

UNIVERSIDADE ESTADUAL DE SANTA CRUZ

ELIVALDO LOZER FRACALLOSSI RIBEIRO

**IMPLEMENTAÇÃO DO MÓDULO DE URGÊNCIA E EMERGÊNCIA DO SISTEMA
GERENCIADOR DO HOSPITAL GERAL LUIZ VIANA FILHO**

ILHÉUS – BA

2013

ELIVALDO LOZER FRACALLOSSI RIBEIRO

**IMPLEMENTAÇÃO DO MÓDULO DE URGÊNCIA E EMERGÊNCIA DO SISTEMA
GERENCIADOR DO HOSPITAL GERAL LUIZ VIANA FILHO**

Monografia apresentada, para obtenção do título
de Bacharel em Ciência da Computação, à
Universidade Estadual de Santa Cruz.

Área de concentração: Engenharia de Software

Orientador: Prof. Dr. Marcelo Ossamu Honda

ILHÉUS – BA

2013

ELIVALDO LOZER FRACALLOSSI RIBEIRO

**IMPLEMENTAÇÃO DO MÓDULO DE URGÊNCIA E EMERGÊNCIA DO SISTEMA
GERENCIADOR DO HOSPITAL GERAL LUIZ VIANA FILHO**

Ilhéus – BA, 25/11/2013.

Marcelo Ossamu Honda – Dr
UESC
(Professor orientador)

Edgar Alexander – MSc
UESC
(Professor da disciplina de Estágio)

Paulo Eduardo Ambrósio – Dr
UESC
(Professor convidado)

DEDICATÓRIA

A Deus, por tudo.

A minha família, em especial meus pais e irmão, pelo carinho e apoio, tornando essa conquista real.

Ao meu sol de um metro e sessenta e três, Adriadna, pelo amor, pela existência e por lavar alguns pratos na minha vez, rs.

AGRADECIMENTOS

A Deus, eu agradeço pela vida, pela saúde e pelas pessoas existentes na minha história. Mestre, muitíssimo obrigado.

Aos meus pais, Elizabeth e Olivaldo, e meu irmão, Elivelton, agradeço pelo companheirismo, pelo afeto e por sempre sonharem junto comigo. Eu amo vocês!

A toda minha família, fisicamente próximos ou distantes, eu agradeço pela confiança, pelo carinho e pela atenção.

A minha amada, Adriadna, agradeço pelo amor e pelo incentivo. Nada disso seria possível sem você.

A todos os meus amigos, agradeço pela torcida e por toda cumplicidade.

A todos os professores, eu agradeço pelos conhecimentos compartilhados e por toda a atenção dispensada. Serei sempre grato.

A cinco professores, especialmente dedico esse momento. Aos que, ao longo da vida acadêmica, fui mais próximo. Eu agradeço, principalmente pela paciência, a Álvaro Degas, José Adolfo, Marcelo Honda, Nestor Castañeda e Sergio Alvarez.

A todos os colaboradores da UESC, que de alguma forma, foram responsáveis por essa conquista. Obrigado.

*"Quando o jogo termina, o rei e o
peão voltam sempre para a mesma
caixa."*

Provérbio italiano

RESUMO

Com potencial crescimento na área de gestão hospitalar, o uso das TICs (Tecnologias da Informação e Comunicação) se tornou um dos responsáveis pela evolução dos SIH (Sistema de Informação Hospitalar), possibilitando o compartilhamento de informações de forma eficiente. Sendo assim, o objetivo desse trabalho foi a implementação de um módulo de urgência e emergência para o sistema gerenciador do Hospital Geral Luiz Viana Filho, localizado na cidade de Ilhéus/BA, com a finalidade de reduzir a utilização de papel, tornando o processo mais rápido, fácil e seguro. Para alcançar este objetivo, criamos um sistema baseado no modelo Web com uso exclusivo de ferramentas livres que, dentre outros motivos, tornam a utilização do sistema financeiramente viável em estabelecimentos hospitalares com pouco, ou nenhum, recurso financeiro destinado as TICs. O módulo foi projetado e desenvolvido, atendendo funcionalidades e características que visaram cumprir requisitos específicos de um sistema para gerenciamento do setor de urgência e emergência.

Palavras-chave: Tecnologias da Informação e Comunicação, Sistema de Informação Hospitalar, Prontuários.

ABSTRACT

Potential growth in the area of hospital management, the use of ICTs (Information and Communication Technologies) became one of the responsible for the evolution of HIS (Hospital Information System), enabling the information share efficiently. Therefore, the goal of this work was the implementation of a module of urgency and emergency for the management system of the General Hospital Luiz Viana Filho, located in the Ilhéus/BA city, in order to reduce the use of paper, making the process faster, easier and safer. To accomplish this goal, we created a Web-based system with exclusive use of free tools that, among other reasons, make the use of the system financially viable in hospitals with little or no financial resources for ICTs. The module was designed and implemented, given functionalities and features that aimed to fulfill specific requirements of a management system for urgent and emergency sector.

Keywords: Information Technologies and Communications, Hospital Information System, Medical Records.

LISTA DE SIGLAS

CFM	Conselho Federal de Medicina
CSS	<i>Cascading Style Sheets</i>
FPDF	<i>Free PDF</i>
HTML	<i>HyperText Markup Language</i>
MD5	<i>Message-Digest algorithm 5</i>
MVC	<i>Model-Controller-View</i>
NGS	Níveis de Garantias de Segurança
NSA	<i>National Security Agency</i>
OMS	Organização Mundial de Saúde
PDF	<i>Portable Document Format</i>
PEP	Prontuário Eletrônico do Paciente
PHP	<i>PHP: Hypertext Preprocessor</i>
SBIS	Sociedade Brasileira da Informática da Saúde
SGBD	Sistema Gerenciador de Banco de Dados
SGML	<i>Standard Generalized Markup Language</i>
SHA	<i>Secure Hash Algorithm</i>
SIH	Sistema de Informação Hospitalar
SIMPAS	Sistema Integrado de Material, Patrimônio e Serviços da Bahia
SO	Sistema Operacional
SUS	Sistema Único de Saúde
SVN	<i>Subversion</i>
TI	Tecnologia da Informação
TICs	Tecnologias da Informação e Comunicação
W3C	<i>World Wide Web Consortium</i>
XAMPP	<i>X, Apache, MySQL, PHP, Perl</i>
XML	<i>eXtensible Markup Language</i>

LISTA DE FIGURAS

Figura 1 – Tela principal do primeiro atendimento.	30
Figura 2 – Tela para preenchimento dos valores da anamnese do primeiro atendimento.....	31
Figura 3 – Tela principal do segundo atendimento.....	31
Figura 4 – Tela para preenchimento dos valores da anamnese do segundo atendimento.....	32
Figura 5 – Tela principal do terceiro atendimento (para a sala 2).	32
Figura 6 – Tela para preenchimento dos valores da anamnese do terceiro atendimento.....	33
Figura 7 – Modelo, em branco, da ficha de classificação.	34
Figura 8 – Tela para cadastro de pacientes.	35
Figura 9 – Tela para cadastro de usuários.	35
Figura 10 – Tela para cadastro de medicamentos.	36
Figura 11 – Exemplo de máscara com JQuery no campo de cadastro do CPF.	38
Figura 12 – Exemplo de validação com HTML5: campos obrigatórios.....	39
Figura 13 – Exemplo da busca automática.	41
Figura 14 – Exemplo do uso da busca automática para edição de um paciente cadastrado.....	41

LISTA DE TABELAS

Tabela 1 – Cotação dos preços de alguns requisitos para implantação do e-SUS ...	21
Tabela 2 – Algumas características para obtenção da certificação NGS1	40

SUMÁRIO

RESUMO	VII
ABSTRACT	VIII
LISTA DE SIGLAS	IX
LISTA DE FIGURAS	X
LISTA DE TABELAS	XI
1. INTRODUÇÃO	14
1.1. Contextualização	14
1.1.1. TICs	14
1.1.2. SIH	14
1.1.3. Prontuários	16
1.1.4. Certificação SBIS/CFM	17
1.2. Situação atual	18
1.3. Objetivos	19
1.3.1. Objetivo Geral	19
1.3.2. Objetivos específicos	19
1.4. Organização do trabalho	19
2 MATERIAL E MÉTODOS	21
2.1. HTML	22
2.2. CSS	23
2.3. PHP	23
2.4. JavaScript	24
2.5. XAMPP	24
2.6. PostgreSQL	25
2.7. Orientação a Objetos	25
2.8. MVC	26

2.9.	SVN	26
2.10.	Documentação do código	27
2.11.	Scripts externos.....	27
2.11.1.	Active Record	27
2.11.2.	FPDF	28
2.11.3.	FancyBox	28
2.11.4.	Formee.....	29
2.11.5.	Smart Paginator	29
3	RESULTADOS E CARACTERÍSTICAS DO SISTEMA	30
3.1.	Atendimentos.....	30
3.2.	Ficha de classificação.....	33
3.3.	Cadastros	34
3.4.	Separação MVC.....	36
3.5.	Segurança dos dados	37
3.6.	Ajustável em diferentes resoluções.....	37
3.7.	Validação dos formulários.....	38
3.8.	Certificação	39
3.9.	Busca automática	40
3.10.	Ferramentas livres.....	42
4	CONCLUSÃO.....	43
4.1.	Considerações finais.....	43
4.2.	Trabalhos futuros	43
	REFERÊNCIAS	45

1. INTRODUÇÃO

1.1. Contextualização

1.1.1. TICs

As Tecnologias da Informação e Comunicação (TICs) podem ser definidas como um conjunto de recursos tecnológicos integrados, que por meio de hardwares, softwares e telecomunicações, proporcionam a automatização dos processos informacionais e comunicação das pessoas. Para Rosini e Palmisano (2003), as TICs abrangem uma variedade de soluções computacionais que se aplicam na capacidade de coletar, processar, armazenar e acessar informações.

