1. Introdução

MOTIVAÇÃO

A cada ano a chamada indústria da saúde cresce no mundo, tornando-se um negócio de investimentos bilionários, a era da caridade, por parte de organizações sociais e até mesmo do governo, acabou.

Para que as corporações se mantenham no mercado, o qual exige grandes investimentos em alta tecnologia, tanto eletrônica quanto farmacêutica, são necessários profissionais dedicados à gestão, e esses devem dominar as técnicas e conhecerem as melhores práticas. Durante muito tempo a gestão dos hospitais foi feita por médicos que compartilhavam outras funções, e em países como o Brasil isso ainda é realidade.

A informatização dos processos ocorre em todos os setores comerciais e industriais, em organizações de cuidado à saúde, por todo mundo, essa tendência vem sendo incorporada, e cada vez mais empresas de software oferecem soluções para gestão das diversas áreas hospitalares. No entanto, no Brasil, programas com essa finalidade são pouco desenvolvidos.

OBJETIVOS

O objetivo geral do trabalho é criar uma ferramenta de apoio a gestão de equipamentos médicos e hospitalares, que forneça dados sobre custos e disponibilidade dos equipamentos, a fim de auxiliar decisões de contratação de manutenção, bem como decisões de compra, além de elucidar questões gerenciais.

Alguns objetivos específicos podem ser citados:

* Cadastro e organização do parque tecnológico de instituições de saúde, em geral;
* Geração de relatórios de custos;
* Geração de relatórios de disponibilidade;
* Avaliação da possibilidade de manutenção interna;
* Controle do estado do equipamento;
* Gerenciamento de ordens de serviço.

PROPOSTA

Esse trabalho propõe a criação de um sistema web, open-source, para o atendimento dos objetivos supracitados, que possa ser implantado em qualquer instituição de saúde que disponha de uma rede e um servidor, internos, ou conexão dos setores gerenciais e de manutenção à internet. Esse sistema deve mudar a cultura das instituições de forma amigável e positiva, provendo maior controle sobre os equipamentos.

JUSTIFICATIVA

Profissionais especializados devem entrar nas instituições de saúde, a fim de gerenciar os recursos financeiros e os recursos humanos, essas instituições devem ser geridas como indústrias, guardadas as devidas proporções. A produção está diretamente relacionada com as vidas e a satisfação dos pacientes, no entanto, todos os gastos, esforços e padrões de qualidade devem ser conhecidos.

As preocupações dos profissionais de saúde, dentre eles médicos e enfermeiros, devem ser diretamente relacionados aos pacientes, as questões administrativas devem caber a outras pessoas especializadas neste assunto, tais como os Engenheiros Clínicos. Estes deverão organizar dados sobre o cotidiano hospitalar, e esperam-se obter informações importantes a partir da análise da relação entre esses dados gerenciados.

Uma pesquisa, realizada em São Paulo, levantou que nos últimos 10 anos, 20 hospitais particulares fecharam na cidade, três unidades deixaram de existir no ano de 2011, dentre as principais causas está má gestão. O fechamento mais recente ocorreu depois de uma ação de despejo, por causa disso, alguns pacientes internados tiveram que ser transferidos de forma não planejada, e os prontuários deles foram perdidos, gerando desconforto e retrabalho [1].

1. METODOLOGIA

2.1 TECNOLOGIAS

2.1.1 RUBY ORIENTAÇÃO A OBJETOS

Em um sistema web, o programa é separado em duas partes principais, o lado servidor e o lado cliente. No lado servidor são realizadas as interações com banco de dados e processamento de dados, o servidor se encarrega de enviar os dados resultantes das operações para o cliente. O lado cliente pode ser acessado pelo usuário e até mesmo manipulado, isso não acontece com o programa do lado servidor [7].

Nesse projeto o lado servidor foi programado usando uma linguagem de script, relativamente nova, Ruby em sua versão 1.9. O Ruby é uma linguagem multi-paradigma, no entanto muito utilizada em razão de orientação a objetos, a linguagem prima a legibilidade e a simplicidade do código.

O Ruby é uma linguagem dinâmica, ou seja, não utiliza declaração de tipos, os quais são determinados durante a execução. A linguagem apresenta uma gramática complexa, no entanto muito expressiva, além de biblioteca dotada de muitas funcionalidades pré-construídas [8].

