formato eletrônico e fazem parte dessa configuração. Isso inclui todos os arquivos fontes, todos os documentos eletrônicos, as ferramentas de software utilizadas para construir ou mesmo ler estes arquivos, o sistema operacional utilizado, as bibliotecas de software, etc.

Essa configuração varia com o tempo, pois novos arquivos são incluídos, e arquivos existentes são alterados ou removidos. O objetivo da Gerência de Configuração como um todo é organizar todos estes elementos de forma a saber em qual estado o sistema se encontrava nos momentos chave do desenvolvimento (por exemplo, quando o sistema foi entregue ao cliente, quando o sistema passou por uma mudança de versão, quando o sistema foi enviado para auditoria, etc). A Gerência de Configuração como um todo trata dos elementos, incluindo hardware, necessários para a manutenção apropriada do sistema. A Gestão de Configuração de Software trata especificamente dos elementos necessários a construção de sistemas de software, e em geral, controla apenas os elementos em formato computadorizado.

Em Sistemas de controle de versão as configurações específicas são geralmente identificadas pelo uso de tags ou labels (placas ou etiquetas, em inglês).

Item de configuração, ou artefato de configuração, ou ainda apenas artefato é o elemento básico da gerência de configuração. O Item de configuração é um elemento unitário que será gerenciado: um arquivo de código fonte, um documento de texto, um projeto de uma placa eletrônica, uma planta feita em papel, um CD-ROM de instalação de um sistema operacional, etc. A configuração de um sistema é basicamente a lista de todos os itens de configuração necessários para reproduzir um determinado estado de um sistema. Em geral números de versão são associados aos itens de configuração de forma a podermos identificar também a evolução destes itens.

O controle de revisões é o controle que se faz em cima de cada item de configuração para armazenar todas as mudanças que foram aplicadas nele. Em geral, na Gerência de Configuração de Software, usamse Sistemas de controle de versão para automatizar esta tarefa. O controle de revisões deve permitir que se tenha acesso a todas as formas anteriores de um artefato e também saber quem fez as alterações e quando (para fins de auditoria). Dessa maneira podemos facilmente recompor uma configuração antiga (para identificar problemas que ocorrem em versões específicas do sistema, mas não em outras) e auditar as mudanças realizadas para ver se estão de acordo com o que foi solicitado originalmente.

Um Conjunto de mudanças (do inglês: change set) é o conjunto de todas as alterações que ocorreram no sistema para atender um determinado fim, ou num determinado período de tempo. Associados com a Gerência de Mudanças, os conjuntos de mudanças mapeiam os itens que foram mudados para uma dada versão do sistema com os motivos das mudanças.

Linhas-base ou Baseline é um conjunto de configurações passadas e futuras que devem ser seguidas para se alcançar determinado objetivo. Elas podem ser simultâneas ou consecutivas: a versão 1.0 pode estar ao mesmo tempo sendo corrigida para falhas de segurança, gerando a versão 1.1 e evoluída com novas funcionalidades gerando a versão 2.0 do sistema. Nesse caso específico podemos identificar duas linhas-base simultâneas (versão 1.1 e versão 2.0) e uma linha base passada (versão 1.0). Linhas-base não são configurações fixas, mas sim toda a evolução de determinadas configurações. Todo o trabalho realizado para se chegar a versão 2.0 e todo o trabalho realizado depois em cima desta versão para se testar e corrigir problemas são parte da linha base, e não apenas a versão 2.0 em si.

Em Sistemas de controle de versão uma linha-base é geralmente identificada pelo uso de *branches* (galhos, em inglês).

A Gerência de mudanças é uma parte geralmente negligenciada da Gerência de configuração. Como ela não tem resultados imediatos para os desenvolvedores e engenheiros de software envolvidos no projeto, estes acabam por não perceber sua importância. Gerência de mudanças entretanto é uma parte importante da Gerência de configuração, pois é a atividade que permite se saber o motivo de uma configuração ter sido mudada para outra configuração. Esta atividade também pode ser parcialmente automatizada, e diversos Sistemas de controle de versão já são integrados com sistemas de gerência de mudanças. A gerência de mudanças tem por objetivo mapear, para cada mudança efetuada no sistema, qual foi o motivo que gerou esta mudança. É comum vermos em sistemas de software arquivos que listam as melhorias e mudanças entre duas versões. Estes arquivos são resultado da gerência de mudanças, identificando o que mudou entre uma versão e outra.

