

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**DIEGO VIERO
JULIAN DALLA CORTE MATUCHAK**

**SISTEMA WEB PARA GERENCIAMENTO DE TREINAMENTOS
DE IMPLANTAÇÃO DE SISTEMAS**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2015**

**DIEGO VIERO
JULIAN DALLA CORTE MATUCHAK**

**SISTEMA WEB PARA GERENCIAMENTO DE TREINAMENTOS
DE IMPLANTAÇÃO DE SISTEMAS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Campus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

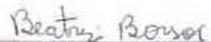
Orientadora: Profa. Beatriz Terezinha Borsoi

**PATO BRANCO
2015**

ATA Nº: 268

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DOS ALUNOS DIEGO VIERO e JULIAN DALLACORTE MATUCHAK.

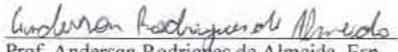
Às 14:30 hrs do dia 25 de junho de 2015, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (Orientadora), Robison Cris Brito (Convidado) e Anderson Rodrigues de Almeida (Convidado), para avaliar o Trabalho de Diplomação do aluno Diego Viero, matrícula 1116657 e do aluno Julian Dallacorte Matuchak, matrícula 1116711, sob o título **Sistema Web para gerenciamento de treinamentos na implantação de sistemas**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação os candidatos foram entrevistados pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 15:25 hrs foi encerrada a sessão.



Profa. Beatriz Terezinha Borsoi, Dr.
Orientadora



Prof. Robison Cris Brito, M.Sc.
Convidado



Prof. Anderson Rodrigues de Almeida, Esp.
Convidado



Prof. Soelaine Rodrigues Ascari, M.Sc.
Coordenador do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

RESUMO

MATUCHAK, Julian Dalla Corte; VIERO, Diego. Sistema *web* para gerenciamento de treinamentos de implantação de sistemas. 2015. 55 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2015.

As atividades dentro de uma empresa de desenvolvimento de software consistem em várias etapas, uma delas é a implantação. Nessa atividade, o cliente passa por um processo de treinamento e adaptação ao software. O processo de treinamento pode ser realizado via Internet, utilizando ferramentas de comunicação *web* como Skype, ou podem ser realizados presencialmente, com os consultores dirigindo-se até o estabelecimento do cliente. Para que o treinamento ocorra é necessária a definição de alguns parâmetros como, data, horário e conteúdo a ser abordado, entre outros itens. Todo esse processo precisa ficar registrado para controle da empresa e para possibilitar consultas futuras caso necessário. Se não houver um sistema específico ou um sistema de gerenciamento com funcionalidades que permitam o controle das atividades de treinamento, os dados do processo ficam armazenados em locais diferentes. É, assim, dificultada sua recuperação, o controle das atividades sendo realizadas, a agenda dos funcionários que realizam as capacitações e o agendamento dos treinamentos com os clientes. Verificou-se, assim, a possibilidade de implementar um sistema *web* que permita que todos os processos sejam feitos e armazenados em um único local. Esse sistema, implementado utilizando a linguagem PHP e o banco de dados MySQL, foi desenvolvido como resultado deste trabalho.

Palavras-chave: Aplicativo web. Gerenciamento de treinamentos. Linguagem PHP.

LISTA DE FIGURAS

Figura 1 – Diagrama de casos de uso.....	20
Figura 2 – Diagrama de entidades e relacionamentos.....	21
Figura 3 - Tela inicial do sistema	26
Figura 4 - Tela de <i>login</i>	27
Figura 5 - Cadastro de usuário.....	28
Figura 7 - Configuração de e-mail	29
Figura 8 - Cadastro de módulo - sem registros	30
Figura 9 - Cadastro de módulos	30
Figura 10 – Listagem dos registros dos módulos	31
Figura 11 - Edição de módulos	31
Figura 12 – Relatórios.....	32
Figura 13 - Relatório de agendamento	32
Figura 14 – Relatório filtros	33
Figura 15 - <i>Grid</i> de agendamentos – visão consultor	34
Figura 16 - Cadastro de agendamentos – visão consultor.....	34
Figura 17 – Agendamentos – visão cliente.....	35
Figura 18 – Visualização agendamento – visão cliente.....	35

LISTA DE QUADROS

Quadro 1 – Ferramentas e tecnologias utilizadas	16
Quadro 2 – Requisitos funcionais.....	19
Quadro 3 – Requisitos não funcionais	19
Quadro 4 – Campos da tabela Usuarios	21
Quadro 5 – Campos da tabela Consultores.....	22
Quadro 6 – Campos da tabela OpcoesEmail.....	22
Quadro 7 – Campos da tabela Cidades	22
Quadro 8 – Campos da tabela Estados	23
Quadro 9 – Campos da tabela Tipo_Treinamento	23
Quadro 10 – Campos da tabela Agendamentos	23
Quadro 11 – Campos da tabela Clientes	24
Quadro 12 – Campos da tabela Modulos	24
Quadro 13 – Campos da tabela Tipo_Contato.....	24
Quadro 14 – Campos da tabela Contatos.....	25
Quadro 15 – Campos da tabela LogEnvioEmail	25

LISTAGENS DE CÓDIGO

Listagem 1 – Código da página inicial (home)	39
Listagem 2 – Insert usuários.....	40
Listagem 3 – Código para exclusão de usuários.....	41
Listagem 4 - Edição de usuários.....	42
Listagem 5 – Código para mensagem de confirmação de exclusão.....	42
Listagem 6 – Chamada dos métodos/funções	43
Listagem 7 – Verifica se há registros no banco de dados	43
Listagem 8 – Código para carregamento de dados no <i>grid</i>	44
Listagem 9 – Insert módulos.....	44
Listagem 10 – Relatório	45
Listagem 11 – E-mail.....	46
Listagem 12 – Tela agendamentos	51

LISTA DE SIGLAS

AJAX	<i>Asynchronous JavaScript and XML</i>
CRUD	<i>Create, Retrieve, Update and Delete</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
IoS	<i>Internet of Services</i>
ODBC	<i>Open Database Connectivity</i>
PDF	<i>Portable Document Format</i>
PHP	<i>Hypertext Preprocessor</i>
RIA	<i>Rich Internet Applications</i>
RIA	<i>Rich Internet Applications</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SQL	<i>Structured Query language</i>
TI	<i>Tecnologias de Informação</i>
WWW	<i>World Wide Web</i>
XML	<i>Extensible Markup Language</i>
XUL	<i>XML User Interface Language</i>

SUMÁRIO

1 INTRODUÇÃO.....	9
1.1 CONSIDERAÇÕES INICIAIS	9
1.2 OBJETIVOS	9
1.2.1 Objetivo Geral.....	10
1.2.2 Objetivos Específicos	10
1.3 JUSTIFICATIVA	10
1.4 ORGANIZAÇÃO DO TEXTO	11
2 REFERENCIAL TEÓRICO	12
2.1 DESENVOLVIMENTO DE APLICAÇÕES WEB	12
3 MATERIAIS E MÉTODO	16
3.1 MATERIAIS.....	16
3.2 MÉTODO	17
4 RESULTADO	18
4.1 ESCOPO DO SISTEMA	18
4.2 MODELAGEM DO SISTEMA.....	19
4.3 APRESENTAÇÃO DO SISTEMA	25
4.4 IMPLEMENTAÇÃO DO SISTEMA	36
5 CONCLUSÃO.....	52
REFERÊNCIAS.....	54

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa da realização deste trabalho. No final do capítulo é apresentada a organização do texto por meio de uma breve apresentação dos seus capítulos.

