

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

RELATÓRIO DE ESTÁGIO

DESENVOLVIMENTO DO MÓDULO CADASTRO DE ALUNOS
DO SISTEMA SISLEGAL E SUPORTE TÉCNICO EM
INFORMÁTICA

ANTONIO ALEXANDRE MOURA COSTA

Estagiário

TIAGO LIMA MASSONI

Orientador Acadêmico

UIRÁ VASCONCELOS ALENCAR DE ASSIS

Supervisor Técnico

Campina Grande – PB

1º de julho de 2011

**DESENVOLVIMENTO DO MÓDULO CADASTRO DE ALUNOS
DO SISTEMA SISLEGAL E SUPORTE TÉCNICO EM
INFORMÁTICA**

APROVADO EM _____

BANCA EXAMINADORA

Prof. Tiago Lima Massoni, DSc.

ORIENTADOR ACADÊMICO

Prof.^a Joseana Macêdo Fechine, DSc.

MEMBRO DA BANCA

Prof. José Eustáquio Rangel de Queiroz, DSc.

MEMBRO DA BANCA



Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

AGRADECIMENTOS

Agradeço ao Ministério Público da Paraíba pela grande oportunidade de participação no estágio, que me proporcionou alcançar um novo patamar tanto na vida acadêmica quanto na profissional.

Agradeço ao meu atual orientador acadêmico, Prof. Tiago Lima Massoni, pela disposição e apoio durante esse período.

Agradeço ao meu antigo orientador acadêmico, Prof. Eustáquio Rangel, pois os frutos desse trabalho são também consequência do processo anterior de estágio.

Ao chefe do departamento de desenvolvimento do Ministério Público da Paraíba, que orientou durante o processo de desenvolvimento do sistema em questão.

A minha família, por todo incentivo, amor e condições fornecidas para permanência no estágio.

A Mithylenny, pela compreensão, companheirismo, carinho e total apoio que sempre demonstrou, durante todo o tempo do período de estágio.

Aos colegas de estágio que constituíram uma grande equipe de desenvolvimento e suporte técnico, auxiliando na resolução dos problemas enfrentados.

APRESENTAÇÃO

Como parte das exigências do curso de Ciência da Computação, da Universidade Federal de Campina Grande, para cumprimento da disciplina de Estágio Integrado II, apresenta-se o relatório de estágio, **DESENVOLVIMENTO DO MÓDULO CADASTRO DE ALUNOS DO SISTEMA SISLEGAL E SUPORTE TÉCNICO DE INFORMÁTICA**.

O estágio foi realizado no 2º Centro de Apoio Operacional às Curadorias do MPPB, situado à Rua Promotora Terezinha Lopes de Moura S/N, Liberdade, 58.410-064, Campina Grande Paraíba.

O conteúdo do relatório está distribuído conforme segue:

Capítulo 1 – Introdução: aborda a importância do estágio na formação do aluno, a delimitação e os objetivos do estágio.

Capítulo 2 – Ambiente de estágio: aborda o local de trabalho, identificando a empresa na qual o estágio foi realizado e a infraestrutura do mesmo.

Capítulo 3 – Fundamentação teórica: aborda a fundamentação teórica, trazendo informações sobre as tecnologias e aplicativos utilizados.

Capítulo 4 – Atividades do estágio: aborda detalhadamente as tarefas realizadas durante o tempo de estágio.

Capítulo 5 – Considerações finais.

Referências bibliográficas.

O Apêndice contém o plano de estágio proposto inicialmente.

RESUMO

No presente relatório, documentam-se as atividades realizadas no estágio integrado no período de 08 de março de 2011 a 08 de junho de 2011. Na área de desenvolvimento, foi implementado o módulo *Cadastro de Alunos*, que consiste no registro dos alunos de escolas municipais, estaduais e federais de todos os níveis de ensino, sem reconhecimento de maternidade ou paternidade ou sem registro. Em relação ao suporte técnico, em que as tarefas surgiam com mais frequência, foram realizadas basicamente as seguintes atividades: detecção e resolução de problemas relativos tanto a *hardware* quanto *software*, manutenção da rede interna da instituição e suporte ao usuário no uso dos aplicativos adotados pelo órgão.

SUMÁRIO

LISTA DE SIGLAS E ABREVIATURAS	8
LISTA DE FIGURAS	9
LISTA DE QUADROS.....	10
1 Introdução.....	12
1.1 Delimitação do estágio.....	12
1.2 Objetivos	12
1.2.1 Objetivo geral	12
1.2.2 Objetivos específicos	12
2 Ambiente de estágio.....	15
2.1 Identificação da empresa	15
2.2 Infraestrutura de <i>hardware</i> e <i>software</i>	16
3 Fundamentação teórica.....	18
3.1 Linguagem de programação: <i>Java</i>	18
3.2 <i>Hibernate</i>	19
3.3 <i>JavaServer Faces(JSF)</i>	20
3.4 <i>Spring</i>	22
3.5 Aplicativos utilizados	23
3.5.1 Sistema operacional: <i>Ubuntu 10.02</i> (64 bits)	24
3.5.2 <i>Apache Tomcat 7.0.4</i>	24
3.5.3 IDE: <i>Eclipse Helios</i>	25
3.5.4 <i>PostgreSQL 8.4.0</i>	25
4 Atividades realizadas.....	28
4.1 Descrição geral das atividades	28
4.2 Detalhamento das atividades.....	29
4.2.1 Estudo das tecnologias	29
4.2.2 Familiarização com a estrutura do código.....	30
4.2.3 Desenvolvimento do módulo Cadastro de Alunos	30
4.2.4 Integração com o módulo Cadastro de Escolas	33
4.2.5 Criação de testes	33

4.2.6 Suporte técnico	34
5 Considerações finais	38
REFERÊNCIAS BIBLIOGRÁFICAS	40
Apêndice	41

LISTA DE SIGLAS E ABREVIATURAS

<i>API</i>	<i>Application Programming Interface</i>
<i>BSD</i>	<i>Berkeley Software Distribution</i>
<i>CD</i>	<i>Compact Disc</i>
<i>GUI</i>	<i>Graphical User Interface</i>
<i>HD</i>	<i>Hard Disk</i>
<i>HTML</i>	<i>HyperText Markup Language</i>
<i>HTTP</i>	<i>HypertText Tranfer Protocol</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>IoC</i>	<i>Inversion of Control</i>
<i>JDT</i>	<i>Java Development Tooling</i>
<i>JNDI</i>	<i>Java Naming and Diretory Interface</i>
<i>JSF</i>	<i>Java Server Faces</i>
<i>JSP</i>	<i>Java Server Pages</i>
<i>JVM</i>	<i>Java Virtual Machine</i>
<i>MP</i>	<i>Ministério Público</i>
<i>MPPB</i>	<i>Ministério Público da Paraíba</i>
<i>MVC</i>	<i>Model View Controller</i>
<i>MVCC</i>	<i>MultiVersion Concurrency Control</i>
<i>PC</i>	<i>Personal Computer</i>
<i>POO</i>	<i>Programação Orientada a Objeto</i>
<i>RAM</i>	<i>Random Access Memory</i>
<i>SGBD</i>	<i>Sistema de Gerenciamento de Banco de Dados</i>
<i>SQL</i>	<i>Structure Query Language</i>
<i>TI</i>	<i>Tecnologia da Informação</i>
<i>UI</i>	<i>User Interface</i>
<i>XML</i>	<i>eXtensible Markup Language</i>
<i>XP</i>	<i>eXtreme Programming</i>

LISTA DE FIGURAS

Figura 01	Organograma administrativo do MPPB	15
Figura 02	Arquitetura <i>Hibernate</i>	20
Figura 03	Arquitetura padrão	21
Figura 04	Exemplo injeção de dependência 01	23
Figura 05	Exemplo injeção de dependência 02	23
Figura 06	Diagrama de casos de uso geral	31
Figura 07	Página de <i>login</i>	32
Figura 08	Menu principal	32
Figura 09	<i>Tela inicial de Cadastro de Alunos</i>	32
Figura 10	Página Cadastrar de Aluno	33

LISTA DE QUADROS

Quadro 01	Atividades realizadas	28
-----------	-----------------------	----

CAPÍTULO I

INTRODUÇÃO

1 Introdução

A participação em um estágio mostra-se de fundamental importância, pois é mediante tal elemento acadêmico que se tem a oportunidade de aplicar e evoluir o conhecimento adquirido durante a vida acadêmica e, desta forma, conseguir crescer e conseguir reconhecimento profissional no mercado de trabalho.

