

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Pós-Graduação em Arquitetura de Software Distribuído

Fabício Teixeira Nascimento

**SISTEMA INTEGRADOR DOS DADOS TRANSFUSIONAIS PARA MELHOR  
EFICIÊNCIA DO CONTROLE TRANSFUSIONAL DE HEMOCOMPONENTES  
PRODUZIDOS E DISTRIBUIDOS PELA FUNDAÇÃO HEMOMINAS**

Belo Horizonte  
2014

Fabício Teixeira Nascimento

**SISTEMA INTEGRADOR DOS DADOS TRANSFUSIONAIS PARA MELHOR  
EFICIÊNCIA DO CONTROLE TRANSFUSIONAL DE HEMOCOMPONENTES  
PRODUZIDOS E DISTRIBUIDOS PELA FUNDAÇÃO HEMOMINAS**

Trabalho apresentado ao Curso de Pós-Graduação em Arquitetura de Software Distribuído da Pontifícia Universidade Católica de Minas Gérias, como requisito parcial para obtenção do título de Especialista em Arquitetura de Software Distribuído.

Orientador: Zenilton Kleber Gonçalves do Patrocínio Júnior

Belo Horizonte

2014

## RESUMO

Este trabalho tem por objetivo apresentar uma solução proposta para integração entre o sistema interno de controle do “ciclo do sangue” na Fundação Hemominas e os sistemas dos demais hospitais atendidos pela instituição, para obter uma gestão e controle mais efetivos sobre os hemocomponentes produzidos na Fundação Hemominas e distribuídos para os mais diversos hospitais de Minas Gerais. Foi desenvolvido um *Web Service* utilizando a linguagem Java, para prover o serviço de inclusão dos dados do paciente receptor do hemocomponente, que será consumido pelos sistemas dos hospitais. Para este trabalho foi desenvolvido um protótipo de clientes para simular a integração com os sistemas dos hospitais desenvolvido utilizando a linguagem de programação Java.

Palavras-chave: *Web Service*. Fundação Hemominas. Hemocomponente. Ciclo do sangue. XML. W3C.

## **ABSTRACT**

This work aims to present a proposal for integration between the internal control system of the "cycle of blood" in Hemominas Foundation and the systems of other hospitals served by the institution solution for a more effective management and control over blood products produced in the Foundation Hemominas and distributed to various hospitals of Minas Gerais. We developed a Web Service using the Java language, to provide the service inclusion of patient data receiver of blood components that will be consumed by the systems of the hospitals. For this work a prototype client was developed to simulate the integration of hospitals with systems developed using the Java programming language.

Keywords: Web Service. Hemominas Foundation. Blood components. Blood cycle. XML. W3C.

## LISTA DE FIGURAS

FIGURA 1 – Modelo do Banco de Dados.....	10
FIGURA 2 – <i>JavaDoc</i> da Classe IntegraWS.....	12
FIGURA 3 – Método RecebeDadosHemocomponente.....	12
FIGURA 4 – Método RecebeDadosPaciente.....	13
FIGURA 5 – Método GravaPaciente.....	14
FIGURA 6 – Método GravaUtilizacao.....	15
FIGURA 7 – Método ListaReacoes.....	15
FIGURA 8 – Método GravarReacoes.....	16
FIGURA 9 – Tela inicial do Cliente Java.....	17
FIGURA 10 – Tela de Cadastro de Pacientes - Cliente Java.....	17
FIGURA 11 – Tela de Registro dos Procedimentos - Cliente Java.....	18
FIGURA 12 – Tela de Registro de Reações Adversas - Cliente Java.....	19
FIGURA 13 – Tela de Listagem de Reações Adversas - Cliente Java.....	19