A finalidade da política de Gerência de Configuração de Software consiste em definir a maneira como as atividades de Gerência de Configuração de Software serão executadas, o momento adequado, os responsáveis em executa-las e os conceitos envolvidos no processo. Entre as definições que devem contar nas políticas de Gerência de configuração de software podemos citar:

Ferramentas para automatização do controle de revisões (Sistema de controle de versão) caso seja usada. Caso não seja usada ferramenta, deve se definir o procedimento manual para o controle de revisões. Caso existam elementos que não estejam em formato eletrônico (ferramentas de hardware, por exemplo), os procedimentos de controle de revisões para estes elementos devem, também, ser definidos.

Ferramentas para o controle de mudanças. Caso não seja usada ferramenta, deve também ser definido o procedimento manual.

Controle de acesso às ferramentas de controle de revisão e controle de mudanças

Nível de integração entre as ferramentas caso sejam ferramentas distintas. Alguns fabricantes fornecem ferramentas que já desempenham os papeis de controle de revisão e controle de mudanças num único sistema, enquanto outros fabricantes as fornecem em separado.

Periodicidade e granularidade do controle de revisões. Recomenda-se um controle diário para elementos em formato eletrônico. A granularidade em geral depende do tipo de item e da ferramenta utilizada.

## 2.2 Computação em Nuvens

Computação em nuvem é uma tendência recente de tecnologia cujo objetivo é proporcionar serviços de Tecnologia da Informação (TI) sob demanda com pagamento baseado no uso. Tendências anteriores à computação em nuvem foram limitadas a uma determinada classe de usuários ou focadas em tornar disponível uma demanda específica de recursos de TI, principalmente de informática [Buyya et al. 2009b].

Para utilizarem os serviços, os usuários necessitam apenas ter em suas máquinas um sistema operacional, um navegador e acesso à Internet. Todos os recursos e processamentos computacionais estão disponíveis na Internet. Assim, as máquinas dos usuários não necessitam ter altos recursos computacionais, diminuindo assim o custo na aquisição de máquinas por parte destes usuários. Todo hardware pode ser utilizado para realizar alguma tarefa que seja adequada ao seu poder de processamento. Novos recursos de hardware podem ser adicionados a fim de aumentar o poder de processamento e cooperar com os recursos existentes.

O modelo de computação em nuvem foi desenvolvido com o objetivo de fornecer serviços de fácil acesso e de baixo custo e garantir características tais como disponibilidade e escalabilidade. Este modelo visa fornecer, basicamente, três benefícios. O primeiro benefício é reduzir o custo na aquisição e composição de toda infra-estrutura requerida para atender as necessidades das empresas, podendo essa infra-estrutura ser composta sob demanda e com recursos heterogêneos e de menor custo. O segundo é a flexibilidade que esse modelo oferece no que diz respeito à adição e troca de recursos computacionais, podendo assim, escalar tanto em nível de recursos de hardware quanto software para atender as necessidades das empresas e usuários. O último benefício é prover uma abstração e facilidade de acesso aos usuários destes serviços. Neste sentido, os usuários dos serviços não precisam conhecer aspectos de localização física e de entrega dos resultados destes serviços.

**3. Estudo de Caso: Prontuário Eletrônico de Paciente**

# 3. 1 Prontuário Eletrônico de Paciente

O PEP - Prontuário Eletrônico do Paciente - é definido como um sistema de prontuário médico padronizado e digital. Segundo informações do Institute of Medicine (IOM), o prontuário eletrônico consiste em um registro eletrônico elaborado com especificidade para apoiar o usuário, oferecendo acesso prático à inúmeras informações de banco de dados, recursos de apoio à decisão, alertas e diversos outros recursos.

Além disso, o sistema é um meio físico onde fica gravado todo o histórico clínico do paciente, o que gera discussões éticas entre profissionais de saúde. A implantação do Prontuário Eletrônico Digital foi implementada no Brasil no ano de 2002, época em que o Conselho Federal de Medicina (CFM) definiu suas características gerais na resolução 1639. A implantação do sistema só se tornou possível em aspecto jurídico após a criação do Certificado Digital pelo Governo Brasileiro. [SILVA et al, 2015]