1.1 CONSIDERAÇÕES INICIAIS

Ao adquirir um software o cliente passa por um processo denominado implantação. Nesse processo podem ser realizados vários treinamentos para adaptação ao novo sistema. As atividades realizadas nesse processo precisam ser registradas pelos consultores que são os responsáveis por acompanhar o novo cliente. Para melhor organização todos os treinamentos e visitas são agendadas com prévia antecedência, seguindo um cronograma que é definido em comum acordo entre consultor responsável e cliente. O consultor deve relatar, após o término do treinamento, tudo o que ocorreu na ocasião, dúvidas, sugestões, conteúdo abordado.

Diante deste contexto, percebeu-se que um sistema computacional que permitisse aos envolvidos no processo de treinamento registrar todas as ações necessárias para a capacitação no uso do sistema contribuiria para facilitar o gerenciamento desse tipo de atividade. Um sistema *web* facilitaria o acesso pelos consultores e por todos que estejam envolvidos direta ou indiretamente no processo.

Neste texto é apresentada a modelagem elaborada para o sistema e é mostrado o uso das tecnologias utilizadas na implementação do mesmo.

1.2 OBJETIVOS

O objetivo geral está relacionado ao resultado principal que é esperado da realização deste trabalho. E os objetivos específicos complementam o objetivo geral em termos de funcionalidades do sistema.

1.2.1 Objetivo Geral

Modelar um sistema que permita aos responsáveis pela implantação de software ter o controle dos treinamentos que são realizados nos clientes.

1.2.2 Objetivos Específicos

- Facilitar o controle sobre os registros dos treinamentos realizados nos processos de implantação de sistema.
- Agilizar o processo de registro do treinamento desde seu agendamento até sua efetiva realização.
- Facilitar a consulta a informações decorrentes do processo de treinamento.
- Facilitar a visualização de resultados das atividades relacionadas ao treinamento.

1.3 JUSTIFICATIVA

A falta de organização e a descentralização das informações relacionadas ao processo de treinamentos de implantações é a justificativa principal para a realização deste trabalho. Para as pessoas que realizam o processo de treinamento do agendamento até a atualização dos registros após a realização dos treinamentos, é importante haver uma centralização do controle das tarefas que são realizadas, envolvidas no processo de implantação do sistema.

Para os consultores envolvidos nesses treinamentos, um sistema *web* tornaria mais fácil o processo de agendamento, atualização e controle sobre o conteúdo aplicado. É bem mais prático para os envolvidos possuírem todas essas informações em um mesmo local, facilitando consultas de relatórios e agilizando o acesso aos dados.

A justificativa de aplicabilidade do resultado deste trabalho se fundamenta na necessidade percebida de facilitar o controle das informações e agilizar o acesso a elas. Em termos de tecnologias, a escolha de implementação de um sistema *web* decorre da facilidade de acesso possibilitada por meio da Internet.

1.4 ORGANIZAÇÃO DO TEXTO

Este texto está organizado em capítulos. No Capítulo 2 está o referencial teórico que é sobre desenvolvimento de aplicações *web*. O aplicativo desenvolvido como resultado deste trabalho é para *web*. No Capítulo 3 estão os materiais e o método utilizado no desenvolvimento do trabalho. O Capítulo 4 apresenta o resultado da realização do trabalho que é a modelagem do sistema e a implementação das funcionalidades básicas de um cadastro. Por fim está a conclusão com as considerações finais.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico do trabalho. E está centrado no desenvolvimento de aplicações para Internet. A modelagem do aplicativo que é obtida como resultado desse trabalho é para uma aplicação *web*.

2.1 DESENVOLVIMENTO DE APLICAÇÕES WEB

A Internet, com um pouco mais de três bilhões de usuários, 42,3 % da população mundial (INTERNET..., 2014), é considerada para Vaz *et al* (2012) o artefato de engenharia mais bem sucedido que o ser humano já criou.

Da sua concepção inicial e acadêmica para ser uma rede de interconexão de computadores com fins militares, rapidamente evoluiu para uma plataforma global, provendo ampla comunicação e acesso a informação por meio do uso de protocolos padronizados e abertos.

O surgimento da *Word Wide Web* (WWW), popularizou a Internet que sendo utilizadas com as mais diversas finalidades, de infraestrutura para gerenciamento de grandes redes de produção e distribuição de bens e serviços a um imenso repositório de conteúdo e provedor de ferramentas de interação social e entretenimento.

O uso da Internet nos negócios é visto como um processo irreversível e crescente. Para Reding (2008), a Internet do futuro desempenhará um papel ainda mais vital em todos os processos de negócio. Vaz *et al.* (2012, p. 2292) ressaltam as Tecnologias de Informação (TI), com a Internet de forma particular, têm contribuído significativamente para a evolução do setor de serviços e, conseqüentemente, para a evolução da economia mundial.

A evolução tecnológica do final dos anos 90 levou o conceito de serviço para a *web*, por meio dos chamados *Web Services*. *Web Service* pode ser definido como uma aplicação que expõe (divulga) suas funcionalidades por meio da descrição da sua interface e torna essa descrição disponível para outros programas (BENHARREF; SERHANI; BOUKTIF, 2010). *Web Services* permitem que computadores e dispositivos possam automaticamente interagir uns com os outros usando a Internet para trocar e obter dados. A perspectiva futura da Internet está sendo guiada por um novo conceito conhecido como tecnologias *Web Services* (W3C, 2006).

Os *Web Services* estão estreitamente vinculados ao paradigma da Internet de Serviços (*Internet of Services (IoS)*). A IoS visa desenvolver novas teorias, modelos, arquiteturas e tecnologias para a criação eficiente e eficaz de serviços na Internet, permitindo que usuários não profissionais criem, comercializem e consumam serviços disponíveis na *web* (CARDOSO *et al.*, 2009).

Com o desenvolvimento de ferramentas e tecnologias para *Rich Internet Applications* (RIA) a interface com o usuário de aplicações *web* atuais estão migrando do tradicional hipertexto para a semântica das aplicações *desktop*. As aplicações *desktop* possuem vantagens em relação a interface com o usuário e a característica de comunicação multimídia em tempo real. Contudo as RIAs integram todas essas vantagens (PANG; WEN; PAN; LUI, 2010).