1.1 Delimitação do estágio

O estágio foi realizado no setor de informática do Ministério Público da Paraíba (MPPB), na cidade de Campina Grande, tendo início no mês de março de 2011 e conclusão no mês de junho de 2011, período correspondente ao semestre letivo de 2011.1 da Universidade Federal de Campina Grande.

1.2 Objetivos

A seguir, são descritos os objetivos, geral e específicos, identificados no estágio integrado.

1.2.1 Objetivo geral

Desenvolver o sistema SisLegal, que constitui o projeto *Nome Legal* que faz parte do planejamento estratégico do MPPB (ver Apêndice).

1.2.2 Objetivos específicos

O objetivo geral é dividido de seguinte forma:

- o Implementar o módulo *Cadastro de Alunos*, que é responsável pelo cadastro de todos os alunos sem reconhecimento de maternidade ou paternidade ou sem registro;
- o Integrar o módulo *Cadastro de Alunos* com *Cadastro de Escolas*;
- o Criar testes automatizados para o módulo *Cadastro de*

Procedimentos;

- Detectar e resolver todos os problemas relacionados a *hardware* e *software*, e realizar freqüentemente a manutenção da rede interna do órgão, além de fornecer toda a orientação necessária aos servidores da instituição.

CAPÍTULO II

AMBIENTE DE ESTÁGIO

2 Ambiente de estágio

Neste capítulo, é apresentada a empresa na qual o estágio foi vivenciado, bem como a estrutura de *hardware* e *software* disponível para realização das atividades.

2.1 Identificação da empresa

O MPPB atua no campo dos interesses daqueles que por si só são insuficientes para se defender e vela pela aplicação da lei. O órgão é "instituição permanente, essencial à função jurisdicional do Estado, incumbindo-lhe a defesa da ordem jurídica, do regime democrático e dos interesses sociais e individuais indisponíveis" (BRASIL, constituição 1988).

Na Figura 01 mostra-se como é dividida a estrutura administrativa do órgão na área de Tecnologia da Informação (TI), no qual todo departamento de TI está subordinado a Procuradoria Geral de Justiça.

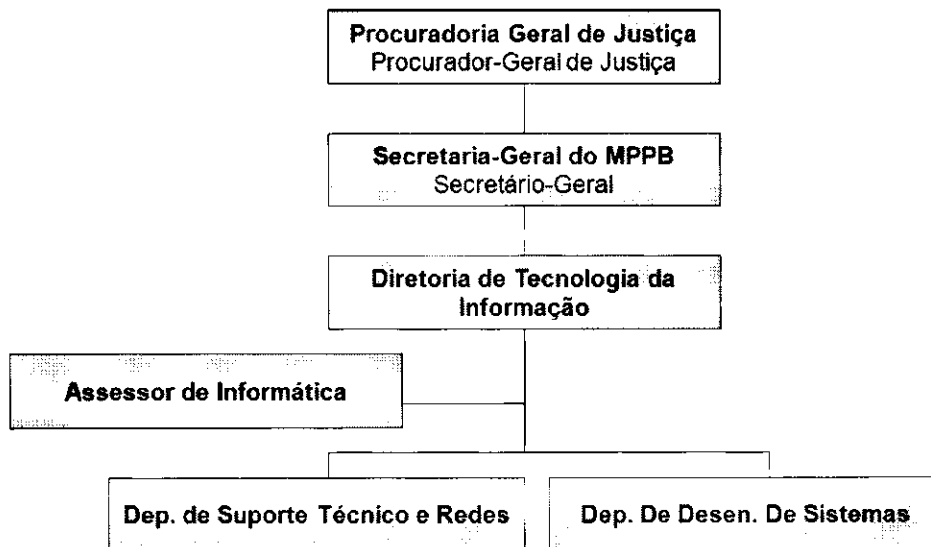


Figura 01 - Organograma Administrativo do MPPB.

Infelizmente, a instituição situada em Campina Grande, não compartilha da mesma estrutura física representada pela Figura 01,

pois tal realidade existe apenas na sede em do órgão em João Pessoa. Na capital, tem-se uma equipe de 16 (dezesesseis) estagiários, além de analistas de sistemas, gerentes de suporte técnico e redes, que constituem equipes específicas para as áreas de desenvolvimento, suporte técnico e redes. Em Campina Grande conta-se com apenas 4 (quatro) estagiários e um técnico em informática, trabalhando simultaneamente em todas as áreas. Portanto, tal situação dificulta consideravelmente as condições de trabalho.

2.2 Infraestrutura de *hardware* e *software*

O ambiente de estágio, no que diz respeito à estrutura física, se mostrou inadequado para as atividades de desenvolvimento. Infelizmente, o *hardware* disponível não apresenta configuração desejável para a prática de programação, faltando principalmente memória principal, fundamental para a execução de aplicativos mais exigentes. Diante dessa situação, foi necessário utilizar uma máquina pessoal (notebook), que contava com a seguinte configuração:

- Processador Core i5 450M 2,4 GHz;
- 4 GB de memória RAM DDR3 1066 MHz;
- 500 GB de espaço em disco;
- Sistemas Operacionais *Windows 7* e *Ubuntu 10.02*, ambos

64 bits.

Uma distribuição Linux e vários aplicativos necessários para o desenvolvimento foram fornecidos pelo chefe do departamento de desenvolvimento para dar início às atividades relacionadas ao sistema SisLegal.

CAPÍTULO III

FUNDAMENTAÇÃO
TEÓRICA

3 Fundamentação teórica

Neste capítulo, descrevem-se as tecnologias e aplicativos utilizados durante o desenvolvimento do sistema em questão.

O SisLegal foi implementado com a linguagem de programação *Java*. A arquitetura do sistema é baseada em *JavaServer Faces(JSF)*, que é um *framework* para desenvolver aplicações web de forma ágil. A persistência de dados foi realizada com o auxílio do *PostgreSQL* e *Hibernate*.

3.1 Linguagem de programação: Java

Tendo sido originalmente concebida para o desenvolvimento de pequenos aplicativos e programas de controle de aparelhos eletrodomésticos e eletroeletrônicos, a linguagem de programação *Java* mostrou-se ideal para ser usada na Internet. O que a torna tão atraente é o fato de programas escritos em *Java* poderem ser executados virtualmente em qualquer plataforma, mas principalmente em *Windows*, *Unix* e *Mac*. Soma-se a isso o fato de programas *Java* poderem ser embutidos em documentos *HTML (HiperText Markup Language)*, podendo assim ser divulgados pela rede. Diferentemente da linguagem *C*, não é apenas o código fonte que pode ser compartilhado pela rede, mas o próprio código executável compilado, chamado *bytecode* (LARA, 2010).

