



Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Sistemas e Computação – DSC

Relatório de Estágio

Ministério Público da Paraíba

**Desenvolvimento do Módulo de Cadastro de Escolas do
Sistema SisLegal e Suporte Técnico em Informática**

Aluno: Lenin da Nóbrega Medeiros (21011093)
Supervisor Acadêmico: José Eustáquio Rangel de Queiroz
Supervisor Técnico: Uirá Vasconcelos Alencar de Assis
Coordenadora de Estágio: Joseana Macêdo Fachine
Semestre Letivo: 2011.1
Disciplina: Estágio Integrado
Créditos e Carga Horária: 10 créditos e 300 horas

Campina Grande, julho de 2011.

Desenvolvimento do Sistema SisLegal e Suporte Técnico em Informática

Aprovado em _____

Banca Examinadora

Prof. José Eustáquio Rangel de Queiroz, D.Sc.
Orientador Acadêmico

Prof.^a Joseana Macêdo Fechine, D.Sc.
Membro da Banca

Prof. Leandro Balby Marinho, D.Sc.
Membro da Banca



Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

Agradecimentos

A ajuda recebida foi imprescindível para a realização deste estágio. Portanto, gostaria de externar sinceramente meus agradecimentos a todos aqueles que contribuíram para a conclusão deste trabalho.

Ao professor e orientador Prof. Eustáquio Rangel, que esteve sempre disponível e cujos ensinamentos foram cruciais.

Agradeço à professora Prof.^a: Joseana Fechine sua cativante preocupação e, também, todas as contribuições dadas.

A Jefferson Ferreira, que, no que diz respeito ao desenvolvimento do projeto, sempre esteve disposto a me ajudar.

Aos meus colegas de estágio Antônio Alexandre Costa, Felipe Barbosa e Luís Paulo Lima, que tornaram o ambiente de trabalho bem mais agradável e um tanto quanto divertido.

Apresentação

O exercício da prática profissional oferece uma participação efetiva no contexto do planejamento e execução de projeto, além de enriquecer o cenário da formação acadêmica. É justamente no contexto citado que se inclui a disciplina *Estágio Integrado I*, oferecida pelo curso de Ciência da Computação da Universidade Federal de Campina Grande.

Como parte das exigências para o cumprimento da disciplina citada anteriormente, apresenta-se este relatório de estágio. O mesmo refere-se ao estágio de Lenin Medeiros no MPPB (Ministério Público da Paraíba). As atividades começaram em março de 2011 e se encerraram em junho do mesmo ano.

O conteúdo do presente documento está distribuído como segue:

1. Introdução;
2. Ambiente de Estágio;
3. Fundamentação Teórica;
4. Atividades Realizadas;
5. Considerações Finais.

Resumo

O estágio teve dois objetivos: suporte técnico em informática ao prédio do Ministério Público da Paraíba em Campina Grande e o desenvolvimento de um projeto proposto pelo Departamento de Desenvolvimento de Sistemas do MPPB. O suporte técnico se deu mediante à manutenção dos recursos tecnológicos disponíveis e da rede de computadores do prédio onde foi realizado o estágio, além do auxílio no uso de aplicativos e dispositivos. No que diz respeito ao desenvolvimento, foi implementado o módulo de cadastro de escolas do sistema SisLegal, além de terem sido realizados testes no módulo de cadastro de alunos do mesmo sistema.

Sumário

LISTA DE SIGLAS E ABREVIATURAS	7
LISTA DE QUADROS E TABELAS	8
LISTA DE FIGURAS	9
1. INTRODUÇÃO.....	10
1.1 DELIMITAÇÃO DO ESTÁGIO	10
1.2 OBJETIVOS	10
1.3 ORGANIZAÇÃO DO RELATÓRIO.....	12
2. AMBIENTE DE ESTÁGIO.....	12
2.1 LOCALIZAÇÃO GEOGRÁFICA	13
2.2 IDENTIFICAÇÃO DA EMPRESA	13
2.3 ATIVIDADES DESENVOLVIDAS.....	14
2.4 INFRAESTRUTURA DE <i>HARDWARE</i> E <i>SOFTWARE</i>	15
2.5 DADOS DO ORIENTADOR E SUPERVISOR.....	15
3. FUNDAMENTAÇÃO TEÓRICA	15
3.1 PROCESSO DE DESENVOLVIMENTO	16
3.1.1 <i>Engenharia de Requisitos</i>	16
3.1.2 <i>Planejamento</i>	17
3.1.3 <i>Implementação</i>	18
3.1.4 <i>Validação</i>	18
3.2 TECNOLOGIAS UTILIZADAS	18
3.2.1 <i>Hibernate</i>	19
3.2.2 <i>Spring</i>	21
3.2.3 <i>JSF</i>	23
3.2.4 <i>MVC</i>	24
4. ATIVIDADES REALIZADAS	24
4.1 PLANEJAMENTO.....	26
4.2 IMPLEMENTAÇÃO.....	27
4.3 VALIDAÇÃO E INSPEÇÃO	31
4.4 INTEGRAÇÃO	31
4.5 SUPORTE TÉCNICO EM INFORMÁTICA	31
5. CONSIDERAÇÕES FINAIS.....	34
REFERÊNCIAS BIBLIOGRÁFICAS	36
APÊNDICE.....	37

Lista de Siglas e Abreviaturas

MPPB	Ministério Público da Paraíba
MVC	<i>Model-View-Controller</i>
JSF	<i>Java Server Faces</i>
JSP	<i>Java Server Pages</i>
Spring	<i>Framework open source</i> para injeção de dependência
Hibernate	<i>Framework</i> para mapeamento objeto-relacional
EJB	<i>Enterprise JavaBeans</i>
JNDI	<i>Java Naming and Directory Interface</i>
FreeMaker	Ferramenta para geração de texto baseado em <i>templates</i>
PostGres	Banco de dados <i>open source</i>
SQL	<i>Structured Query Language</i>
HQL	<i>Hibernate Query Language</i>

Lista de Quadros

Quadro 1 – Descrição das atividades e das horas gastas.....	25
Quadro 2 – Novo cronograma de atividades.....	27

Lista de Figuras

Figura 1 - Prédio do MPPB em Campina Grande.....	13
Figura 2 - Organograma simplificado do MPPB	14
Figura 3 - Módulos do Spring [JOHNSON, 2010].....	21
Figura 4 - Diagrama geral de casos de uso do SisLegal.....	26
Figura 5 - Trecho de código da classe <i>Escola</i>	28
Figura 6 - Tela de <i>login</i> do SisLegal.....	29
Figura 7 - Página inicial.....	29
Figura 8 - Página principal da seção <i>Escola</i>	30
Figura 9 - Página de cadastro de escolas.....	30
Figura 10 - Exibição do campo a ser preenchido referente ao nome de uma escola e da <i>combo box</i> referente aos possíveis níveis de ensino.....	31
Figura 11 - Verificando tempo de resposta do servidor.....	33
Figura 12 - Visualização de rede.....	33

1. INTRODUÇÃO

O estágio integrado é uma atividade acadêmica de grande importância, pois oferece uma oportunidade de se aplicar os conhecimentos adquiridos durante a formação acadêmica. Além disso, há também a possibilidade de se aprender conceitos novos que não foram citados na academia.

1.1 Delimitação do Estágio

O estágio foi realizado no prédio do MPPB em Campina Grande durante o período letivo 2011.1 da UFCG. A carga horária foi de 25 horas semanais. Portanto, as 300 horas requeridas por esta instituição para que a disciplina *Estágio Integrado I* fosse cursada foram alcançadas.