Tem havido uma crescente migração de uma multiplicidade de aplicações *desktop* para a *web*. Essas aplicações migradas, denominadas aplicações internet rica, as RIA (DUHL, 2003) são vislumbradas para ser a unificação do melhor das aplicações *desktop* tradicionais, tais como a interface de interatividade com o usuário e a rapidez de resposta dessas interfaces sem recarregamento da página, com o melhor das aplicações *web*, tais como uso de *download* progressivo para recuperação de conteúdo (POWELL; NAKAMURA; AKAMA, 2009). Ainda, podem ser citadas como vantagens das aplicações *web* a facilidade de acesso que ocorre por meio da Internet, não necessitando de redes proprietárias e a não necessidade de instalação da aplicação no cliente porque o acesso é realizado por meio de um navegador *web*.

Powell, Nakamura e Akama (2009) destacam que várias metodologias utilizadas para desenvolver aplicações *web* comuns não são rapidamente transferidas para RIA devido às características dessas aplicações (DUHL, 2003; PRECIADO *et al.*, 2005), tais como:

- a) carregamento da aplicação do servidor para o cliente em camadas;
- b) redução da quantidade de dados transferidos entre o servidor e o cliente por meio de atualização parcial da página;
- c) distribuição da carga de computação da página entre o servidor e o cliente, isto é, há processamento no servidor e no cliente;
- d) comunicação assíncrona entre o cliente e o servidor.

Também tem sido mostrado que modelos padrões de comportamento usados para especificar o comportamento comum de aplicações *web* são inadequados para

RIAs uma vez que elas podem exibir comportamentos ricos e mais flexíveis, tais como computações envolvendo somente partes da interface (COMAI; CARUGHI, 2007).

Além dessa certa ineficiência, os modelos tradicionais de desenvolvimento *web* destaca-se a necessidade de uma multiplicidade de tecnologias - como, por exemplo, para JavaScript, *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS) para o cliente; JSP, Perl, PHP, ASP, Java para o servidor; *Structured Query Language* (SQL), XQuery para o banco de dados, para citar algumas – para implementar uma aplicação *web* tradicional e RIA, em particular. Isso resulta em uma ampla diversidade de dificuldades, tais como problema de incompatibilidades (COOPER *et al.*, 2007), segurança, sincronia e relacionados às falhas nas tecnologias de implementação (POWELL; NAKAMURA; AKAMA, 2009).

Em adição às dificuldades para a construção de RIAs está o fato que há pelo menos quatro grandes categorias de RIAs com um caminho de transação não fácil entre elas (BOZZON *et al.*, 2006, STEARN, 2007). Essas quatro categorias são:

- a) baseadas em *script* – a lógica de negócio do lado do cliente é implementada por meio de linguagens de *script*, tais como JavaScript e as interfaces são baseadas na combinação de HTML e CSS;
- b) baseadas em *plugins* – processamento de eventos e atualização da tela são avançados por meio de *plugins Extensible Markup Language* (XML) interpretados no navegador *web*. Para o seu desenvolvimento são utilizadas tecnologias como Flash, Flex, Laszlo, entre outras;
- c) baseadas em *browser* – a interação com o usuário é nativamente suportada por alguns navegadores *web* que interpretam linguagens de definição de interface declarativas (*XML User Interface Language* (XUL));
- d) baseadas em tecnologias *desktop* – as aplicações são baixadas da *web* e executadas fora de um navegador *web* (Java Web Start, Window Smart Client, por exemplo).

Linguagens de programação como PHP podem ser utilizadas para gerar páginas HTML dinâmicas. Contudo, HTML por si só não permite o desenvolvimento de aplicações Internet ricas. Isso é possível por meio do uso de tecnologias adicionais como *Asynchronous JavaScript and XML* (AJAX) (PAVLIÆ; PAVLIÆ; JOVANOVIÆ, 2012). Para esses autores, a necessidade de usar várias tecnologias distintas torna o desenvolvimento de tais aplicações ineficientes em termos de tempo e custo, se comparado com o desenvolvimento de aplicações *desktop* tradicionais. Eles

ainda ressaltam que o principal problema está na necessidade de criar o lado servidor (implementado usando linguagens de programação para gerar HTML) separadamente do lado cliente (que é implementado usando tecnologias AJAX para dinamicamente atualizar apenas partes da tela e CSS para definir a interface gráfica com o usuário). E que uma das maneiras de resolver esse problema poderia ser pela adoção de HTML na sua versão 5.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados para a realização deste trabalho. Os materiais estão relacionados às tecnologias e ferramentas utilizadas e o método apresenta a sequência das principais atividades realizadas.

3.1 MATERIAIS

O Quadro 1 apresenta as ferramentas e as tecnologias que foram utilizadas para modelar e implementar o sistema.

Ferramenta / Tecnologia	Versão	Referência	Finalidade
Visual Paradigma	10.0	http://www.visual-paradigm.com/	Documentação da modelagem baseada na UML
Case Studio 2	2.25	http://www.casestudio.com	Modelagem do diagrama de entidades e relacionamentos do banco de dados
Linguagem PHP	PHP 5.6.6	http://php.net/	Linguagem de programação
NetBeans	8.0.2	https://netbeans.org/	Ambiente de desenvolvimento
MySQL	5	http://www.mysql.com/	Banco de dados
MySQL WorkBench	5.2 CE	http://www.mysql.com/products/workbench/	Administrador do banco de dados
phpMyAdmin		http://www.phpmyadmin.net/home_page/index.php	Gerenciamento do MySQL em aplicações <i>web</i>
Xampp	3.2.1	https://www.apachefriends.org/pt_br/	Servidor <i>web</i> para a aplicação
HTML	5.0.0	http://www.w3.org/TR/html5/	Desenvolvimento da interface da aplicação
CSS	3.0.0	http://www.w3.org/Style/CSS/Overview.en.html	Estilização da aplicação
Bootstrap	2.2	http://getbootstrap.com/	<i>Framework front-end</i>

Quadro 1 – Ferramentas e tecnologias utilizadas

3.2 MÉTODO

O levantamento e a modelagem dos requisitos foram realizados como atividade de estágio supervisionado pelos autores deste trabalho. Os requisitos foram revisados e complementados incluindo funcionalidades como uma área de acesso ao cliente e envio de *email*. Contudo, a ênfase deste trabalho esteve centrada na implementação do sistema. A implementação foi realizada utilizando as tecnologias apresentadas no Quadro 1.

4 RESULTADO

Este capítulo apresenta o resultado deste trabalho que é implementação de um sistema para registrar o agendamento e controlar o conteúdo relacionado a ele em treinamentos. A modelagem do sistema foi realizada como atividade de estágio pelos autores deste trabalho. A documentação da modelagem produzida consta aqui para facilitar o entendimento das funcionalidades do sistema.

4.1 ESCOPO DO SISTEMA

O sistema modelado como resultado deste trabalho automatizará o procedimento de registro e controle de conteúdo de treinamentos realizados por empresas de Software para os seus clientes. Como exemplos desses treinamentos estão videoconferências e visitas técnicas. A solução proposta considera o contexto apresentado a seguir.