Java foi desenvolvida por um grupo de pesquisadores da *SUN Microsystems* nos anos 90, pouco antes da explosão da Internet. Esta linguagem possui estrutura muito semelhante à da linguagem *C*, da qual descende imediatamente. O *Java* tem em comum com a linguagem *C++* o fato de ser orientada a objetos e mantém com esta um alto grau de semelhança. O paradigma de programação orientada a objetos consiste de um grau a mais na abstração da programação,

em comparação com a programação estruturada, tendo se mostrado extremamente útil na produção de programas cada vez mais sofisticados, em menor tempo e com maior qualidade. A programação orientada a objetos (POO) é hoje universalmente adotada como padrão de mercado.

3.2 *Hibernate*

O *Hibernate* é um *framework* para o mapeamento objeto-relacional escrito na linguagem *Java*. Ele facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (*XML*) para estabelecer esta relação.

Seu objetivo é diminuir a complexidade entre os programas *Java*, baseado no modelo orientado a objeto, que precisam trabalhar com um banco de dados do modelo relacional, presente na maioria dos Sistemas de Gerenciamento de Banco de Dados (SGDBs), em especial, no desenvolvimento de consultas e atualizações dos dados.

O *Hibernate* tem como principal característica a transformação das classes em *Java* para tabelas de dados (e dos tipos de dados *Java* para os da *SQL*), gerando as chamadas *SQL* e liberando o desenvolvedor do trabalho manual da conversão dos dados resultante, o que mantém o programa portátil para quaisquer bancos de dados *SQL*. Tal característica em contrapartida, causa um pequeno aumento no tempo de execução (OLIVEIRA, 2010).

Na Figura 02, é apresentado um exemplo da arquitetura do *Hibernate*. Esta tecnologia usa reflexão para acessar as propriedades de um objeto *Java* persistido. As classes são mapeadas em documentos a partir de um arquivo *XML*. Esses arquivos são compilados no início da aplicação e servem para a criação de esquemas do banco de dados e código-fonte *Java*.

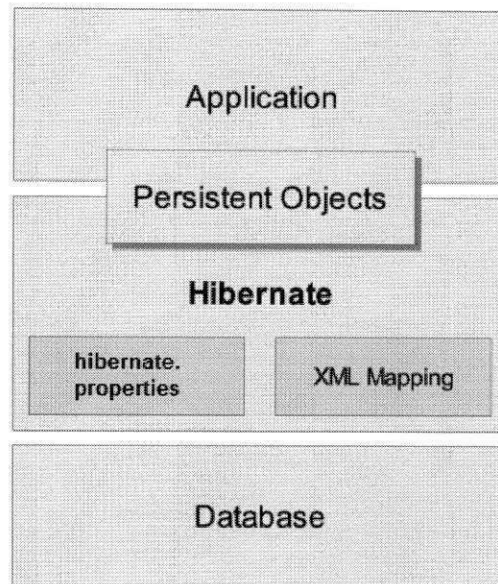


Figura 02- Arquitetura *Hibernate*

Com o uso do *Hibernate*, a produtividade foi melhorada, pois se evitou a escrita de consultas ao banco de dados, o que acabaria se tornando algo muito tedioso. Portanto, com o auxílio desta tecnologia foi possível acelerar o desenvolvimento do SisLegal.

3.3 *JavaServer Faces(JSF)*

JSF é uma tecnologia que incorpora características de um *framework MVC (Model View Controller)* para *Web* e de um modelo de interfaces gráficas baseado em eventos. Por basear-se no padrão de projeto *MVC*, uma de suas melhores vantagens é a clara separação entre a visualização e regras de negócio (modelo).

A idéia do padrão *MVC* é dividir uma aplicação em três camadas: modelo, visualização e controle. O modelo é responsável por representar os objetos de negócio, manter o estado da aplicação e fornecer ao controlador o acesso aos dados. A visualização representa a interface com o usuário, sendo responsável por definir a forma como os dados serão apresentados e encaminhar as ações dos

usuários para o controlador. Por sua vez, a camada de controle é responsável por fazer a ligação entre o modelo e a visualização, além de interpretar as ações do usuário e as traduzir para uma operação sobre o modelo, no qual são realizadas mudanças e, então, gerar uma visualização apropriada.

No *JSF*, o controle é composto por um *servlet* denominado *FacesServlet*, por arquivos de configuração e por um conjunto de manipuladores de ações e observadores de eventos. O *FacesServlet* é responsável por receber requisições da *Web*, redirecioná-las para o modelo e então remeter uma resposta. Os arquivos de configuração são responsáveis por realizar associações e mapeamentos de ações e pela definição de regras de navegação. Os manipuladores de eventos são responsáveis por receber os dados vindos da camada de visualização, acessar o modelo, e então devolver o resultado para o *FacesServlet* (PITANGA, 2010).

Na Figura 03, mostra-se a arquitetura do *JavaServer Faces* baseada no modelo *MVC*, que é fornecida como padrão pelo departamento de desenvolvimento do MPPB.

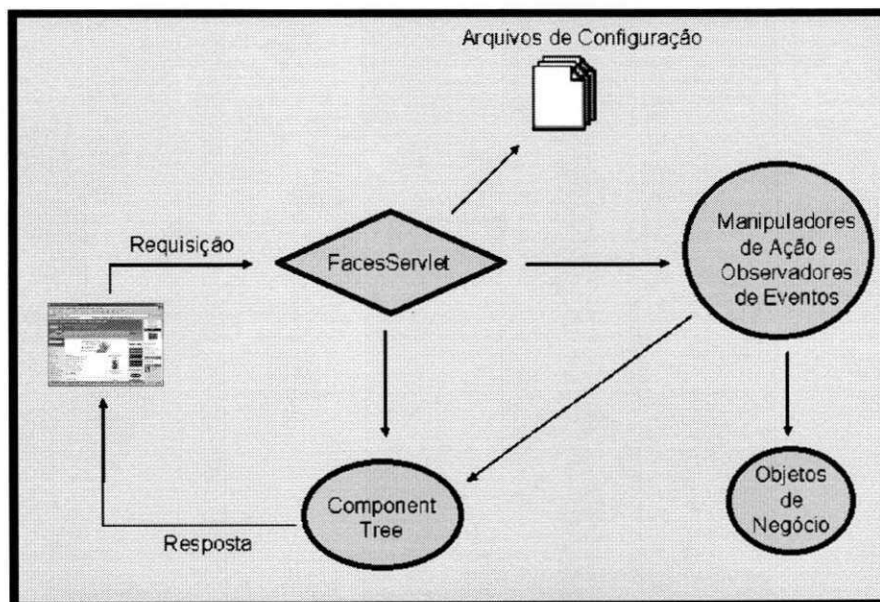


Figura 03 – Arquitetura padrão.

O modelo representa os objetos de negócio e executa uma lógica de negócio ao receber os dados vindos da camada de visualização. Finalmente, a visualização é composta por *component trees* (hierarquia de componentes *UI*), tornando possível unir um componente ao outro para formar interfaces mais complexas.

JSF possui dois principais componentes: *Java API* para a representação de componentes *UI* e o gerenciamento de seus estados, manipulação/observação de eventos, validação de entrada, conversão de dados, internacionalização e acessibilidade; e *taglibs JSP* que expressam a interface *JSF* em uma página *JSP* e que realizam a conexão dos objetos no lado servidor.