1.2 Objetivos

Neste tópico, estão descritos todos os objetivos do estágio. Os mesmos são obrigações impostas ao estagiário para que o trabalho realizado seja reconhecido por esta instituição.

1.2.1 Geral

O objetivo geral e principal do estágio é desenvolver um módulo do sistema SisLegal.

Existe um objetivo secundário, que é oferecer suporte técnico em informática para todo o prédio do Ministério Público da Paraíba em Campina Grande.

1.2.2 Específicos

O sistema SisLegal tem três objetivos, a saber:

1. Criação de um cadastro para as informações de nomes de escolas, diretores e relação de alunos e respectivas mães com os

registros dos alunos e respectivos pais para o projeto *Nome Legal*¹;

2. 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;

3. Criação de um módulo gerencial no qual a pessoa responsável pela gestão do projeto poderá acompanhar o andamento dos vários casos cadastrados no sistema e tomar as devidas providências, caso necessário.

Sendo assim, o sistema supracitado está dividido em cinco módulos, assim discriminados:

1. Cadastro das escolas existentes em um dado município (particulares, municipais, estaduais e federais), de todos os níveis de ensino;

2. Cadastro da relação de alunos sem maternidade ou paternidade reconhecida ou sem registro;

3. Cadastro da relação de procedimentos por escola originados a partir dos cadastros do item anterior;

4. Registro das notificações expedidas para marcação das oitivas (audição; ato de ouvir 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 respectivos motivos, além das relações de escolas e alunos.

O trabalho realizado teve como foco o desenvolvimento e os testes dos 1º e 2º módulos, respectivamente.

Outra atribuição do estágio foi a de suporte técnico em informática. Foi feita a manutenção dos recursos tecnológicos (computadores e dispositivos) e da rede em que estão inseridas as

¹ Informações mais detalhadas sobre o projeto *Nome Legal* se encontram no apêndice deste documento.

máquinas do prédio do MPPB em Campina Grande, além do auxílio aos usuários em relação aos aplicativos de *software* instalados nos computadores.

1.3 Organização do Relatório

O que se segue é uma descrição de como foi dividido conteúdo deste documento.

Na Seção 1, aborda-se: a importância do estágio, a sua delimitação e os objetivos delineados para tal.

A Seção 2 contém uma descrição do ambiente de estágio, identificando a instituição, as atividades desenvolvidas e a infraestrutura disponibilizada.

A Seção 3 trata dos fundamentos teóricos. Nela se encontram as informações necessárias sobre as tecnologias que foram utilizadas no desenvolvimento dos módulos do sistema SisLegal.

Estão descritas, na Seção 4, as atividades realizadas ao longo do estágio, tanto as de desenvolvimento quanto as de suporte técnico em informática.

A Seção 5 destina-se às considerações finais.

Finalmente, o Apêndice contém o plano de estágio proposto no início das atividades da disciplina.

2. AMBIENTE DE ESTÁGIO

Nesta seção está descrito, além da organização da empresa na qual o estágio foi realizado, o espaço físico onde o mesmo aconteceu. A forma como estágio foi planejado pelos supervisores do estagiário também se encontra nesta parte do relatório.

2.1 Localização Geográfica

O estágio aconteceu no prédio do Ministério Público da Paraíba em Campina Grande, que está inserido num complexo conhecido como Pólo Jurídico. Os estagiários de informática do MPPB de Campina Grande (são quatro estagiários) trabalham numa sala que está dentro do *2º Centro de Apoio Operacional às Curadorias*. O endereço é: Rua Promotora Terezinha Lopes de Moura S/N, Liberdade, 58.410-064, Campina Grande, Paraíba.

Na Figura 1 é apresentada uma foto de satélite do prédio em questão.

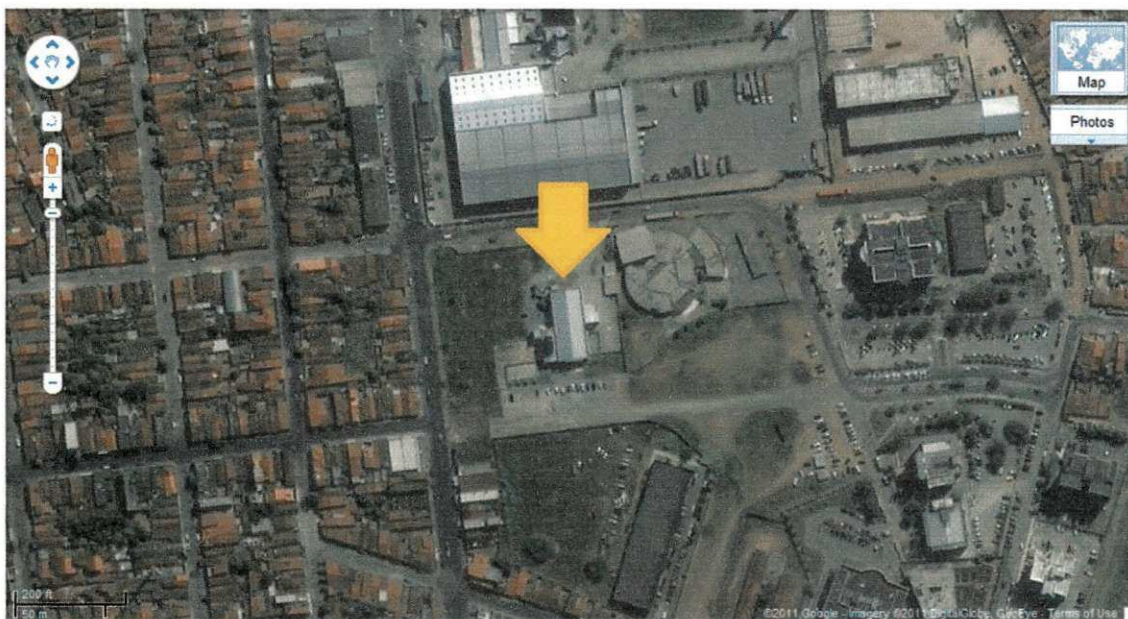


Figura 1 - Prédio do MPPB em Campina Grande, destacado pela seta amarela.

2.2 Identificação da Empresa

O MPPB é um órgão da esfera jurídica e tem como eixo central o desenvolvimento de políticas institucionais que atendam, de maneira efetiva, os anseios da sociedade, definindo para tanto: estratégias, programas e ações que permitam uma gestão prioritária na área social.

O órgão anteriormente mencionado está dividido nas seguintes

diretorias: Financeira, Planejamento, Tecnologia da Informação, Administrativa e Corregedoria-Geral. No organograma da Figura 2, exibe-se, de forma simplificada, esta estrutura. O estágio transcorreu junto à Diretoria de Tecnologia da Informação, cuja sede se encontra na cidade de João Pessoa.