Um treinamento pode ter a participação de mais de uma pessoa e pode ser ministrado por mais de um consultor. Cada treinamento é agendado com prévia antecedência levando em consideração a disponibilidade por parte do consultor e cliente. A duração do mesmo é pré-determinada pelo consultor, de acordo com o seu tipo e buscando seguir uma padronização, contudo podem-se abrir exceções quanto à duração dependendo do cliente. O conteúdo abordado em cada treinamento segue o módulo do sistema selecionado (Financeiro, Pedagógico), cada módulo já possui seu conteúdo parametrizado para melhor organização e melhor desempenho pelo consultor.

O consultor responsável pelo cliente faz os agendamentos e define juntamente com o mesmo os módulos a serem abordados em cada encontro. Após a finalização de cada treinamento o consultor será responsável por atualizar o registro do treinamento, descrevendo os acontecimentos ocorridos no encontro.

Assim que o treinamento for agendado, o consultor poderá encaminhar ao cliente uma mensagem eletrônica informando a data, o horário, o tipo do treinamento e seu módulo, além de informações sobre o treinamento, como dicas e sugestões.

4.2 MODELAGEM DO SISTEMA

O Quadro 2 apresenta a listagem dos requisitos funcionais identificados para o sistema.

Identificação	Nome	Descrição
RF01	Cadastrar Consultores	Responsáveis pelo agendamento e execução do treinamento.
RF02	Cadastrar Clientes	Os clientes precisam estar devidamente cadastrados para que os treinamentos possam ser agendados e para que possam receber sua confirmação via e-mail após cada agendamento.
RF03	Cadastrar Tipos de Treinamento	Necessário para identificar como o treinamento será realizado (<i>in-loco, online</i>).
RF04	Cadastrar configuração de e-mail	Necessário para encaminhar a notificação de agendamento ao cliente
RF05	Cadastrar Agendamentos	Os agendamentos (treinamentos) serão cadastrados e atualizados pelos consultores.
RF06	Cadastrar Módulos	Cadastro com os módulos de treinamento.
RF07	Cadastrar Usuários	Usuários com acesso ao sistema. Os usuários possuem permissões distintas. Administrador com acesso total ao sistema. Cada consultor terá acesso a todos os agendamentos aos quais está em seu nome (tipo de projeto).
RF08	Emitir relatório de Agendamentos	Emitir um relatório de agendamentos por período de data e/ou por consultor. Cada consultor poderá emitir o seu relatório com seus respectivos agendamentos.
RF09	Emitir Relatório de Clientes	Consultar dados dos clientes.

Quadro 2 – Requisitos funcionais

A listagem do Quadro 3 apresenta os requisitos não-funcionais identificados para o sistema como um todo, também denominados de requisitos suplementares. Os requisitos não funcionais explicitam regras de negócio, restrições ao sistema de acesso, requisitos de qualidade, desempenho e segurança, dentre outros.

Identificação	Nome	Descrição
RNF01	Acesso ao sistema	O acesso ao sistema será realizado por meio de <i>login</i> e senha.
RNF02	Agendamento	Cada consultor poderá visualizar apenas seus treinamentos, visitas, etc.

Quadro 3 – Requisitos não funcionais

A Figura 4 apresenta o diagrama de casos de uso. Nesse diagrama estão os três autores representados. O ator Administrador possui permissões de manter usuários, com o cadastro e exclusão de usuários no sistema. O ator Gerente realiza as funcionalidades de manter os cadastros do sistema, monitorar os agendamentos realizados e emitir relatórios de controle dos treinamentos agendados e realizados. O ator Consultor realiza as atividades relacionadas ao agendamento e ao registro de realização dos treinamentos. E o ator Cliente realiza a visualização dos treinamentos agendados.

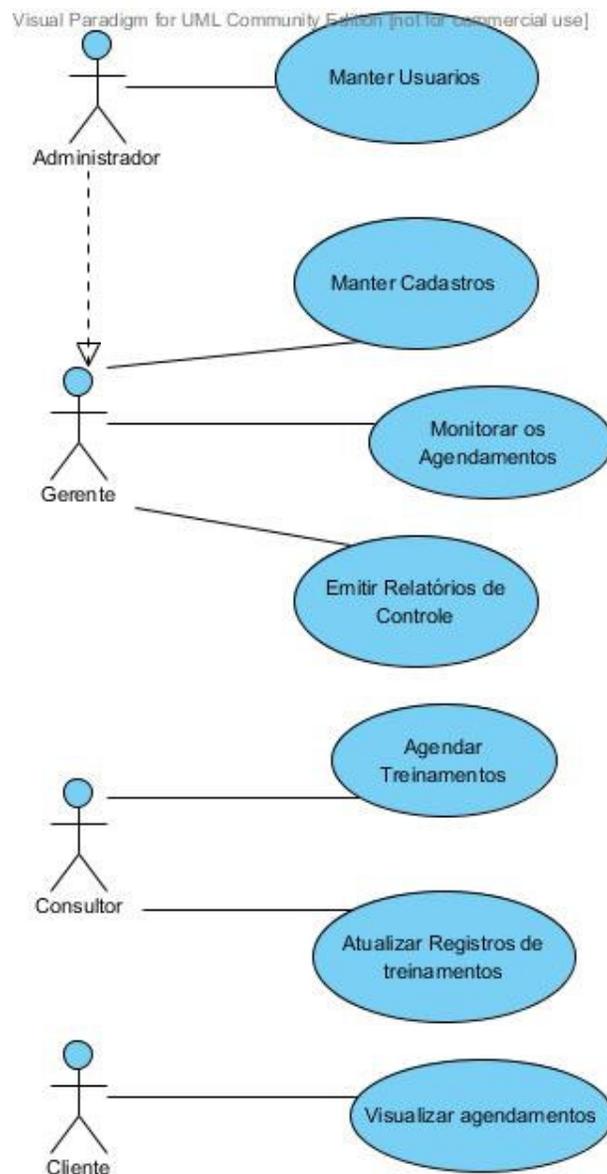


Figura 1 – Diagrama de casos de uso

A Figura 2 apresenta o diagrama de entidades e relacionamentos do banco de dados.