## **LISTA DE SIGLAS**

XML – Extensible Markup Language

W3C – World Wide Web Consortium

TI – Tecnologia da Informação

SGBD – Sistema de Gerenciamento de Banco de Dados

HTTP – HyperText Transfer Protocol

SMTP – Simple Mail Transfer Protocol

FTP – File Transfer Protocol

JSON – JavaScript Object Notation

SGML – Standard Generalized Markup Language

ISO – International Organization for Standardization

SOA – Arquitetura Orientada a Serviços

IDE – Integrated Development Environment

## SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>8</b>
1.1 Justificativa	8
<b>2. DESENVOLVIMENTO</b>	<b>9</b>
2.1 Banco de dados	9
2.2 Web Service	11
2.3 Cliente Java	16
<b>3. CONCLUSÃO</b>	<b>20</b>
3.1 Trabalhos futuros	20
<b>REFERÊNCIAS</b>	<b>21</b>

## 1. INTRODUÇÃO

A Fundação Centro de Hematologia e Hemoterapia de Minas Gerais, Fundação Hemominas, é o responsável por receber as doações de sangue e pela produção da maior parte dos hemocomponentes [1] utilizados nas cirurgias e nos diversos tratamentos hemoterapicos realizados em todo estado de Minas Gerais. Estes hemocomponentes são distribuídos para uma grande variedade de hospitais, cada um com uma estrutura de TI muito diferente uma da outra. A Fundação Hemominas é responsável por todos estes hemocomponentes e por qualquer reação transfusional ou contaminação que possa ocorrer em algum tratamento hemoterapico ou transfusão de sangue, por isso existe necessidade de se ter o controle total de todos estes hemocomponentes desde a sua doação até a transfusão nos pacientes, sabendo exatamente qual bolsa de sangue foi utilizada, em qual paciente, em qual instituição e por qual motivo.

Devido à variedade das estruturas de TI, para ser mais preciso, estrutura de sistemas informatizados e pela dificuldade de integrar estas informações, pois cada hospital utiliza uma plataforma diferente uma da outra, a Fundação Hemominas ainda não consegue ter de forma prática e ágil a rastreabilidade dos hemocomponentes, sendo necessário buscar as informações de fontes não confiáveis ou de forma manual e desconhecida com os próprios hospitais.

No decorrer deste trabalho é apresentado um protótipo de um *Web Service* que terá como função principal fornecer aos hospitais os dados necessários referente às bolsas de sangue e principalmente receber todos os dados do paciente que recebeu o hemocomponente e o registro de possíveis reações adversas.

### 1.1 Justificativa

Para que a Fundação Hemominas possa ter a rastreabilidade total de seus hemocomponentes distribuídos, é necessária a realização da integração entre os sistemas dos hospitais e o da Fundação Hemominas.

Devido a diversidade de sistemas e plataformas existentes, a solução escolhida foi a utilização da tecnologia *Web Service*, pois esta permite às aplicações enviar e receber dados em formato *Extensible Markup Language*, XML [2]. Cada sistema pode utilizar sua própria linguagem de programação, pois as mensagens são



trocadas em formato universal, o formato XML, que é uma recomendação da *The World Wide Web Consortium*, W3C para representação e transferência de dados.

## 2. DESENVOLVIMENTO

O *Web Service* do Sistema Integrador funcionará como uma ponte entre o atual sistema de controle e gerenciamento do Ciclo do Sangue e os sistemas dos demais hospitais, ele será o responsável por prover as informações referentes aos hemocomponentes e por receber as informações referente aos procedimentos realizados com estes hemocomponentes e de seus respectivos pacientes.

