

Luiz Antonio Viana da Silva

INFORMATIZAÇÃO DA SALA DE VACINAS DO SUS (SISTEMA ÚNICO
DE SAÚDE)

Assis
2008

INFORMATIZAÇÃO DA SALA DE VACINAS DO SUS (SISTEMA ÚNICO DE SAÚDE)

Luiz Antonio Viana da Silva

Trabalho de Conclusão de Curso apresentado ao Instituto municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora.

Orientador: DOMINGOS DE CARVALHO VILLELA JÚNIOR

Analisador (1): DOUGLAS SANCHES DA CUNHA

Analisador (2): ALEX SANDRO ROMEO DE SOUZA POLETTO

Luiz Antonio Viana da Silva

INFORMATIZAÇÃO DA SALA DE VACINAS DO SUS (SISTEMA ÚNICO DE SAÚDE)

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora.

Orientador: DOMINGOS DE CARVALHO VILLELA JÚNIOR

Área de Concentração: DESENVOLVIMENTO DE SISTEMA EM ASP

Assis
2008

DEDICATÓRIA

Dedico o presente trabalho em primeiro lugar a Deus que tem estado comigo em todos os momentos de minha vida, me dando capacidade e condições para chegar até aqui, todos que de certa forma contribuíram para minha formação acadêmica, particularmente aos professores desta instituição de ensino superior, ao meu professor orientador, aos companheiros de sala, que por diversos atos me ajudaram e incentivaram minha caminhada e formação. E em especial, minha esposa que me ajudou em todos os momentos, e também minha família que sempre me incentivou e esteve ao meu lado.

RESUMO

Este trabalho tem por objetivo informatizar a sala de vacinas da unidade do SUS (Sistema Único de Saúde) oferecendo uma maior facilidade em trabalhar no setor com o desenvolvimento das tarefas e realização das rotinas. O sistema a ser desenvolvido e implantado visa em melhorar a qualidade no atendimento dos pacientes com o ganho de tempo e qualidade no atendimento. O sistema irá facilitar de modo geral todas as funções que são feitas manualmente garantindo, redução das falhas de ortografia em relação ao preenchimento de cadastros manuais e datas, rapidez em encontrar registros e atualizá-los e também obter um relatório completo dos pacientes e vacinas. Enfim, a informatização da sala de vacinas trará muitos benefícios tanto aos pacientes atendidos quanto para os responsáveis que atuam no departamento local.

Palavras Chave: ASP: Active Server Pages

ABSTRACT

This work has the objective of to computerize the SUS vaccine's unity, offering more facility in working in the task's development and the realization of the routines.

The system to be developed and implanted tries to become better the attendiment of the patients, using less time for this.

The system will become lazy all the functions that are realized in manual style, maintaining reduction of spelling's faults related to the fulfilness the manual's registration and dates.

This will also offer fastness to find register and actualize them and even getting full reports and vaccines. So, the computerizing of the vaccine's unity will bring so many benefits to the patients and also for the responsible in the act of the local department.

Keywords: ASP Active Server Pages

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de Use Case *efetuar login de acesso*	22
Figura 2 – Diagrama de Use Case *cadastrar pacientes*	23
Figura 3 – Diagrama de Use Case *cadastrar vacinas*	24
Figura 4 – Diagrama de Use Case *cadastrar insumos*	25
Figura 5 – Diagrama de Use Case *cadastrar funcionários*	26
Figura 6 – Diagrama de Use Case *manter peso*	27
Figura 7 – Diagrama de Use Case *atualizar cadastros*	28
Figura 8 – Diagrama de Use Case *efetuar buscas*	29
Figura 9 – Diagrama de Use Case *atualizar estoque insumos*	30
Figura 10 – Diagrama de Use Case *atualizar estoque vacinas*	31
Figura 11 – Diagrama de Use Case *vacinar paciente*	32
Figura 12 – Diagrama de Use Case *relatório vacinas por paciente*	33
Figura 13 – Diagrama de Use Case *relatório vacinas por paciente*	34
Figura 14 – Diagrama de Use Case *relatório vacinas por insumos*	34
Figura 15 – Diagrama de Use Case *histórico de pacientes*	35
Figura 16 – Diagrama de Use Case *histórico de vacinas*	35
Figura 17 – Diagrama de Use Case *histórico insumos*	36
Figura 18 – Relacionamento entre os diagramas de classe	37

LISTA DE TABELAS

Tabela 01 – Elementos da linguagem SQL.....	17
Tabela 02 – Descrição da lista de eventos.....	21
Tabela 03 – Cronograma do Projeto.....	40
Tabela 04 – Tabela final dos custos.....	41

LISTA DE ABREVIATURAS E SIGLAS

HTML	HyperText Markup Language
ASP	Active Server Pages
VB SCRIPT	Visual Basic Script Language
ASCII	American Standard Code for Information Interchange
IIS	Internet Information Server
SGBD	Sistema de Gerenciamento de Banco de dados
SQL	Structured Query Language
SUS	Sistema Único de Saúde
DCL	Data Control Language
DDL	Data Definition Language
DML	Data Manipulation Language
CSS	Cascading Style Sheets

SUMÁRIO

1. INTRODUÇÃO.....	12
2. DESCRIÇÃO DO AMBIENTE DE DESENVOLVIMENTO.....	13
2.1 LINGUAGEM DE PROGRAMAÇÃO ADOTADA.....	13
2.1.1 ASP.....	13
2.1.2 Aplicações Cliente Servidor.....	14
2.1.3 Client Side Script.....	14
2.1.4 Server Side Script.....	14
2.1.5 Definição de Script.....	15
2.1.6 Visual Basic Script Language (VBScript).....	15
2.1.7 Definição de Html.....	16
2.1.8 Definição de Javascript.....	16
2.2 FERRAMENTA DE DESENVOLVIMENTO.....	18
2.2.1 Macromedia Dreamweaver CS3.....	18
2.2.2 Banco de Dados SQL Server 2000.....	19
3. ENTREVISTA PADRÃO.....	21
3.1 DESCRIÇÃO DO PROBLEMA.....	21
3.2 RESULTADO ESPERADO.....	21
4. ANÁLISE DE REQUISITOS.....	22
4.1 LEVANTAMENTO DE REQUISITOS.....	22
5. ANÁLISE.....	23
5.1 LISTA DE EVENTOS.....	23
6. DIAGRAMAS.....	24
6.1 DIAGRAMAS DE USE CASE.....	24
7. DIAGRAMAS DE CLASSE.....	39
8. ESPECIFICAÇÃO DE CUSTOS.....	40
8.1 RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO.....	40
8.2 ESTIMATIVAS DE CUSTOS.....	40
8.3 CRONOGRAMA DO PROJETO.....	42
8.4 ORÇAMENTO DO PROJETO.....	43

8.5 CONCLUSÃO.....	44
9. BIBLIOGRAFIA.....	45
10. ANEXOS.....	46
10.1 INTERFACES DO SISTEMA.....	46

1. INTRODUÇÃO

O SUS (Sistema Único de Saúde) possui unidades em todo país, mas ainda com pouca informatização.

Na cidade de Palmital os métodos manuais de coleta e armazenamento de dados causam demora no atendimento, pessoas impacientes nas filas e falhas no procedimento de preenchimento de cadastros. Além disso, todas as informações armazenadas em fichas podem facilmente ser alteradas ou até mesmo perdidas pelo fato de todos os funcionários do local terem acesso livre à sala de vacinação.

Diante disso, verificou-se a necessidade da criação de um sistema de informação para a sala de vacinação.

O desenvolvimento desse sistema será uma excelente ferramenta na área da saúde, já que no momento o departamento da sala de vacinas do SUS da unidade de Palmital não possui informatização e todos os procedimentos são realizados manualmente.

O sistema irá facilitar de modo geral todas as funções que são feitas manualmente com ganho de tempo, redução das falhas de ortografia em questão de preenchimento de cadastros manuais e datas, rapidez em encontrar registros e atualizá-los.