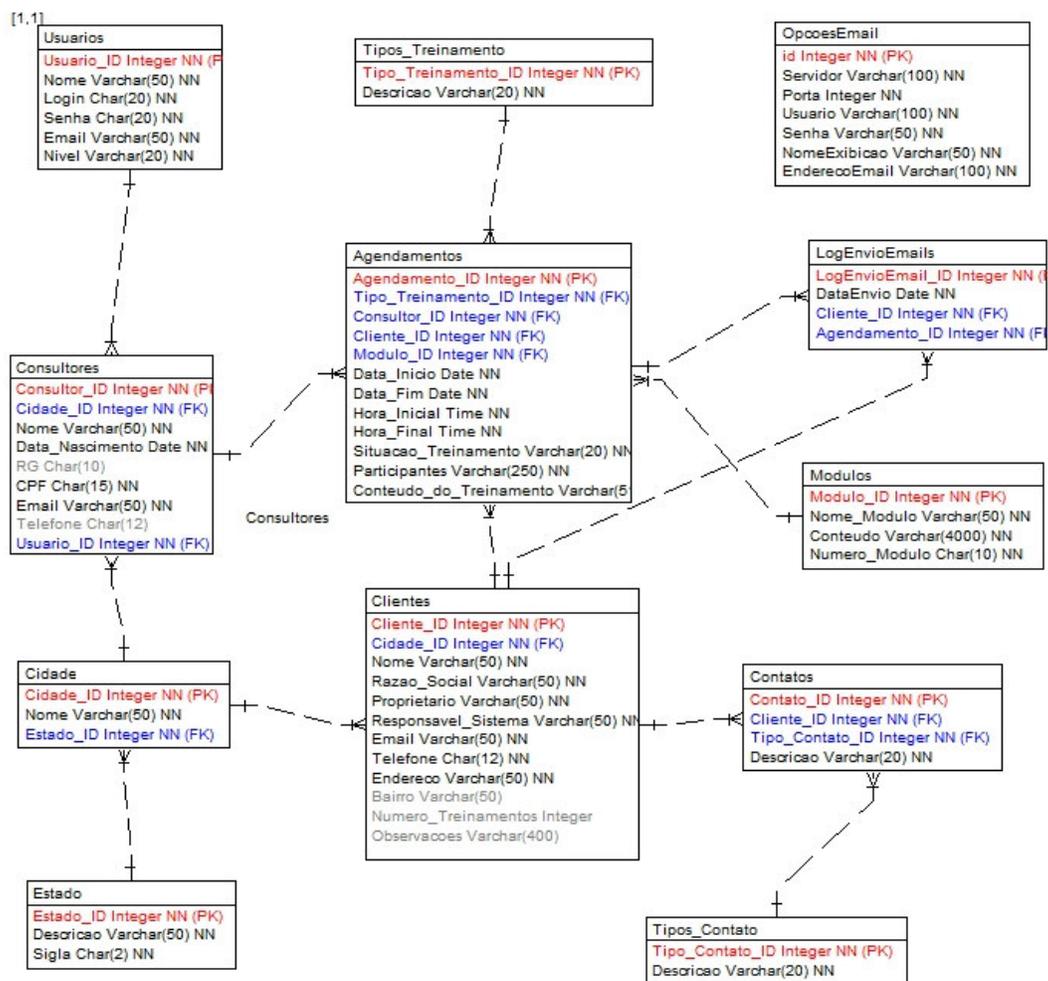


Figura 2 – Diagrama de entidades e relacionamentos

A seguir são descritas as tabelas apresentadas no diagrama da Figura 2. No Quadro 4 estão os campos da tabela de Usuários.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Usuario_id	Numérico	Não	Sim	Não	
Nome	Texto	Não	Não	Não	
Login	Texto	Não	Não	Não	
Senha	Texto	Não	Não	Não	
Email	Texto	Não	Não	Não	
Nivel	Texto	Não	Não	Não	

Quadro 4 – Campos da tabela Usuarios

O Quadro 5 apresenta os campos da tabela de Consultores. Os consultores serão responsáveis pelo agendamento e execução do treinamento.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Consultor_id	Numérico	Não	Sim	Não	
Usuario_id	Numérico	Não	Não	Sim	Da tabela Usuários
Cidade_id	Númérico	Não	Não	Sim	Da tabela Cidades
Nome	Texto	Não	Não	Não	
Data_Nascimento	Data	Não	Não	Não	
Rg	Texto	Sim	Não	Não	Preenchimento opcional
Cpf	Texto	Não	Não	Não	
E-mail	Texto	Não	Não	Não	
Telefone	Numérico	Sim	Não	Não	Preenchimento Opcional

Quadro 5 – Campos da tabela Consultores

O Quadro 6 apresenta os campos da tabela de OpcoesEmail.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira
id	Numérico	Não	Sim	Não
Servidor	Texto	Não	Não	Não
Porta	Numérico	Não	Não	Não
Usuario	Texto	Não	Não	Não
Senha	Texto	Não	Não	Não
Nome_Exibicao	Texto	Não	Não	Não
Endereco_Email	Texto	Não	Não	Não

Quadro 6 – Campos da tabela OpcoesEmail

O Quadro 7 apresenta os campos da tabela de Cidades.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Cidade_id	Numérico	Não	Sim	Não	
Estado_id	Numérico	Não	Não	Sim	Da tabela Estado
Nome	Texto	Não	Não	Não	

Quadro 7 – Campos da tabela Cidades

O Quadro 8 apresenta os campos da tabela de Estados.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira
Estado_id	Numérico	Não	Não	Sim
Descricao	Texto	Não	Não	Não
Sigla	Texto	Não	Não	Não

Quadro 8 – Campos da tabela Estados

O Quadro 9 apresenta os campos da tabela de Tipo_Treinamento.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira
Tipo_Treinamento_id	Numérico	Não	Sim	Não
Descricao	Texto	Não	Não	Não

Quadro 9 – Campos da tabela Tipo_Treinamento

O Quadro 10 apresenta os campos da tabela de Agendamentos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Agendamento_id	Numérico	Não	Sim	Não	
Tipo_Treinamento_id	Numérico	Não	Não	Sim	Da tabela Tipo_Treinamento. Define o tipo de treinamento a ser realizado.
Consultor_id	Numérico	Não	Não	Sim	Consultor Responsável
Cliente_id	Numérico	Não	Não	Sim	Da tabela Cliente
Modulo_ID	Númerico	Não	Não	Sim	Carrega o módulo e conteúdo do treinamento
Data_inicio	Data	Não	Não	Não	
Data_Fim	Data	Não	Não	Não	
Hora_Inicial	Time	Não	Não	Não	
Hora_Final	Time	Não	Não	Não	
Situacao_treinamento	Texto	Não	Não	Não	Situação real do treinamento.
Participantes	Texto	Não	Não	Não	
Conteudo_do_Treinamento	Texto	Não	Não	Não	Local de realização do treinamento.

Quadro 10 – Campos da tabela Agendamentos

O Quadro 11 apresenta os campos da tabela de Clientes.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Cliente_id	Numérico	Não	Sim	Não	
Cidade_id	Numérico	Não	Não	Sim	Da tabela Cidades
Nome	Texto	Não	Não	Não	
Razao_Social	Texto	Não	Não	Não	
Proprietario	Texto	Não	Não	Não	
Responsavel_Sistema	Texto	Não	Não	Não	Responsável pelo sistema por parte do cliente
E-mail	Texto	Não	Não	Não	
Telefone	Texto	Não	Não	Não	
Numero_Treinamentos	Numérico	Não	Não	Não	Número de treinamentos disponibilizados ao cliente
Endereco	Texto	Não	Não	Não	
Bairro	Texto	Não	Não	Não	
Observacoes	Texto	Não	Não	Não	Campo livre

Quadro 11 – Campos da tabela Clientes

O Quadro 12 apresenta os campos da tabela de Modulos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Modulo_id	Numérico	Não	Sim	Não	
NomeModulo	Texto	Não	Não	Não	Módulo do treinamento
Conteudo	Texto	Não	Não	Não	Conteúdo relacionado a cada módulo.
NumeroModulo	Texto	Não	Não	Não	