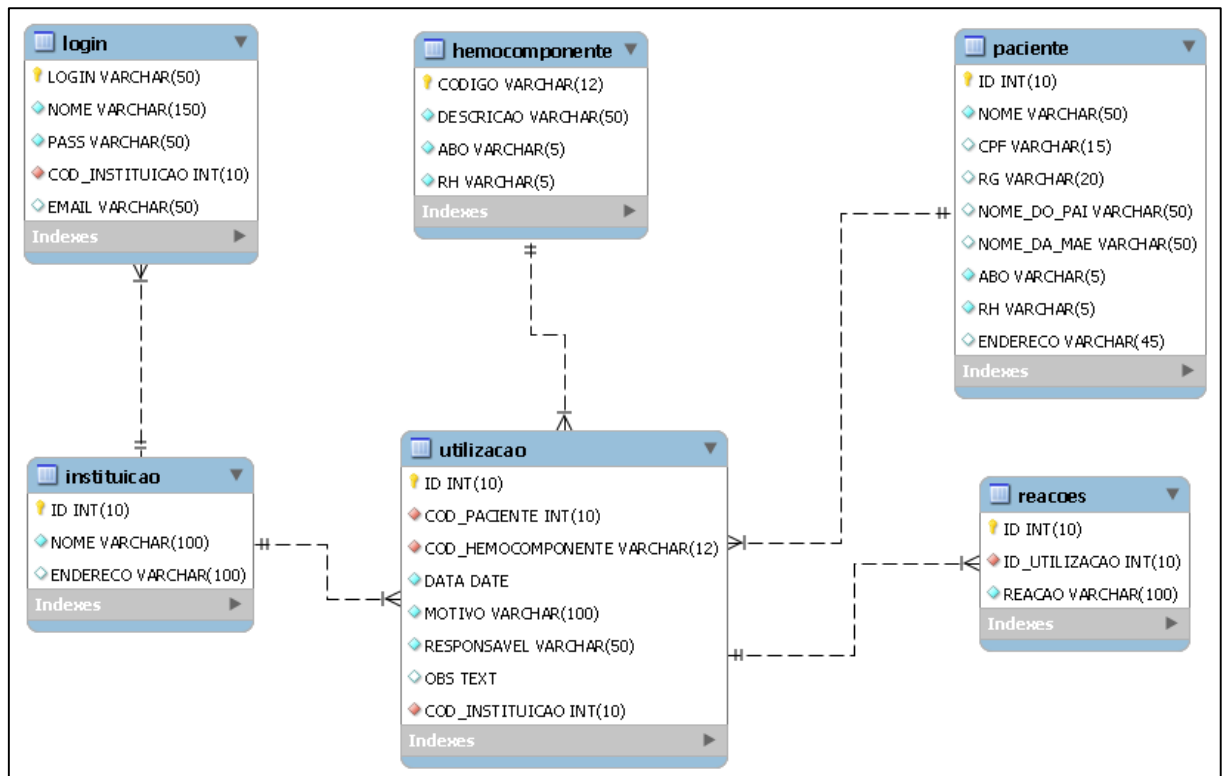
### 2.1. Banco de dados

Existem diversos tipos de Sistemas Gerenciadores de Banco de Dados, SGBD, disponíveis no mercado mas a escolha do SGBD para este trabalho foi o MySQL, por ser um SGBD gratuito, devido já ser utilizado na empresa, pela baixa complexidade para o desenvolvimento e facilidade de manutenção e gerenciamento do mesmo.

Foi utilizada a ferramenta MySQL *Workbench* para realizar a modelagem e inserção dos dados pré-definidos. O MySQL *Workbench* é uma ferramenta de gerenciamento de SGBD que possui diversas funcionalidades, como *backup* e *restore* de banco de dados, execução de *scripts* e geração de modelos.

O banco de dados do sistema Integrador foi modelado para simular a integração com o sistema de gerenciamento do Ciclo do Sangue, e é constituído das seguintes tabelas: Login, Instituicao, Hemocomponente, Paciente, Utilizacao e Reacoes. A Figura 1 mostra o diagrama de entidade-relacionamento do banco de dados do sistema Integrador, gerado com o MySQL *Workbench*.

**Figura 1: Modelo do banco de dados**



**Login:** Armazena as informações referentes ao usuário que estará utilizando o sistema no lado servidor.

**Instituicao:** Armazena os dados referentes à instituição (hospital), que recebeu o hemocomponente para a realização do procedimento.