Todos os valores em Ruby são objetos, por esse motivo a linguagem é classificada de puramente orientada a objetos, mesmo contendo capacidades procedimentais e funcionais. Os programas são organizados em Classes e Módulos, uma classe é uma coleção de métodos (funções) relacionados, os quais operam sobre o estado dos objetos, que são instâncias das classes. Um módulo é um conjunto de classes e métodos relacionados, os métodos de um módulo são estáticos, ou seja, não são invocados por objetos, os módulos funcionam como espaço de nomes, nomes utilizados em um módulo podem ser repetidos em outro [8].

O uso de Ruby tem sido cada vez maior, principalmente para aplicações web, devido a expansão da framework Rails, no entanto aplicativos desktop também podem ser desenvolvidos nessa linguagem, que tem sido promovida como a linguagem amiga do programador, devido a boa legibilidade proporcionada e sua API bastante útil. O uso de Ruby com Rails permite às equipes maior concentração nas peculiaridades do projeto desenvolvido, uma vez que as funções comumente utilizadas já estão todas implementadas na API [7, 8].

2.1.2 BANCO DE DADOS DOCUMENTOS

Os bancos de dados orientados a documentos existem desde o início do arquivamento digital de dados, no entanto nos últimos anos esse tipo de arquitetura de dados foi aprimorado, e tem sido usada em várias aplicações web, principalmente por sua escalabilidade, ou seja, a capacidade de atender mais, ou menos, clientes de acordo com a oscilação da demanda.

O banco de dados utilizado nesse projeto foi o MongoDB, sistema bastante novo, o qual lançou sua primeira versão estável em 2005. MongoDB apresenta algumas funcionalidades úteis de modelos relacionais, como índices e ordenação, além de fornecer algumas funções pré-estabelecidas para agrupamento de dados que estejam separados em vários documentos.

A principal vantagem do modelo orientado a documentos é a flexibilidade, uma vez que diferentemente do relacional, não há esquema definido, não existem tabelas com colunas fixas para cada linha. Documentos que representem o mesmo tipo de objeto podem apresentam dados diferentes com tipos diferentes, o que facilita a administração e a manutenção. Esse modelo é muito bom para representação de hierarquias complexas, uma vez que documentos podem ser inseridos uns dentro dos outros, de tal forma que uma consulta apenas no banco de dados pode trazer a hierarquia completa, o que significa muitas vezes todos os dados necessários para as operações.

Os documentos são conjuntos de pares “chave-valor”, no caso do MongoDB os documentos são objetos JavaScript, JSON, e a linguagem de comunicação com o banco é também JavaScript em substituição ao SQL. O conjunto de documentos é chamado de coleção, e cada coleção pode ser armazenada em um arquivo separado, o qual é encriptado, o conjunto dessas coleções forma um banco de dados. O MongoDB interpreta ações escritas em javascript, e em seguida manipula os arquivos das coleções de acordo com a ação desejada.

Nesse projeto foi utilizado o driver de Ruby para MongoDB, e ainda uma biblioteca chamada MongoMapper para o uso do modelo Rails, de forma não ser necessário a escrita de código em JavaScript para a manipulação do banco de dados.

2.1.3 RAILS FRAMEWORK MVC

O Rails é um framework para construções de aplicações web, baseada em Ruby, esse framework é a compilação de várias gemas, espécie de biblioteca do Ruby, as quais apresentam funções e características comuns às aplicações já implementadas. Para a melhor utilização do framework é necessário seguir o “jeito Rails”, ou seja, é preciso seguir uma série de convenções, a fim de evitar novas configurações. Com o uso de Rails é possível construir aplicações web poderosas em alta velocidade e com muito agilidade, as quais podem ser utilizadas sem a necessidade de download de plugins.

A grande vantagem da construção de aplicações na internet é a facilidade de distribuição. Aplicações web podem ser construídas aos poucos e cada nova funcionalidade estará disponível para todos os clientes, uma vez que o código está instalado no servidor, e os clientes não precisam fazer nenhuma atualização em seus computadores.