A popularização da internet foi a principal responsável pelo crescimento e potencialização das TICs em diversas áreas, e, segundo Pinto (2006), a Organização Mundial da Saúde (OMS) entende que as TICs devem fazer parte de todas as estratégias voltadas para a área. Para tanto, podem ser utilizadas para melhorar as condições de saúde da população, visto que contribuem para troca de informações e cooperam na tomada de decisões. Neste sentido, podem ser utilizadas para a informatização na área médica, que apresenta um grande potencial de crescimento (BEZERRA, 2009).

1.1.2. SIH

A utilização da informática na gestão hospitalar evoluiu nos últimos anos. Antes, os computadores eram utilizados apenas na realização de tarefas rotineiras e isoladas, atualmente procura-se unir, através de um sistema único, os diversos setores, como administração, contabilidade, almoxarifado, dentre outros. (JOHANSTON, 1993).

Rocha, Costa e Otero (2001) asseguram que a utilização de softwares para a gestão hospitalar requer cuidados adicionais e adaptações, em razão da

especificidade da prestação de serviços e da importância dos dados. Além disso, deve contribuir para a melhoria contínua do atendimento e da gerência das informações (MARIN, 2010).

Conforme Perez (2006), em um ambiente hospitalar, é necessário um sistema informatizado que possibilite a segurança, disponibilidade e acesso aos dados dos pacientes e que possa ter informações compartilhadas entre todos os setores do hospital. Este modelo de sistema descrito é a definição do que hoje se conhece como Sistemas de Informação Hospitalar (SIH).

O desenvolvimento do SIH começou nos anos 60, e, devido aos altos custos investidos, somente grandes instituições dos EUA e Europa tinham recursos suficientes para a obtenção desse sistema (CAVALCANTE; SILVA; FERREIRA, 2011). Johanston (1993) afirma ainda que, a solução adotada na época, ainda utilizada em vários países, inclusive no Brasil, foi a de um sistema totalmente centralizado, onde bases de dados eram disponibilizadas a partir de uma estação central, que por sua vez, estava conectada aos computadores dos diversos setores. Posteriormente, graças aos avanços tecnológicos, os preços dos hardwares e software baixaram consideravelmente, fazendo assim, com que alguns SIH genéricos e especializados fossem disponibilizados a menores custos, tornando-os acessíveis às empresas com menor poder aquisitivo (CAVALCANTE; SILVA; FERREIRA, 2011).

Ainda de acordo com Johanston (1993), um SIH atualmente integra diversos módulos, sendo cada um responsável por um setor específico do hospital. Por exemplo, podemos ter um sistema que contenha módulos que auxiliem no gerenciamento de farmácia, almoxarifado, enfermarias, ambulatórios, urgência e emergência, entre outros.

A OMS descreve que o SIH é um instrumento composto pelas etapas necessárias em um atendimento hospitalar, sendo: (i) a coleta de dados; (ii) o processamento; (iii) a análise; e (iv) a transmissão dos elementos necessários. Provocando, dessa maneira, a organização das informações (CAVALCANTE; SILVA; FERREIRA, 2011).

É importante destacar que, a interação do SIH com o usuário deve acontecer de tal forma que todas as modificações feitas devem estar instantaneamente disponíveis para consultas feitas por outros departamentos hospitalares, assim que as mesmas forem inseridas nos prontuários.

1.1.3. Prontuários

Antigamente, a identificação de cada paciente era feita através de uma etiqueta fixada ao punho, contendo apenas as informações básicas. Com o passar do tempo, ficou evidente a importância desse artefato e o quanto ele era mal empregado. Este fato resultou na adaptação da identificação de punho, para uma ficha mais elaborada, surgindo assim, o prontuário do paciente (PATRÍCIO et al., 2011; BRAGA et al, 2013).

O CFM (Conselho Federal de Medicina) define prontuário como um documento que contém um conjunto de informações do paciente, relativas à sua saúde e a assistência prestada. Sendo de caráter legal e sigiloso, bastante útil na continuidade do processo de atendimento (CFM, 2002).

Os prontuários em papel apresentam algumas limitações, como a ineficiência na organização e armazenamento, o difícil compartilhamento, problemas na pesquisa coletiva, entre outros (PINTO, 2006). Assim, surgiu a necessidade de um prontuário em formato eletrônico, para padronizar o trabalho feito por diversos profissionais, em locais e tempos diferentes, o denominado Prontuário Eletrônico do Paciente (PEP) (MARIN, 2010).

Ainda de acordo com o Marin (2010), o PEP é um conjunto de informações que deverão ser armazenadas em um sistema projetado com a finalidade específica de armazenar e disponibilizar esses dados de forma rápida e completa. Essas informações podem ser relacionadas em diferentes aspectos, como por exemplo, na identificação, dados sobre sua saúde (receitas, diagnósticos, radiografias, entre outros), aspectos socioeconômico e assim por diante (PINTO, 2006).

Uma vez coletado dados de um paciente, eles devem ser inseridos em um formato que possa ser facilmente recuperado ou distribuído. A problemática dessa

situação ocorre, dentre outros aspectos, a necessidade de sistemas e profissionais integrados (MARIN, 2010).

O uso do PEP traz algumas vantagens, dentre elas, a facilidade do cumprimento da exigência, estipulada pelo CFM, do tempo mínimo que um prontuário deve ser preservado desde a última alteração, que são 20 anos (CFM, 2002), bem mais fácil de ser executado se utilizando o PEP. Segundo Bezerra (2009), as organizações hospitalares do Brasil devem obedecer às resoluções do CFM e da SBIS (Sociedade Brasileira da Informática da Saúde).

É válido destacar que, obter todas as informações contidas em um único documento, seja em formato digital ou em papel, não é uma tarefa trivial, pois um paciente pode ter atendimentos em diferentes organizações, em intervalos de tempo impreciso e cada instituição de atendimento armazenar suas informações de maneiras distintas, tornando assim, a padronização um desafio tecnológico a ser considerado (NARDON; FURUIE; TACHINARDI, 2000).

1.1.4. Certificação SBIS/CFM

Com intuito de incentivar que sistemas na área da saúde implementem características que a SBIS e o CFM pontuam importantes em produtos desse gênero, foi elaborado um processo denominado de certificação de software para a saúde, podendo ser aplicado, desde que sigam as recomendações, a qualquer sistema que armazene, capture, apresente ou imprima informações relacionadas à saúde. Dentre os objetivos da certificação, podemos destacar a legalidade da utilização dos sistemas hospitalares na substituição dos prontuários de papel, caso os requisitos obrigatórios sejam atendidos (VARGAS et al., 2010; SBIS, 2009).

O processo de certificação SBIS/CFM é dividido em alguns requisitos, podendo ser sobre: (i) segurança; (ii) conteúdo; (iii) estrutura e (iv) funcionalidades. Para cada item, é exigido que o *software* implemente algumas características, e, em cada uma, é indicada se é obrigatória ou não. Além disso, em alguns casos, é descrito como uma atividade deve ser realizada, por exemplo, indicando que

imagens coloridas não devem ser visualizadas em monitores monocromáticos (VARGAS et al., 2009; SBIS, 2009).

As regras elencadas visam padronizar os sistemas e, de forma clara, destacar as regras para que as informações sejam transmitidas com segurança e eficiência (VARGAS et al, 2009).

1.2. Situação atual

Localizado na cidade de Ilhéus-BA, o Hospital Geral Luiz Viana Filho, utiliza planilhas eletrônicas para controle de cadastros e um montante de papel para controle dos dados das consultas, que segue manualmente o fluxo dos atendimentos. Depois de preenchidos, os funcionários encaminham os formulários para o próximo responsável.