2. DESCRIÇÃO DO AMBIENTE DE DESENVOLVIMENTO

2.1 LINGUAGEM DE PROGRAMAÇÃO ADOTADA

2.1.1 ASP

ASP (Active Server Pages) é uma linguagem utilizada para a criação de páginas dinâmicas. É uma tecnologia criada pela Microsoft em um ambiente de programação que fornece a habilidade de combinar HTML, *scripts* e componentes para desenvolver aplicações que rodam em seu servidor IIS (Internet Information Server). As páginas contêm arquivos de extensão (.asp) que utiliza combinações de Server-Side script e tags Html, já que o código de programação existente em Asp é executado no servidor, que retorna respostas em HTML padrão. Com isso os códigos ficam protegidos, visto que o servidor retorna somente códigos HTML. Uma aplicação em ASP pode conter linhas de Client-Side e Server-Side script que serão executados. (<http://www.apostilando.com>)

Com isso se tem uma vantagem em aplicações Asp que não possuem Client-side script pode ser acessado por qualquer browser existente no mercado. As páginas em ASP podem ser hospedadas em um servidor Windows NT server com o IIS a partir da versão 3 ou superior. ASP na verdade é uma junção de VBSCRIPT e Objetos Activex, é uma linguagem criada a partir do Visual Basic, mas com limitações por motivos de segurança.

Objetos Activex são funções prontas já existentes, ou seja, várias bibliotecas reunidas, essas funções captam parâmetros de entrada manipula-os de acordo com sua função e envia-os para saída. Quando em uma programação a linguagem ASP for usada em alguns pedaços de páginas HTML usa-se um símbolo `<% asp %>` para diferenciar ASP de HTML.

ASP veio para complementar à linguagem HTML, trazendo todas as vantagens de acesso a banco de dados, acesso a arquivos de texto, captação de informações de

formulários, captação de informações sobre cliente servidor e uso de variáveis loops etc.

2.1.2 Aplicações Cliente Servidor

É uma divisão de processos entre cliente e servidor, com a finalidade de obter melhor desempenho, menor tempo de resposta e facilidade de manutenção.

Na arquitetura cliente servidor, o servidor é responsável por armazenar e gerenciar os dados da aplicação. O cliente solicita dados e/ou processos ao servidor que retorna para o cliente o que lhe foi solicitado.

O cliente conterá a aplicação visual do sistema, que poderá ser uma aplicação Web, o servidor conterá as regras de negócio.

2.1.3 Client Side Script

São códigos processados pela estação cliente (local), geralmente em aplicações voltadas para Internet o código que é executado no local que gerencia apenas consistências simples como telas e validações de entrada de dados. Em aplicações Web os client-side scripts terão de ser processados em um *browser*, um dos problemas enfrentados hoje é a incompatibilidade de interpretação da linguagem entre os *browsers*.

2.1.4 Server Side Script

São os códigos de programação processados no servidor, dessa forma, não é necessário se preocupar com a linguagem que o código foi desenvolvido, o servidor

é que se encarrega de interpretá-lo e devolver uma resposta. Em páginas ASP são esses códigos os responsáveis pelos resultados apresentados.

2.1.5 Definição de Script

É um programa escrito em uma determinada linguagem de programação que não é compilada, mas sim interpretada, seus comandos são lidos em tempo de execução processados e seus resultados passados para a saída padrão da aplicação (monitor, impressora, servidor web etc), são miniprogramas interpretados e voltados para funções específicas. As linguagens scripts foram desenvolvidas para permitir a criação de aplicações para Internet rapidamente, permitindo criar funções como:

- Alterar linhas do *browser*.
- Definir *timeout* para procedure.
- Rodar telas de alerta, confirmação e *inputs*.
- Alterar cor de fundo e barras de títulos.
- Executar funções no *browser*.

2.1.6 Visual Basic Script Language (VBScript)

VBSCRIPT é uma linguagem criada a partir do Visual Basic, mas com algumas limitações, não sendo possível criar uma página, por exemplo, que tenha acesso ao sistema de arquivos do computador e possa apagar ou modificar informações. (<http://www.aspbrasil.com.br>)

Porém diversos cuidados foram tomados pela Microsoft, para tornar a VBSCRIPT uma linguagem segura baseados nas funcionalidades do Visual Basic é uma das muitas possibilidades de linguagem *script* que roda em um servidor, e no IIS ela é uma linguagem padrão. Algumas características do VBSCRIPT:

- Permite manipulação de *strings*, datas, numéricos.
- Permite a utilização de todos os comandos do *Visual Basic*
- O VB SCRIPT é interpretado pelo servidor quando estiver entre tags `< > </>`.

Exemplo de uma função em VBSCRIPT:

```
< SCRIPT LANGUAGE = VBScript runat = server >  
Function RetornaData ()  
RetornaData = Date  
End Function</SCRIPT>.
```

2.1.7 Definição de Html

HTML é uma linguagem de marcação hiper texto especializado na qual se definem as páginas web. Trata-se de um conjunto de etiquetas, códigos especiais chamados (tags) que tratam de definir as formas nas quais as páginas e seus conteúdos serão apresentados, que na verdade são os comandos da linguagem HTML. (RAMALHO, 1996)

Quando o *browser* exibe uma página ele lê somente o texto da página e procura os símbolos especiais que são as tags que dizem como as informações devem ser exibidas. A tag especifica como o texto será apresentado, fonte, tamanho, cor etc. Pode-se também determinar endereços de outras páginas a serem acessadas quando o texto for selecionado. O documento em HTML é um arquivo texto comum ASCII (*American Standard Code for Information Interchange*) que pode ser escrito em qualquer editor de texto comum e tem a extensão (. htm ou . html), e por ser uma linguagem interpretada ao salvar o arquivo já pode ser aberto em qualquer navegador.

As tags são identificadas por serem envolvidos por: < > ou </ >. Entre os sinais < >, são especificados os comandos propriamente ditos.

2.1.8 Definição de Javascript

Javascript é uma linguagem de script utilizada para desenvolver aplicações, em páginas Web ou servidores Web. Os comandos e funções de *Javascript* são

inseridos em um documento da web, com tags HTML e texto. Quando o navegador de um usuário acessa o documento, ele formata a página executando o programa nela inserido. Para acessar uma página que possui scripts, o navegador deve ser capaz de interpretar a linguagem. *Javascript* é uma linguagem baseada em objetos com um conjunto de objetos já embutidos. Sempre que acontece algo em uma página, ocorre um evento. *Javascript* é uma linguagem dirigida por eventos, projetada para reagir quando um evento ocorre. (<http://www.criarweb.com>)

A linguagem *Javascript* foi projetada para manipular e apresentar informações através de um navegador, ela não é capaz de recuperar informações de outro arquivo ou salvar dados em um servidor ou no computador do usuário. Com *Javascript* podem-se criar efeitos nas páginas e definir interatividade com o usuário. Também é compatível com a maioria dos navegadores. *Javascript* é na verdade uma excelente linguagem de programação que oferece ao programador Web muitos recursos, os quais podem ser executados facilmente.

2.2. FERRAMENTA DE DESENVOLVIMENTO

2.2.1 Macromedia Dreamweaver CS3

O Macromedia Dreamweaver é um editor profissional que cria e gerencia sites e páginas Web. Com esse programa é fácil criar e editar páginas compatíveis com diversos navegadores e plataformas, usando ferramentas avançadas de desenho e layout. A verificação do navegador de destino detecta os problemas potenciais que seu trabalho possa apresentar em todas as plataformas e navegadores. A tecnologia Roundtrip HTML da Macromédia importa documentos HTML sem reformatar o código e o Dreamweaver pode ser definido para limpar e reformatar o HTML quando você quiser. O Dreamweaver pode ser totalmente personalizado, você pode criar os seus próprios objetos e comandos, modificar os menus e atalhos de teclado, além de gravar código JavaScript para estender os recursos do Dreamweaver com novos comportamentos e inspetores de propriedades. (YNEMINE, 2001)

Trabalhar com Dreamweaver é tão fácil quanto abrir ou criar um novo documento HTML. Para experimentar amplamente o Dreamweaver, devem-se compreender os conceitos básicos subjacentes à área de trabalho do programa e saber escolher as opções mais adequadas ao seu estilo de trabalho. A área de trabalho do Dreamweaver é flexível, os códigos gerados são também automaticamente gerados na parte visual assim possibilitando ao desenvolvedor ver como a parte da interface esta ficando ao mesmo tempo em que desenvolve seus códigos, as formatações de fontes cores botões e muitos outros detalhes de interfaces podem ser facilmente gravados no estilo CSS (*Cascading Style Sheets*) que o Dreamweaver possui e utilizá-los no decorrer da criação da página usando um mesmo padrão sem a necessidade de ter que formatar a todo o momento, possui também diversos estilos de trabalho e níveis de experiência.