Quadro 12 – Campos da tabela Modulos

O Quadro 13 apresenta os campos da tabela de Tipo_Contato.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira
Tipo_Contato_id	Numérico	Não	Sim	Não
Descricao	Texto	Não	Não	Não

Quadro 13 – Campos da tabela Tipo_Contato

O Quadro 14 apresenta os campos da tabela de Contatos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Contato_id	Numérico	Não	Sim	Não	
Tipo_Contato_id	Numérico	Não	Não	Sim	Da tabela Tipo_Contato
Cliente_id	Numérico	Não	Não	Sim	Da tabela Clientes
Descricao	Texto	Não	Não	Não	

Quadro 14 – Campos da tabela Contatos

O Quadro 15 apresenta os campos da tabela de LogEnvioEmail.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira
LogEnvioEmailID	Numérico	Não	Sim	Não
DataEnvio	Date	Não	Não	Não
Cliente_ID	Numérico	Não	Não	Sim
Agendamento_ID	Numérico	Não	Não	Sim

Quadro 15 – Campos da tabela LogEnvioEmail

4.3 APRESENTAÇÃO DO SISTEMA

O sistema desenvolvido tem por objetivo auxiliar os consultores em relação ao controle dos agendamentos de treinamento, disponibilizados pelas empresas de software aos seus clientes por meio do processo de Implantação.

O leiaute do sistema é composto por três setores: o setor superior que contém o menu de navegação para acesso às ações do sistema; o setor central que irá apresentar o conteúdo da página que está sendo navegada; e o setor inferior que apresentará um rodapé no qual será exibido o usuário que está logado no sistema e o seu nível. No setor central está a área de “edição” dos formulários.

Todo o leiaute foi construído utilizando o *framework front-end* Bootstrap, o qual possibilita a criação de interfaces sem a necessidade de utilização das folhas de estilo (CSS), porém por mais completo que seja o *framework*, ainda sim é necessário realizar ajustes para completar a criação da interface. A Figura 3 apresenta a tela inicial do sistema na visão do administrador.

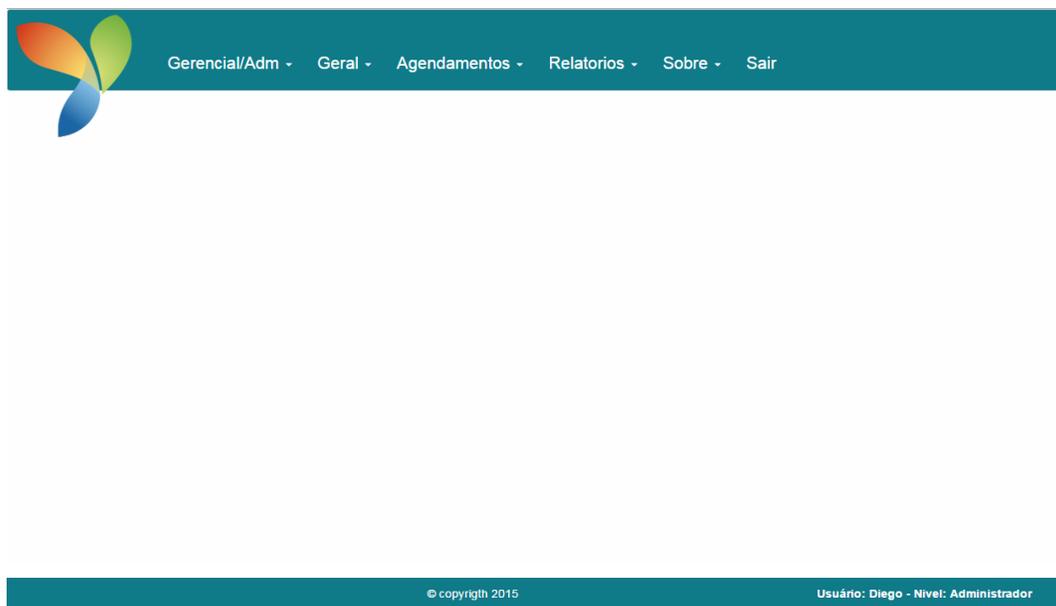


Figura 3 - Tela inicial do sistema

A divisão dos menus é feita em Gerencial/Adm. e Geral que contém cadastros básicos para o bom funcionamento do sistema, agendamentos e relatórios. A visibilidade das opções de menu depende do nível de acesso do usuário logado que podem ser Administrador, Consultor ou Gerente.

A Figura 4 exibe a tela utilizada pelos usuários para acesso ao sistema. Possuindo *login* e senha previamente cadastrados, os usuários terão acesso às funcionalidades de acordo com seu respectivo nível. Por meio da mesma tela, o cliente poderá consultar seus agendamentos. O *login* para acesso à essa área é feito pelo *link* “Área do Cliente” localizado abaixo do botão Entrar.



Figura 4 - Tela de login

Os *grids* das telas de listagem utilizam em sua estrutura o ‘TableSorter’, que é um *plugin* do JQuery responsável por ordenar as tabelas por colunas. Para isto basta criar o cabeçalho da coluna e ao clicar no mesmo, todos os registros são ordenados, de maneira crescente ou decrescente.

Como na maioria dos sistemas, o passo inicial para começar a sua utilização é a definição dos usuários e suas respectivas senhas, identificando por meio do seu cadastro qual o seu nível de acesso dentro do sistema. Esta definição é realizada exclusivamente pelo usuário administrador e com isto é possível criar um nível de organização dentro do setor, impedindo que usuários acessem informações indevidamente.

A Figura 6 apresenta esse cadastro que possui informações objetivas para o cadastro de um usuário, como nome, *login*, senha e o nível de acesso. Utilizando o campo “Confirma Senha” apenas para fins de validação, não sendo realizada a gravação dessa informação no banco de dados.

Gerencial/Adm - Geral - Agendamentos - Relatorios - Sobre - Sair

Novo - Usuário

Registros Cadastrados

Nome

Login

Senha

Confirmar Senha

E-mail

Administrador
 Gerente
 Consultor

© copyright 2015 Usuário: Diego - Nivel: Administrador

Figura 5 - Cadastro de usuário

Após preencher o cadastro e clicar no botão ‘Salvar’ o usuário é direcionado para um outro formulário. Nesse formulário são exibidos, em um *grid*, os registros já cadastrados, assim como as ações de ‘Edição’ e ‘Exclusão’. A Figura 7 mostra a tela de exibição dos registros.