**Hemocomponente:** Armazena os dados dos hemocomponentes que foram distribuídos para os hospitais.

**Pacientes:** Armazena os dados do paciente que recebeu a transfusão ou tratamento. Estes dados serão recebidos diretamente dos hospitais via arquivos XML.

**Utilizacao:** Armazena os dados referentes ao procedimento realizado no paciente e suas devidas particularidades. Estes dados serão recebidos diretamente dos hospitais via XML.

**Reacoes:** Armazena os dados referentes a possíveis reações transfusionais que possam ter ocorrido em algum dos procedimentos. Estes dados serão recebidos diretamente dos hospitais via XML.

## 2.2. Web Service

Web Service consiste de uma camada de software baseada em uma arquitetura orientada a serviços em que as interfaces são baseadas em protocolos da Internet (HTTP, SMTP, FTP). Além disso, as mensagens enviadas podem ser em XML ou JSON, *JavaScript Object Notation*. Essa última apresenta crescente popularidade, embora a primeira ainda seja a mais utilizada e difundida.

O XML, Extensible Markup Language, de acordo com o W3C, é um formato de texto simples, muito flexível derivado do SGML, *Standard Generalized Markup Language* [3] (ISO 8879). Originalmente concebido para enfrentar os desafios da publicação eletrônica em larga escala, o XML também está desempenhando um papel cada vez mais importante na troca de uma ampla variedade de dados na Web e em outros lugares.

Devido a essas características, o XML tornou-se amplamente utilizado na integração de sistemas, já que um documento XML é escalável e facilmente lido por aplicações sendo executadas em plataformas diferentes. Ele também se tornou uma ferramenta essencial em sistemas que utilizam Arquitetura Orientada a Serviços (SOA), sobretudo na troca de mensagens entre Web Services e seus clientes.

O *Web Service* desenvolvido para este trabalho, denominado IntegraWS, possui as funções básicas e necessárias para a perfeita integração com os sistemas dos hospitais, ele possui seis métodos disponíveis para utilização: *getData*, *getPaciente*, *insertPaciente*, *insertUtilizacao*, *listarReacoes* e *insertReacoes*. Estes métodos foram expostos no *Web Service* utilizando a anotação *@WebMethod*, respectivamente com os seguintes nomes: *RecebeDadosHemocomponente*, *RecebeDadosPaciente*, *GravaPaciente*, *GravaUtilizacao*, *ListaReacoes* e *GravaReacoes*, conforme demonstrado na Figura 2.

**Figura 2: JavaDoc da Classe IntegraWS**

Method Summary	
Methods	
Modifier and Type	Method and Description
<b>Hemocomponente</b>	<b>getData</b> (java.lang.String numeroBolsa) Método que envia os dados do hemocomponente para o hospital client.
<b>Paciente</b>	<b>getPaciente</b> (int id) Método que envia os dados do paciente para o hospital client.
boolean	<b>insertPaciente</b> (java.lang.String nome, java.lang.String cpf, java.lang.String rg, java.lang.String pai, java.lang.String mae, java.lang.String aboPac, java.lang.String rhPac, java.lang.String endereco) Método que recebe os dados dos pacientes para gravar
boolean	<b>insertReacoes</b> (Reacoes reacao) Método responsável pela gravação das reações adversas nas transfusões
boolean	<b>insertUtilizacao</b> (int codPaciente, java.lang.String codHemocomponente, java.lang.String data, java.lang.String motivo, java.lang.String responsavel, java.lang.String obs, int codInstituicao) Método que grava os dados referente a utilização do hemocomponente
java.util.List<Reacoes>	<b>listReacoes</b> (java.lang.String nomePaciente) Método que envia os dados de reações transfusionais para os hospitais client.