2.2.2 Banco de Dados SQL Server 2000

O SQL Server é um SGDB (Sistema de Gerenciamento de Banco de Dados) muito robusto baseado na arquitetura cliente servidor. Com a expansão das redes locais de computadores e do modelo cliente servidor, a utilização dos chamados bancos de dados relacionais cresceu bastante. Hoje a maioria das aplicações se baseia nesse tipo de banco de dados. (BATTISTI, 2001)

A linguagem SQL é subdividida em três grupos: DCL, DDL, DML.

Esses comandos são relativamente simples, mas possuem certas cláusulas que podem torná-los complexos. Os comandos da linguagem DCL são utilizados para tratar as permissões que o usuário terá dentro de um sistema. Os DDL são para a criação dos objetos no sistema, já os DML são utilizados para tratar os dados do sistema.

A linguagem SQL é muito eficiente e muito poderosa, mas somente com ela não se pode construir uma aplicação completa, já que a SQL não possui comandos para a construção de programas. Sendo assim, as empresas que produzem os gerenciadores de banco de dados e que utilizam a linguagem padrão SQL precisaram desenvolver uma outra linguagem que contivesse além da SQL comandos de uma linguagem de programação. A Microsoft criou a linguagem Transact-SQL com esse objetivo. Ela contém a linguagem SQL e o comando de uma linguagem procedural.

DCL – Data Control Language – Linguagem de Controle de Dados	
GRANT	Concede permissões
DENY	Nega permissões
REVOKE	Revoga a concessão ou a negação de permissões

DDL - Data Definition Language – Linguagem de Definição de Dados	
CREATE	Cria objetos no sistema
ALTER	Altera a estrutura dos objetos no sistema
DROP	Elimina objetos do sistema

DML – Data Manipulation Language – Linguagem de Manipulação de Dados	
SELECT	Lê dados de tabelas e views
INSERT	Inserir dados em tabelas
UPDATE	Altera dados das tabelas
DELETE	Exclui dados da tabela
BACKUP	Realiza backup dos dados
RESTORE	Restaura dados de um backup

Elementos Procedurais da Linguagem Transact-SQL
--

DECLARE	Declara variável na memória
SET	Atribui um determinado valor a uma variável
SELECT	Lê dados de tabelas e views e acrescenta estes dados em variáveis de memória
IF/ELSE IF/ELSE	Realiza testes de condição em um programa
WHILE	Realiza laço de repetição em um programa
BEGIN/END	Cria um bloco de comandos
PRINT	Escreve na tela do servidor
EXECUTE	Cria Transações
BEGIN TRANSACTION	Fecha transações e desfaz as operações com dados realizados dentro delas
COMMIT TRANSACTION	Fecha transações e confirma as operações com dados realizadas dentro delas

Tabela 01 – Elementos da linguagem SQL

3. ENTREVISTA PADRÃO

Empresa:

Unidade do SUS (Sistema Único de Saúde)

Software a ser desenvolvido:

Informatização da sala de vacinas

3.1 DESCRIÇÃO DO PROBLEMA

O sistema tem por objetivo facilitar o cadastramento de pacientes, atualizar seus dados sempre que necessário, acompanhar o peso dos pacientes recém nascidos, ter um histórico de seus pacientes e seus dados guardados em um banco de dados. Fazer um controle dos funcionários do setor através de solicitação de senhas para atuar na sala de vacinas, terem um controle total do estoque de vacinas através de entradas e saídas e também através de relatórios.

3.2 RESULTADO ESPERADO

Organização dos cadastros dos pacientes e dos funcionários, facilidade e rapidez em obter informações dos pacientes, um ótimo controle do estoque podendo ser consultado a qualquer momento sua quantidade, acesso através de senhas proibindo assim o uso de pessoas sem autorização no local e um banco de dados guardando todas essas informações sem ter a preocupação de perdê-las.

4. ANÁLISE DE REQUISITOS

4.1 LEVANTAMENTO DE REQUISITOS

O levantamento dos requisitos permite conhecer e obter as necessidades que o sistema irá suprir, dados que são fornecidos pela pessoa que o solicitou ou por seus usuários para atender as necessidades no sistema a ser desenvolvido.

Dados necessários que foram solicitados a estar no sistema.

- Cadastro de pacientes
- Cadastro de vacinas
- Cadastro de funcionários
- Cadastro de Insumos
- Vacinar pacientes
- Efetuar login de acesso
- Acompanhamento de pesos dos recém nascidos
- Atualização de cadastros
- Controle do estoque de vacinas e insumos
- Atualização de estoque de insumos e vacinas
- Efetuar busca de pacientes.
- Histórico e relatórios

5. ANÁLISE

5.1 LISTA DE EVENTOS

Nº.	Descrição	Use Cases
01	Funcionário Solicita Acesso	Efetuar Login de acesso
02	Paciente Solicita Cadastro	Cadastrar Paciente
03	Funcionário Cadastra Vacinas	Cadastrar Vacinas
04	Funcionário Solicita Cadastro	Cadastrar Funcionário
05	Funcionário Cadastra Insumos	Cadastrar Insumos
06	Funcionário Acompanha Peso	Manter Controle de Peso
07	Funcionário Atualiza Cadastros	Atualizar Cadastros
08	Funcionário Faz Procuras	Efetuar Buscas
09	Funcionário Atualiza estoque Insumos	Atualizar Estoque Insumos
10	Funcionário Atualiza estoque Vacinas	Atualizar Estoque Vacinas
11	Paciente Solicita Vacina	Vacinar Paciente
12	Emitir Relatório Vacinas por Pacientes	Relatório Vacina por Paciente
13	Emitir Relatório Vacinas por Funcionário	Relatório Vacina por Funcionário
14	Emitir Relatório Insumos por Vacina	Relatório Insumos por Vacina
15	Emitir Histórico Paciente Cadastrados	Emitir Histórico Pacientes
16	Funcionário Emite Histórico Vacinas	Emitir Histórico Vacinas
17	Funcionário Emite Histórico Insumos	Emitir Histórico Insumos

Tabela 02 – Descrição da lista de eventos

6. DIAGRAMAS

6.1 DIAGRAMAS DE USE CASE

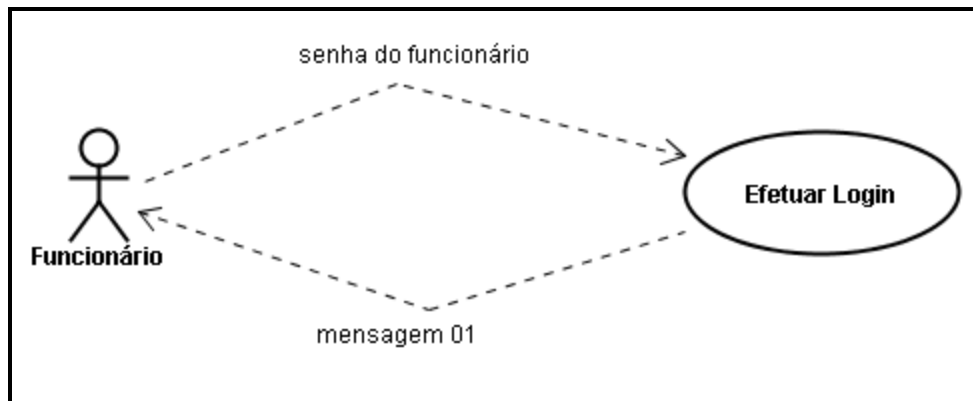


Figura 01 – Use Case *efetuar login *

Curso Normal

- 1 – preencher o campo usuário
- 2 – preencher o campo senha
- 3 – atender solicitação do acesso
- 4 – mensagem 01 = “acesso”

Curso alternativo - 01

- 1 – se campo usuário e senha não estiver preenchidos
- 2 – mensagem 01 = “Favor preencha o Campo”

Curso alternativo – 02

- 1 – verificar se campo usuário e senha estão corretos
- 2 – mensagem 01 = “usuário ou senha inválido tente novamente”

Curso alternativo 02

- 1 – se dados usuário e senha já existirem
- 2 – mensagem 01 = “já existe este funcionário cadastrado”