3.4 Spring

O *Spring* é um *framework open source* criado, não intrusivo, baseado nos padrões de projeto inversão de controle (*IoC*) e injeção de dependência.

A inversão de controle é o que torna uma simples biblioteca de classes diferente de um *framework*. Uma biblioteca consiste em um conjunto de classes que um usuário instancia e utiliza seus métodos. Após a chamada ao método, o controle do fluxo da aplicação retorna para o usuário. Entretanto, em um *framework* este fluxo é diferente. Para utilizar um *framework*, um código próprio da aplicação deve ser criado e mantido acessível ao mesmo, podendo ser a partir de classes que estendem classes do próprio código.

O padrão injeção de dependência consiste em uma especialização do padrão inversão de controle. Aplicações como *Spring* e *PicoContainer*, denominados de *lightweight containers*, adotam a inversão de controle.

Existem dois tipos de injeção de dependência: *Constructor Injection* e *Setter Injection*. No primeiro tipo, *Constructor Injection*, a dependência é resolvida através de um construtor do objeto a receber o objeto dependente.

```
1 public class MovieLister {
2     public MovieLister(MovieFinder finder) {
3         this.finder = finder;
4     }
5 }
```

Figura 04 – Exemplo injeção de dependência 01.

Na Figura 04, apresenta-se este tipo de injeção de dependência, na qual a classe *MovieLister* define como parâmetro do seu construtor padrão um objeto do tipo *MovieFinder*. Neste contexto, o objeto *Assembler* resolverá a dependência entre os dois objetos passando para *MovieLister* uma implementação concreta de *MovieFinder* através do seu construtor.

```
1 class MovieLister {
2     private MovieFinder finder;
3     public void setFinder(MovieFinder finder) {
4         this.finder = finder;
5     }
6 }
```

Figura 05 - Exemplo injeção de dependência 02.

Na Figura 05, por sua vez, apresenta-se o tipo *Setter Injection*, no qual a dependência entre os objetos é resolvida pelo *Assembler* a partir de um método *Setter* no objeto *MovieFinder* (CARVALHO, 2010).

3.5 Aplicativos utilizados

A utilização das aplicações de *software* descritas neste capítulo foi estabelecida pelo chefe do departamento de desenvolvimento do MPPB.

Os aplicativos utilizados são *open-source* ou *softwares* livres,

pois esse tipo de exigência faz parte de uma das políticas do departamento. Uma pequena parte é composta de aplicações de software *freeware* e *shareware*, por questões de utilidade específica ou de licitações realizadas.

3.5.1 Sistema operacional: *Ubuntu 10.02 (64 bits)*

A proposta do *Ubuntu* é oferecer um sistema operativo que qualquer pessoa possa utilizar sem dificuldades, independentemente de nacionalidade, nível de conhecimento ou limitações físicas. A distribuição deve ser constituída totalmente de *software* gratuito e livre, além de isenta de qualquer taxa. Atualmente, uma organização cuida para que cópias sejam remetidas em *CD* para todo o mundo sem custos.

Focando na qualidade, o *Ubuntu* produz um ambiente de computação rico e robusto que é flexível para usar tanto em casa como em ambientes empresariais. O projeto tem mais tempo para despender em detalhes e é capaz de lançar uma versão que inclui as melhores e mais atuais aplicações gratuitas conhecidas a cada seis meses.

Durante o processo desenvolvimento esse sistema operacional serviu como base para a implementação do projeto, possibilitando o gerenciamento dos recursos de *hardware* e *software* necessários e fornecendo maior segurança contra ataques de vírus.

3.5.2 *Apache Tomcat 7.0.4*

O *Apache Tomcat* é um servidor de aplicações *Java* para Web que implementa as tecnologias *JavaServlets* e *JavaServer Pages*, também podendo comportar-se como um servidor Web (*HTTP*) ou funcionar integrado a um servidor Web dedicado (como o *Apache* ou o *IIS*).

O *Apache Tomcat* é um *software* livre, inteiramente escrito em *Java* e, portanto, para ser executado ele necessita de uma *Java Virtual Machine (JVM)* instalada. A instalação do servidor é simples, porém a configuração requer conhecimento prévio sobre o assunto ou

uma leitura criteriosa nos manuais encontrados tanto na página do oficial do aplicativo como em vários sites na internet.

O Tomcat foi utilizado para hospedar localmente o sistema SisLegal durante o desenvolvimento. A transferência do sistema para o servidor web do MPPB será feita apenas após a finalização do projeto.

3.5.3 IDE: Eclipse Helios

O *Eclipse* é um *framework* para integrar diferentes tipos de aplicações. Uma de suas aplicações é a *JDT (Java Development Tooling)*, a qual já vem com o Eclipse.

Essas aplicações são oferecidas em forma de *plugins* e automaticamente reconhecidas e integradas pela plataforma, tendo seus próprios recursos para gerenciamento de mecanismo, que são geralmente arquivos no *HD*, residentes em seu *workspace*, uma pasta especial localizada no sistema de arquivos. As aplicações instaladas comunicam-se entre si; assim, se uma aplicação altera um recurso qualquer, todas as outras aplicações instaladas serão notificadas sobre essa mudança, garantindo uma consistência e integração em todo o seu ambiente de desenvolvimento. (VALENTIM, 2010).

Toda a codificação do sistema foi feita com auxílio do *Eclipse* e seus *plugins*, como o de Controle de Versão, que possibilitava fazer *commits* e *checkouts* no código do sistema.

3.5.4 PostgreSQL 8.4.0

O banco de dados *PostgreSQL* nasceu na Universidade de *Berkeley*, nos anos 80, como um projeto acadêmico, sendo mantido pela comunidade *Open Source*. A coordenação do desenvolvimento do *PostgreSQL* é executada pelo *Global Development Group*, formado por um amplo grupo de desenvolvedores ao redor do mundo, o que faz o *PostgreSQL* ter uma constante evolução no que se refere à correção de falhas e

implementação de novas funcionalidades (SILVA, 2010).

O *PostgreSQL* é um *SGBDR*, ou seja, um Sistema de Gerência de Bancos de Dados Relacional estendido e livre. "R" porque ele adota uma abordagem relacional. A abordagem em um *SGBD* é caracterizada pela forma como os dados estão organizados nesse *SGBD*. Em um *SGBD* relacional, os dados estão organizados em forma de tabelas (linhas e colunas) e suas relações (chaves estrangeiras).

O *PostgreSQL* foi adotado como *SGBD* padrão da instituição e foi utilizado para a criação das tabelas e colunas do banco de dados do sistema SisLegal.

CAPÍTULO IV

ATIVIDADES REALIZADAS

4 Atividades realizadas

Neste capítulo são descritas todas as atividades desempenhadas durante o período de estágio.

4.1 Descrição geral das atividades

Basicamente, dois tipos de atividades foram realizadas durante o processo: suporte técnico e desenvolvimento.

O suporte técnico compôs-se de tarefas como detecção e resolução de problemas ligados a *hardware* e *software*, reciclagem de cartuchos, suporte ao usuário e manutenção da rede interna.

A etapa de desenvolvimento caracterizada pela construção do sistema SisLegal se dividiu em: estudo das tecnologias, desenvolvimento de uma entidade para familiarização com o sistema, criação do módulo *Cadastro de Alunos*, integração com o módulo *Cadastro de Escolas* e criação de testes para o módulo *Cadastro de Procedimentos*.

No Quadro 01, estão listadas as tarefas executadas, o responsável por cada uma delas, assim como a quantidade de horas necessárias e mês em que foram feitas.