No dia 20 de Agosto de 2013, a Câmara Municipal de Ilhéus aprovou em sessão o projeto de Lei número 118/2013, enviado pelo vereador Fábio Magal, que entre outras disposições: (i) o PEP ficaria instituído na rede pública municipal de saúde; (ii) que o mesmo seria identificado pelo número do cadastro no Sistema Único de Saúde (SUS) do paciente; (iii) que estava autorizado o registro, a comunicação e a transmissão do prontuário no formato digital; (iv) e que o sistema para gerenciamento do PEP fosse, preferencialmente, gerado a partir de softwares de código fonte aberto (ILHÉUS, 2013).

Atualmente, o Governo do Estado da Bahia dispõe de um software com a mesma finalidade do sistema que esse módulo tem o objetivo integrar, o e-SUS. No entanto alguns requisitos de infraestruturas são necessários, dentre eles um servidor *Oracle* e um Sistema Operacional (SO) *Windows Server*, o que se torna um agravante, devido os custos necessários para obtenção e manutenção desses equipamentos.

1.3. Objetivos

1.3.1. Objetivo Geral

Implementar um módulo de um sistema de gerenciamento do processo de urgência e emergência do Hospital Geral Luiz Viana Filho, com intuito de reduzir a utilização de papel, tornando o processo mais rápido, fácil e seguro.

1.3.2. Objetivos específicos

- Disponibilizar dados de forma remota e simultânea.
- Permitir a integração com outros módulos no mesmo sistema.
- Agilizar o processo de atendimento de urgência e emergência.
- Garantir o processamento contínuo dos dados.
- Evitar a replicação e perda de dados em prontuários.
- Auxiliar a tomada de decisão sobre o tipo de tratamento ao qual o paciente deverá ser submetido.

1.4. Organização do trabalho

No primeiro capítulo deste relatório, com o intuito de demonstrar a relevância do tema, temos uma contextualização acerca do objeto de estudo. Após isso, existe uma breve descrição sobre a situação em que o hospital piloto se encontra atualmente e a apresentação dos objetivos almejados.

O segundo capítulo apresenta a fundamentação teórica do trabalho, isto é, exhibe e conceitua toda a metodologia que foi utilizada no desenvolvimento desse sistema.

No terceiro capítulo são exibidos, de forma detalhada, os resultados e as funcionalidades do sistema. Em alguns casos, algumas telas do sistema estão disponíveis de modo a tornar o entendimento mais rápido.

Por fim, no quarto e último capítulo são apresentadas as conclusões do projeto, além de uma relação de trabalhos futuros que poderão ser desenvolvidos como uma maneira de aperfeiçoamento do módulo.

2 MATERIAL E MÉTODOS

Neste trabalho, propomos a criação de um módulo de urgência e emergência, que posteriormente será integrado a um sistema gerenciador de hospitais. Neste módulo princípios do PEP serão aplicados. A implantação será feita no Hospital Geral Luiz Viana Filho, localizado na cidade de Ilhéus-BA.

Como citado, é de propriedade do Governo Estadual da Bahia um sistema denominado e-SUS, que tem a mesma finalidade do sistema que este módulo será integrado. Porém, um dos requisitos necessários para sua utilização é que o hospital beneficiado tenha um servidor com Sistema Gerenciador de Banco de Dados (SGBD) proprietário da *Oracle*, o que, muitas vezes, inviabiliza sua utilização devido ao alto custo da ferramenta. Em contrapartida, propusemos a utilização de ferramentas livres, como por exemplo, HTML, CSS, PHP, *JavaScript*, etc., tornando assim, o sistema baseado no modelo *Web*. Alguns custos necessários para a implantação do e-SUS podem ser vistos na Tabela 1.

Tabela 1 – Cotação dos preços de alguns requisitos para implantação do e-SUS

Item	Descrição do produto	Preço
01	Windows Server 2012 R2 Standard com IIS	US\$ 882,00 ¹
02	Oracle Database 12c Enterprise Edition	US\$ 27526,25 ²
03	Microsoft SQL Server 2012 Enterprise	US\$ 6874,00 ³
TOTAL		US\$ 35282,25

Fontes: ¹ <http://www.microsoft.com/en-us/server-cloud/products/windows-server-2012-r2/default.aspx#fbid=h1PgADbhrxa>

² <https://shop.oracle.com/pls/ostore/f?p=700:6:0::NO::>

³ <http://www.microsoft.com/sqlserver/pt/br/get-sql-server/how-to-buy.aspx>

Os detalhes dos valores pagos variam de acordo com o produto. Para o item 01, a licença descrita (US\$ 882,00) é unitária por processador, acrescida de um valor, não considerado, que é adicionado de acordo com a quantidade de usuários, além disso, suporta *upgrade* gratuito de versões por dois anos. No item 02, o custo apresentado (US\$ 27526,25) é calculado com base no total de usuários, para isso, computa-se US\$ 950,00 por usuário (com venda mínima para 25), mais o valor do suporte por 01 ano, sendo a licença de prazo perpétuo. Por fim, no produto 03 o valor apresentado (US\$ 6874,00) é unitário por núcleo.

A escolha pelo padrão *Web* justifica-se por: (i) alta curva de aprendizagem, ou seja, a aprendizagem em um curto intervalo de tempo; (ii) portabilidade, visto que seu uso independe do SO, do *hardware*, necessitando apenas de um navegador; (iii) manutenção e suporte, pois, como o sistema se encontrará em um servidor, uma mudança feita remotamente, estará disponível para todos os usuários.

2.1. HTML

Como descrito por Souza e Alvarenga (2004), HTML (*HyperText Markup Language*, em português, linguagem de marcação em hipertexto) é uma linguagem interpretada derivada do padrão SGML (*Standard Generalized Markup Language*), que, por sua vez, é uma metalinguagem, isto é, uma linguagem utilizada para descrever outras linguagens. Foi inicialmente concebida como uma solução para a publicação de documentos científicos em meios eletrônicos, no entanto, foi bastante popularizada com o crescimento da internet sendo hoje o padrão dos documentos da *Web* (ALMEIDA, 2002).

Desenvolvida em 1989 por Tim Berners-Lee, HTML é utilizado atualmente para a construção da maioria das páginas *Web* e baseia-se em um conjunto de *tags*, que são marcações que descrevem os dados e comandos para manipulação de um documento, fazendo, assim, que os navegadores renderizem, estruturem e mostrem as páginas (SOUZA; ALVARENGA, 2004).

Mantido pelo *World Wide Web Consortium* (W3C) desde 1996, o HTML encontra-se atualmente na versão 5 (HTML5). Apesar da maioria das páginas encontrarem-se disponíveis na versão 4.01, esse módulo possui algumas funcionalidades da versão atual, visando uma maior interação com o usuário, o que é amplamente visto com uma das características dessa versão.

2.2. CSS

Criado após estudo de Hakon Wium Lie e Bert Bos, *Cascading Style Sheets* (Folhas de Estilo em Cascata), acrônimo para CSS, referem-se ao conjunto de declarações que especificam a apresentação do documento. Trata-se de uma linguagem de estilo utilizada para definir a apresentação de páginas escritas em uma linguagem de marcação, como HTML ou *eXtensible Markup Language* (XML). Considerada bastante flexível, permite que páginas sejam estilizadas por arquivos externos, contribuindo assim, para a legibilidade dos códigos, uma vez que separa o estilo do conteúdo (DANTAS et al., 2007; SONZA; SANTAROSA; CONFORTO, 2008; TESTA, 2010).

A versão mais utilizada é a 2.1, no entanto, a atual, versão 3, compreende alguns novos recursos e já pode ser utilizada, apesar de não estar totalmente desenvolvida. Como consequência, alguns navegadores não suportam todos os recursos existentes na versão atual. Nesse projeto utilizamos alguns recursos do CSS3, porém, a sua não visualização (se renderizados por navegadores que ainda não o suportam), não acarretará em perda de conteúdo, por se tratar apenas de detalhes de *layout*.

2.3. PHP

Segundo Niederauer (2004), PHP (acrônimo recursivo para PHP: *Hypertext Preprocessor*), é uma das linguagens mais utilizadas na internet e surgiu em 1995, graças a Rasmus Lerdorf. Uma das maiores vantagens dessa linguagem é a capacidade de tornar as páginas mais interativas, seja por meio de banco de dados ou através de outros sites.

Dentre as características que fazem com que PHP seja uma das linguagens mais utilizadas, podemos destacar: (i) o fato de ser gratuita e de código aberto; (ii) seu código é embutido no HTML; (iii) é executado no lado do servidor; (iv) portabilidade, sendo utilizado na maioria dos SO; (v) existência de funções que

permitem a comunicação entre vários SGBDs e (vi) suporte a programação orientada a objetos (ACHOUR, 2013).