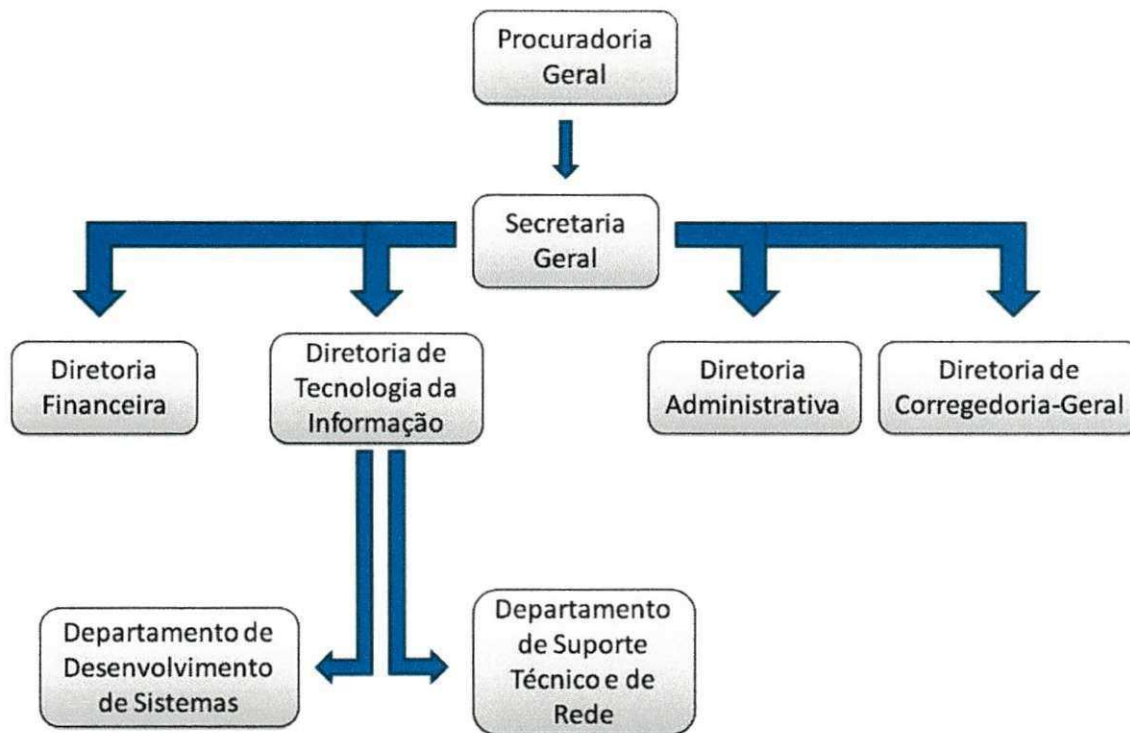


Figura 2 - Organograma simplificado do MPPB.

2.3 Atividades Desenvolvidas

As atividades foram realizadas em parceria com Antônio Alexandre Costa, Felipe Barbosa e Luís Paulo Lima. As atividades relativas ao desenvolvimento do sistema SisLegal foram propostas pelo MPPB na seguinte sequência:

1. Levantamento dos requisitos do sistema com o responsável pela gestão do projeto;
2. Análise e arquitetura do projeto;
3. Implementação dos módulos do sistema;
4. Testes;

5. Implantação; e
6. Manutenção.

Vale salientar que a atividade de suporte técnico também foi definida como dever dos estagiários do MPPB.

2.4 Infraestrutura de *Hardware* e *Software*

A instituição cedeu uma sala com três computadores, cada um dos quais contendo um processador Pentium Dual Core 2.5 GHz, com 250 GB de espaço em disco e 2 GB de RAM, além de uma distribuição Linux com todos as aplicações de *software* necessárias à realização das atividades dos estagiários.

2.5 Dados do Orientador e Supervisor

Orientador Acadêmico: *José Eustáquio Rangel de Queiroz*

Endereço: Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Avenida Aprígio Veloso, 882 – CEP: 58.109-970
Bodocongó, Campina Grande, PB – Brasil.

Supervisor Técnico: *Uirá Vasconcelos Alencar de Assis*

Endereço: Rua Promotora Terezinha Lopes de Moura, S/N
CEP: 58.410-064
Liberdade, Campina Grande, PB – Brasil.

3. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, encontram-se os conceitos que, por determinação dos supervisores do estágio, tiveram que ser estudados antes do início do desenvolvimento do sistema SisLegal.

Inicialmente, houve um estudo sobre o processo de desenvolvimento adotado pelo MPPB. Em seguida, foram estudadas as tecnologias que seriam utilizadas no projeto.

3.1 Processo de Desenvolvimento

O Departamento de Desenvolvimento de Sistemas do MPPB adota um processo de desenvolvimento de *software* baseado no *Scrum*. Para segui-lo corretamente, é preciso que se realizem reuniões diárias para acompanhamento de projeto. Além disto, há geração de artefatos como, por exemplo, o *BurnDown Chart*: um gráfico que serve de monitoramento com o intuito de se estabelecer a quantidade de trabalho que falta para terminar um certo *sprint* (nome dado às iterações do processo em questão).

Como nem todos os desenvolvedores do SisLegal trabalham juntos (uns em João Pessoa e outros em Campina Grande), seria impossível ter reuniões diárias, o que impossibilitou o uso do processo supramencionado. Sendo assim, optou-se pelo processo *Extreme Programming*, também conhecido como *XP*.

3.1.1 Engenharia de Requisitos

As atividades descritas a seguir tiveram como objetivo principal a realização de uma documentação relativa aos requisitos do sistema, valendo salientar que elas se deram de forma simultânea e incremental.

É importante realizar um estudo de viabilidade, para que a equipe desenvolvedora conheça melhor o problema a ser resolvido. Com isso é possível avaliar se o sistema é viável do ponto de vista tecnológico e organizacional. Deve-se tentar obter confirmação de que o sistema contribui para os objetivos da empresa. Também é preciso saber se o sistema pode ser implementado baseado nas restrições, que podem ser políticas ou econômicas. Este estudo pode ser feito mediante reuniões e entrevistas com as partes interessadas.

Na etapa em que ocorre levantamento e análise dos requisitos, a equipe consegue identificar as estratégias de desenvolvimento do produto. Algumas técnicas são usadas com profissionais dos mais variados níveis hierárquicos do órgão, com o intuito de conhecer as necessidades de cada um. As mais utilizadas para descobrir os requisitos de um sistema são entrevistas e utilização de questionários.

Existe uma fase na qual os requisitos são especificados. Ela consiste em organizar e converter os requisitos acumulados até então para um documento formal contendo detalhes e especificações dos mesmos. É bem interessante quando se usa diagramas de casos de uso, pois estes facilitam o entendimento tanto do cliente quanto de quem estiver desenvolvendo o projeto. Tais diagramas também são úteis quando se há dúvidas quanto às funcionalidades implementadas.

É claro que, em algum momento, deve haver a validação dos requisitos que foram levantados e especificados. Os mesmos devem estar estruturados corretamente e sem ambiguidade. Além disso, deve-se ter a garantia de que os mesmos refletem exatamente o sistema que será implementado. Isso tudo aumenta a probabilidade de sucesso do projeto, certificando que está sendo desenvolvida a solução mais correta.

3.1.2 Planejamento

Esta etapa tem como objetivo realizar um planejamento do que será feito (a curto e longo prazo). Dessa forma é possível acompanhar a evolução do projeto e saber se as atividades definidas estão sendo cumpridas de acordo com o que era esperado a princípio ou não. Tarefas de avaliação de riscos, estabelecimento de prazos e definição de cronograma devem estar envolvidas.

Com os requisitos em mãos é possível planejar as tarefas que serão desenvolvidas. Na definição do cronograma de atividades, deve-se levar sempre em consideração a disponibilidade dos recursos no momento do desenvolvimento, as dependências entre tarefas e a possibilidade de surgimento de riscos desconhecidos até então. Esses últimos podem acarretar no atraso da entrega do produto.

3.1.3 Implementação

Na fase de implementação, ocorre a codificação do sistema. Nessa etapa os desenvolvedores escrevem os códigos e os armazenam num repositório, o qual deve estar sempre atualizado para que a integração seja feita constantemente. É importante seguir padrões de design que possam ser aplicados as necessidades do sistema, facilitando futuras manutenções, principalmente se forem feito por outra equipe.