Quadro 01 – Atividades realizadas.

Atividade	Descrição da atividade	Executor	Realizadas	Mês	Horas
1	Estudo das tecnologias utilizadas no desenvolvimento	Equipe	SIM	Abril	50h
2	Desenvolvimento de uma entidade para familiarização com a estrutura do código	Equipe	SIM	Abril	25h
3	Desenvolvimento do módulo <i>Cadastro de Alunos</i>	Individual	SIM	Maio	50h
4	Integração com o módulo <i>Cadastro de Escolas</i>	Individual	SIM	Maio	15h
5	Criação de testes para <i>Cadastro de Procedimentos</i>	Individual	SIM	Junho	25h

6	Suporte técnico	Equipe	SIM	Março a Junho	85h
7	Elaboração do relatório de estágio	Individual	SIM	Maio a Junho	50h
8	Treinamento	Equipe	NÃO	-	-
9	Implantação	Equipe	NÃO	-	-
Total					300h

Houve um atraso no cronograma e duas das atividades descritas no plano de estágio não puderam ser realizadas no tempo previsto devido ao surgimento de dificuldades durante a implementação do projeto e também na prestação de suporte técnico, uma vez que essas atividades foram desempenhadas diariamente em revezamento e como consequência pequenos atrasos foram se acumulando. As tarefas referem-se ao treinamento e implantação do sistema, as quais estão planejadas para o mês de julho de 2011.

4.2 Detalhamento das atividades

A seguir serão descritas com maior nível de detalhadamente as tarefas desempenhadas durante o período de estágio integrado.

4.2.1 Estudo das tecnologias

O departamento de desenvolvimento do MPPB adota um conjunto de tecnologias como padrão para construção de seus projetos na área.

O estudo se deu sobre as tecnologias *Hibernate*, *Spring* e *JSF*, que são as principais técnicas utilizadas na criação dos sistemas para a instituição. Mediante pesquisas na Internet, foi possível conhecer e se aprofundar um pouco mais sobre cada um dos mecanismos utilizados.

A etapa de configuração das tecnologias não foi necessária, pois o departamento exige que todos os sistemas desenvolvidos pela instituição sigam o padrão estabelecido pelo *JSF*, portanto, um

esqueleto do sistema já é previamente fornecido seguindo os conceitos desse *framework*.

4.2.2 Familiarização com a estrutura do código

Para obter familiarização com a estrutura padrão, foi realizada a implementação de uma aplicação *CRUD* (*Create, Retrieve, Update e Delete*). Essa tarefa envolveu a programação de funcionalidades que, embora básicas, forneceram uma excelente visão da estrutura e funcionamento da arquitetura adotada (ver Figura 03) pelo departamento.

4.2.3 Desenvolvimento do módulo Cadastro de Alunos

O projeto SisLegal tem como objetivo a construção de um cadastro para as informações de nomes de escolas, diretores e relação de alunos e mães de alunos com os registros dos alunos e seus pais para o projeto Nome. Outro intuito do projeto é fornecer a disponibilização de um módulo para que o Promotor participante do projeto possa acessar e cadastrar as informações de acompanhamento relativas ao seu município. Por fim, intenta-se a criação de um módulo gerencial, a partir do qual a gestora do projeto poderá acompanhar o andamento dos vários casos cadastrados no sistema e tomar as devidas providências, caso necessário.

A metodologia de desenvolvimento estipulada pelo departamento de desenvolvimento do MPPB foi *Scrum*, porém não foi possível utilizá-lo, pelo fato de que algumas práticas da mesma não puderam ser seguidas, como por exemplo, *Daily Scrum*, que são reuniões diárias que tem como objetivo disseminar o conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia. Portanto, SCRUM foi substituída por *eXtreme Programming (XP)*, mediante decisão da própria equipe e com devida autorização do chefe do departamento, que se mostrou mais adequada para a realidade de desenvolvimento enfrentada, devido a sua maior praticidade de aplicação

(WUESTEFELD, 2010). Basicamente, as práticas utilizadas do XP foram:

- Software em funcionamento mais que documentação abrangente;
- Indivíduos e interações entre eles mais que processos e ferramentas;
- Colaboração com o cliente mais que negociação de contratos;
- Resposta a mudanças mais cumprimento de um plano.

O sistema é composto por cinco módulos a saber: *Cadastro de Alunos*, *Cadastro de Escolas*, *Cadastro de Procedimentos*, *Registro de Notificação de Oitivas* e *Emissão de Relatórios*. Dos módulos citados, apenas *Cadastro de Alunos* será detalhado, pois este foi desenvolvido individualmente.

Na Figura 06, tem-se um diagrama geral de casos de uso do sistema SisLegal. O diagrama engloba os módulos *Cadastro de Escolas*, *Cadastro de Alunos* e *Cadastro de Procedimentos*. É possível perceber que os módulos apresentam funcionalidades básicas de uma aplicação *CRUD*.

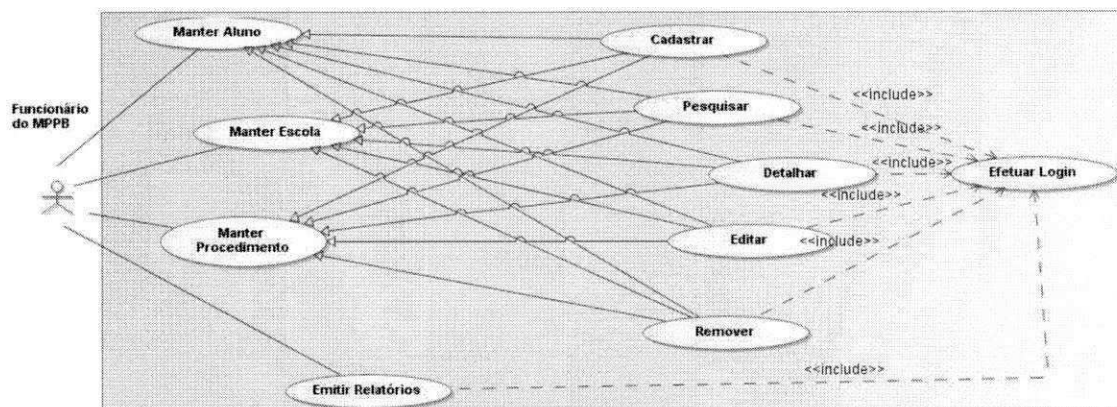


Figura 06 – Diagrama de casos de uso geral.

Na Figura 07, é mostrada a tela de *login* do sistema. Toda a questão de segurança na autenticação de usuários já vem previamente implementada no sistema.

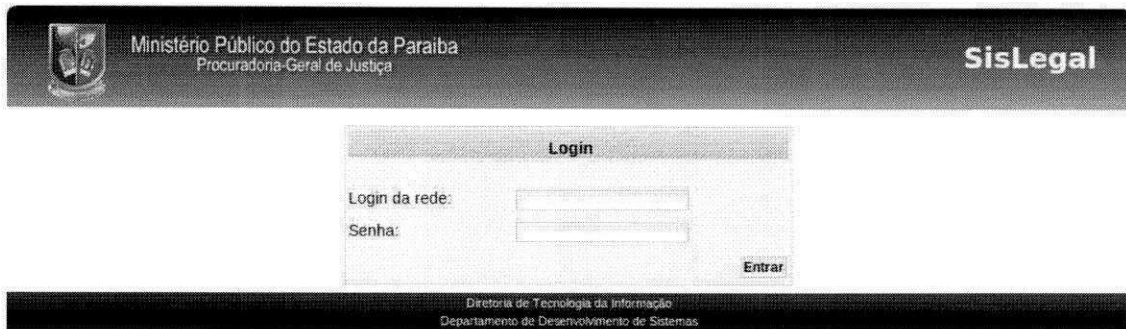


Figura 07 – Página de login.