No projeto, utilizamos recursos disponíveis na linguagem para programação orientada a objetos e a versão utilizada foi a 5.4.4.

2.4. JavaScript

Criada em 1995, pelo desenvolvedor Brendan Eich, *JavaScript* é uma linguagem de programação com recursos de orientação a objetos que pode ser interpretado pela maioria dos navegadores, e que, em conjunto com elementos do HTML, pode tornar aplicações mais flexíveis e simples de serem utilizadas (STOTHARD, 2000).

De acordo com Carvalho e Almeida (2010), *JavaScript* é composta por bibliotecas, que são utilizadas a depender do propósito. Dentre elas, existem, por exemplo, *DOJO*, *MooTools*, *ExtJS*, *SmartClient*, além da utilizada nesse projeto, a *jQuery*.

Ainda de acordo com o autor, *jQuery* é uma biblioteca leve e compatível com vários navegadores que foi criada por John Resig e tornada disponível em 2006, sendo gratuito e código aberto.

2.5. XAMPP

Para que páginas PHP sejam interpretadas pelo navegador é necessário que estejam em um servidor que ofereça tal serviço, podendo ser online ou local. Servidores online são aqueles localizados em máquinas externas. Em contrapartida, existem os servidores locais, que funcionam na própria estação de trabalho e sem a necessidade da conexão com a internet.

Dentre as opções, utilizamos o XAMPP (abreviatura para: X - indicando o uso para qualquer SO -, *Apache*, *MySQL*, *PHP*, *Perl*) na versão 3.0.12, um servidor local,

livre, desenvolvido pelo *Apache Friends*, que se caracteriza pela facilidade de configuração e de uso, com suporte a diversos serviços e tecnologias, como, PHP, *phpPgAdmin*, *MySQL*, *Perl*, *FileZilla FTP Server*, entre outros (VOGELGESANG, 2013).

2.6. PostgreSQL

Segundo Niederauer (2004), PHP pode ser integrado com vários SGBDs, como por exemplo, *MySQL*, *PostgreSQL*, *Sybase*, *Oracle*, *Microsoft SQL Server*, entre outros. Para isso, em cada um estão presentes características e funcionalidades que são utilizadas pelo PHP, com intuito de resolver algum problema específico.

A utilização do *PostgreSQL* nesse projeto, que teve sua implementação iniciada em 1986, é justificada por se tratar de um SGBD gratuito, de código fonte aberto, que pode ser executado tanto por UNIX quanto por *Windows* (PAVAN et al., 2000) e que é amplamente utilizado em todo mundo, uma vez que o mesmo pode ser modificado e distribuído por qualquer pessoa, para qualquer finalidade, sem custo algum (BORGES; CONY, 2005).

2.7. Orientação a Objetos

A programação orientada a objetos é um paradigma de desenvolvimento de sistema. Em contrapartida com a programação estruturada, que por sua vez, é composta por um conjunto de procedimentos e variáveis, que nem sempre são agrupadas de acordo com o contexto, a programação orientada a objetos visa uma maior proximidade com mundo real, tratando, por sua vez, objetos e estruturas já conhecidas pelo cotidiano, e que, conseqüentemente, temos uma maior compreensão, fugindo da abstração criada pela programação estruturada e tornando o código reutilizável, evitando a repetição de procedimentos (DALL'OGGIO, 2009).

Para esse estudo, utilizamos o suporte fornecido pelo PHP para a programação orientada a objetos, com o intuito de tornar o código mais claro, eficiente, fácil de ser corrigido, documentado e testado.

2.8. MVC

Dall'Oglio (2009) afirma que *design pattern* descreve um problema que ocorre com frequência, e, conclui que a solução para esse problema deve ocorrer de tal forma que a solução possa ser utilizada várias vezes, sem que haja repetição.

Ainda de acordo com o autor, no contexto da orientação a objetos, *design pattern* descreve o modo de comunicação e relacionamento entre objetos e classes, de modo a aperfeiçoar o processo de criação e manutenção de projetos.

Dentre as possibilidades, *Model View Controller* (MVC) é o *design pattern* mais conhecido de todos. Um projeto baseado em MVC é basicamente dividido em três camadas: (i) *Model* (M) é o objeto que representa as informações do negócio do projeto, geralmente representado por um *Active Record*; (ii) *View* (V) é a definição da interface e (iii) *Controller* (C) é a manipulação dos dados e a execução da tarefa correspondente, isto é, a definição da maneira como a interface (V) deve agir a partir das informações fornecidas pelo usuário (DALL'OGGIO, 2009).

2.9. SVN

Subversion (SVN) é um mecanismo para controle de versão que gerencia arquivos e diretórios, de modo a controlar as modificações feitas ao longo do tempo. Trata-se de um sistema de código aberto, que pode ser utilizado em rede por diferentes pessoas, sendo comumente chamado pela literatura de “máquina do tempo” (COLLINS-SUSSMAN; FITZPATRICK; PILATO, 2007).

2.10. Documentação do código

Esse projeto é um módulo de um sistema maior, que futuramente será acrescida de várias outras funcionalidades a fim de obter um produto capaz de gerir por completo um hospital. O sucesso dessa continuidade está inteiramente conectado a um dos princípios básicos da programação: a documentação. Esta é uma maneira dos desenvolvedores irem descrevendo detalhadamente tudo já desenvolvido, com o intuito de que essa documentação sirva como referência para os próximos desenvolvedores deste projeto.

A documentação deste módulo foi gerada a partir de uma ferramenta gratuita e bastante conhecida na literatura, o *PHPDocumentor* (ou simplesmente, *PHPDoc*). Com uma interface *Web* intuitiva, o *PHPDoc* reconhece, no código PHP, *tags* pré-definidas que são utilizadas pelo programador a depender do trecho de código documentado, gerando, por fim, um documento detalhando o que foi desenvolvido.

2.11. Scripts externos

Além das ferramentas e metodologias acima citadas, com o intuito da reutilização de código, a depender do propósito, comumente se fazia necessário à utilização de algum script externo desenvolvido por terceiros. A seguir, listaremos os utilizados, bem como uma breve descrição sobre a funcionalidade e suas respectivas localizações.

2.11.1. Active Record

Active Record, disponível em <http://www.phpactiverecord.org>, é descrito por Dall'Oglio (2009) como um padrão de projetos que, por meio de um mapeamento objeto-relacional, implementa um mecanismo de acesso ao banco de dados. Através desse recurso, é possível acessar os dados de maneira orientada a objeto, além

disso, de modo intuitivo, fazer uso de operações comumente utilizadas, como por exemplo, *create*, *read*, *insert*, entre outros.

Semelhante ao *Row Data Gateway*, *Active Record* é um *pattern* que provê uma estrutura para a classe que representará o objeto do modelo conceitual do banco de dados. A diferença entre ambos se dá pelo fato de que no *pattern* utilizado nesse projeto, o *Active Record*, existem métodos que implementam características específicas do modelo de negócio (DALL’OGLIO, 2009).

2.11.2. FPDF

Não apenas neste sistema, mas é comumente desejável obter relatórios em formato PDF. Para tanto, existem diversas bibliotecas com funcionalidades semelhantes que geram, a partir de um script PHP, um documento PDF, por exemplo, mPDF, FPDF, TCPDF, R&OS PDF, entre outros. Podendo ser obtida de forma gratuita em <http://www.fpdf.org>, utilizamos nesse projeto o FPDF, que se caracteriza pela fácil utilização e por dispensar qualquer configuração externa, bastando, extrair os arquivos da classe no diretório do projeto, e de forma totalmente orientada a objetos, utilizar métodos já desenvolvidos (MANCINI; ARA-SOUZA; LOUZADA NETO, 2010).

2.11.3. FancyBox

Recurso que oferece, de modo agradável, um modal que flutua por cima da página web, *FancyBox* é uma ferramenta elegante e gratuita (<http://fancyapps.com/fancybox/>) que, em contraposição aos *popups* já conhecidos, oferece aos desenvolvedores uma maneira de tornar a página bem mais atraente. Altamente personalizável, o *FancyBox* permite modais para vários tipos de multimídia, como vídeos e imagens, além de textos e formulários, como utilizados para a construção desse módulo.

2.11.4. Formee

Desenvolvido por brasileiros, *formee* (<http://formee.org/>) é um *framework* gratuito que foi desenvolvido com o propósito de criação e estruturação de formulários, de modo a tornar o desenvolvimento mais rápido e o resultado mais elegante. Para isso, utiliza *grids* responsivos que ajudam a formatar a largura dos campos, tornando o mesmo variável, a depender da configuração utilizada pelo usuário.

O pacote fornece ainda, formatação elegante para os botões e, além disso, estilo e estrutura para construção de mensagens de sucesso, informação, alerta e erro.