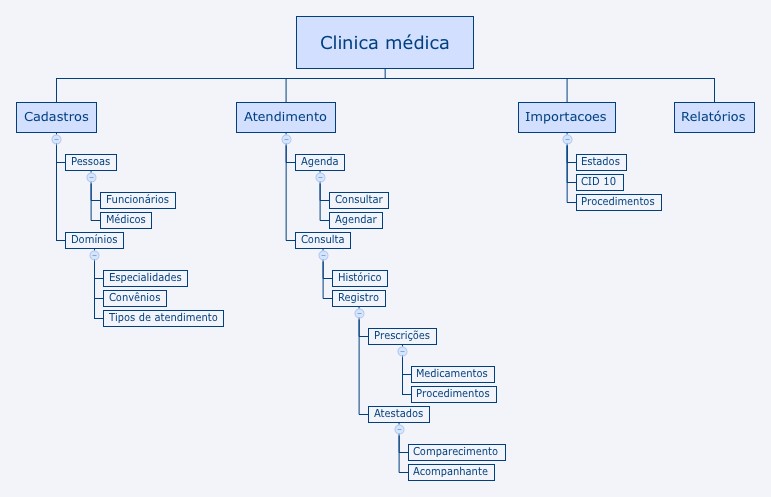
Segundo a Organização Mundial de Saúde (OMS), a quantidade de informações na área da saúde dobra a cada três anos e atinge o médico de diversas maneiras, como novos métodos de diagnóstico e tratamento, novos princípios químicos e inovações em biologia molecular e genética. O crescimento exponencial do conhecimento médico levou ao aparecimento de diversas especialidades clínicas. Hoje, a responsabilidade pelo cuidado do paciente é transferida para diferentes equipes de profissionais. A associação entre a crescente geração e demanda por informações estruturadas e acessíveis, concomitante ao desenvolvimento da área da informática despertou o interesse para o desenvolvimento do prontuário eletrônico do paciente (PEP). [STOLF, 2007]

## 3.2 Estrutura Analítica de Projeto

A representação utilizada para a definição do escopo, que é a saída desse processo, é a estrutura analítica do projeto (EAP), tradução para o português de Work Breakdown Structure (WBS). A Estrutura Analítica do Projeto (EAP) é a base para o detalhamento do trabalho do projeto. Depois de elaborada e aprovada, ela passa ser a base de referência do escopo do projeto (scope baseline). [SIQUEIRA, 2007]

A EAP não é um instrumento novo no gerenciamento de projetos. Conforme XAVIER (2005), há registros de sua utilização desde o início da década de 60, nos EUA, pelo Departamento de Defesa. Porém, algumas áreas criaram a cultura de representar o escopo do projeto através de uma lista organizada das atividades que seriam executadas no mesmo. Essa forma de representar o escopo enfocava o que seria feito e não o que seria entregue pelo projeto. Com o passar do tempo, cada vez mais os gerentes de projetos foram verificando que, para melhor atender os anseios dos clientes, era mais importante representar o escopo em função dos resultados que seriam obtidos, sendo uma melhor prática representar as entregas dos projetos em uma EAP. O PMI (2001) consolidou esse conceito ao definir a EAP como “um agrupamento de elementos componentes do projeto, orientado a subprodutos, que organiza e define o escopo de trabalho de um projeto”.

A EAP é uma estrutura hierárquica podendo ser representada na forma gráfica, (semelhante a um organograma), ou como uma lista identada. Segue abaixo o EAP deste projeto (figura 01):

 Figura 01: Estrutura Analítica de Projeto da Clínica Médica

# 3. 3 Requisitos Funcionais através da UML

Análise de requisitos resulta na especificação das características operacionais dos softwares; indica a interface do software com outros elementos do sistema e estabelece restrições a que o software deve satisfazer. [PRESSMAN, 2006]

A UML (Unified Modeling Language) é uma linguagem-padrão para a elaboração da estrutura de projetos de software. A UML poderá ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software. [BOOCH, 2000].

Dentre os vários diagramas da UML, foi escolhido o Diagrama de Caso de Uso (DCU) para definição dos requisitos funcionais. Um caso de uso é uma descrição de um conjunto de sequencia de ações, inclusive variantes, que um sistema executa para produzir um resultado de valor observável por um ator. Graficamente, o caso de uso é representado por uma elipse. [BOOCH, 2000].

Veja o Diagrama de Caso de Uso do projeto Prontuário Eletrônico do Paciente (PEP), conforme figura 02, figura 03, figura 04 e figura 05.