**RecebeDadosHemocomponente:** recebe o código da bolsa do hemocomponente, busca as informações na base de dados e retorna um objeto do tipo Hemocomponente, que contém as seguintes informações: Código, Descrição, ABO e RH. Estas informações serão recebidas pelos sistemas dos hospitais. (Ver Figura 3)

**Figura 3: Método RecebeDadosHemocomponente**

```
@WebMethod(operationName = "RecebeDadosHemocomponente")
public Hemocomponente getData(@WebParam(name = "numeroBolsa") String numeroBolsa) {
    HemocomponenteDAO hemoDao = new HemocomponenteDAO();
    Hemocomponente bolsa = new Hemocomponente();

    bolsa = hemoDao.getHemocomponente(numeroBolsa);

    return bolsa;
}
```

**RecebeDadosPaciente:** recebe o código do paciente que irá receber o hemocomponente, busca as informações na base de dados e retorna um objeto do tipo Paciente, que contém as seguintes informações: Código, Nome, CPF, RG, Nome do Pai, Nome da Mãe, ABO, RH e Endereço do paciente. (Ver Figura 4)

**Figura 4: Método RecebeDadosPaciente**

```
@WebMethod(operationName = "RecebeDadosPaciente")
public Paciente getPaciente(@WebParam(name = "id") int id) {
    PacienteDAO pacDao = new PacienteDAO();
    Paciente paciente = new Paciente();

    paciente = pacDao.getPaciente(id);

    return paciente;
}
```

**GravaPaciente:** método responsável por efetuar o cadastro dos pacientes no banco de dados. Recebe os dados: Nome, CPF, RG, Nome do Pai, Nome da Mãe, ABO, RH e Endereço do paciente e efetua o cadastro no banco de dados. Se o cadastro for efetuado corretamente, o método retorna um valor *booleano* verdadeiro, no caso de acontecer algum erro, retorna falso. (Ver Figura 5)

**Figura 5: Método GravaPaciente**

```

@WebMethod(operationName = "GravaPaciente")
public boolean insertPaciente(
    @WebParam(name = "nome") String nome,
    @WebParam(name = "cpf") String cpf,
    @WebParam(name = "rg") String rg,
    @WebParam(name = "pai") String pai,
    @WebParam(name = "mae") String mae,
    @WebParam(name = "aboPac") String aboPac,
    @WebParam(name = "rhPac") String rhPac,
    @WebParam(name = "enderco") String endereco) {

    PacienteDAO pacienteDao = new PacienteDAO();
    Paciente paciente = new Paciente();

    paciente.setNome(nome);
    paciente.setCpf(cpf);
    paciente.setRg(rg);
    paciente.setNomeDoPai(pai);
    paciente.setNomeDaMae(mae);
    paciente.setAbo(aboPac);
    paciente.setRh(rhPac);
    paciente.setEndereco(endereco);

    boolean retorno = pacienteDao.insert(paciente);

    return retorno;
}

```

**GravaUtilizacao:** principal método do *Web Service IntegraWS*, pois após realizar o cadastro do paciente ele vincula o hemocomponente utilizado com paciente que o recebe. Inclui também todos os dados referente à utilização do hemocomponente. O método GravaUtilização recebe os seguintes parâmetros: Código do Paciente, Código do Hemocomponente, Data do procedimento, Motivo do procedimento, Responsável pelo procedimento, Código da instituição e um campo para Observações. Se o cadastro for efetuado corretamente, o método retorna um valor *booleano* verdadeiro, no caso de acontecer algum erro, retorna falso. (Ver Figura 6)