É preciso que haja uma inspeção nos códigos. Sugere-se que um desenvolvedor inspecione códigos implementados por outros. Deve-se analisá-los, visando encontrar defeitos, possíveis inconsistências e sugerindo mudanças para a melhoria da qualidade do projeto.

3.1.4 Validação

A validação serve para determinar se o sistema atende as necessidades e requisitos do usuário. É ideal que seja feita antes da entrega final do produto. Ela pode ser feita com a utilização de testes funcionais, em que um usuário começa a usar o sistema desenvolvido, facilitando a detecção de possíveis problemas (erros e inconsistências).

3.2 Tecnologias Utilizadas

A arquitetura padrão sugerida para uso no desenvolvimento de sistemas dentro do MPPB é a união de três frameworks. São eles:

1. *Hibernate* (camada de persistência);
2. *Spring* (injeção de dependências);
3. *JSF* (camada de apresentação).

3.2.1 Hibernate

Hibernate é um framework de mapeamento objeto/relacional para *Java* que possibilita desenvolver classes persistentes. É possível configurar o *Hibernate* de três maneiras:

1. Instanciando um objeto *Configuration* e inserindo as suas propriedades programaticamente;
2. Usando um arquivo *.properties* com as suas configurações e indicando os arquivos de mapeamento programaticamente;
3. Utilizando o arquivo *hibernate.cfg.xml*, que contém as propriedades de inicialização e os caminhos dos arquivos de mapeamento.

O *Hibernate* possui cinco interfaces fundamentais que são utilizadas em praticamente todos os aplicativos do mesmo, possibilitando o armazenamento e recuperação de objetos persistentes e o controle de transações.

O que se segue é uma descrição das tais interfaces do *framework Hibernate*.

I. Interface *Session*

É a principal interface usada por aplicativos do *Hibernate*. Criar e destruir uma instância da sessão não são atividades muito custosas em termos de eficiência. Esta é uma importante característica, pois a aplicação terá de criar e destruir as sessões todo o tempo, talvez em cada solicitação feita ao banco de dados [BAUER, 2005].

Uma sessão é como um *cache*, ou uma coleção de objetos carregados relativa à uma única unidade de trabalho. O *Hibernate*

pode detectar mudanças nos objetos nesta unidade de trabalho. É comum chamar uma sessão de gerenciador de persistência, porque é também uma interface para efetuar as operações relacionadas com a persistência, como armazenar e recuperar objetos [BAUER, 2005].

II. Interface *SessionFactory*

Classe responsável por abrir as instâncias de trabalho do *Hibernate*, ou seja, as *sessions*. Existe normalmente uma única *SessionFactory* para toda a aplicação que é criada durante a inicialização do aplicativo.

O *cache* da *SessionFactory* gera instruções *SQL* e metadados de mapeamento que o *Hibernate* usa em tempo de execução. Caso a classe especifique que um *cache* de segundo nível é desejável, então os dados lidos em uma unidade de trabalho e que poderão ser reutilizados futuramente são mantidos nesta *cache* [BAUER, 2005].

III. Interface *Transaction*

É uma interface opcional. Permite que o aplicativo controle os limites de transações utilizando uma *API* consistente, tornando assim as aplicações portáteis entre diferentes tipos de ambientes de execução [BAUER, 2005].

IV. Interface *Configuration*

É utilizada para configurar e inicializar o *Hibernate* especificando a localização dos documentos de mapeamento e as propriedades específicas.

V. Interfaces *Query* e *Criteria*

A interface *Query* permite executar consultas no Banco de Dados. As consultas são executadas em *HQL* ou no *SQL* nativo do seu banco de dados. Uma instância de *Query* é usada para vincular os parâmetros, o limite da quantidade de resultados retornados e para executar a consulta [BAUER, 2005].

A interface *Criteria* é muito semelhante, permitindo criar e executar consulta orientada a objetos.

3.2.2 Spring

O *Spring* é um *framework* utilizado em aplicações Java que permite configurar os componentes de sua aplicação e gerenciar o tempo de vida dos objetos.

I. Injeção de Dependência

Injeção de dependência é um padrão que não permite que os objetos se configurem por eles mesmos. Ao invés disso, as dependências devem aparecer explicitadas em uma entidade externa (um arquivo de configuração do *framework Spring*, por exemplo).

II. Módulos

Abaixo se tem a Figura 3 onde a distribuição dos módulos do *framework Spring* pode ser visualizada.

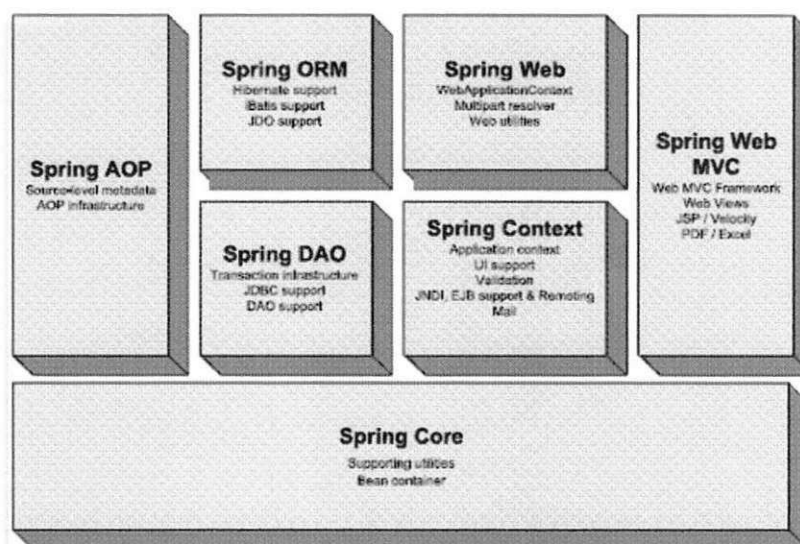


Figura 3 - Módulos do *Spring* [JOHNSON, 2010].

Tais módulos podem ser definidos como segue.

a. *Spring Core*

Fornece a funcionalidade fundamental do *Spring*. Possui como elemento principal o *BeanFactory*, que é uma implementação

do padrão *Factory*, que aplica inversão de controle para diminuir o acoplamento entre a especificação de dependências e o código da aplicação [WALLS, 2005].

b. Spring DAO

Possui uma camada de abstração para o *JDBC*, eliminando grande parte da codificação necessária para comunicação com um banco de dados.

c. Spring ORM

Permite a integração do *Spring* com *frameworks* para persistência de objetos, como o *Hibernate*.

d. Spring AOP

Este módulo provê suporte a uma implementação orientada a aspectos, possibilitando a programação de metadados que permitem adicionar anotações em seus códigos fonte para especificar onde e como aplicar aspectos (definição de *pointcuts*).

e. Spring Context

Esse módulo aumenta o conceito de *BeanFactory*, adicionando eventos de ciclos de vida e de validação, além de fornecer serviços como integração com *EJB*, acesso *JNDI* e permitir o suporte para integração com outros *frameworks* como, por exemplo, o *FreeMaker*.

f. Spring Web

Provê funcionalidades específicas para projetos *web* como, por exemplo, facilitar as tarefas de manipulação de múltiplas requisições e vincular parâmetros de requisição a objetos de domínio. Permite integração com *Jakarta Struts* e *JSF* [WALLS, 2005].

g. Spring Web MVC

O *Spring* pode ser facilmente integrado com outros *frameworks MVC*, mas esse módulo fornece uma implementação de

framework web que utiliza inversão de controle para separar o controlador da lógica dos objetos de negócio [WALLS, 2005].