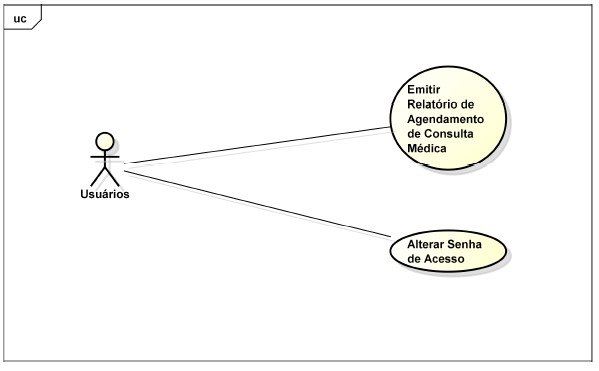


Figura 02 – DCU Usuário

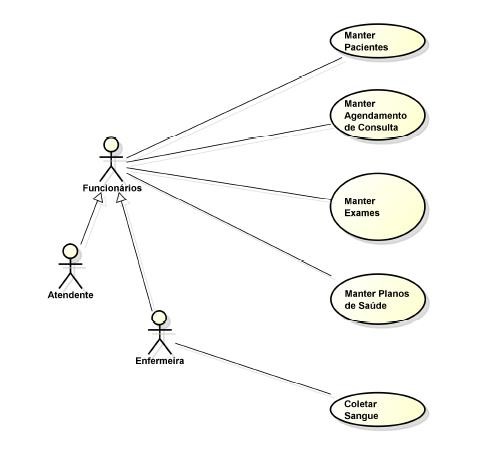


Figura 03 – DCU Funcionários

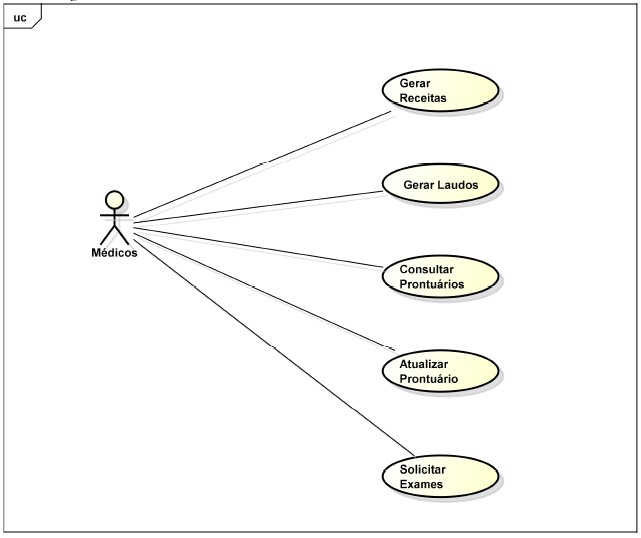


Figura 04 – DCU Médicos

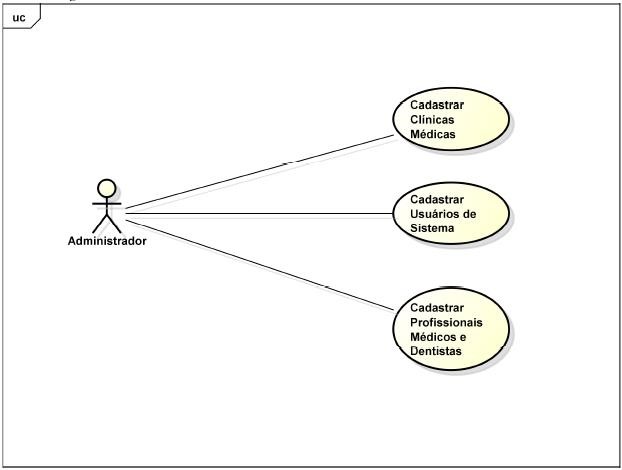


Figura 05 – DCU Administrador

Para complementar os diagramas de caso de uso, achou-se necessário estudar o conceito de História de Usuário, que segundo [....] é uma ou mais sentenças na linguagem de negócio ou cotidiana do [usuário final](https://pt.wikipedia.org/wiki/Usu%C3%A1rio_final) ou [usuário do sistema](https://pt.wikipedia.org/w/index.php?title=Usu%C3%A1rio_(sistema)&action=edit&redlink=1) que captura o que um usuário faz ou necessita fazer como parte de sua função de trabalho. Histórias de usuário são usadas com [metodologias ágeis](https://pt.wikipedia.org/wiki/Desenvolvimento_%C3%A1gil_de_software) de [desenvolvimento de software](https://pt.wikipedia.org/wiki/Desenvolvimento_de_software) como a base para definir as funções que um sistema de negócio deve fornecer e para facilitar o [gerenciamento de requisitos.](https://pt.wikipedia.org/wiki/Gerenciamento_de_requisitos) Ela captura o "quem", "o quê" e "por quê" de um requisito em uma forma concisa e simples, geralmente limitada em detalhes, de forma que possa ser escrita a mão em um pequeno cartão de notas de papel.