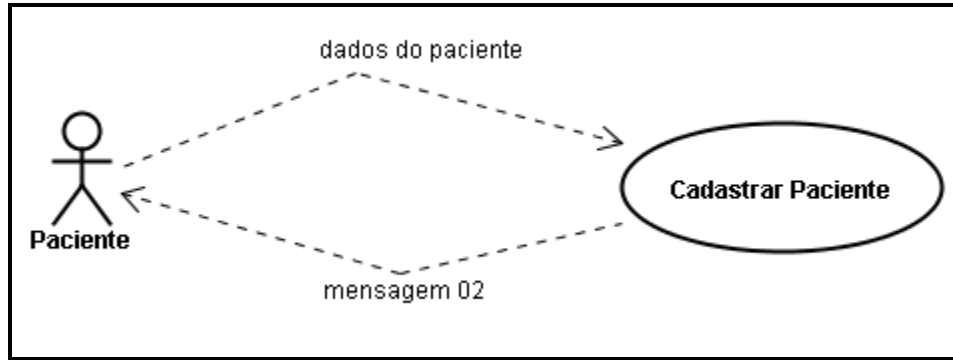


Figura 02 – Use Case *cadastrar pacientes*

Curso Normal

- 1 – verificar dados completos do paciente (nome paciente, nome mãe, data nascimento, local nascimento, endereço, bairro, numero, cidade, telefone, cep, estado).
- 2 – verificar campos obrigatórios preenchidos (nome paciente, nome mãe, data nascimento, cidade).
- 3 – atender solicitação do cadastro
- 4 – mensagem 02 = “paciente cadastrado”

Curso Alternativo 01

- 1 – se campos obrigatórios não forem preenchidos
- 2 – mensagem 02 = “Favor preencha o Campo”

Curso alternativo 02

- 1 – se (nome do paciente, nome da mãe) do paciente já existirem
- 2 – mensagem 02 = “já existe este paciente cadastrado”

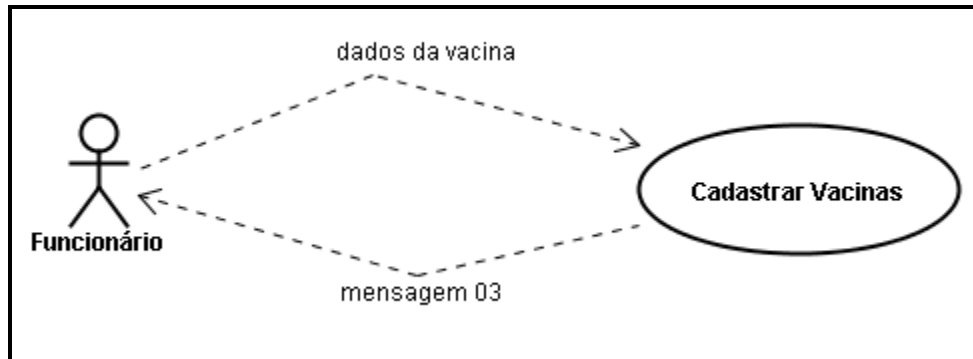


Figura 03 – Use Case *cadastrar vacinas*

Curso Normal

- 1 – verificar dados completos da vacina no cadastro (descrição, lote, quantidade, saldo anterior, saldo atual, data chegada, data vencimento).
- 2 – verificar campos obrigatórios preenchidos (descrição, lote, quantidade, saldo anterior, saldo atual, data chegada, data vencimento).
- 3 – atender solicitação do cadastro
- 4 – mensagem 03 = “vacina cadastrada”

Curso Alternativo 01

- 1 – se campos obrigatórios não forem preenchidos
- 2 – mensagem 03 = “Favor preencha o Campo”

Curso alternativo 02

- 1 – se descrição da vacina já existir
- 2 – mensagem 03 = “já existe esta vacina cadastrada”

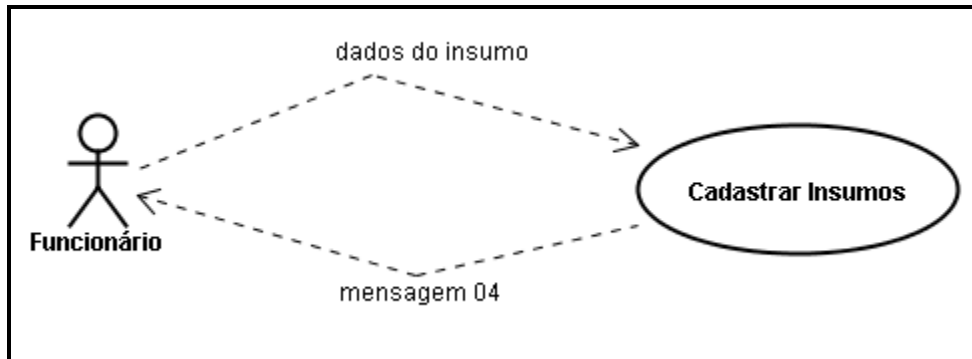


Figura 04 – Use Case *cadastrar insumos*

Curso Normal

- 1 – verificar dados completos do insumo no cadastro (descrição, lote, quantidade, saldo anterior, saldo atual, data chegada, data vencimento).
- 2 – verificar campos obrigatórios preenchidos (descrição, lote, quantidade, saldo anterior, saldo atual, data chegada, data vencimento).
- 3 – atender solicitação
- 4 – mensagem 04 = “insumo cadastrado”

Curso alternativo 01

- 1 – se campos obrigatórios não forem preenchidos
- 2 – mensagem 04 = “Favor preencha o Campo”

Curso alternativo 02

- 1 – se descrição do insumo já existir
- 2 – mensagem 04 = “já existe este insumo cadastrado”

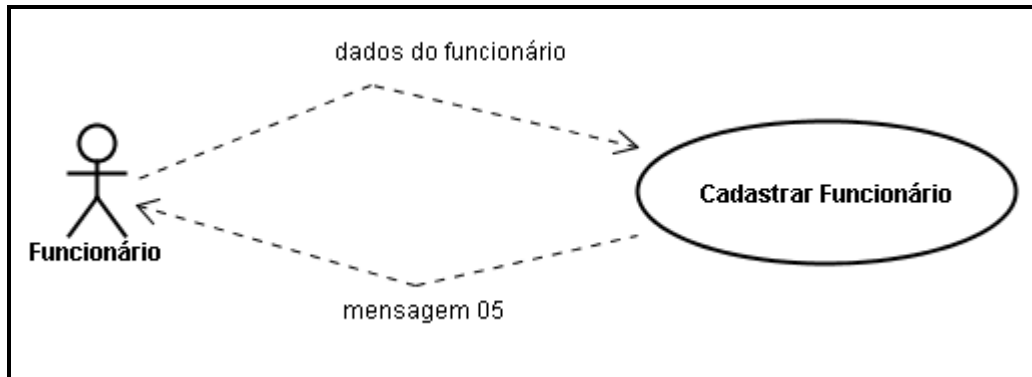


Figura 05 – Use Case *cadastrar funcionários*

Curso Normal

- 1 – verificar dados completos do funcionário (nome, cpf, rg, telefone, endereço, e-mail, bairro, cidade, usuário, senha).
- 2 – verificar campos obrigatórios preenchidos (nome, cpf, rg, usuário, senha).
- 3 – atender solicitação do cadastro
- 4 – mensagem 05 = “funcionário cadastrado”

Curso alternativo 01

- 1 – se campos obrigatórios não forem preenchidos
- 2 – mensagem 05 = “Favor preencha o Campo”

Curso alternativo 02

- 1 – se (RG) do funcionário já existir
- 2 – mensagem 05 = “já existe este funcionário cadastrado”

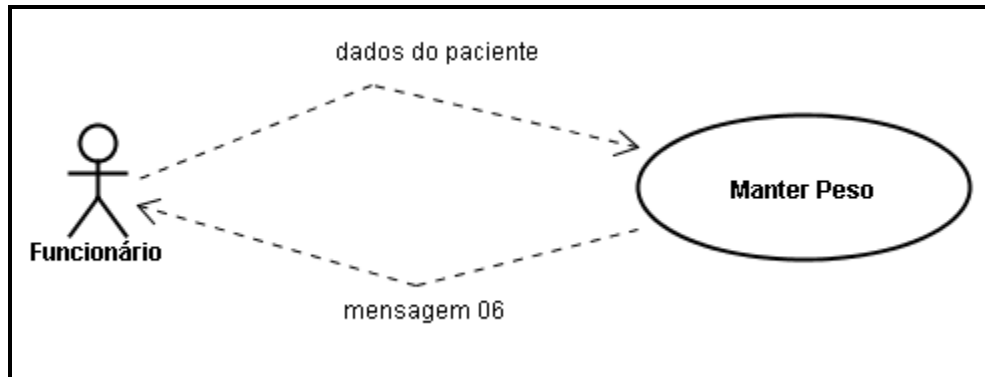


Figura 06 – Use Case *manter peso*

Curso Normal

- 1 – verificar (nome) do paciente
- 2 – verificar peso anterior
- 3 – se o peso atual for igual ao peso anterior
- 4 – mensagem 06 = “atenção não ganhou peso”