Na Figura 08, é apresentada a tela inicial do SisLegal. À esquerda, está o menu principal que dá acesso aos módulos desenvolvidos. Observa-se que apenas três dos cinco módulos estão visíveis, pois os outros dois módulos ainda não foram concluídos.

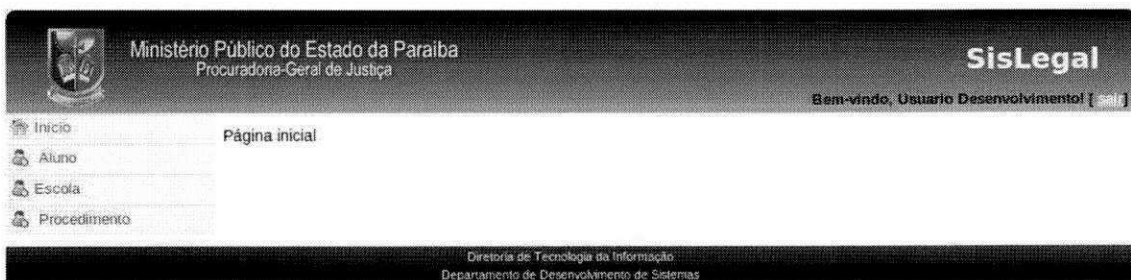


Figura 08 – Menu principal.

Na Figura 09, tem se a tela inicial do módulo Cadastro de Alunos. Nesta pagina, são mostrados os alunos registrados no sistema. Também é possível realizar o cadastro de um novo aluno. A partir da ativação do botão *novo* ou fazer uma busca, digitando o nome de um aluno e ativando o botão *pesquisar*.

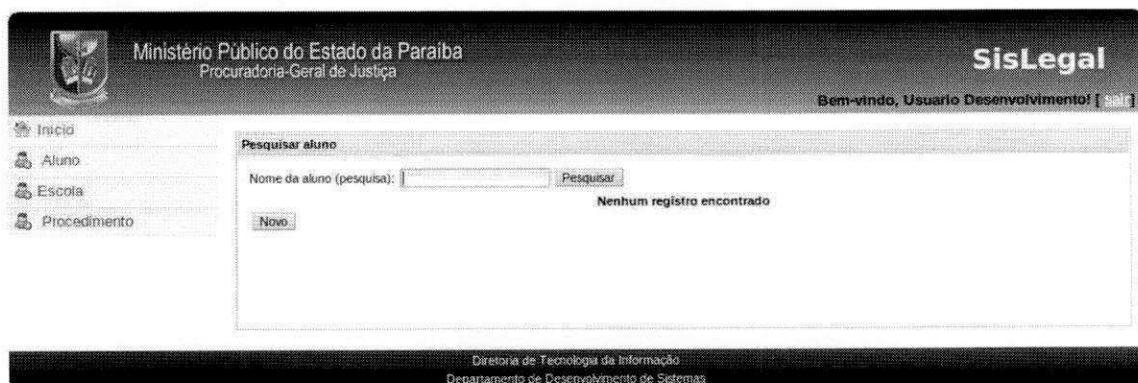


Figura 09 – Tela inicial de Cadastro de Alunos.

Na Figura 10, são mostradas as opções de preenchimento dos dados para cadastramento de um novo aluno.

The screenshot displays a web interface for the 'Ministério Público do Estado da Paraíba' with the 'SisLegal' logo. A navigation menu on the left includes 'Início', 'Aluno', 'Escola', and 'Procedimentos'. The main content area is titled 'Cadastrar aluno' and contains the following form fields: 'Nome' (text input), 'CPF' (text input), 'Escola' (dropdown menu with '-- seletione --' selected), 'Nome da mãe' (text input), 'Nome do pai' (text input), 'Endereço' (text input), 'CEP: (Informe o CEP para preencher o endereço)' (text input with a 'Pesquisar CEP' button), 'Rua' (text input), 'Telefone' (text input), 'Cidade' (text input), 'Estado' (text input), and a 'Salvar' button. The footer of the page reads 'Diretoria de Tecnologia da Informação' and 'Departamento de Desenvolvimento de Sistemas'.

Figura 10 – Página Cadastrar Aluno.

4.2.4 Integração com o módulo Cadastro de Escolas

Após a finalização do módulo *Cadastro de Alunos*, foi necessário realizar uma integração com o módulo *Cadastro de Escolas*, uma vez que, ao cadastrar um novo aluno, é preciso especificar sua escola e, neste caso, é feito um acesso ao banco de dados uma lista com o nome das escolas cadastradas é retornada.

4.2.5 Criação de testes

Foi utilizada o *JUnit* que é um *framework open-source*, com suporte à criação de testes de unidade automatizados na linguagem de programação *Java*. O módulo testado foi *Cadastro de Procedimentos*, que foi desenvolvido por outro membro da equipe.

Espera-se realizar testes funcionais na interface gráfica ao final do projeto a partir da ferramenta *Selenium*, utilizada para testar aplicações *web* pelo *browser* de forma automatizada.

A utilização das ferramentas de teste não foi decidida antecipadamente no plano de estágio, pois o MPPB não adota nenhuma metodologia específica de testes. Portanto, ficou a critério da equipe de desenvolvimento definir como essa atividade seria realizada.

4.2.6 Suporte técnico

As atividades de suporte foram as mais freqüentes durante todo o período de estágio. Em sua maioria, tratavam-se de tarefas não planejadas, ou seja, o trabalho era realizado sob demanda.

Frequentemente, as máquinas do MPPB apresentavam algum tipo de problema no *hardware*, como por exemplo, computadores que não conseguiam inicializar seus sistemas operacionais devido à falha na memória *principal*, que em grande parte das vezes era causado por acúmulo de sujeira dentro do gabinete. Algumas vezes não era possível resolver situações que ou exigiam ferramentas indisponíveis ou conhecimento mais específico de *hardware* como, por exemplo, monitores mais antigos que começavam a apresentar imagens distorcidas e impressoras que não mais imprimiam devido a problemas mecânicos. Nesses casos, os equipamentos eram encaminhados à sede do Ministério Público em João Pessoa para que pudessem receber o tratamento necessário.

Um grande impasse enfrentado pelo suporte técnico era a alta incidência de vírus nos computadores e nos *pen drives*. Quase em 100% dos casos era sempre o mesmo vírus que atacava usando o seguinte artifício: todas as pastas do *pen drive* mudavam para ocultas e atalhos dessas pastas eram criados na raiz do dispositivo, porém tais atalhos não davam acesso à pasta original. Outro detalhe

que piorava a situação era que o vírus alterava alguma propriedade das pastas que impedia que elas pudessem ser definidas como visíveis novamente. Após longa pesquisa na Internet, foi possível descobrir uma técnica para combater tal malefício, que consistia basicamente em restringir as permissões de escrita na raiz do dispositivo, possibilitando tal ação apenas em uma determinada pasta da unidade. Resolvido este problema, ainda restava a questão dos vírus que atingiam os computadores do MPPB, pois o antivírus adotado pelo órgão (*Trend Micro*) era extremamente ineficiente, além do fato de ser um *software shareware*. A solução para tal problema foi solicitar autorização para do antivírus existente por um outro *freeware* (*Avira*), porém muito mais eficiente. Depois disso, o índice de reclamação relacionado a esse assunto reduziu-se quase completamente.