Algumas das características relevantes do Rails são [7]:

* Inclui toda estrutura necessária para a construção da aplicação, em muitos casos dispensando instalação e configuração de bibliotecas;
* Código aberto, sob a licença .....;
* Independente de plataforma;
* Uso do padrão Model, View, Controller(MVC), cada parte do programa tem seu lugar apropriado, incentivo ao uso de boas práticas;
* Camada de manipulação de dados, estruturada de forma que para qualquer banco de dados utilizado o código é o mesmo;
* Comunidade de programadores extensa e ativa, permitindo o esclarecimento rápido de dúvidas;

O Rails utiliza o padrão MVC, um padrão de projeto que separa claramente a lógica, a manipulação de dados e a apresentação. Nessa arquitetura, o model representa o dados e lida com o banco, a view representa a interface com o usuário, e o controller direciona as ações relacionando os dados com a interface. Dessa forma, o código fica funcionalmente divido e o trabalho pode ser separado em especialidades [7].

A Figura, abaixo, ilustra o fluxo de interações proporcionado pelo padrão MVC. Geralmente o fluxo se dá seguinte forma, uma interface é apresentada ao usuário, o usuário gera um evento na interface, esse evento é enviado ao controller que sabe a qual model deve requisitar dados, então os dados são passados para a view que é renderizada com as informações atualizadas para o usuário [7].



Retirada de 7

2.1.4 jQuery, HTML e CSS

Em uma aplicação web o lado cliente, ou seja, a parte do programa que é transferida para o computador do usuário, pode ser construída através da agregação de vários elementos. Geralmente três elementos são utilizados, o primeiro fornece a estrutura da página, o segundo personaliza a página com cores, tamanhos, bordas e distâncias, e o terceiro adiciona interatividade e movimento. Nesse projeto o primeiro elemento é o HTML, e o segundo é o CSS, e o terceiro é o jQuery um framework JavaScript, que oferece de maneira amigável várias funções comumente usadas, que permite o aumento da portabilidade e principalmente da agilidade dos projetos.

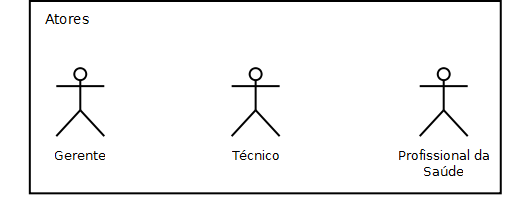
Os elementos citados são utilizados para a construção das interfaces com usuário, que são as Views no padrão de projeto utilizado. Para a utilização dessas ferramentas algumas convenções são seguidas, as quais são estabelecidas pelo W3C, órgão que regulamenta as ferramentas para desenvolvimento web, o uso dessas convenções tenta garantir que as páginas sejam renderizadas da mesma forma nos diferentes browsers. As maneiras diferentes dos browsers de interpretar os códigos de personalização e interatividade tem sido grande fonte de problemas para a portabilidade das aplicações web.

Através do uso daqueles elementos, e possível criar páginas ricas e intuitivas que funcionem em qualquer sistema operacional, sem o uso de nenhum um plugin adicional, como Flash Player.