2.11.5. Smart Paginator

Smart Paginator (<http://www.egrappler.com/jquery-pagination-plugin-smart-paginator/>) é uma classe que implementa uma ferramenta bastante útil, porém, complexa de ser desenvolvida. Trata-se de paginação, recurso que mostra parte dos dados na tela, respeitando definições impostas pelo desenvolvedor, com vantagem da várias possibilidades de customização.

3 RESULTADOS E CARACTERÍSTICAS DO SISTEMA

A seguir, serão listadas algumas características do módulo desenvolvido com o objetivo de descrever as funcionalidades, e em alguns casos, quando necessário, mostrando algumas imagens ilustrando essas características.

3.1. Atendimentos

O fluxo básico do atendimento no hospital se baseia em três formulários.

No primeiro atendimento, o responsável seleciona um paciente previamente cadastrado e solicitante do atendimento médico, como na Figura 1, e preenche alguns dados referentes à sua anamnese básica (peso, altura, temperatura, entre outros) como pode ser visto na Figura 2. Quando estes dados são salvos, o paciente é transferido para a fila do segundo atendimento.

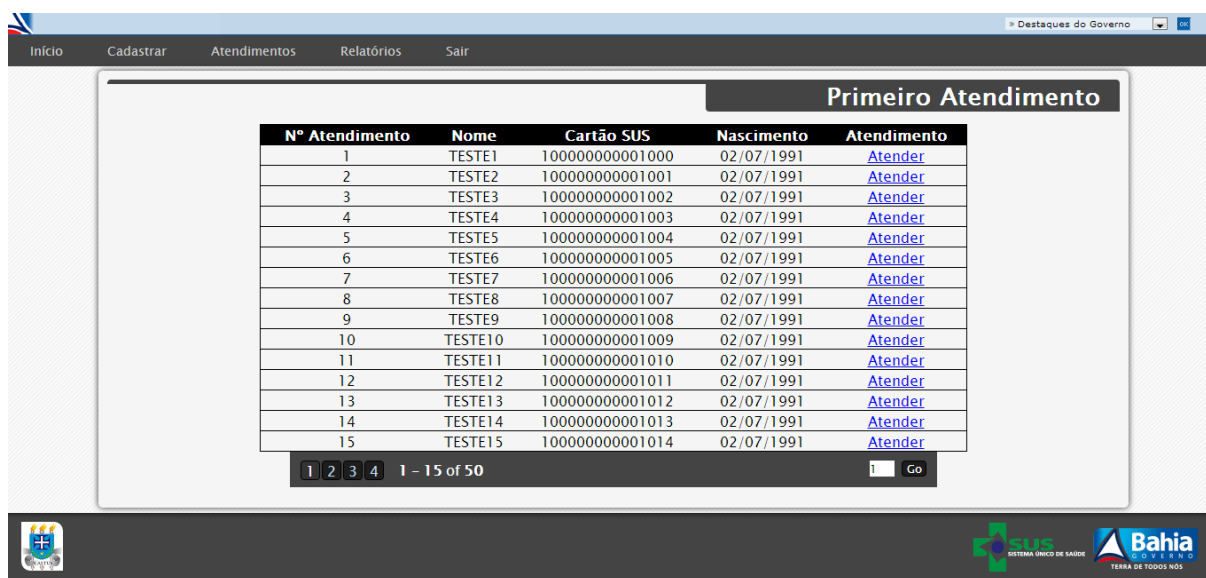


Figura 1 – Tela principal do primeiro atendimento.

Realização do atendimento

Dados do paciente

Nome: TESTE7

Data de Nascimento: 02/07/1991

Cartão SUS: 100000000001006

Anamnese

Altura (m): 1.7 Peso (kg): 70 Pressão (mmHg): Temperatura (°C): 35

Pulsção (bpm): 60 HGT (mg/dl): 35 Sat. O2 (%): 50

SALVAR **LIMPAR**

Figura 2 – Tela para preenchimento dos valores da anamnese do primeiro atendimento.

Nesse segundo momento, geralmente feito por outro profissional, após a escolha do paciente na lista, como na Figura 3, é atribuída uma descrição sobre o estado do paciente, seguido de uma classificação que se baseia em um padrão nacional referente ao risco clínico, podendo ser vermelho, amarelo, verde e azul (variando do mais grave, ao menos grave, na ordem), e por fim é imposta ao paciente uma sala, como na Figura 4. Ao salvar esse formulário, o mesmo é enviado para o terceiro atendimento, realizado pelo médico.

Segundo Atendimento

N° Atendimento	Nome	Cartão SUS	Nascimento	Atendimento
202	TESTE202	100000000001201	02/07/1991	Atender
205	TESTE205	100000000001204	02/07/1991	Atender
206	TESTE206	100000000001205	02/07/1991	Atender
207	TESTE207	100000000001206	02/07/1991	Atender
208	TESTE208	100000000001207	02/07/1991	Atender
209	TESTE209	100000000001208	02/07/1991	Atender
210	TESTE210	100000000001209	02/07/1991	Atender
211	TESTE211	100000000001210	02/07/1991	Atender
212	TESTE212	100000000001211	02/07/1991	Atender
213	TESTE213	100000000001212	02/07/1991	Atender
214	TESTE214	100000000001213	02/07/1991	Atender
215	TESTE215	100000000001214	02/07/1991	Atender
216	TESTE216	100000000001215	02/07/1991	Atender
217	TESTE217	100000000001216	02/07/1991	Atender
218	TESTE218	100000000001217	02/07/1991	Atender

1 2 3 4 1 - 15 of 50 Go

Figura 3 – Tela principal do segundo atendimento.

Realização do atendimento

Dados do paciente

Nome: TESTE208

Data de Nascimento: 02/07/1991

Cartão SUS: 100000000001207

Anamnese

Altura (m): 1.70 Peso (kg): 70 Pressão (mmHg): 12/8 Temperatura (°C): 35

Pulsção (bpm): 60 HGT (mg/dl): 35 Sat. O2 (%): 50

Descrição:
O paciente tem febre e dor de cabeça

Classificação: VERDE Sala: SALA 2

SALVAR **LIMPAR**

Figura 4 – Tela para preenchimento dos valores da anamnese do segundo atendimento.

No terceiro, e último, atendimento, as filas são agrupadas de acordo com a sala, como visto na Figura 5, e, para cada atendimento, o médico tem um campo para o preenchimento da descrição médica, como na Figura 6, completando a anamnese do paciente atendido, e, conseqüentemente, finalizando o fluxo básico.

Terceiro Atendimento

Sala: SALA 2

Nº Atendimento	Nome	Nascimento	Classificação	Atendimento
310	TESTE310	02/07/1991	VERMELHO	Atender
317	TESTE317	02/07/1991	VERMELHO	Atender
316	TESTE316	02/07/1991	LARANJA	Atender
306	TESTE306	02/07/1991	VERDE	Atender
318	TESTE318	02/07/1991	VERDE	Atender
325	TESTE325	02/07/1991	VERDE	Atender
329	TESTE329	02/07/1991	VERDE	Atender
344	TESTE344	02/07/1991	VERDE	Atender
346	TESTE346	02/07/1991	VERDE	Atender
350	TESTE350	02/07/1991	VERDE	Atender
376	TESTE376	02/07/1991	VERDE	Atender
301	TESTE301	02/07/1991	AZUL	Atender

1 2 3 4 5 1 - 12 of 50 Go

Figura 5 – Tela principal do terceiro atendimento (para a sala 2).

Realização do atendimento

Dados do paciente

Nome: TESTE318

Data de Nascimento: 02/07/1991

Cartão SUS: 100000000001317

Anamnese

Altura (m): 1.7 Peso (kg): 70 Pressão (mmHg): 12/8 Temperatura (°C): 35

Pulsção (bpm): 60 HGT (mg/dl): 35 Sat. O2 (%): 50

Descrição:
O paciente tem febre e dor de cabeça

Classificação: VERDE Sala: SALA 2

Descrição Médica:

SALVAR **LIMPAR**

Figura 6 – Tela para preenchimento dos valores da anamnese do terceiro atendimento.

3.2. Ficha de classificação

No terceiro atendimento, realizado pelo médico, existe a necessidade da geração de uma ficha. Esse documento é chamado de ficha de classificação, e nela contêm os dados pessoais dos pacientes, informações sobre a anamnese, exames solicitados, dentre outros.

Atualmente, essa ficha é impressa em branco e preenchida manualmente por um profissional do hospital, gerando um gasto de tempo desnecessário. Uma vez que as informações dos pacientes e dos atendimentos estão armazenadas de tal forma que não é possível haver uma comunicação entre a ficha e esses dados, pensamos em uma característica que fosse possível diminuir o tempo gasto no preenchimento deste documento.

É valido ressaltar que, o módulo engloba menos dados do que aqueles disponíveis na ficha, por exemplo, exames não são considerados nesse momento. Com isso, apenas parte da ficha terá geração automatizada, restando outros dados que devem ter suas anotações manuais inalteradas.