Curso alternativo 01

- 1 – se o peso atual for menor ao peso anterior
- 2 – mensagem 06 = “cuidado perdeu peso”

Curso alternativo 02

- 1 – se o peso atual for maior ao peso anterior
- 2 – mensagem 06 = “ótimo ganhou peso”

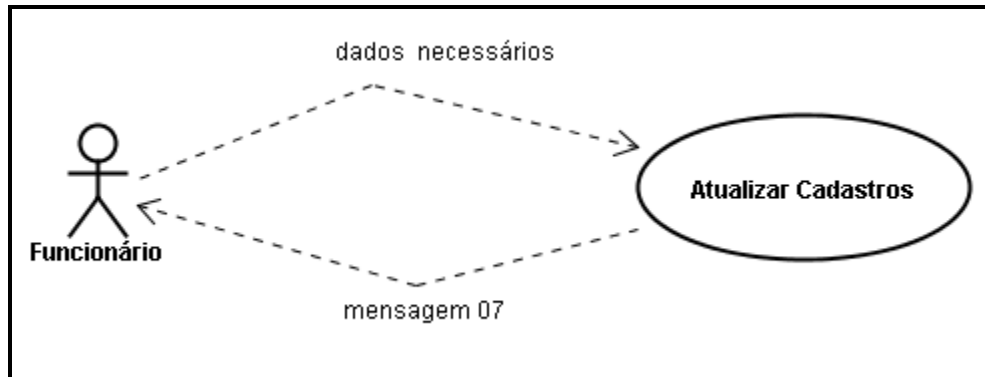


Figura 07 – Use Case *atualizar cadastros*

Curso Normal

- 1 – verificar dados anteriores do paciente (nome paciente, nome mãe, data nascimento, local nascimento, endereço, bairro, numero, cidade, telefone, cep, estado).
- 2 – se dados do paciente for diferentes aos dados atuais
- 3 – atender solicitação
- 4 – mensagem 07 = “dados atualizados”

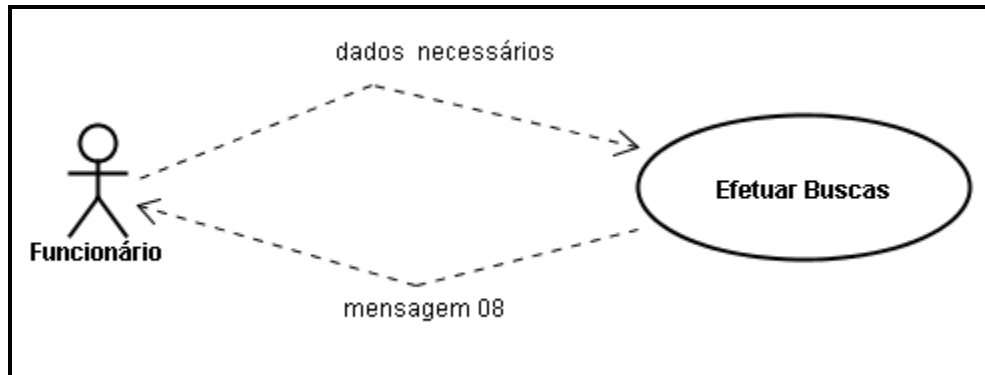


Figura 08 – Use Case *efetuar buscas*

Curso Normal

- 1 – informar dados a ser pesquisado (nome do paciente, ou, nome do funcionário, ou, nome da vacina, ou, nome do insumo)
- 2 – se dados forem encontrados
- 3 – mensagem 08 = “busca completa”

Curso alternativo 01

- 1 – se dados não forem encontrados
- 2 – mensagem 08 = “este cadastro não existe”

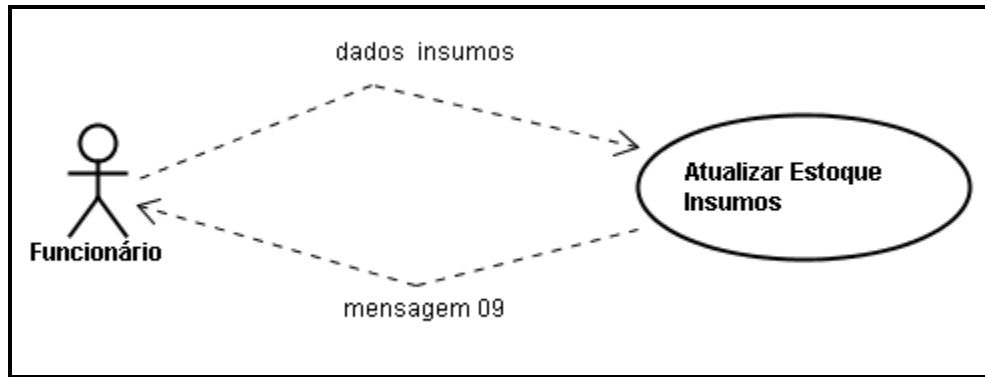


Figura 09 – Use Case *atualizar estoque de insumos*

Curso Normal

- 1 – consultar nome do insumo no estoque
- 2 – conferir quantidade a ser lançada no estoque
- 3 – atualizar quantidade no estoque
- 4 – mensagem 09 = “atualizado”

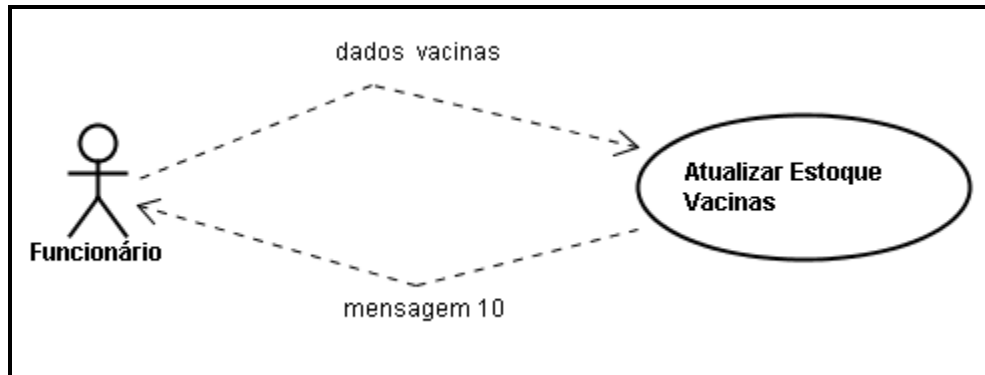


Figura 10 – Use Case *atualizar estoque vacinas*

Curso Normal

- 1 – consultar nome da vacina no estoque
- 2 – conferir quantidade a ser lançada no estoque
- 2 – atualizar quantidade no estoque
- 3 – mensagem 10 = “atualizado”

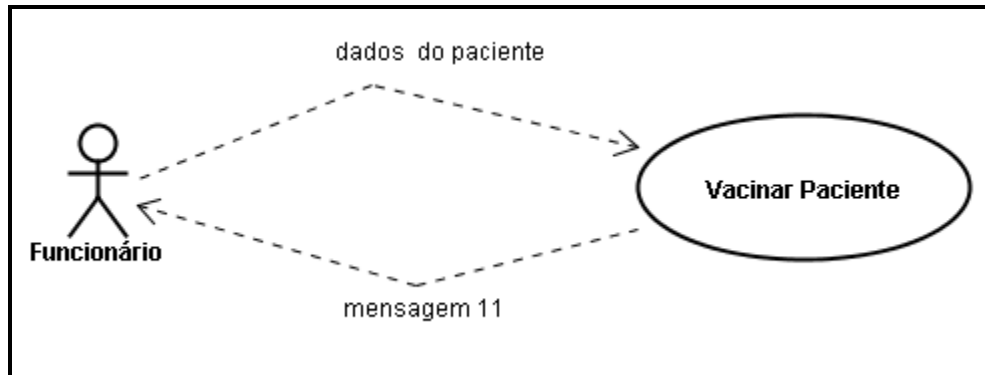


Figura 11 – Use Case *vacinar paciente*

Curso Normal

- 1 – verificar dados do paciente (nome paciente, nome mãe, data nascimento, local nascimento, endereço, bairro, numero, cidade, telefone, cep, estado).
- 2 – verificar se vacina solicitada possui quantidade disponível em estoque
- 3 – atender solicitação
- 4 – mensagem 11 = “paciente vacinado”