2.1.5 GIT CONTROLE DE VERSÃO

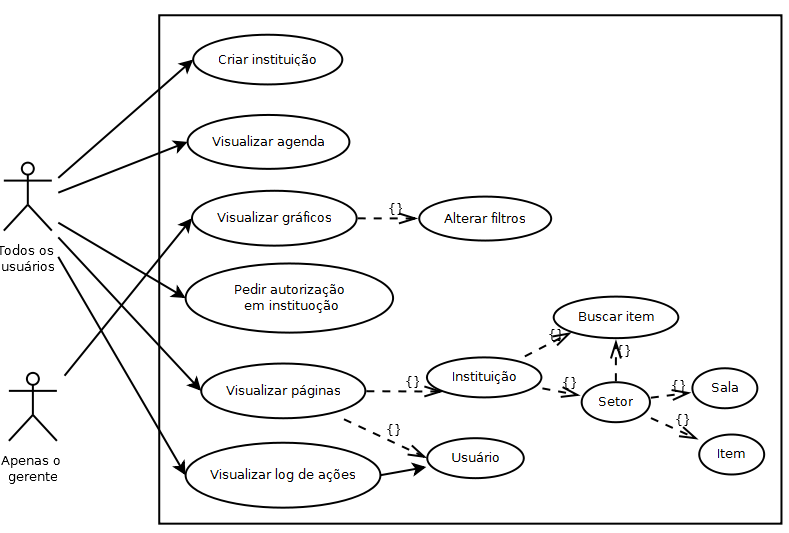
2.2 CASOS DE USO

O sistema apresenta três tipos de usuários, para os casos de uso considerados atores, os quais são: gerente, técnico e profissional de saúde. Esses atores podem realizar suas atividades no mesmo ambiente de trabalho ou não. Os três tipos de usuário estão relacionados entre si através de uma instituição. A Figura X mostra os atores.



O diagrama de casos de uso foi separado em três partes, as quais contêm ações relacionadas. Na primeira parte (Figura X) estão ações genéricas, comuns aos atores. Na segunda parte (Figura X) encontram-se as ações para população da base de dados. Na última parte (Figura X) estão as ações específicas dos envolvidos.

2.2.1 DIAGRAMA PARTE 1



2.2.1.1 Criar instituição

A idéia por trás do sistema é criar um perfil de usuário envolvido com equipamentos médicos, dessa forma cada pessoa que possui um perfil pode participar de diferentes instituições em diferentes funções.

Qualquer pessoa que tenha um perfil pode criar uma instituição, dessa forma se tornará o gerente desta. Todas as ações do sistema dependem da existência de uma instituição. A ação de criar instituição estará disponível no menu da página do usuário

A Figura X mostra o diagrama de sequência para esse caso de uso.

2.2.1.2 Visualizar agenda

Os usuários poderão visualizar a agenda da instituição, a fim de verificar se os equipamentos poderão estar indisponíveis em alguma data.

Na agenda estarão registrados os seguintes eventos: aquisição de itens, vencimento de garantia de itens e manutenções preventivas. A adição de calibrações com freqüência em dias está prevista para a agenda.

A Figura X mostra o diagrama de sequência para esse caso de uso.

2.2.1.3 Visualizar gráficos

O gerente de uma instituição poderá fazer uma análise rápida da mesma através de gráficos.

Na página de gráficos ele poderá ver: a situação dos equipamentos, os custos de manutenção dos últimos três meses, a quantidade de serviço nos últimos três meses, o tempo médio em dias para realização de um serviço também nos últimos três meses.

Para realização de uma análise mais apurada, alguns filtros estão previstos para os dados dos gráficos. Os filtros seriam: por setor, por marca e por modelo. Além disso, poderá ser adicionada a possibilidade de escolha da janela de tempo, em substituição aos últimos três meses.