Gerencial/Adm - Geral - Agendamentos - Relatorios - Sobre - Sair

Usuários

[Novo Usuário](#)
[Visualizar Relatório](#)

Pesquisar

	Nome	Login	E-mail	Ação
<input type="checkbox"/>	Julian Dalla Corte M	julian	juliandallacorte@gmail.com	
<input type="checkbox"/>	Mariane	mari	mariane@teste.com.br	
<input type="checkbox"/>	Diego Viero	Diego	diego@gmail.com	

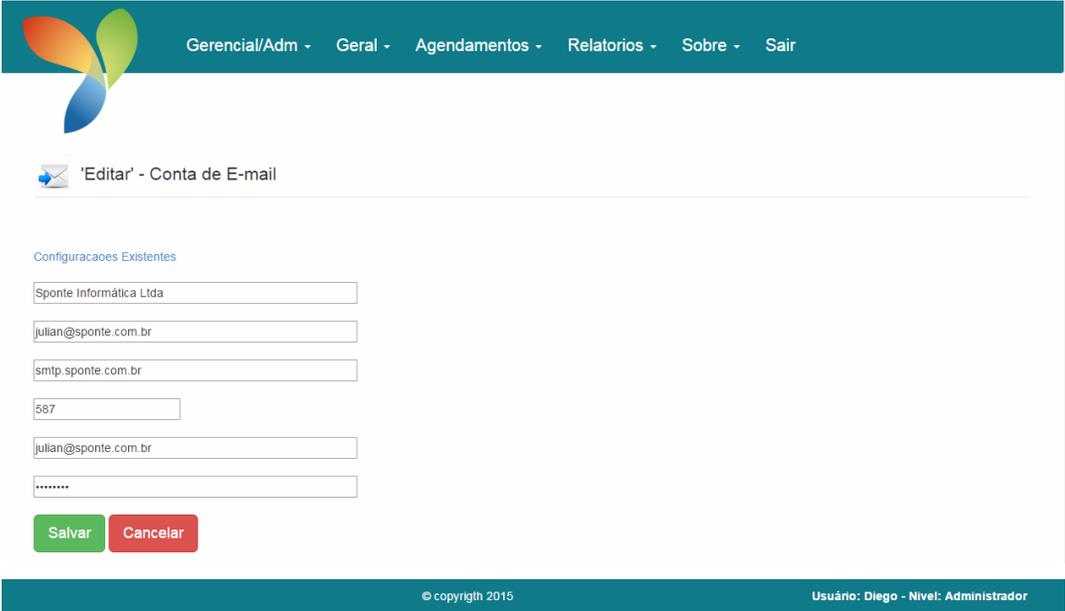
Exibir 10 Registros

© copyright 2015 Usuário: Diego - Nivel: Administrador

Figura 6 - Listagem dos registros de usuários

Acima do Grid o usuário pode efetuar o cadastro de um novo registro acessando o link “Novo Usuário” e acessar o relatório de usuários por meio do botão “Visualizar Relatório”.

O Administrador será o responsável também pela configuração da conta de e-mail que será utilizada pelos consultores no envio de e-mails aos clientes após a realização do agendamento de treinamento. Esta configuração poderá ser acessada através do menu Gerencial/Adm opção Mailing. A Figura 8 exibe a tela de configuração.



The screenshot shows a web application interface for editing an email account. At the top, there is a teal navigation bar with a logo on the left and menu items: 'Gerencial/Adm', 'Geral', 'Agendamentos', 'Relatorios', 'Sobre', and 'Sair'. Below the navigation bar, the page title is 'Editar' - Conta de E-mail. The main content area is titled 'Configuracoes Existentes' and contains several input fields: 'Sponte Informática Ltda', 'julian@sponte.com.br', 'smtp.sponte.com.br', '587', 'julian@sponte.com.br', and a password field with asterisks. At the bottom of the form are two buttons: 'Salvar' (green) and 'Cancelar' (red). The footer of the page is teal and contains the text '© copyright 2015' and 'Usuário: Diego - Nivel: Administrador'.

Figura 7 - Configuração de e-mail

No *grid* de agendamentos constará o link de envio que ao ser acionado enviará automaticamente um e-mail padrão ao cliente, constando os dados do treinamento como dia, horário, consultor e algumas informações sobre o processo.

O Gerente será responsável por efetuar e manter o restante dos cadastros do sistema atualizados. O seu usuário somente não terá acesso aos usuários e a configuração da conta de e-mail que são específicos do administrador.

Um dos principais cadastros é o de módulos, que inicialmente ao ser acessado exibe ao usuário uma mensagem informativa, pois ainda não possui nenhum registro cadastrado. A Figura 8 exibe esta tela quando é acessada pela primeira vez.

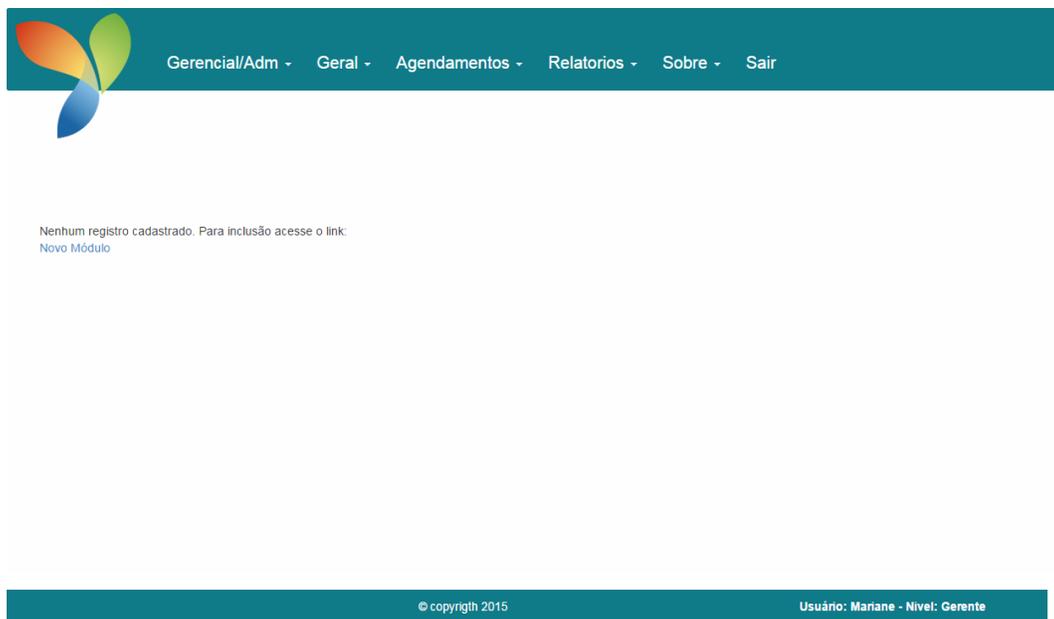


Figura 8 - Cadastro de módulo - sem registros

Acionando o link “Novo Módulo”, exibido na Figura 8, o usuário é direcionado para um novo formulário, dessa vez, contendo os campos a serem preenchidos. Para o cadastro de módulos é necessário informar três campos: número do módulo, nome/descrição e conteúdo. As informações desse formulário são necessárias e relevantes para o agendamento do treinamento pelo consultor, portanto, o usuário deve alimentar os registros de módulos antes mesmo de realizar o agendamento. A Figura 9 exibe a tela de cadastro dos módulos.