A manutenção da rede interna era uma das tarefas que exigia mais agilidade na resolução, pois interrompiam o trabalho dos servidores. Inserir máquinas e dispositivos na rede, compartilhar pastas e arquivos, resolver problema de restrições de acesso entre as máquinas, fazer manutenção das contas dos usuários na rede. Vários comandos de redes do sistema operacional Windows foram aprendidos como o *ping* que é *um* comando que usa o protocolo ICMP para testar a conectividade entre equipamentos, *net view* que exibe uma lista de recursos que estão sendo compartilhados em um computador, dentre outros comandos.

No Ministério Público, a utilização de dispositivos de impressão é constante e bastante acentuada, então o setor de informática conta com uma estrutura para a realização de recarga de tinta dos cartuchos. Esta atividade é realizada diariamente pelos estagiários de informática do órgão e é fundamental para manter o funcionamento do órgão.

O suporte ao usuário no uso dos aplicativos adotados pelo MPPB foi constante, pois o órgão conta com a presença de diversos servidores, que, na maioria das vezes, por possuírem idade mais avançada, não dispunham de grande habilidade para o manuseio dos programas, como por exemplo, o *BrOffice*, que é o principal aplicativo utilizado no MPPB.

CAPÍTULO V

CONSIDERAÇÕES

FINAIS

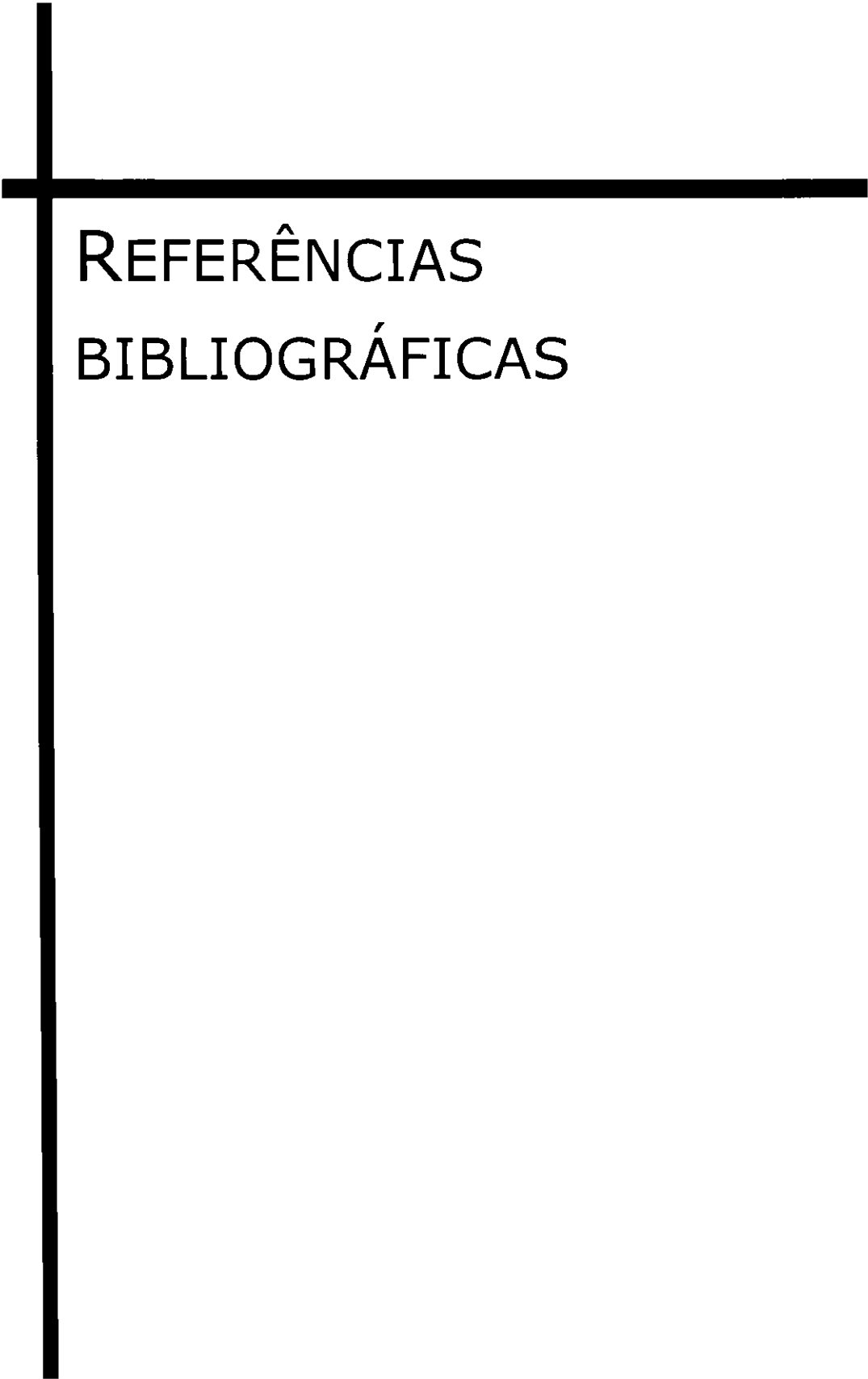
5 Considerações finais

O estágio é uma ferramenta de aprendizagem essencial a qualquer profissional que deseja estar preparado para enfrentar os desafios de uma carreira, principalmente numa área que exige profissionais cada vez mais qualificados. Participar desse processo pela segunda vez foi uma experiência inegavelmente mais enriquecedora. Foi possível aprender com erros cometidos no período anterior e, assim, evitar que pudessem ser repetidos no estágio atual.

A oportunidade de participação da equipe de desenvolvimento do Ministério Público da Paraíba foi muito proveitosa, pois foi possível ter contato com profissionais com bastante experiência na área de TI e também com pessoal de área jurídica.

Um ponto negativo que persistiu da experiência anterior foi a alta carga burocrática que atinge desde a solicitação de peças simples como *mouses* e teclados para reposição, até a requisição de senhas de acesso a administração da rede. Tal situação limitou a execução de algumas tarefas e, por várias, vezes prejudicou o funcionamento da instituição.

O resultado dos dois semestres de estágio no MPPB foi extremamente valioso. No âmbito pessoal, foi muito gratificante ter a chance de trabalhar com os colegas de curso, que se mostraram competentes e dedicados. No âmbito profissional, foi importante ter a oportunidade de aprender e aprimorar novas técnicas e tecnologias de programação e suporte ao usuário.



REFERÊNCIAS
BIBLIOGRÁFICAS

REFERÊNCIAS BIBLIOGRÁFICAS

LARA, Sílvio Garbes. **Tutorial de programação Java**. Disponível em: <<http://sites.google.com/site/silviogarbes/desenvolvimento-web/tutorial-de-programacao-java>>. Acessado em: 10 abr. 2011.

OLIVEIRA, Alison Cardoso. **Tecnologia java para desktop utilizando Hibernate**. Disponível em: <<http://web.unipar.br/~seinpar/artigos/alison-cardoso-de-oliveira.pdf>>. Acessado em: 12 abr. 2011.

CARVALHO, Marlon. **Spring Framework**. Disponível em: <http://imasters.com.br/artigo/4497/spring_framework_introducao>. Acessado em: 12 abr. 2011.