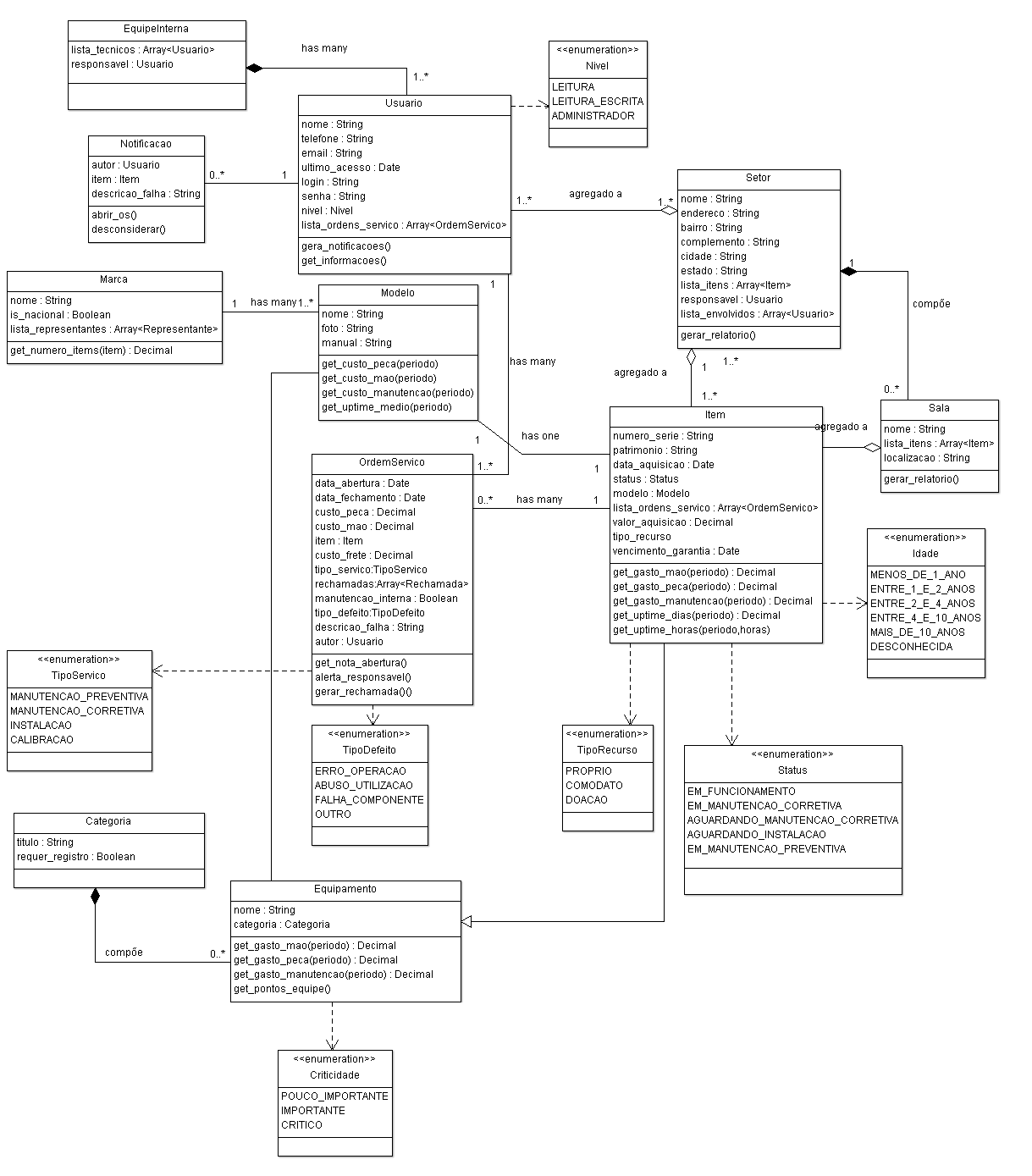
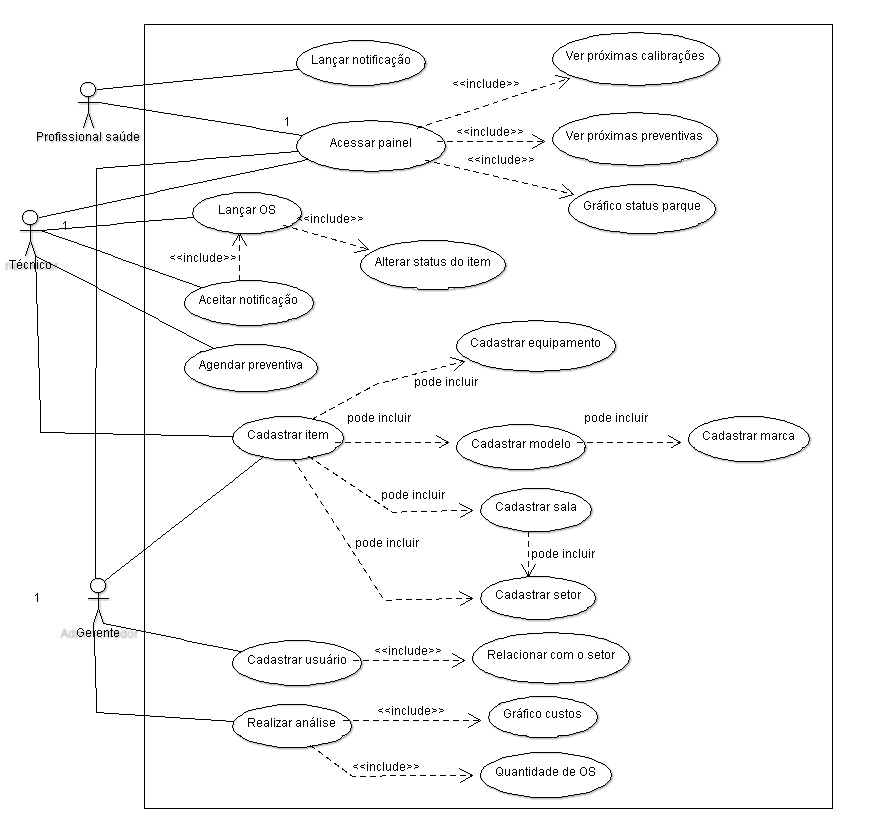
A Figura X mostra o diagrama de sequência para esse caso de uso.