3.2.3 JSF

Uma aplicação *web* é composta por duas partes: a camada de apresentação e a lógica de negócios. A camada de apresentação se preocupa com a aparência do aplicativo. No contexto de uma aplicação baseada em *browser*, o visual é determinado pelas *tags HTML* que especificam o *layout*, fontes, imagens, e assim por diante. A lógica de negócio é executada em um código *Java* que determina o comportamento do aplicativo [GEARY, 2007].

No contexto do *JSF*, o código do aplicativo está contido nos *beans*, e a parte do design está contida nas páginas *web*.

I. Bean

É uma classe que expõe propriedades e eventos para um ambiente como o *JSF*. A propriedade é um valor nomeado de um determinado tipo que pode ser lido e/ou escrito. A maneira mais simples para definir uma propriedade é a utilização de uma convenção de nomenclatura padrão para os métodos de leitura e escrita, que são *get/set*. A primeira letra do nome da propriedade é alterada para letra maiúscula no nome dos métodos.

Os métodos *get/set* podem realizar ações arbitrárias. Em muitos casos, eles simplesmente definem um campo de instância, mas eles também podem acessar um banco de dados. Nas aplicações *JSF*, você pode usar um *bean* para todos os dados que precisam ser acessíveis a partir de uma página.

II. Páginas JSF

É necessária uma página *JSF* para cada tela do navegador. Dependendo do seu ambiente de desenvolvimento, as páginas *JSF* normalmente tem a extensão *.jsf* ou *.jsp*.

III. Navegação

A navegação pode ser estática ou dinâmica. A estática é uma navegação entre páginas pré-determinadas, ou seja, sempre que o usuário clicar em um determinado botão será redirecionado para uma página definida de forma fixa no código. Na dinâmica o resultado de navegação é gerado por um método que foi invocado e então os resultados são mapeados para páginas de saída.

3.2.4 MVC

O *MVC* é um padrão de arquitetura de *software* que descreve uma forma coerente de estruturar uma aplicação. Seu objetivo é separar a lógica de negócio (*Model*) da interface do usuário (*View*) e do fluxo da aplicação (*Controller*).

Esse modelo foi descrito pela primeira vez em 1979, na comunidade de *Smalltalk*, mas seu contexto mudou um pouco, pois o conceito de aplicações web ainda não existia.

Existem três diferentes responsabilidades no MVC e as mesmas podem ser descritas como:

1. Controlador ou *controller*: é responsável pelas requisições feitas pelo usuário. Sua função é chamar os recursos e objetos necessários para atender essas requisições. Ele deve chamar o *Model* apropriado, requisitar os dados e enviar para o *View*;
2. *Model*: é a lógica de negócio. Representa os dados e as regras que se aplicam aos dados;
3. *View*: permite que os dados possam ser apresentados em diferentes formas (tem muito a ver com interface gráfica).

4. ATIVIDADES REALIZADAS

Nesta seção, encontram-se descritas as atividades realizadas no decorrer deste estágio.

Vale ressaltar que tais atividades não são exatamente aquelas previstas no plano de estágio. São elas:

1. Estudo das tecnologias e da arquitetura do sistema SisLegal;
2. Desenvolvimento de um protótipo utilizando as tecnologias aprendidas;
3. Desenvolvimento do módulo 1 (cadastro das escolas) do sistema SisLegal;
4. Integração dos módulos do sistema desenvolvido;
5. Implementação de testes para o módulo de cadastro alunos;
6. Suporte técnico em informática;
7. Geração de relatório sobre as atividades do estágio.

Maiores informações estão no Quadro 1.

Quadro 1 – Descrição das atividades e do tempo gasto em cada uma delas.

ATIVIDADE	DESCRIÇÃO	TEMPO GASTO (h)
1	Estudos.	50
2	Desenvolvimento de um protótipo.	35
3	Desenvolvimento do módulo 1.	50
4	Integração.	25
5	Testes.	30
6	Suporte Técnico.	80
7	Relatório.	30
TOTAL		300

As discrepâncias entre as atividades planejadas e as realizadas de fato serão comentadas no decorrer desta seção.

4.1 Planejamento

O cronograma proposto no Plano de Estágio, Apêndice A, sofreu algumas alterações. O novo cronograma pode ser visualizado no Quadro 2.

Os requisitos foram, na verdade, elicitados e validados pelos funcionários do MPPB de João Pessoa. Sendo assim, a equipe desenvolvedora de Campina Grande já começou a desenvolver o projeto com os requisitos validados em mãos. Durante uma das únicas reuniões realizadas no desenvolvimento do projeto, foi apresentado o diagrama de casos de uso abaixo.

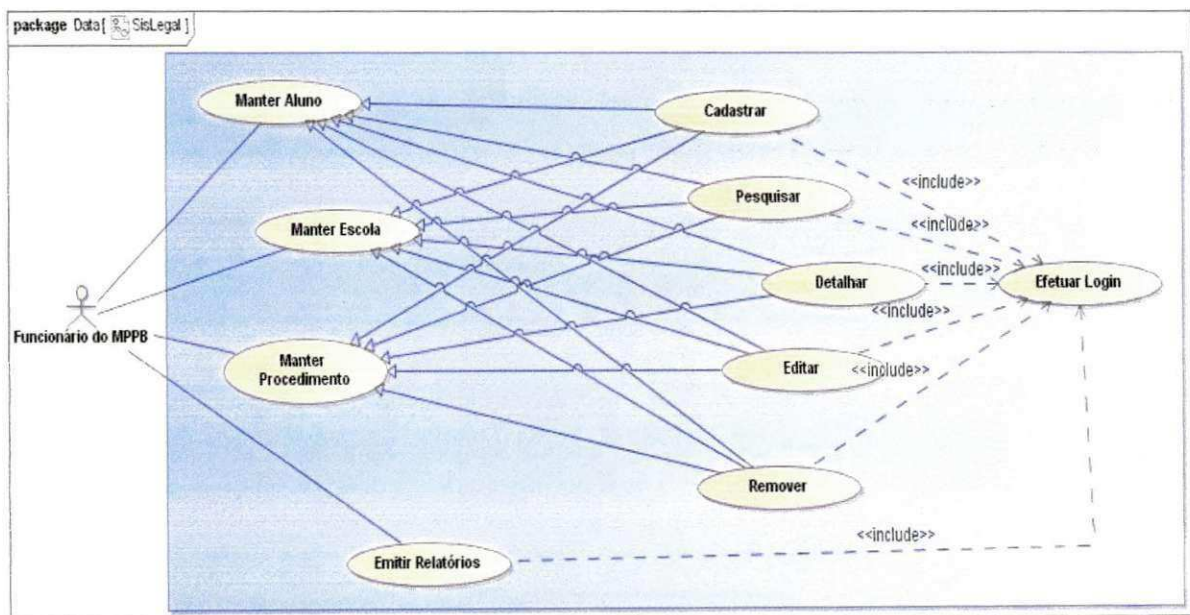


Figura 4 – Diagrama geral de casos de uso do SisLegal.

Assim sendo, ficaram claras quais seriam as funcionalidades do sistema.

A equipe de João Pessoa também ficou responsável pelas etapas de treinamento e implantação do SisLegal.

O novo cronograma pode ser visualizado no Quadro 2.

Quadro 2 – Novo cronograma de atividades.