**Figura 6: Método GravaUtilizacao**

```

@WebMethod(operationName = "GravaUtilizacao")
public boolean insertUtilizacao(
    @WebParam(name = "codPaciente") int codPaciente,
    @WebParam(name = "codHemocomponente") String codHemocomponente,
    @WebParam(name = "data") String data,
    @WebParam(name = "motivo") String motivo,
    @WebParam(name = "responsavel") String responsavel,
    @WebParam(name = "obs") String obs,
    @WebParam(name = "codInstituicao") int codInstituicao) {

    UtilizacaoDAO procedimentoDao = new UtilizacaoDAO();
    Utilizacao procedimento = new Utilizacao();

    procedimento.setCodPaciente(codPaciente);
    procedimento.setCodHemocomponente(codHemocomponente);
    procedimento.setData(data);
    procedimento.setMotivo(motivo);
    procedimento.setResponsavel(responsavel);
    procedimento.setObs(obs);
    procedimento.setCodInstituicao(codInstituicao);

    boolean retorno = procedimentoDao.insert(procedimento);

    return retorno;
}

```

**ListaReacoes:** método responsável por buscar no banco de dados e listar todas as reações de um paciente específico, exibindo os dados do paciente, do procedimento e a reação ocorrida. Este métodos recebe como parâmetro o nome ou parte do nome do paciente e retorna uma lista de reações que ocorrerem com o paciente. O objeto retornado é uma lista do tipo List<Reacoes>. O objeto Reacoes possui os seguintes dados que serão retornados: código do paciente, nome do paciente, código do procedimento, data do procedimento e reação ocorrida. (Ver Figura 7)

**Figura 7: Método ListaReacoes**

```

@WebMethod(operationName = "ListarReacoes")
public List<Reacoes> listReacoes(@WebParam(name = "nomePaciente") String nomePaciente) {
    ReacoesDAO reacoesDao = new ReacoesDAO();
    return reacoesDao.getReacoes(nomePaciente);
}

```

GravarReacoes: método responsável pela gravação dos dados referentes às reações transfusionais ocorridas nos pacientes durante a realização ou mesmo após algum procedimento. Este método recebe como parâmetro um objeto do tipo Reacoes, que contém os dados do paciente, do procedimento que foi realizado e também a reação que ocorreu. Os dados contidos no objeto Reacoes são: código do paciente, código do procedimento, data do procedimento e reação ocorrida. Se o cadastro for efetuado corretamente, o método retorna um valor *booleano* verdadeiro, no caso de acontecer algum erro, retorna falso. (Ver Figura 8)

**Figura 8: Método GravarReacoes**

```
@WebMethod(operationName = "GravarReacoes")
public boolean insertReacoes(@WebParam(name = "reacao") Reacoes reacao) {
    ReacoesDAO reacoesDao = new ReacoesDAO();
    return reacoesDao.insert(reacao);
}
```

### 2.3. Cliente Java

O primeiro protótipo do cliente foi desenvolvido simulando o sistema de um hospital que hoje já trabalha com a Fundação Hemominas mas não possui nenhuma integração. Este protótipo foi desenvolvido na linguagem JavaEE e utilizando a ferramenta NetBeans como IDE de desenvolvimento.

Foram criadas algumas páginas *web* para simular a utilização do sistema por parte do hospital, que neste caso foi utilizado o exemplo do Hospital Madre Tereza. As páginas desenvolvidas são para as funções de Cadastro de Pacientes, Registro de Procedimentos, Registro de Reações Adversas e Listagem de Reações Adversas, conforme apresentado nas figuras abaixo:



Figura 9: Tela inicial do Cliente Java



A screenshot of the initial screen of the Java Client. At the top, there is a header with the logo of 'HOSPITAL MADRE TERESA' and the text 'INSTITUTO DAS PEQUENAS MISSIONÁRIAS DE MARIA IMACULADA'. Below the header, the main content area has a title 'Hospital Madre Tereza' and a subtitle 'Protótipo do sistema para integração com o Hemominas'. Underneath, there is a list of four links: 'Cadastrar paciente', 'Registra procedimentos', 'Registra Reações Adversas', and 'Visualiza Reações Adversas'.