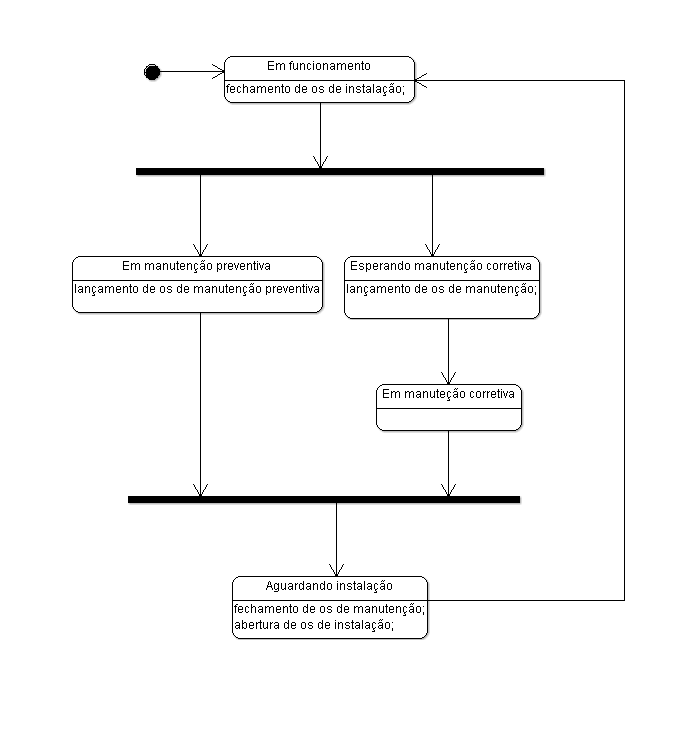
2.2.1.4 Alterar filtros

Para realização de uma análise mais apurada, alguns filtros estão previstos para os dados dos gráficos. Os filtros seriam: por setor, por marca e por modelo. Além disso, poderá ser adicionada a possibilidade de escolha da janela de tempo, em substituição aos últimos três meses.

A Figura X mostra o diagrama de sequência para esse caso de uso.

2.2.1.4 Alterar filtros





* 1. DESCREVER O PASSO A PASSO DO DESENVOLVIMENTO DO SISTEMA.
  2. EXPLICAR AS ETAPAS E EXEMPLIFICAR COM OS DIAGRAMAS

1. RESULTADOS
   1. O SISTEMA – TELAS E FUNCIONAMENTO COM SIMULACAO DE DADOS
2. CONCLUSÃO

REFERENCIAS

[1] Artigo revista sobre fechamento hospitais

[3] ANTUNES, E.; VALE, M.; MORDELET, P.; GRABOIS, V.; Gestão da Tecnologia Biomédica – Tecnologia e Engenharia Clínica. Paris – França – ACODESS, 2002.

[5] CALIL, S.J.; TEIXEIRA, M.S.; Gerenciamento de Manutenção de Equipamentos Hospitalares, Coleção Saúde & Cidadania, Vol. 11, São Paulo, IDS – EFP, 1998.

[6] MINISTÉRIO DA SAÚDE, Equipamentos Médico-Hospitalares e o Gerenciamento da Manutenção, Projeto REFORSUS – GEMA, Brasília, Editora MS, 2002.

[2] MALAGÓN-LONDOÑO, G.; MORERA, R.G.; LAVERDE,G.P.; Administração Hospitalar, Rio de Janeiro, Editora Guanabara Koogan, 2ª ed. 2003.

[4] Portaria Nº 648 GM/2006

[7] Livro de Rails

[8] Livro de Ruby