Segue abaixo as histórias de usuário levantadas neste projeto, baseadas nos casos de uso (veja figura figuras 6 e 7).

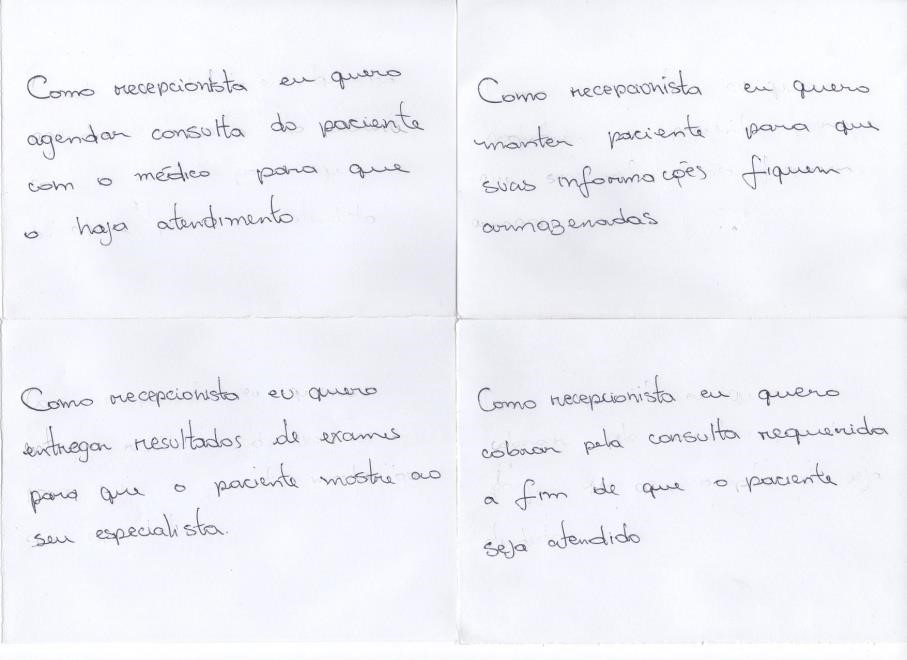


Figura 06 – Frente do cartão de História de Usuário

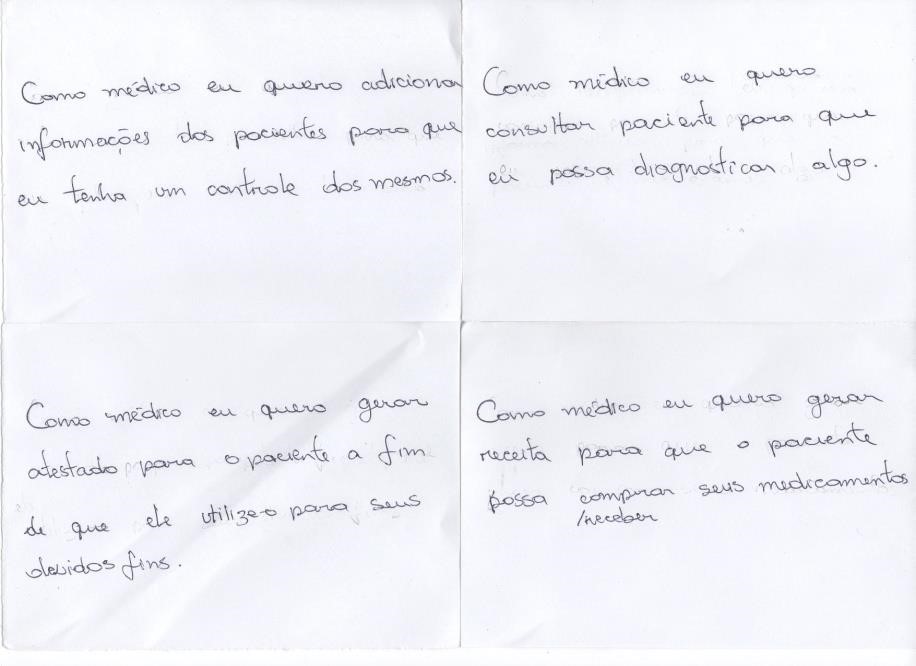


Figura 07 – Verso do cartão de História de Usuário

# 4. Considerações Finais

Com o desenvolvimento deste projeto pode-se aprender como se configura um ambiente de desenvolvimento de software, como se controla suas versões, as mudanças e a auditoria das configurações.