Com essa funcionalidade, ao finalizar o terceiro atendimento, automaticamente ocorre a geração desse documento, com os dados já conhecidos

devidamente preenchidos. Na figura 7, a ficha de classificação é exemplificada com valores em branco.


 HOSPITAL GERAL LUIZ VIANA FILHO FICHA DE ATENDIMENTO ENTRADA		Registro:	
NOME:		IDADE:	NASC:
ENDEREÇO:		BAIRRO:	CIDADE:
Nº DOC:	FONE:	EST. CIVIL:	SEXO:
RESPONSÁVEL:		CARTÃO SUS:	
ANAMNESE			
Altura (m):	Peso (kg):	Pressão (mmHg):	
Temperatura (°C):		Pulsção (bpm):	
HGT (mg/dl):		Saturação O2 (%):	
SUSPEITA DIAGNÓSTICA			
HORA	EXAME SOLICITADO	RESULTADO	HORA
TRATAMENTO SUBMETIDO			
HORAS	OBSERVAÇÕES DA ENFERMAGEM		
ALTA:	<input type="checkbox"/> OBS. <input type="checkbox"/> P/ RES <input type="checkbox"/> TRANSFERENCIA P/ _____ <input type="checkbox"/> INTERNAMENTO <input type="checkbox"/> ÓBITO		
Médico Plantonista:			
Documento gerado em 06/11/2013 às 21:12:14			

Figura 7 – Modelo, em branco, da ficha de classificação.

3.3. Cadastros

Uma funcionalidade básica em qualquer sistema é o cadastro. No sistema proposto, há três cadastros fundamentais para que o módulo alcance os objetivos propostos: cadastros de pacientes, de usuários e de medicamentos.

No primeiro cadastro (Figura 8), são cadastrados dados do paciente, como, dados gerais, localização, contato e outras informações. Nesse formulário, ainda existem um campo para que seja preenchido o nome do acompanhante, isso é relevante quando, ao invés de um cadastro de paciente, a intenção é que seja gerado um atendimento.

A interface de cadastro de pacientes apresenta uma barra de navegação superior com links para Início, Cadastrar, Atendimentos, Relatórios e Sair. O formulário principal, intitulado 'Paciente', contém as seguintes seções:

- Dados gerais:** Campos para Nome, Nascimento, CPF, Nome mãe, Cartão SUS, Apelido, Naturalidade, N° prontuário, RG, Emissor (menu suspenso) e Sexo (menu suspenso).
- Localização:** Campos para Rua, Bairro, Complemento, Tipo endereço (menu suspenso), Cidade, CEP, Número, Estado (menu suspenso) e Referências.
- Contato:** Campos para DDI, DDD, Operadora (menu suspenso), Número, Tipo contato (menu suspenso) e E-mail.
- Outras informações:** Campos para Tipo Sanguíneo (menu suspenso), Escolaridade (menu suspenso), Profissão (menu suspenso), Estado Civil (menu suspenso), Etnia (menu suspenso) e Faleceu? (menu suspenso com opções 'Sim' e 'Não').

Na base do formulário, há botões 'SALVAR' e 'LIMPAR' em azul, um campo de texto para 'Acompanhante:' em laranja e um botão 'ATENDER' em laranja.

Figura 8 – Tela para cadastro de pacientes.

No cadastro de usuários, são inseridas informações condizentes à inserção do usuário, como o nome, senha, pergunta e resposta secreta, entre outros, como podemos visualizar na Figura 9.

A interface de cadastro de usuários, intitulada 'Novo Usuário', possui uma barra de navegação superior idêntica à da Figura 8. O formulário contém os seguintes campos:

- Informações:** Campos para Nome, Senha, Confirmação da senha, E-mail, Tipo (menu suspenso), Pergunta secreta (menu suspenso) e Resposta.

Na base do formulário, há botões 'SALVAR' e 'LIMPAR' em azul.

Figura 9 – Tela para cadastro de usuários.

Por fim, na inclusão de medicamento, como na Figura 10, é incluído o nome e o código referente ao Sistema Integrado de Material, Patrimônio e Serviço (SIMPAS) para cada item cadastrado.

Figura 10 – Tela para cadastro de medicamentos.

3.4. Separação MVC

A evolução dos padrões de projetos ocorreu de modo paralelo à melhoria da separação das camadas que compõe um sistema, de modo a tornar o desenvolvimento mais eficiente e o produto mais fácil de ser aperfeiçoado e de ser mantido. Com a divisão das camadas nesse projeto (*Model/View/Controller*), as atribuições de cada nível tornam-se claras e isoladas, de modo que ambas façam tarefas estritamente específicas.

Ilustração a utilização, podemos citar um cadastro de paciente: o que é visto pelo usuário no seu navegador está definido na *View*. Assim que o formulário estiver devidamente preenchido e o botão para salvar for acionado, a *View* chamará o *Controller* enviando o objeto a ser salvo e a ação desejada (nesse caso, salvar um paciente). Por fim, o *Controller* tendo posse do objeto e da ação, envia ao *Model* o que deve ser incluído no banco de dados, completando assim o ciclo de requisições entre as camadas.

Como exemplo de benefício do MVC, podemos imaginar que um novo campo deve ser inserido no formulário. Desse modo, sabe-se, de imediato, o que deve ser modificado no código fonte: o novo campo deve ser adicionado ao formulário (alterando a *View*), e mais um atributo deve ser salvo no banco de dados (alterando o *Model*).

3.5. Segurança dos dados

Com o objetivo de manter a confidencialidade dos dados, característica necessária não apenas na área da Tecnologia da Informação (TI), utilizamos a criptografia da senha. Para essa finalidade, existem alguns algoritmos *Hashs*, como MD5 (*Message-Digest Algorithm 5*) e SHA (*Secure Hash Algorithm*), fazendo uso de uma codificação chamada “de mão única”. A expressão citada na literatura faz alusão a uma característica dos métodos, pois uma vez criptografada, a palavra não pode sofrer a operação inversa. A senha é conhecida exclusivamente pelo usuário, não sendo acessível pelos administradores do sistema. Outro atributo do processo é que se duas palavras distintas são criptografadas, então é praticamente impossível terem o mesmo resultado, além do mais, dadas duas entradas iguais, as saídas serão sempre iguais (ROSA, 2004).

Projetada pela *National Security Agency* (NSA), nesse projeto as senhas foram criptografadas com utilizando o SHA-1, função mais utilizada dentre as variantes dos SHA.

3.6. Ajustável em diferentes resoluções

Por questões de acessibilidade, um quesito importante é a capacidade do sistema, principalmente web, se ajustar a resolução do monitor e ao *zoom* do browser utilizado pelo usuário.

Para tal propósito, utilizamos um recurso do CSS3 (*Media Queries*), que, a depender das configurações do dispositivo de saída utilizado, ajusta alguns valores

da folha de estilo para que a página se adeque ao formato esperado. Além disso, como fizemos uso do *framework formee*, o mesmo ajusta o formulário de acordo com a definição utilizada pelo usuário.

3.7. Validação dos formulários

A validação de campos de um formulário é um aspecto importante em qualquer sistema, isso por que garante que os dados recebidos pelo servidor seja exatamente como esperado.

Como exemplo, podemos supor que enviemos letras, ao invés de números, para o campo CPF no cadastro de pacientes. Certamente, o respectivo campo no banco de dados para armazenar essa informação espere apenas dígitos, com isso, visto a incompatibilidade dos dados, ocorrerá algo inesperado. Para resolver esse problema, usamos a validação do lado do cliente a partir de *JQuery* e HTML5, para que, de certo modo, fizesse com que os usuários enviassem dados compatíveis com os esperados pela aplicação.

Do *JQuery* fizemos uso de dois mecanismos. O primeiro são as máscaras (*mask*) e o exemplo do CPF pode ser visto na Figura 11. Esse mecanismo faz com que o campo tenha exatamente uma quantidade de dígitos e que os mesmos sejam de apenas um tipo. Além disso, utilizamos regras (*rules*), útil para customização de um campo, que pode depender de diversos fatores, como por exemplo, que o mesmo tenha um tamanho máximo e/ou mínimo, tenha apenas letras e/ou dígitos, que seja um e-mail válido, que seja idêntico a outro campo, que respeite a uma expressão regular, entre outros.



Figura 11 – Exemplo de máscara com JQuery no campo de cadastro do CPF.

Empregamos recursos do HTML5 com propósito de indicar a obrigatoriedade de algum campo. No *JQuery* também é possível realizar tal propósito, porém, o uso

do HTML5 se justifica uma vez que uma mensagem de aviso de obrigatoriedade está previamente implementada, e que surge ao tentar enviar um campo obrigatório sem valor algum, como pode ser visto na Figura 12. Nesse exemplo, ao clicar no botão para entrar no sistema sem inserir valor algum nos campos de usuário e senha.