PITANGA, Talita. **JavaServer Faces: A mais nova tecnologia Java para desenvolvimento Web**. [S.L]. Disponível em: <<http://www.guj.com.br/content/articles/jsf/jsf.pdf>>. Acessado em: 11 abr. 2011.

SILVA, Jackson Luiz. **Banco de dados comunitário**. Disponível em: <http://projetos.inf.ufsc.br/arquivos_projetos/projeto_635/TCC-Jackson-Final-BDC.pdf>. Acessado em: 15 abr. 2011.

VALENTIM, Ricardo A. M. **Eclipse: uma IDE Para Java**. Disponível em: <http://www.cefetrn.br/~valentim/disciplinas/poo/material/Eclipse_Tutorial.pdf>. Acessado em: 21 abr. 2011.

WUESTEFELD, Klaus. **Xispê: Extreme Programming**. Disponível em: <<http://www.xispe.com.br/index.html>>. Acesso em: 19 abr.2011

BRASIL. Constituição (1988). Constituição da República Federativa do Brasil. Brasília, DF, Senado, 1998.



Apêndice



UFPG - UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CEEI - CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DSC - DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO

Plano de Estágio Integrado

Ministério Público

Desenvolvimento do Sistema SisLegal e Suporte Técnico em Informática

Antonio Alexandre Moura Costa

Mat.: 20921282

Prof. Tiago Lima Massoni

Orientador Acadêmico

Campina Grande,

Março 2011

Informações Pessoais

Nome: Antonio Alexandre Moura Costa

Curso: Ciência da Computação

Matricula: 20921282

CPF: 048.753.964.81

RG: 2923843 SSP/PB

Endereço: Rua Malaquias Souza do Ó, nº 34, Mirante

Campina Grande – PB

CEP: 58407563

Email: alexandrecostapb@gmail.com

Ambiente de Estágio

O estágio será realizado no 2º Centro de Apoio Operacional às Curadorias, situada à Rua Promotora Terezinha Lopes de Moura S/N, Liberdade, 58.410-064, Campina Grande, Paraíba.

O trabalho será realizado por uma equipe de 5 (cinco) pessoas, sendo quatro desenvolvedores e um gerente, na qual a atuação se dará como desenvolvedor e testador.

Supervisão

Supervisor Acadêmico

Nome: Tiago Lima Massoni

Endereço: Departamento de Sistemas e Computação

Universidade Federal de Campina Grande

Avenida Aprígio Veloso, 882 – CEP: 58.109-970

Bodocongó, Campina Grande, PB – Brasil.

Email: massoni@dsc.ufcg.edu.br

Supervisor Técnico

Nome: Uirá Vasconcelos Alencar de Assis

(Chefe de Suporte Técnico do Ministério Público da Paraíba)

Endereço: Rua Promotora Terezinha Lopes de Moura, S/N

CEP: 58.410-064

Liberdade, Campina Grande, PB – Brasil.

Email: uira@mp.pb.gov.br

Resumo do Problema do Estágio

A Paraíba conta hoje com um número muito alto de crianças sem o reconhecimento de paternidade em seu registro de nascimento. Esse tipo de situação pode causar diversos tipos de deficiências psicológicas interferindo na socialização e na vida dessas crianças. Além de ser importante do ponto de vista psicológico, a regularização do registro, também é um direito da criança.

Com a implantação deste projeto o Ministério Público da Paraíba visa reduzir o número de crianças sem o reconhecimento de paternidade em seu registro de nascimento, por meio do levantamento, em cada comarca, das pessoas que têm seu registro civil incompleto, notificando os respectivos responsáveis (geralmente as mães) para esclarecimento acerca das medidas que poderão ser utilizadas para a regularização da filiação.

Objetivo do Estágio

O projeto SisLegal visa à criação de um cadastro para as informações de nomes de escolas, diretores e relação de alunos e mães de alunos com os registros dos alunos e seus pais para o projeto Nome Legal (Planejamento estratégico do Ministério Público da Paraíba 2010-2016). Como também a disponibilização de um módulo desse sistema para que o Promotor participante do projeto possa acessar e cadastrar as informações de acompanhamento relativas ao seu município. E finalmente a criação de um módulo gerencial onde a gestora do projeto poderá acompanhar o andamento dos vários casos cadastrados no sistema e tomar as devidas providências, caso necessário.

Criação de relatórios gerenciais informando da atuação dos vários membros participantes do projeto e dos registros dos casos terminados com sucesso (quando o pai aceita registrar o filho) ou pendentes.

Este projeto está alinhado ao planejamento estratégico do Ministério Público da Paraíba através do objetivo: Incrementar ações de atenção à família e aos incapazes.

I. Módulos do sistema SisLegal:

1. Cadastro das escolas existentes no município (particulares, municipais, estaduais, federais), de todos os níveis de ensino;
2. Cadastro da relação dos alunos sem maternidade ou sem paternidade reconhecida ou sem registro;
3. Cadastro da relação de procedimentos por escola originados a partir dos cadastros realizados o item anterior;
4. Registro das notificações expedidas para marcação das oitivas (audição, ato de ouvir as testemunhas ou as partes de um processo judicial);
5. Módulo gerencial com a emissão de relatórios de casos resolvidos (paternidade aceita), casos não resolvidos e seus motivos, relação de escolas e relação de alunos;

Metodologia

O Departamento de Desenvolvimento de Sistemas do MPPB, subordinado a Diretoria de Tecnologia da Informação, está utilizando para o desenvolvimento de seus sistemas uma metodologia ágil com práticas de SCRUM[1] e Kanban para acompanhamento e transparência das tarefas (estórias ou requisitos do sistema) atribuídas ao desenvolvedor e práticas de RUP[2] para a coleta e documentação dos sistemas.

Atividades Planejadas

- I. Levantamento dos requisitos do sistema com a gestora do projeto;
- II. Análise e arquitetura do sistema;
- III. Implementação dos módulos descritos no item 2.
- IV. Testes
- V. Implantação
- VI. Manutenção

Participarei efetivamente do desenvolvimento do módulo 2 e atuarei na elaboração de testes do módulo 3.

Cronograma de Atividades

Atividades planejadas	1- 2/Abril	3- 4/Abril	1- 2/Maio	3- 4/Maio	1- 2/Junho	3- 4/Junho
Levantamento dos requisitos detalhados do sistema com a gestora do projeto	X					
Estudo das tecnologias utilizadas no departamento de Desenvolvimento de Sistemas - DITEC	X					
Análise da arquitetura do sistema		X				
Módulo Cadastro de Alunos			X	X		
Testes de Cadastro de Escolas				X	X	X
Treinamento						X
Implantação						X
Suporte Técnico	X	X	X	X	X	X
Elaboração do Relatório	X	X	X	X	X	X
Defesa do Estágio						X

Resultados Esperados

Após a implantação, treinamento e correta alimentação do sistema por parte dos servidores e promotores envolvidos no projeto esperamos fornecer condições para que a promotora responsável, Dra. Renata Carvalho, possa apresentar relatórios gerenciais de acompanhamento dos vários casos de registro de paternidade cadastrados no sistema.

Bibliografia

[1] SCHWABER, K., Agile Project Management with Scrum ed. Microsoft Press, 2004. ISBN 978-0-735-61993-7.

[2] KRUCHTEN, P., Introdução ao RUP: Rational Unified Process ed. Ciência Moderna, 2003. ISBN 8-57393-275-9

Aprovação

Uirá Vasconcelos Alencar de Assis
Supervisor Técnico

Tiago Lima Massoni
Orientador Acadêmico

Joseana Macêdo Fachine
Coordenadora da disciplina Estágio Integrado