Em relação à computação nas nuvens, observou-se a tendência de utilização, dado suas vantagens econômicas, reduzindo o custo de infra-estrutura. Os usuários necessitam apenas ter em suas máquinas um sistema operacional, um navegador e acesso a Internet. Todos os recursos e processamentos computacionais estão disponíveis na Internet. Assim, as máquinas dos usuários não necessitam ter altos recursos computacionais, diminuindo assim o custo na aquisição de máquinas por parte destes usuários.

Para o desenvolvimento do ciclo de vida de desenvolvimento de software, foi realizado o levantamento de requisitos funcionais através do diagrama de caso de uso da UML, foi realizada a estrutura analítica de projeto, e ainda a utilização de história de usuários.

Como limitações deste trabalho, não foi modelado todas as instâncias de digramas possíveis pela UML, como diagrama de classe, diagrama de sequencia e de atividades.

Como trabalho futuro, pretende-se criar uma interface gráfica via web com acesso a banco de dados para que os usuários listados nos diagramas de caso de uso, possam acessar o prontuário eletrônico do paciente (PEP).

# Referências Bibliográficas

BOOCH, GRADY; RUMBAUGH, JAMES, JACOBSON, IVAR. UML, GUIA DO USUÁRIO. 7ª TIRAGEM. RIO DE JANEIRO: CAMPUS, 2000.

FURUIE S. S. **PRONTUÁRIO ELETRÔNICO EM AMBIENTE DISTRIBUÍDO E HETEROGÊNEO: A EXPERIÊNCIA DO INCOR**. IN: CONGRESSO BRASILEIRO DE INFORMÁTICA EM SAÚDE 8. 2002 NATAL. ANAIS... DISPONÍVEL EM:

HTTP://WWW.SBIS.ORG.BR/INDEXFRAME.HTML. ACESSO EM: 2 ABR 2006.

PRESSMAN, ROGER S. **ENGENHARIA DE SOFTWARE**.6ED. SÃO PAULO: MCGRAW-HILL,

2006.

Silva, Fábia Gama; Tavares Neto, José. **Avaliação dos prontuários médicos de hospitais de ensino do Brasil**. [Revista Brasileira de Educação Médica.](http://www.scielo.br/scielo.php?script=sci_serial&pid=0100-5502&lng=en&nrm=iso) On-line version ISSN 1981-5271. Rev. bras. educ.

med. vol.31 no.2 Rio de Janeiro May/Aug. 2007.

SIQUEIRA, RODRIGO GEORGE PIUBELLO. **PLANEJAMENTO DE ESCOPO DE PROJETOS: O CASO DE UMA CONSULTORIA.** MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE

PRODUÇÃO. UNIVERSIDADE Federal de Juiz de Fora. Pesquisada em janeiro de 2015 na url [http://www.fmepro.org/XP/editor/assets/DownloadsEPD/TCC\_dez2007\_RodrigoPiubello.pdf,](http://www.fmepro.org/XP/editor/assets/DownloadsEPD/TCC_dez2007_RodrigoPiubello.pdf) 2007.

Sousa, Flávio R. C.; Moreira, Leonardo e Machado, Javam. **Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios**. Minicurso realizado na Universidade Federal do Piauí.

Pesquisa realizada em janeiro de 2015 na url http://www.ufpi.br/subsiteFiles/ercemapi/arquivos/files/minicurso/mc7.pdf

STOLF, GIULIANO MÁRCIO. **SISTEMA WEB GERENCIADOR DE CLÍNICA MÉDICA: AUTOMATIZANDO A CLÍNICA CARDIOMED**. MONOGRAFIA DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO. UNIVERSIDADE FEDERAL DE BLUMENAU. Pesquisada em janeiro de 2015 na url [http://dsc.inf.furb.br/arquivos/tccs/monografias/2007-1giulianomarciostolfvf.pdf,](http://dsc.inf.furb.br/arquivos/tccs/monografias/2007-1giulianomarciostolfvf.pdf) 2007.

XAVIER, Carlos Magno da Silva. **Gestão de projetos: como definir e controlar o escopo do projeto**. 1 ed. São Paulo: Saraiva, 2005.