The image is a screenshot of a web browser displaying the login page for 'SGHospitar'. The page has a light gray background. At the top center is a logo consisting of a red heart with a white ECG line passing through it. Below the logo, the text 'SGHospitar' is displayed in a bold, black, sans-serif font. In the center of the page is a dark gray rectangular box with rounded corners. Inside this box, the word 'Login' is written in white. Below 'Login', there are two input fields. The first is labeled 'Usuário:' and is empty. The second is labeled 'Senha:' and is also empty. To the right of the 'Senha:' label, there is a small yellow warning icon with an exclamation mark and the text 'Preencha este campo.' in red. Below the input fields, there is a blue button with the word 'ENTRAR' in white. To the right of the button is a link that says 'Esqueci a senha' in a smaller, lighter font. At the bottom of the page, there is a dark gray footer bar. On the left side of the footer bar is a small logo of the state of Bahia. On the right side, there are two logos: one for 'SUS' (Sistema Único de Saúde) and another for 'Bahia' with the tagline 'TERRA DE TODOS NÓS'.

Figura 12 – Exemplo de validação com HTML5: campos obrigatórios.

3.8. Certificação

Atualmente existem dois tipos de certificação, isto é, dois Níveis de Garantias de Segurança (NGS): NGS1 e NGS2. Essa classificação de baseia na segurança da informação, pois, enquanto para NGS1 é opcional o uso de certificados digitais para a assinatura digital das informações, no NGS2 essa característica é obrigatória.

De modo geral, para que um software do gênero seja certificado pela SBIS/CFM, algumas recomendações devem ser seguidas. Os requisitos são agrupadas em três grupos: (i) Mandatário, indicam os requisitos obrigatórios; (ii) Recomendado, apontam as condições recomendadas, e que provavelmente poderão tornar-se obrigatórias nas próximas versões; (iii) e Opcional, assinalam para as características relevantes.

A seguir, a Tabela 2 mostra algumas características necessárias para obtenção da certificação NGS1, bem como a classificação do requisito e um indicativo se o mesmo foi desenvolvido no presente projeto. Não faz sentido fazer essa análise para NGS2, uma vez que não possuímos o certificado digital.

Tabela 2 – Algumas características para obtenção da certificação NGS1

Característica	Classificação	Implementado
Todos componentes devem possuir versão com única referência	Mandatário	Sim
Resgatar código fonte de uma versão específica	Recomendado	Sim
Histórico descritivo das versões	Recomendado	Sim
Ter um repositório estruturado com todas as versões	Recomendado	Sim
Utilizar para autenticação: usuário e senha	Mandatário	Sim
Senha criptografada com SHA-1	Mandatário	Sim
Qualidade de senha	Mandatário	Não
Bloquear usuário após X tentativas inválidas de login	Mandatário	Não
Sessão encerrada após período de inatividade	Mandatário	Não
Não permitir exclusão de dados já existentes	Mandatário	Sim
Armazenado e protegido por um SGBD	Mandatário	Sim

Fonte: SBIS, 2009.

3.9. Busca automática

Pensando em poupar tempo na edição de um cadastro ou geração de um atendimento, implantamos nesse módulo uma busca automática que funciona de forma simples.

No final do formulário que cadastra um paciente existe um campo para preenchimento do nome do acompanhante, dado que só é relevante quando se tem a intenção de gerar um atendimento para um paciente específico. Como dito, o solicitante do atendimento deve estar previamente cadastrado no sistema, e, para criar o atendimento, seu cadastro deve ser “encontrado”, bastando digitar parte do seu nome no campo específico utilizado para o cadastro do mesmo dado, como visto na Figura 13.

Paciente

Dados gerais

Nome: ESTE20

TESTE20 - CPF: 10000000019
 TESTE200 - CPF: 10000000199
 TESTE201 - CPF: 10000000200
 TESTE202 - CPF: 10000000201
 TESTE203 - CPF: 10000000202

Nascimento: CPF: Nome mãe: Cartão SUS: Apelido: Naturalidade:

Nº prontuário: RG: Emissor: Sexo:

Localização

Rua: Bairro: Complemento: Tipo endereço: Cidade: CEP:

Número: Estado: Referências:

Contato

DDI: DDD: Operadora: Número: Tipo contato: E-mail:

Outras informações

Tipo Sanguíneo: Escolaridade: Profissão: Estado Civil: Etnia: Faleceu?

SALVAR LIMPAR Acompanhante: ATENDER

Figura 13 – Exemplo da busca automática.

Ao clicar no nome do paciente, os dados do mesmo são automaticamente preenchidos com suas informações, como mostra a Figura 14.

Paciente

Dados gerais

Nome: TESTE202

Nascimento: 02/07/1991 CPF: 100.000.002-01 Nome mãe: Mae Cartão SUS: 100 0000 0000 1201 Apelido: Tes202 Naturalidade: Itabuna

Nº prontuário: 202 RG: 2033542 Emissor: SSP Sexo: M

Localização

Rua: Rua 2 de Julho Bairro: Centro Complemento: AP 7 Tipo endereço: Casa Cidade: cidade CEP: 55555-555

Número: 58 Estado: BA Referências: Prox CCEP

Contato

DDI: 55 DDD: 73 Operadora: oi Número: 9999-99999 Tipo contato: Casa E-mail: teste@teste202

Outras informações

Tipo Sanguíneo: O- Escolaridade: Não alfabetizado Profissão: Desempregado Estado Civil: Solteiro(a) Etnia: Brancos Faleceu?: Não

SALVAR LIMPAR Acompanhante: ATENDER

Figura 14 – Exemplo do uso da busca automática para edição de um paciente cadastrado.

Por fim, para gerar o atendimento, basta preencher o nome do acompanhante e clicar em “Atender”, e com isso, o atendimento do paciente é gerado e o mesmo estará disponível na fila do primeiro atendimento.

Para a edição de um cadastro, o processo é semelhante: o profissional realiza a busca no mesmo campo, e, ao encontrar, clica no nome do paciente e o formulário é automaticamente preenchido com os dados do banco. Com isso, para editar, basta alterar algum valor e clicar em “Salvar”.

A busca automática auxilia no processo de pesquisa, pois faz a consulta ao banco de dados procurando um paciente que tenha parte do nome conforme digitado.

3.10. Ferramentas livres

Uma ferramenta é livre quando qualquer pessoa tem autorização para usar, copiar e modificar, seja na forma original ou já modificada, com ou sem intenção de obter lucros (HEXSEL, 2002). Concordando com Pistori e Pereira (2006), a utilização de softwares livres se justifica pela grande disponibilidade e variedade de boas ferramentas e pela facilidade de acesso e modificação dos códigos fonte.

Nesse sentido, em concordância ao projeto de Lei 188/2013 anteriormente citado, optamos pela utilização exclusiva de ferramentas livres, pois, como se trata de um sistema com processo de desenvolvimento contínuo, devemos levar em consideração que outros desenvolvedores darão prosseguimento ao projeto, além disso, sua utilização e manutenção não agregam custos ao produto, uma vez que são softwares gratuitos, em contrapartida ao e-SUS que necessita de ferramentas proprietárias.

4 CONCLUSÃO

4.1. Considerações finais

Com este projeto desenvolvemos um módulo que, pode gerenciar o setor de urgência e emergência do Hospital Geral Luiz Viana Filho, localizado na cidade de Ilhéus-BA. Damos início a um sistema que futuramente irá gerenciar, com maior detalhamento, setores pertinentes a um hospital, como, farmácia, almoxarifado, enfermarias, ambulatórios, entre outros. Utilizamos apenas ferramentas livres, com o objetivo de facilitar a continuidade e manutenção do mesmo, e de modo que os artefatos empregados não tornem o produto inviável de ser utilizado em estabelecimentos com pouca ou nenhuma disponibilidade financeira para as TICs.

Alguns requisitos necessários para obtenção da certificação NGS1 não foram implementados, não sendo possível, no momento, a obtenção da mesma. Isso se deve ao fato do foco ter sido totalmente direcionado ao desenvolvimento do módulo, para que fosse possível a conclusão da primeira versão, e, como estes requisitos (para certificação) estão listados, serão desenvolvidos futuramente.