Curso alternativo 01

- 1 – se quantidade de vacinas solicitada for maior que disponível em estoque
- 2 – mensagem 11 = “vacina no estoque insuficiente”

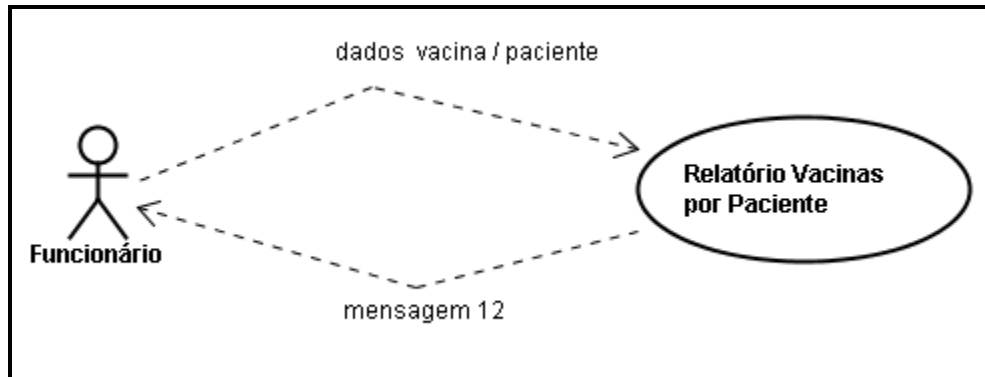


Figura 12 – Use Case *relatório de vacinas por pacientes*

Curso Normal

- 1 – consultar nome do paciente
- 2 – escolher data
- 4 – verificar quantidade de vacinas solicitadas pelo paciente
- 5 – atender solicitação
- 6 – mensagem 12 = “Relatório”

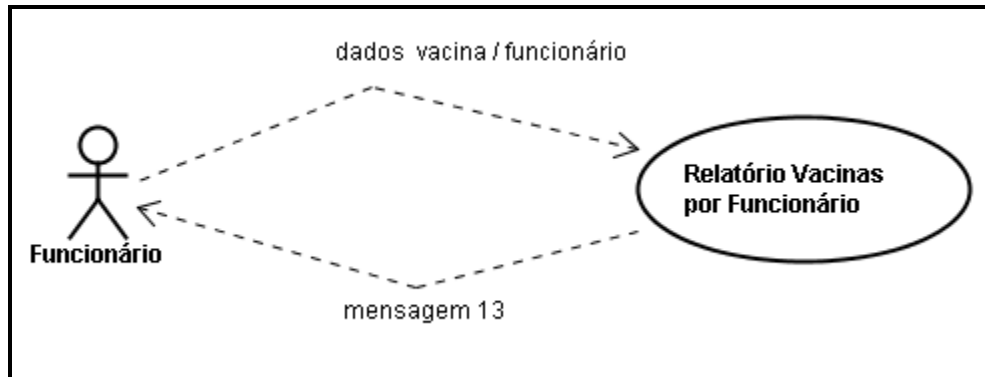


Figura 13 – Use Case *relatório de vacinas por funcionário*

Curso Normal

- 1 – consultar nome do funcionário
- 2 – escolher data
- 3 – verificar quantidade de vacinas aplicadas pelo funcionário
- 4 – atender solicitação
- 5 – mensagem 13 = “Relatório”

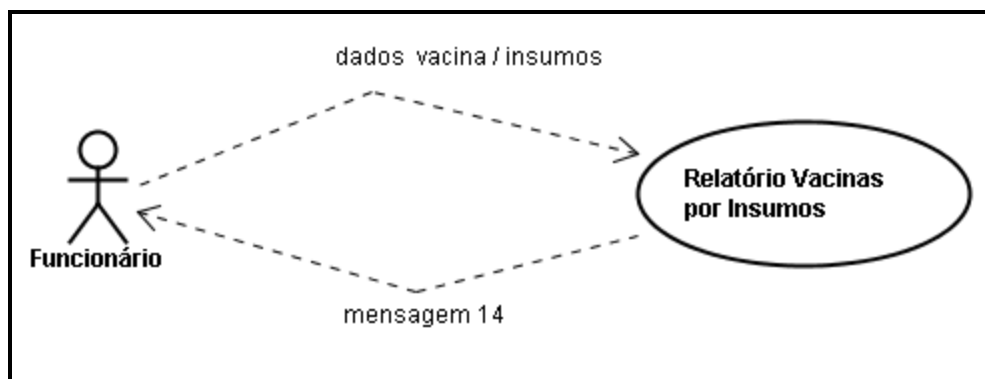


Figura 14 – Use Case *relatório de insumos por vacina*

Curso Normal

- 1 – consultar código da vacina aplicada
- 2 – escolher data
- 3 – verificar quantidade de insumos utilizados na vacina
- 4 – atender solicitação
- 5 – mensagem 14 = “Relatório”

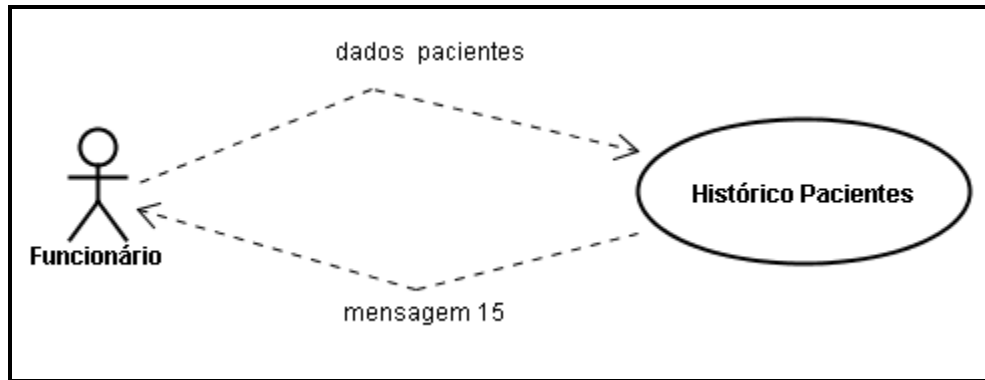


Figura 15 – Use Case *histórico de pacientes*

Curso Normal

- 1 – consultar quantidade de pacientes cadastrados
- 2 – atender solicitação
- 3 – mensagem 15 = “Relatório”.

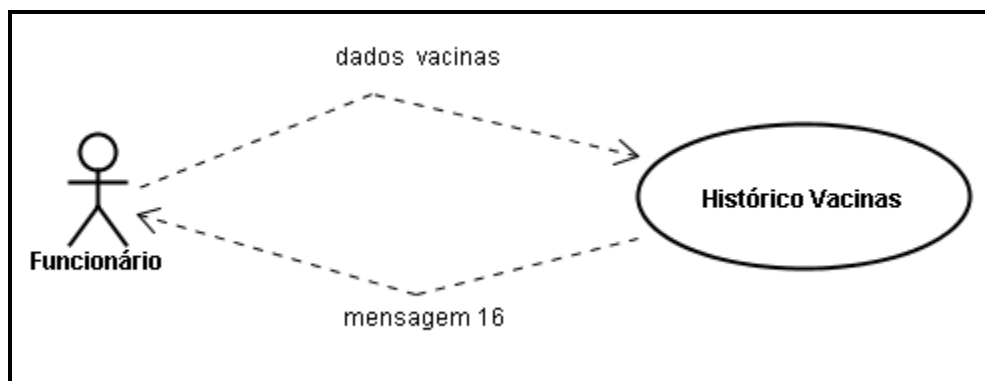


Figura 16 – Use Case *histórico de vacinas*

Curso Normal

- 1 – consultar quantidade de vacinas utilizadas
- 2 – atender solicitação
- 3 – mensagem 16 = “Relatório”