**HOSPITAL MADRE TERESA**  
INSTITUTO DAS PEQUENAS MISSIONÁRIAS DE MARIA IMACULADA

**Hospital Madre Tereza**  
Protótipo do sistema para integração com o Hemominas

- [Cadastrar paciente](#)
- [Registra procedimentos](#)
- [Registra Reações Adversas](#)
- [Visualiza Reações Adversas](#)

A Figura 9 representa a tela inicial do protótipo do Client Java. Na tela principal existe um link para direcionar o usuário para cada um dos procedimentos a serem realizados.

Figura 10: Tela de Cadastro de Pacientes - Cliente Java



A screenshot of the patient registration screen. It features the same header as Figure 9. Below the header, there is a link '<< VOLTAR' and the title 'Hospital Madre Tereza' with the subtitle 'Protótipo do sistema para integração com o Hemominas'. The main form area contains several labels and input fields: 'Nome do paciente:', 'CPF:', 'RG:', 'Nome do pai:', 'Nome da mãe:', 'ABO do paciente:', 'RH do paciente:', and 'Endereço do paciente:'. At the bottom, there are two buttons: 'Gravar' and 'Limpar'.

[<< VOLTAR](#)

**Hospital Madre Tereza**  
Protótipo do sistema para integração com o Hemominas

Nome do paciente:

CPF:

RG:

Nome do pai:

Nome da mãe:

ABO do paciente:

RH do paciente:

Endereço do paciente:

A Figura 10 representa a tela de cadastro de pacientes, onde é realizado todo o cadastro do paciente que receberá o tratamento. Esta tela recebe os seguintes campos: Nome do paciente, CPF, RG, Nome do Pai, Nome da Mãe, ABO do paciente, RH do paciente e Endereço do paciente. Estes dados são transmitidos via Web Service e gravados no Banco de Dados da Fundação Hemominas.

**Figura 11: Tela de Registro dos Procedimentos - Cliente Java**

**HOSPITAL MADRE TERESA**  
INSTITUTO DAS PEQUENAS MISSIONÁRIAS DE MARIA IMACULADA

[<< VOLTAR](#)

**Hospital Madre Tereza**  
Protótipo do sistema para integração com o Hemominas

**Hemocomponente:** 12014002009

**Descrição:** CONCENTRADO DE HEMACIAS  
**ABO:** A  
**RH:** P

**Nome do paciente:** FABRICIO TEIXEIRA NASCIMENTO  
**CPF:** 04746834601  
**RG:** M8243589  
**Nome do pai:** PAI DO FABRICIO  
**Nome da mãe:** MAE DO FABRICIO  
**ABO paciente:** O  
**RH paciente:** P  
**Endereço:** Av. General Carlos Guedes, 205

**Data do procedimento:**

**Motivo:**

**Responsavel:**

**Observações:**

A Figura 11 representa a tela de Registro dos Procedimentos, onde o usuário seleciona o hemocomponente a ser utilizado, este hemocomponente deve estar cadastrado na base de dados do Hemominas. O usuário realiza a busca do paciente, que também deve estar cadastrado no banco de dados do Hemominas. Após selecionar estes dados, são lançados os dados referentes ao procedimento realizado, realizando assim o vínculo entre o paciente e o hemocomponente utilizado.

**Figura 12: Tela de Registro de Reações Adversas - Cliente Java**



**Hospital Madre Tereza**  
INSTITUTO DAS PEQUENAS MISSIONÁRIAS DE MARIA IMACULADA

[<< VOLTAR](#)

Protótipo do sistema para integração com o Hemominas

Código do Paciente:

Nome do paciente: FABRICIO TEIXEIRA NASCIMENTO  
CPF: 04746834601  
RG: M8243589  
Nome do pai: PAI DO FABRICIO  
Nome da mãe: MAE DO FABRICIO  
ABO paciente: O  
RH paciente: P  
Endereço: Av. General Carlos Guedes, 205

Código do Procedimento:

Reação:

A Figura 12 representa a tela de Reações Adversas, no qual o responsável pelo procedimento registra todos os tipos de reações ou intercorrências relacionadas com paciente. Estas reações podem ser listadas na tela de Listagem de Reações adversas, conforme apresentado na Figura 13.