ATIVIDADE	MAR	ABR	MAI	JUN
Atividades de suporte técnico.				
Estudo das tecnologias utilizadas no departamento de Desenvolvimento de Sistemas – DITEC.				
Análise da arquitetura do sistema.				
Implementação do módulo 1.				
Testes unitários, de integração e funcionais.				
Redação do relatório técnico.				

4.2 Implementação

Na etapa de desenvolvimento do módulo 1 do SisLegal (cadastro das escolas existentes em um dado município), foram implementadas classes Java e arquivos *.xhtml* referentes à interface gráfica.

Cada escola tem que possuir: nome, nível de ensino, tipo de administração e endereço completo.

Tendo em vista que o sistema foi desenvolvido sob o padrão MVC, as tarefas foram divididas de acordo com as camadas.

A primeira tarefa tratou da implementação de persistência de dados referentes às escolas. Foi necessária a criação da tabela com seus respectivos atributos no banco de dados. Após a criação foi desenvolvida uma entidade *Escola* que utilizava o *annotation @Entity*, o qual provê o mapeamento para a tabela supracitada.

Foi necessário mapear essa entidade no arquivo *hibernate.cfg.xml* para que o Hibernate conseguisse localizar a classe que referencia a tabela.

Na Figura 5, é apresentado um trecho de código da classe *Escola* que ilustra o que foi anteriormente comentado sobre persistência de dados referentes às escolas. Nota-se que a sintaxe é similar àquela de uma classe *Java* comum que não tem anotações de persistência.

```

13 @Table(name="escola", schema=Constants.PUBLIC_SCHEMA)
14 @Inheritance(strategy=InheritanceType.JOINED)
15 @SequenceGenerator(name="sequence", sequenceName=Constants.PUBLIC_SCHEMA + ".escola_id_seq")
16 @Entity
17 @Where(clause="excluido != 'f'")
18 public class Escola extends AbstractEntity implements IEntidadeExcluiavelLogicamente{
19
20     @Id
21     @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="sequence")
22     @Column(name="id_escola")
23     private Long id;
24
25     @Column(name="nome")
26     private String nome;
27
28     @Column(name="excluido")
29     private Boolean excluido;
30
31     @OneToOne(fetch = FetchType.LAZY, cascade =
32         {javax.persistence.CascadeType.PERSIST,
33          javax.persistence.CascadeType.MERGE,
34          javax.persistence.CascadeType.REFRESH})
35     @JoinColumn(name="id_endereco")
36     @Cascade(value = {org.hibernate.annotations.CascadeType.SAVE_UPDATE, org.hibernate.annotations.CascadeType.MERGE,
37                     org.hibernate.annotations.CascadeType.DELETE, org.hibernate.annotations.CascadeType.DELETE_ORPHAN,
38                     org.hibernate.annotations.CascadeType.PERSIST})
39     private Endereco endereco;
40
41     @Column(name="tipo_administracao")
42     private String tipoAdministracao; //privada, municipal, estadual, federal
43     @Column(name="nivel_ensino")
44     private String nivelEnsino; //infantil, fundamental, médio

```

Figura 5 – Trecho de código da classe *Escola*.

Foi necessário mapear esta entidade no arquivo *hibernate.cfg.xml*, para que o Hibernate conseguisse localizar a classe que referencia a tabela.

A segunda tarefa consistiu em implementar uma classe referente ao papel de controlador do padrão MVC. Tal classe contém os métodos que serão invocados na interface. Os métodos devem retornar várias *strings* que sinalizam o sucesso ou não de uma dada ação. Caso o retorno tenha valor nulo (*null*), a mesma tela deve ser mostrada novamente, pois isto possibilitará o mapeamento correto para as páginas de saída.

A terceira tarefa se deu mediante a implementação da interface gráfica (as visões previstas pelo padrão MVC).

Existe uma página de *login* que deve ser visualizada assim que um dado usuário acesse o sistema. O *print screen* de tal página está apresentado na Figura 6.



Figura 6 – Tela de login do SisLegal.

Logo após efetuar o *login* no sistema, o usuário deve ser redirecionado para a página inicial, ilustrada na Figura 7. Nessa tela, o usuário depara-se com o menu principal do sistema, no qual se encontram as principais opções de navegação do sistema: *Aluno*, *Escola* e de *Procedimento*, além da opção *Início*. O menu principal deve aparecer em todas as telas do SisLegal.



Figura 7 – Página inicial

Ao selecionar a opção *Escola*, o usuário será direcionado para a página principal referente ao cadastro de escolas, encontrando as opções de cadastro e pesquisa de escolas (ver Figura 8).

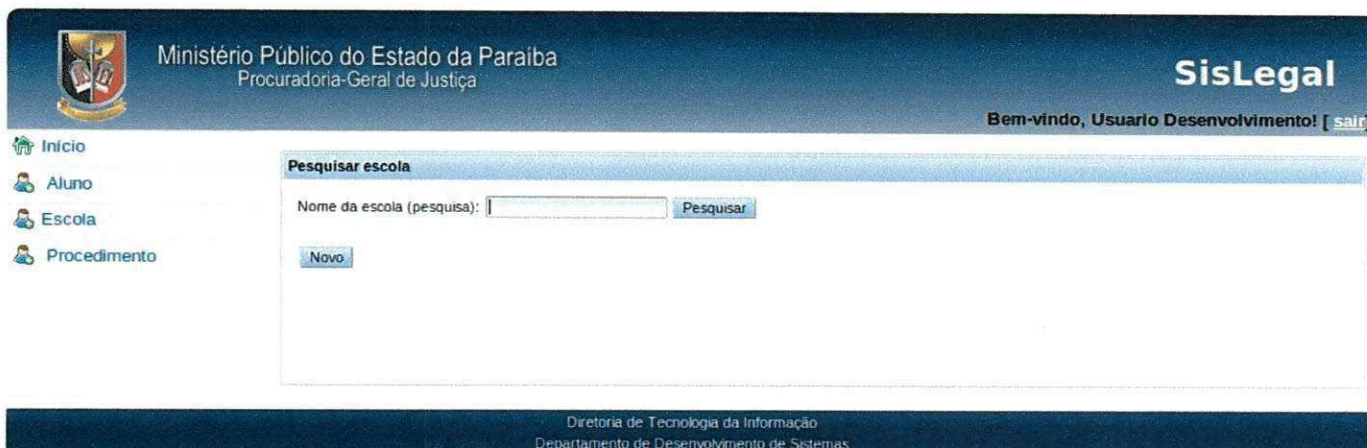


Figura 8 – Página principal da seção *Escola*.

Estando na página principal da seção *Escola*, o usuário pode optar por cadastrar uma nova escola. Para isso, basta clicar no botão *Novo*. Desta forma, será exibida a página de cadastro de escola, na qual se encontram os campos que deverão ser preenchidos pelo usuário. Há tratamento de erros para os atributos, fazendo com que os dados inseridos sejam os mais consistentes possíveis (ver Figura 9).

The image displays the 'Cadastrar escola' form within the SisLegal system. The header and navigation elements are identical to Figure 8. The main content area is titled 'Cadastrar escola' and contains the following fields: 'Nome:' (text input), 'Nível de ensino:' (dropdown menu with '-- seleccione --'), 'Tipo de Administração:' (dropdown menu with '-- seleccione --'), 'Endereço:' (text input), 'CEP: (Informe o CEP para preencher o endereço):' (text input), 'Rua:' (text input), 'Número:' (text input), 'Bairro:' (text input), 'Cidade:' (text input), and 'Estado:' (text input). At the bottom of the form are two buttons: 'Adicionar' and 'Voltar'. The footer text at the bottom of the page is 'Diretoria de Tecnologia da Informação' and 'Departamento de Desenvolvimento de Sistemas'.