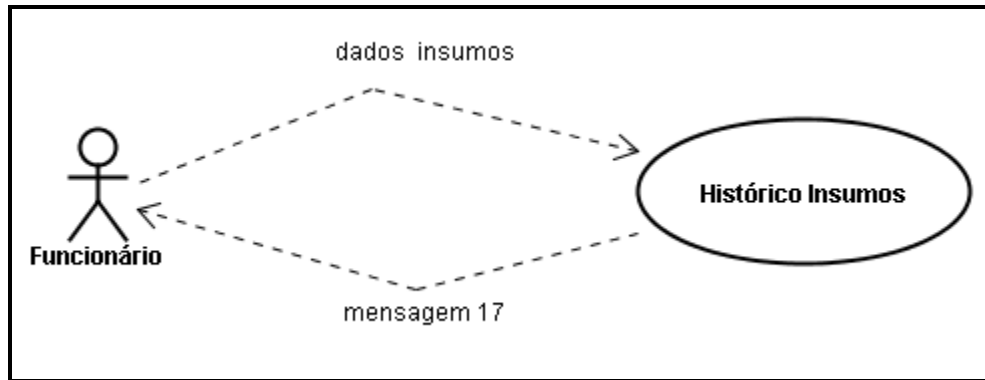


Figura 17 – Use Case *histórico de insumos*

Curso Normal

- 1 – consultar quantidade de insumos utilizados
- 2 – atender solicitação
- 3 – mensagem 17 = “Relatório”

7. DIAGRAMA DE CLASSE

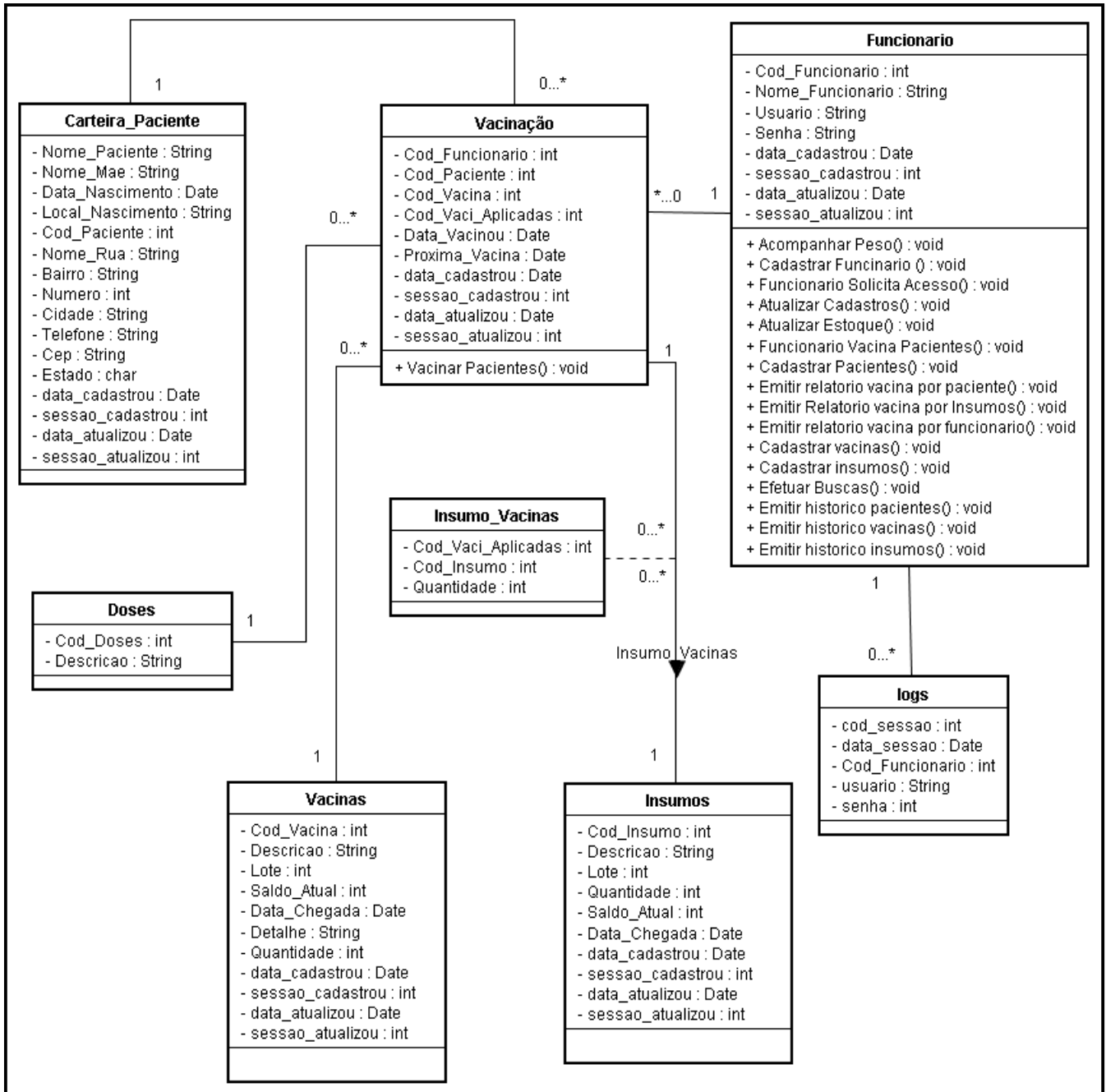


Figura 18 – Relacionamento entre os diagramas de classe

8. ESPECIFICAÇÃO DE CUSTOS

8.1 RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO

1 (um) Analista programador

1 (um) Microcomputador

1 (uma) Impressora

Sistema Operacional Windows XP Professional

IDE: Dreamweaver CS3

Linguagem de Desenvolvimento: ASP com VBSCRIPT

Aplicativos: Microsoft Word, Excel (Pacote Office)

Servidor de Aplicação: IIS

Banco e dados: SQL Server 2000

Software para preparação da análise: Jude Community 5.0

8.2 ESTIMATIVAS DE CUSTOS

Custo Analista-Programador

Custo Diário: R\$ 30,00 (Trinta Reais)

Total de Dias: 240

Custo Total: (Total de Dias * Custo Diário) = R\$ 7200,00 (sete mil e duzentos reais)

Microcomputador: R\$ 2500,00 (Dois mil e quinhentos reais)

Depreciação 2 (dois) anos: $R\$ 2500,00 / 24(\text{meses}) = R\$ 104,17$ (Cento e quatro reais e dezessete centavos) por mês

Custo de um dia: $R\$ 104,17 / 30(\text{Dias}) = R\$ 3,47$ (Três reais e quarenta e sete centavos) por dia

Custo de 240 dias: $R\$ 3,47 * 240 = R\$ 832,80$ (oitocentos e trinta e dois reais e oitenta centavos)

Impressora: R\$ 400,00 (Quinhentos reais)

Depreciação 2 (dois) anos: $R\$ 400,00 / 24(\text{meses}) = R\$ 16,66$ (Vinte reais e oitenta e três centavos) por mês

Custo de um dia: $R\$ 20,83 / 30(\text{Dias}) = R\$ 0,55$ (Setenta centavos) por dia

Custo de 240 dias: $R\$ 0,55 * 240 = R\$ 132$ (Cento e sessenta e oito reais)

Sistema Operacional Microsoft Windows XP: $R\$ 600,00$ (Seiscentos reais)

Depreciação 2 (dois) anos: $R\$ 600,00 / 24(\text{meses}) = R\$ 25,00$ (Vinte e cinco reais) por mês

Custo de um dia: $R\$ 25,00 / 30(\text{Dias}) = R\$ 0,83$ (Oitenta e três centavos) por dia

Custo de 240 dias: $R\$ 0,83 * 240 = R\$ 199,20$ (Cento e noventa e nove reais e vinte centavos)

Macromedia Dreamweaver: $R\$ 500,00$

Depreciar 2 (dois) anos: $R\$ 500,00 / 24 (\text{meses}) = R\$ 20,83$ (Vinte reais e oitenta e três centavos) por mês

Custo de 1 dia $R\$ 20,83 / 30 (\text{dias}) = R\$ 0,70$ (Setenta centavos) por dia

Custo de dias: $R\$ 0,70 * 240 = R\$ 168$ (Cento e sessenta e oito reais)

Microsoft Office (pacote): $R\$ 500,00$ (Quinhentos reais)