**Figura 13: Tela de Listagem de Reações Adversas - Cliente Java**



**Hospital Madre Tereza**  
INSTITUTO DAS PEQUENAS MISSIONÁRIAS DE MARIA IMACULADA

[<< VOLTAR](#)

Protótipo do sistema para integração com o Hemominas

Nome do Paciente:

ID Paciente	Nome Paciente	ID Procedimento	Data Procedimento	Reação
1	FABRICIO TEIXEIRA NASCIMENTO	1	2014-12-01	testeste
2	ANDREIA ARAUJO DE ALMEIDA	2	2013-12-30	asdfsdf
3	PACIENTE 3	4	2014-11-04	dor de cabeça
1	FABRICIO TEIXEIRA NASCIMENTO	1	2014-12-01	asdfsdf

### 3. CONCLUSÃO

A realização deste trabalho como um todo possibilitou a utilização de diferentes conceitos adquiridos durante o curso de pós-graduação em Arquitetura de Software Distribuído e proporcionou um aprofundamento em desenvolvimento de *Web Services* utilizando a plataforma JavaEE.

Com o desenvolvimento do *Web Service* IntegraWS, que será apresentado à diretoria da Fundação Hemominas, a instituição obterá um ganho significativo no tempo referente a pesquisas de Retrovigilância [4] pois diminuirá as pesquisas realizadas manualmente e também obterá um nível de assertividade bem significativo, pois os processos serão registrados de forma informatizada e em tempo real.

Para que o sistema funcione perfeitamente e a Fundação Hemominas consiga atingir os objetivos propostos, é necessário que seja realizada uma campanha com os demais hospitais para que estes utilizem o *Web Service* IntegraWS e realizem a integração com a Fundação Hemominas para manter todos os dados atualizados.

#### 3.1. Trabalhos futuros

Neste projeto há possibilidade de implementação de diversas novas funcionalidades, visto que o projeto ainda não foi apresentado à Fundação Hemominas e nem aos hospitais que consumirão o serviço. Após realizar esta apresentação, serão analisadas mais adequadamente as necessidades dos clientes, e serão desenvolvidas estas funcionalidades solicitadas.

Com o aumento da complexidade do sistema será necessário a utilização de alguns *frameworks* que auxiliarão no desenvolvimento, como Maven, Hibernate e Struts MVC. Com a utilização destes *frameworks*, será necessário um tempo de pesquisa e aprendizado por parte da equipe de desenvolvimento. Este período de aprendizagem será um período muito proveitoso para o desenvolvimento profissional dos membros que estiverem na equipe.

## REFERÊNCIAS

- [1] “Hemocomponente”,  
[http://portal.anvisa.gov.br/wps/wcm/connect/fd337280474597529fcbdf3fbc4c6735/RDC\\_n%C2%BA\\_57.pdf?MOD=AJPERES](http://portal.anvisa.gov.br/wps/wcm/connect/fd337280474597529fcbdf3fbc4c6735/RDC_n%C2%BA_57.pdf?MOD=AJPERES), acessado em 10/11/2014.
- [2] “Extensible Markup Language (XML)”, <http://www.w3.org/XML/>, acessado em 10/11/2014.
- [3] “Standard Generalized Markup Language”, <http://www.w3.org/MarkUp/SGML/>, acessado em 10/11/2014.
- [4] “Retrovigilância”,  
[http://portal.anvisa.gov.br/wps/wcm/connect/fd337280474597529fcbdf3fbc4c6735/RDC\\_n%C2%BA\\_57.pdf?MOD=AJPERES](http://portal.anvisa.gov.br/wps/wcm/connect/fd337280474597529fcbdf3fbc4c6735/RDC_n%C2%BA_57.pdf?MOD=AJPERES), acessado em 10/11/2014.