Figura 9 – Página de cadastro de escolas

A Figura 10 contém um trecho de código em *JavaScript*, o qual segue a estrutura de um arquivo *xml*. Pode-se observar o uso de componentes como *inputText* (utilizado para receber valor de atributos) e *selectOneMenu* (*combo box*), que é utilizado para efetuar a chamada de métodos que exibem na tela informações do banco de dados; um dado usuário só pode escolher uma das opções listadas.

```
137 <h:outputLabel value="Nome: "/>
138 <h:inputText value="#{escolaMBean.escola.nome}" size="35" required="true" id="escola_nome"/>
139
140 <h:outputLabel value="Nível de ensino: "/>
141 <h:selectOneMenu value="#{escolaMBean.escola.nivelEnsino}">
142 <f:selectItem itemValue="null" itemLabel="-- seleccione --"/>
143 <f:selectItems value="#{escolaMBean.niveisEnsino}"/>
144 </h:selectOneMenu>
```

Figura 10 – Exibição do campo a ser preenchido referente ao nome de uma escola e da *combo box* referente aos possíveis níveis de ensino.

4.3 Validação e Inspeção

Testes funcionais de cada módulo desenvolvido foram realizados para verificar possíveis falhas no sistema. Verificou-se também as restrições dos atributos das classes e as mensagens de erro.

A inspeção consistiu em compartilhar o código com outro desenvolvedor da equipe que procurava por de erros e sugeria melhorias no código.

4.4 Integração

A última atividade de desenvolvimento realizada foi a integração dos módulos que foram terminados, a saber: módulo de cadastro de alunos, módulo de cadastro de escolas e módulo de cadastro de procedimentos.

4.5 Suporte Técnico em Informática

As atividades de suporte técnico, que ocorreram durante todo o estágio, podem ser divididas em três:

1. Auxílio no uso de aplicativos;

2. Auxílio no uso dos recursos tecnológicos e manutenção dos mesmos;

3. Manutenção da rede de computadores.

A primeira atividade listada diz respeito, principalmente, ao auxílio no uso dos sistemas operacionais (*Microsoft Vista* e *Microsoft XP*) e da ferramenta *open source BrOffice*, que é a ferramenta de edição de textos utilizada pelos funcionários do MPPB.

Na segunda atividade, se encaixam: reciclagem de cartuchos de impressão, troca de cabos de conexão com a internet e auxílios em geral no uso dos computadores. Tais auxílios foram: troca de cabos de monitores e de fontes de impressoras, formatação de *pendrives* e discos rígidos, etc.

A última atividade listada envolve, basicamente, a resolução de problemas de configuração de *proxy* para a utilização da rede, inserção e remoção de máquinas da rede e de impressoras em grupos de trabalho. As duas figuras presentes na próxima página ilustram parte das atividades de manutenção de rede.

Todas as atividades de suporte técnico ocorreram em paralelo àquelas de desenvolvimento, anteriormente descritas e durante todo o estágio.

Embora fora do escopo do desenvolvimento de *software*, eixo do curso de Ciência da Computação da UFCG, tais atividades serviram de aproximação com o usuário de *software*. Saber as dificuldades dos usuários é muito importante para o processo de desenvolvimento de qualquer sistema, pois o sucesso de um projeto está muito relacionado com a facilidade que um dado usuário encontrará ao usar o sistema desenvolvido.

As saídas geradas por dois comandos típicos de manutenção de rede nos sistemas operacionais da família *Windows* (*ping* e *net view*) podem ser observadas nas Figuras 11 e 12.


```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [versão 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador>ping 10.128.64.10

Disparando contra 10.128.64.10 com 32 bytes de dados:

Resposta de 10.128.64.10: bytes=32 tempo<1ms TTL=128
Resposta de 10.128.64.10: bytes=32 tempo<1ms TTL=128
Resposta de 10.128.64.10: bytes=32 tempo<1ms TTL=128
Resposta de 10.128.64.10: bytes=32 tempo<1ms TTL=128

Estatísticas do Ping para 10.128.64.10:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
    Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 0ms, Máximo = 0ms, Média = 0ms

C:\Documents and Settings\Administrador>

```

Figura 11 – Verificando tempo de resposta do servidor.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [versão 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador>net view

Servidor          Descrição
-----
\\2CAOP-PATPUB-03
\\ASS-JURIDICA1    Assessoria Jurídica 2ª CAOP
\\CAIMP-2CRIMI-2C
\\CAIMP-3CRIMI-2C
\\CAIMP-4CRIM-2C
\\CAIMPCARTORIO2
\\CART-CAIMP-C5
\\CONSU-PROMO-C1
\\CONSUMIDOR-UIMI
\\CONSUMIDOR3      Atendimento da Curadoria do Consumidor
\\COORD-CAOP2-C1
\\COORD-SEC
\\CURAD-FUN-CA-2C
\\CURAD-FUN-PR-2C
\\CURAD-INF-CA-C4
\\CURAD-INF-PR-2C
\\DEFENSORIA_2
\\DELEGADO1
\\INFAN-CART-C1
\\MEIO-AMBIENTE2   Cartório da Promotoria do MA
\\MP-EXEC-PENAL
\\MPPB2
\\NUJCON-CONT1
\\P-3E6303FCF3474
\\PATPUB2
\\PROM-CID-01
\\PROM-CID-02
\\SAUDE1
\\SAUDE2           saude2
\\SERVER-TELE
Comando concluído com êxito.