Depreciação 2 (dois)anos: $R\$ 500,00 / 24(\text{meses}) = R\$ 20,83$ (Vinte reais e oitenta e três centavos) por mês

Custo de um dia: $R\$ 20,83 / 30(\text{Dias}) = R\$ 0,70$ (Setenta centavos) por dia

Custo de 240 dias: $R\$ 0,70 * 240 = R\$ 168$ (Cento e sessenta e oito reais)

8.3 CRONOGRAMA DO PROJETO

Atividades	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro	Novembro
Levantamento Requisitos	■	■	■					
Pesquisa Sql Server		■	■					
Pesquisa ASP (VBScript)		■	■					
Pesquisa Dreamweaver		■	■	■				
Desenho dos Diagramas			■	■	■			
Alteração de alguns requisitos				■	■			
Desenho das Interfaces					■	■		
Implementação					■	■	■	■
Implementação / Testes							■	■

Especificação de semanas utilizadas	■	■	■	■					■	■	■	■
	1 semana								1 mês			

8.4 ORÇAMENTO DO PROJETO

Recursos	Valor
Analista-Programador	R\$ 7200,00
Microcomputador	R\$ 832,80
Impressora	R\$ 132,00
Sistema Operacional	R\$ 199,20
Microsoft Office	R\$ 168,00
Macromedia Dreamweaver CS3	R\$ 168,00
Total	R\$ 8.700,00

Tabela 03 – Tabela final dos custos

8.5 CONCLUSÃO

Após a grande jornada de trabalho sobre esta documentação, venho reconhecer que é importantíssimo tal trabalho, já que sem o mesmo não estaria compreendendo sua importância. Tudo que aprendi será de muita importância para toda a vida, pois pude ver através dos resultados obtidos que sem esforço, pesquisa, estudos e dedicação não são possíveis alcançar objetivos.

Através desse sistema para a sala de vacinas e de seus resultados poderá despertar interesse para implantar sistemas em outros departamentos na área da saúde que ainda não são informatizados, pois a área da saúde ainda está muito precária em relação à informatização.

O desenvolvimento desse sistema trouxe uma grande bagagem de conhecimento em relação a analisar e construir um sistema, já que ainda não tinha tal experiência, pois a prática nos mostra quão grande são os obstáculos e as dificuldades. Mas por outro lado nos faz crescer e aprender cada vez mais.

Em relação ao sistema de vacinas abrem-se grandes oportunidades de se ampliar o sistema com a criação de consulta on-line para os pacientes estarem sempre sabendo e acompanhando suas datas de vacinação e suas vacinas tomadas atualmente. O paciente terá um relatório de seus próprios dados sempre disponíveis a qualquer momento que necessitar.

9 .BIBLIOGRAFIA

MACHADO, Wesley Alves - **Active Server Pages, recursos, técnicas de programação e acesso a dados**; Editora Erica; São Paulo, 2000.

PRADO, Dr. Antonio Francisco **Especialização em Computação**, Desenvolvimento de Aplicações Distribuídas - Pós Graduação “Latu Sensu”, 2006.

YNEMINE, Silvana Tauhata - **Dreamweaver Ultradev 4**, Editora Visual Books, Florianópolis, 2001.

RAMALHO, José Antonio - **Iniciando em HTML**, Editora Makron Books; 1996.

JERKE, Noel – **Desenvolvendo E-Commerce com Visual Basic ASP e SQL Server**, tradução Maria da Penha, Editora Makron Books, 2001.

BATTISTI, Júlio – **SQL Server 2000 Administração e Desenvolvimento**, Editora Axcel Books, 2001.

Disponível em
<<http://www.apostilando.com/download.php?cod=2523&categoria=ASP>>-último acesso em 26 Agosto 2008.

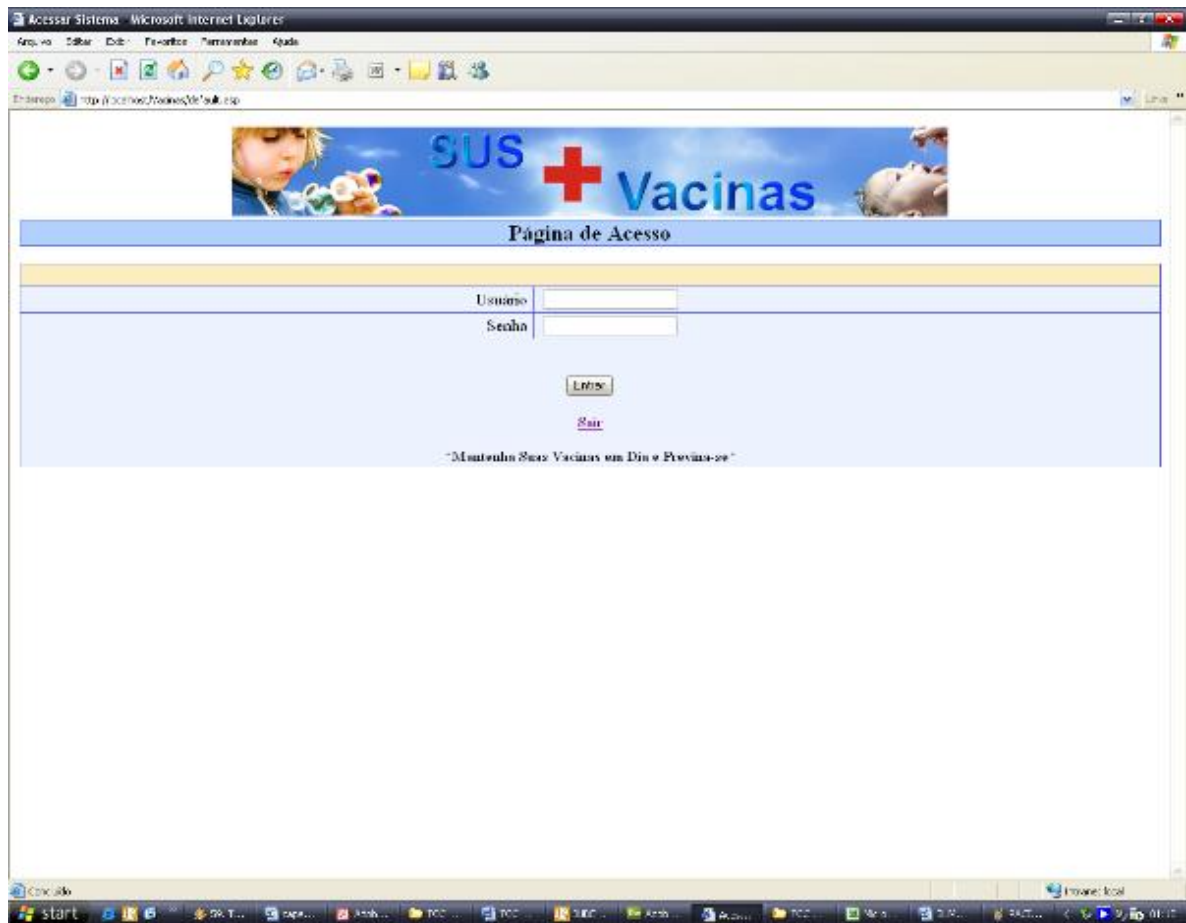
Disponível em
<<http://www.aspbrasil.com.br/conteudo/detalhesCompleta.aspx?codConteudo=3096&Secao=tutoriais>> - acesso em 26 Agosto 2008.

Disponível em <<http://www.criarweb.com/artigos/190.php>>- acesso em 28 Agosto 2008.

10. ANEXOS

10.1 INTERFACES DO SISTEMA

Página de Acesso



Página de Cadastro de Pacientes

Vacinas S.U.S. Microsoft Internet Explorer

Arquivo Editar Ferramentas Favoritos Ferramentas Ajuda

Endereço http://sistemas/usuarios/usuarios/inscricao.asp

SUS + Vacinas

Cadastrar Pacientes

Nome Paciente*	<input type="text"/>
Nome da Mãe*	<input type="text"/>
Data Nascimento*	<input type="text"/>
Local Nascimento	<input type="text"/>
Endereço	<input type="text"/>
Bairro	<input type="text"/>
Número	<input type="text"/>
Cidade*	<input type="text"/>
Telefone	<input type="text"/>
Estado	<input type="text" value="Selecione um"/>
CEP	<input type="text"/>

[Sair](#)

Mantenha suas vacinas em dia e previna-se

Computador

start

01/47