4.2. Trabalhos futuros

Com a finalidade de direcionar as próximas etapas desse projeto, listaremos algumas propostas para que pudessem ser analisadas de acordo com a viabilidade e a real necessidade:

- Validação pelos padrões do W3C, tornando o sistema compatível com o padrão definido pelo consórcio, com objetivo, dentre outros, de facilitar a compatibilidade do produto com os *browsers*;
- Implementação de funcionalidade que preencha dados de localização a partir de um CEP indicado, para isso, o sistema deve se comunicar com uma base de dados dos Correios, mantida para essa finalidade;

- Preenchimento automático de todos os dados da ficha de classificação, tornando o processo menos manual possível;
- Desenvolvimento de funcionalidades para que relatórios detalhados sejam gerados, contribuindo na tomada de decisão dos administradores do hospital;
- Implementação dos demais requisitos necessários para obtenção da certificação NGS1, sob responsabilidade do CFM e da SBIS;
- Criação de outros módulos, a fim de integrar o de urgência e emergência, com intuito de gerenciar, com maior nível de detalhamento, um estabelecimento hospitalar.

REFERÊNCIAS

- ACHOUR, M. et al. **Manual do PHP**. 09 de Agosto de 2013. Disponível em <http://www.php.net/manual/pt_BR/index.php>. Acesso em 16 de Agosto de 2013.
- ALMEIDA, M. B. Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares. **Revista Ciência da Informação**, Brasília, v. 31, n. 2, p.5-13, maio 2002. Trimestral. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/140/1745>>. Acesso em: 15 out. 2013.
- BEZERRA, S. M. Prontuário Eletrônico do Paciente: uma ferramenta para aprimorar a qualidade dos Serviços de Saúde. **Revista Meta: Avaliação**, Rio de Janeiro, v. 1, n. 1, p.73-82, jan. 2009. Disponível em: <<http://metaavaliacao.cesgranrio.org.br/index.php/metaavaliacao/article/view/12>>. Acesso em: 12 nov. 2013.
- BORGES, E. N. **Consultas por similaridade em SGBDs comerciais: Estendendo o PostgreSQL**. 2005. 90 f. TCC (Graduação) - Curso de Engenharia da Computação, Universidade Federal do Rio Grande, Rio Grande, 2005.
- BRAGA, R. D. et al. Validação do prontuário eletrônico do paciente em uma instituição de ensino superior em saúde: relato da experiência no módulo Anamnese. **Journal Health Informatics**, São Paulo, v. 5, n. 1, p.30-36, jan. 2013. Disponível em: <<http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/189>>. Acesso em: 17 out. 2013.
- CARVALHO, F.; ALMEIDA, J. M. **Novas Tecnologias de Animação na Web**. Coimbra: s. n., 2010. Disponível em: <<http://docentes.ismt.pt/~cparis/SMM/TP/Seminario.pdf>>. Acesso em: 20 out. 2013.
- CAVALCANTE, R. B.; SILVA, P. C.; FERREIRA, M. N. Sistemas de Informação em Saúde: possibilidades e desafios. **Revista de Enfermagem da UFSM**, Santa Maria, v. 1, n. 2, p.290-300, maio 2011. Disponível em: <<http://cascavel.ufsm.br/revistas/ojs-2.2.2/index.php/reufsm/article/view/2580>>. Acesso em: 01 nov. 2013.
- COLLINS-SUSSMAN, B.; FITZPATRICK, B. W.; PILATO, C. M. **Controle de Versão com Subversion: Para Subversion 1.4**. Califórnia: s. n., 2007.
- CFM. Define prontuário médico e torna obrigatória a criação da Comissão de Revisão de Prontuários nas instituições de saúde. **Resolução n. 1.638**, de 9 de agosto de 2002. Brasília, p.184-185.
- DALL'OGGIO, P. **PHP: Programando com Orientação a Objetos**. 2. ed. São Paulo: Novatec, 2009. 574 p.
- DANTAS, T. C. B. et al. AprendEAD: Ambiente para Educação à Distância Apoiado em Agentes. **Cadernos do IME: Série Informática**, Rio de Janeiro, v. ?, n. ?, p.13-19, jul. 2007.

HEXSEL, R. A. **Software Livre: Propostas de Ações de Governo para Incentivar o Uso de Software Livre**. Curitiba: s. n., 2002. 53 p. Disponível em: <http://www.inf.ufpr.br/pos/techreport/RT_DINF004_2002.pdf>. Acesso em: 10 set. 2013.

ILHÉUS. Constituição (2013). **Projeto de Lei nº 118**, de 27 de agosto de 2013. Dispõe sobre a implantação do prontuário eletrônico do paciente, na rede pública de saúde do município de Ilhéus. Ilhéus, 2013.

JOHANSTON, H. Sistemas de Informação Hospitalar: Presente e Futuro. **Revista Informédica**, EUA, v. 2, n. 1, p.5-9, jul. 1993.
MANCINI, M. P. M.; ARA-SOUZA, A. L.; LOUZADA NETO, F. **Aperfeiçoamento de procedimentos estatísticos para avaliação institucional online: implantação de relatórios armazenáveis**. São Pedro: s. n., 2010.

MARIN, H. de F. Sistemas de informação em saúde: considerações gerais. **Journal Of Informatics**. São Paulo, p. 20-25. jan. 2010. Disponível em: <<http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/4>>. Acesso em: 30 out. 2013.

NARDON, F. B.; FURUIE, S.; TACHINARDI, U. Anais do VII Congresso Brasileiro de Informática em Saúde. In: CONGRESSO BRASILEIRO DE INFORMÁTICA EM SAÚDE, 7., 2000, São Paulo. **Novas Tecnologias para Construção do Prontuário Eletrônico do Paciente**. São Paulo: s. n., 2000. p. 1 - 5.

NIEDERAUER, J. **Desenvolvimento Websites com PHP: Aprenda a criar Websites dinâmicos e interativos com PHP e banco de dados**. São Paulo: Novatec, 2004. 301 p.

PATRÍCIO, C. et al. O prontuário eletrônico do paciente no sistema de saúde brasileiro: uma realidade para os médicos?. **Scientia Medica**, Porto Alegre, v. 3, n. 21, p.121-131, maio 2011. Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/scientiamedica/article/viewFile/8723/6722>>. Acesso em: 11 out. 2013.

PAVAN, W. et al. **Tecnologia WAP na tomada de decisões no manejo de doenças da cultura do trigo**. Ponta Grossa: s. n., 2000. 7 p.

PEREZ, G. **Adoção de inovações tecnológicas: Um estudo sobre o uso de sistemas de informação na área da saúde**. 2006. 243 f. Tese (Doutorado) - Curso de Administração de Empresas, Universidade de São Paulo, São Paulo, 2006.

PINTO, V. B. Prontuário Eletrônico do Paciente: documento técnico de informação e comunicação do domínio da saúde. **Revista Ciência da Informação**, Florianópolis, v. 1, n. 21, p.34-48, abr. 2006. Disponível em: <<http://www.hmtj.org.br/arquivos/hmtj/prontuario.pdf>>. Acesso em: 30 out. 2013.

PISTORI, H.; PEREIRA, M. C. **Utilização de Ferramentas Livres em um Curso de Visão Computacional**. Campo Grande: s. n., 2006. 6 p.

ROCHA, A. R.; COSTA, R.; OTERO, M. **Avaliação e Seleção de Sistemas de Informação Hospitalares**. Rio de Janeiro: s. n., 2001.

ROSA, A. F. **Garantindo segurança com OPENSSL**. 2004. 73 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Luterana do Brasil, Gravataí, 2004.

ROSINI, A. M.; PALMISANO, A. **Administração de sistemas de informação e a gestão do conhecimento**. São Paulo: Pioneira Thomson Learning, 2003. 219 p.

SBIS. **Manual de Certificação para Sistemas de Registro Eletrônico em Saúde (S-RES)**. s. l.: s. n., 2009. 92 p.

SONZA, A. P.; SANTAROSA, L.; CONFORTO, D. Anais do XIX Simpósio Brasileiro de Informática na Educação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 19., 2008, Fortaleza. **Ambientes Virtuais Acessíveis sob a perspectiva de usuários deficientes visuais**. Fortaleza: Sbie, 2008. p. 74 - 84.

SOUZA, R. R.; ALVARENGA, L. A Web Semântica e suas contribuições para a ciência da informação. **Revista Ciência da Informação**, Brasília, v. 33, n. 1, p.132-142, jan. 2004. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/50>>. Acesso em: 22 out. 2013.

STOTHARD, P. **The Sequence Manipulation Suite**: JavaScript Programs for Analyzing and Formatting Protein and DNA Sequences. Alberta: s. n., 2000.

TESTA, G. G. **PrankDev**: Uma ferramenta para criação e manutenção dinâmica de Websites baseada em templates. 2010. 61 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010.

VARGAS, C. M. et al. **Considerações sobre a Utilização dos Requisitos Mandatórios de Funcionalidades do Processo de Certificação de Software em Saúde Brasileiro para Gerenciamento de Dados Assistenciais**. s. l.: s. n., 2010. 8 p.

VOGELGESANG, Kay. **XAMPP documentation**. Disponível em <http://www.apachefriends.org/pt_br/faq-xampp-windows.html>, acessado em 21 de Outubro de 2013.