C:\Documents and Settings\Administrador>

```

Figura 12 – Visualização de rede.

5. CONSIDERAÇÕES FINAIS

O estágio é uma oportunidade única na qual o graduando tem a chance de por em prática conceitos vistos em sala de aula. Além disso, é possível aprender novas técnicas e tecnologias que não são tratadas na graduação.

Um estudante, no âmbito de um curso de tecnologia, toma conhecimentos dos problemas que podem surgir no decorrer do desenvolvimento de uma aplicação. Porém, a maioria dos exercícios propostos não leva em consideração a utilização de um cliente real. Assim sendo, fica muito difícil, por parte do aluno, entender o quão impactante pode ser um sistema dentro de uma organização real. Daí tem-se a importância da realização de um estágio.

Três dificuldades encontradas no decorrer das atividades, descritas na seção anterior deste documento e que merecem menção, foram:

1. Falta de conhecimento no desenvolvimento de tecnologias *web*;
2. Adaptação à burocracia existente em uma repartição pública; e
3. Falta de habilidade dos funcionários em geral no que diz respeito ao manuseio dos recursos tecnológicos.

Como o curso de Ciência da Computação da UFCG oferece disciplinas em que a prática da programação é estudada, sem levar em conta nenhum tipo específico de linguagem ou tecnologia (como, por exemplo, a disciplina *Paradigmas de Linguagens de Programação*), a primeira dificuldade citada anteriormente foi facilmente superada. Todavia, é importante frisar que o curso supracitado carece de disciplinas optativas relacionadas com tecnologias *web*.

Em relação à segunda dificuldade, sua superação se deu com a ajuda de colegas de trabalho que já estavam na organização há mais tempo. Cada empresa tem as suas próprias peculiaridades. Portanto, é preciso certo tempo para que os novos funcionários se adaptem bem às organizações nas quais os mesmos trabalham.

Por fim, destaca-se a necessidade de um investimento maior, por parte do MPPB, no que diz respeito à educação tecnológica (em virtude da terceira dificuldade listada). Notou-se que a maioria dos funcionários sabe menos do que o necessário sobre os aplicativos que os mesmos necessitam usar no dia-a-dia do trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

BAUER, C., KING, G. Hibernate In Action. Manning Publications, 2005.

GEARY, D. M. Core Java Server Faces. 2.ed. Alta Books, 2007.

JOHNSON, R., HOELLER, J. Spring – Java/J2EE Application Framework. Disponível em:

<<http://static.springsource.org/spring/docs/1.2.9/reference/index.html>>. Acessado pela última vez em: 24 de maio de 2011.

WALLS, C., BREIDENBACH, R. Spring In Action. Manning Publications, 2005.

APÊNDICE

Plano de Estágio



Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Sistemas e Computação – DSC

Plano de Estágio Integrado

Ministério Público da Paraíba

Desenvolvimento do Sistema SisLegal e Suporte Técnico em Informática

Aluno: Lenin da Nóbrega Medeiros (21011093)
Supervisor Acadêmico: José Eustáquio Rangel de Queiroz
Coordenadora de Estágio: Joseana Macêdo Fachine
Semestre Letivo: 2011.1
Disciplina: Estágio Integrado
Créditos e Carga Horária: 10 créditos e 300 horas

Campina Grande, março de 2011.

Informações Pessoais

Nome: Lenin da Nóbrega Medeiros

Curso: Ciência da Computação

Matrícula: 21011093

CPF: 015.741.004-83

RG: 2004029192424 SSP/CE

Endereço: Rua Frei Caneca 275, apto 704

CEP: 58.400-295

Centro, Campina Grande, PB – Brasil.

Telefone: (83) 8709-9263

Email: lenin.medeiros@gmail.com

Ambiente de Estágio

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

As atividades que serão desenvolvidas no âmbito deste estágio durante o período 2011.1 serão realizadas por uma equipe composta por quatro desenvolvedores e um gerente, que será responsável por delegar as atividades de acordo com as atribuições definidas pelo Termo de Compromisso de Estágio.

Supervisão

Supervisor Acadêmico

Nome: José Eustáquio Rangel de Queiroz, DSc.

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: rangel@dsc.ufcg.edu.br

Supervisor Técnico

Nome: Uirá Vasconcelos Alencar de Assis²

Endereço: Rua Promotora Terezinha Lopes de Moura, S/N
CEP: 58.410-064
Liberdade, Campina Grande, PB – Brasil.

Email: uira@mp.pb.gov.br

² Chefe de suporte técnico em informática do Ministério Público da Paraíba.

Sinopse das Atribuições do Estágio

O Ministério Público da Paraíba é um órgão da esfera jurídica, cuja atuação acontece junto ao cidadão paraibano, no tocante à defesa dos seus direitos. Defender os interesses legais das crianças paraibanas é, portanto, um dos deveres da instituição.

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 é também direito de cada indivíduo, de acordo com a nossa Constituição Federal.

Com a implantação do projeto Nome Legal, o Ministério Público da Paraíba visa reduzir o número de crianças sem o reconhecimento de paternidade a partir de um 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. É neste contexto que é inserido o sistema SisLegal (uma atribuição do estágio, no que diz respeito ao desenvolvimento), que tem o intuito de dar suporte, em termos de TI, ao projeto Nome Legal.

O sistema supracitado tem 3 objetivos:

1. Criação de um cadastro para as informações de nomes de escolas, diretores e relação de alunos e respectivas mães com os registros dos alunos e respectivos pais para o projeto Nome Legal;
2. 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;
3. Criação de um módulo gerencial no qual a pessoa responsável pela gestão do projeto poderá acompanhar o andamento dos vários casos cadastrados no sistema e tomar as devidas providências, caso necessário.

O sistema SisLegal é dividido em 5 módulos:

1. Cadastro das escolas existentes em um dado município (particulares, municipais, estaduais e federais), de todos os níveis de ensino;

2. Cadastro da relação de alunos sem maternidade ou paternidade reconhecida ou sem registro;

3. Cadastro da relação de procedimentos por escola originados a partir dos cadastros do item anterior;

4. Registro das notificações expedidas para marcação das oitivas (audição; ato de ouvir 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 respectivos motivos, além das relações de escolas e alunos.

Outra atribuição do estágio é a de suporte técnico em informática. Serão feitas as manutenções dos recursos tecnológicos (computadores e dispositivos) e da rede na qual estão conectadas as máquinas do prédio do Ministério Público da Paraíba em Campina Grande, além do auxílio aos usuários em relação aos *softwares* instalados nos computadores (sistemas operacionais e aplicativos).

Objetivos do Estágio

Os objetivos do estágio são:

1. Desenvolver e implantar o sistema SisLegal;
2. Oferecer suporte técnico em informática para todo o prédio do Ministério Público da Paraíba em Campina Grande.

Vale ressaltar que os módulos do sistema estão especificados nas seções "Sinopse das Atribuições do Estágio". Por outro lado, as tecnologias a serem utilizadas só serão definidas durante o estágio.

Metodologia de Desenvolvimento

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 coleta e documentação dos sistemas.

Pode-se, em síntese, dizer que SCRUM é um processo iterativo e incremental, cujo intuito é gerenciar projetos e desenvolver produtos de *softwares* de maneira ágil. Kanban, por sua vez, sinaliza um controle de fluxo de atividades via registros, permitindo sinalização imediata da demanda que sustenta as prerrogativas da cadeia de execução, inspeção e elaboração de um processo. Assim como foi citado anteriormente, técnicas de RUP estão sob utilização, tendo como pilar a redução do risco (existente em qualquer projeto de *software*), tornando mais eficiente o desenvolvimento com a verificação contínua de qualidade e o controle de mudanças.

Atividades Planejadas

Devem ser desenvolvidas as seguintes atividades no estágio (além daquelas que dizem respeito ao suporte técnico, que não serão planejadas e ocorrerão paralelamente às atividades de desenvolvimento durante todo o estágio):

1. Levantamento dos requisitos do sistema com a pessoa responsável pela gestão do projeto;
2. Análise e arquitetura do projeto;
3. Implementação dos módulos descritos no item 2;
4. Testes;
5. Implantação;
6. Manutenção.

Durante este estágio, em relação ao desenvolvimento do sistema SisLegal, o trabalho será focado no desenvolvimento e nos testes dos 1º e 2º módulos, respectivamente (ambos descritos na seção "Sinopse das Atribuições do Estágio").

Cronograma de Atividades

O cronograma de atividades proposto (de acordo com o calendário acadêmico da UFCG) é o seguinte:

Tarefa	Mar	Abr	Mai	Jun
Atividades de suporte técnico.	X	X	X	X
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		
Implementação do módulo 1.		X		
Implementação do módulo 2.			X	
Implementação do módulo 3.			X	
Implementação do módulo 4.			X	
Implementação do módulo 5.			X	
Testes unitários, de integração e funcionais.		X	X	X
Treinamento.				X
Implantação.				X
Elaboração do relatório técnico.	X	X	X	X
Defesa do estágio.				X

Resultados Esperados

Espera-se que, ao fim do estágio, o sistema SisLegal esteja finalizado, conforme o cronograma apresentado, e sendo utilizado pelo Ministério Público da Paraíba. Espera-se, portanto, que a promotora responsável pelo projeto Nome Legal, Dra. Renata Carvalho da Luz, 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

José Eustáquio Rangel de Queiroz
Orientador Acadêmico

Joseana Macêdo Fchine
Coordenadora da disciplina Estágio Integrado