Gerencial/Adm - Geral - Agendamentos - Relatorios - Sobre - Sair

'Novo' - Módulo

Registros Cadastrados

Nº Módulo

Nome do Módulo

Conteúdo do Módulo

Salvar Cancelar

© copyright 2015 Usuário: Mariane - Nivel: Gerente

Figura 9 - Cadastro de módulos

Assim como no cadastro de usuários, ao submeter o formulário na ação do botão “Salvar”, o usuário é direcionado para um outro formulário contendo o *grid* carregado com os registros. A Figura 10 exibe a tela de listagem.

Gerencial/Adm - Geral - Agendamentos - Relatorios - Sobre - Sair

Módulos

[Novo Módulo](#) [Visualizar Relatório](#)

Pesquisar

	Nº Módulo	Nome	Conteúdo	Ação
<input type="checkbox"/>	A	Gerencial	Usuários Permissões Configurações Iniciais	
<input type="checkbox"/>	B	Pedagógico	Impressão de Boletim Lançamento de Notas	

1/1

Editar 10 Registros

© copyright 2015

Usuário: Mariane - Nivel: Gerente

Figura 10 – Listagem dos registros dos módulos

Na coluna ‘ação’ do Grid, acionando o link Editar um novo formulário é carregado com os dados para que o usuário possa realizar as modificações.

Gerencial/Adm - Geral - Agendamentos - Relatorios - Sobre - Sair

'Editar' - Módulos

Registros Cadastrados

A

Gerencial

Conteúdo do Módulo

Usuários
Permissões
Configurações Iniciais

Salvar Cancelar

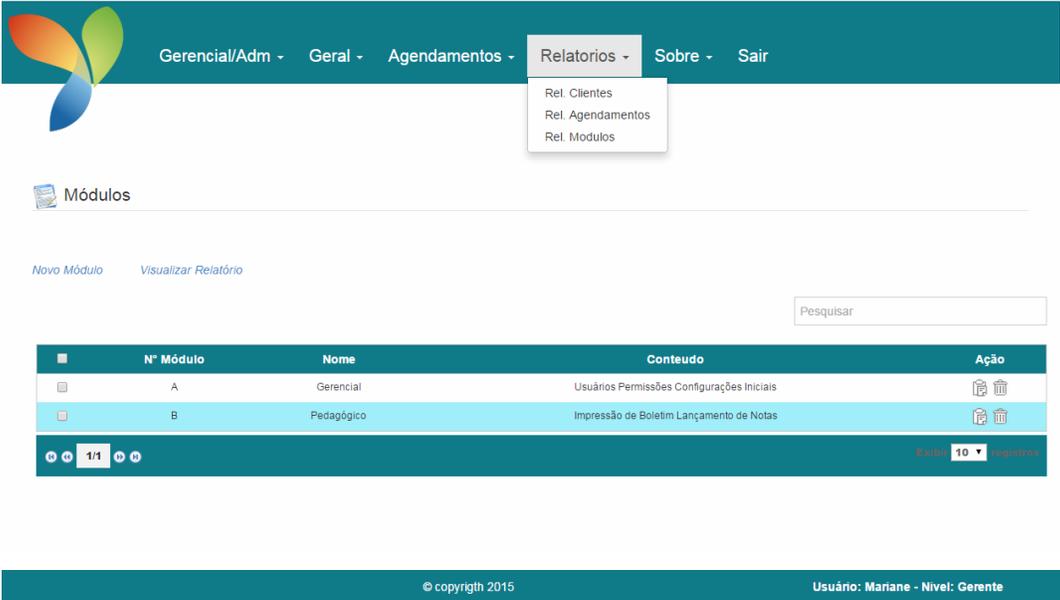
© copyright 2015

Usuário: Mariane - Nivel: Gerente

Figura 11 - Edição de módulos

Após realizado as modificações e salvando-as o usuário é novamente direcionado para o *grid* de listagem.

O gerente, assim como o administrador, poderão visualizar as informações cadastradas por meio de relatórios. A Figura 12 exibe o menu de acesso aos relatórios, bastando que o usuário selecione o relatório desejado para a geração de um arquivo no formato *Portable Document Format* (PDF).



The screenshot shows a web application interface. At the top, there is a navigation menu with the following items: Gerencial/Adm, Geral, Agendamentos, Relatorios, Sobre, and Sair. The 'Relatorios' menu is open, showing three options: Rel. Clientes, Rel. Agendamentos, and Rel. Modulos. Below the menu, there is a section titled 'Módulos' with a search bar labeled 'Pesquisar'. A table lists two modules:

Nº Módulo	Nome	Conteúdo	Ação
A	Gerencial	Usuários Permissões Configurações Iniciais	[Print] [Delete]
B	Pedagógico	Impressão de Boletim Lançamento de Notas	[Print] [Delete]

At the bottom of the page, there is a footer with the text '© copyright 2015' and 'Usuário: Mariane - Nivel: Gerente'.

Figura 12 – Relatórios

A Figura 13 exemplifica o leiaute do relatório de agendamentos. Esse formato exibe os registros em um quadro e sem nenhum tipo de filtro.



The screenshot shows a report titled 'Relatorio de Agendamentos'. The report contains a table with the following columns: Consultor, Cliente, Modulo, Data, and Horario. The data is as follows:

Consultor	Cliente	Modulo	Data	Horario
juca balero	CNA Urupema	Gerencial	2015-06-24	19:00:00 as 20:00:00
juca balero	CNA Urupema	Gerencial	2015-07-02	17:00:00 as 18:00:00
juca balero	CNA Urupema	Pedagógico	2015-06-30	13:00:00 as 14:40:00

Figura 13 - Relatório de agendamento

O usuário poderá visualizar o relatório de agendamentos mais específico através de filtros que podem ser acessados pelo link 'relatório de agendamentos' localizado na tela de listagem dos registros de treinamentos. A Figura 14 exibe a tela de filtros para o relatório.

Gerencial/Adm - Geral - Agendamentos - Relatorios - Sobre - Sair

'Filtros'

[Retornar a Listagem](#)

Periodo

dd/mm/aaaa a dd/mm/aaaa

Código Cliente

Informe o código do cliente

Consultor

Módulo

Visualizar Limpar Filtros

© copyright 2015 Usuário: Mariane - Nivel: Gerente

Figura 14 – Relatório filtros

Os filtros possibilitam ao usuário ter o acesso às informações mais detalhadas sobre os agendamentos feitos pelos consultores.

Agendamentos é a principal funcionalidade do sistema que se refere ao controle dos agendamentos de treinamento. Este recurso será utilizado basicamente pelos consultores. A tela principal desse módulo é composta pelo *grid* de exibição dos registros já existentes. Nesse *grid* é possível ter acesso a edição e exclusão dos mesmos, assim como o envio do *e-mail* para o cliente. A Figura 16 exibe a tela de listagem na visão do consultor.

Agendamentos - Sair

Agendamentos

[Novo Agendamento](#) [Relatório de Agendamentos](#)

Pesquisar

ID	Cliente	Módulo	Consultor	Tipo Treinamento	Data Inicial	Horário Inicial	Ação
8	CNA Urupema	Gerencial	Julian Dalia Corte Matuchak	OnLine via Skype	2015-06-24	19:00:00	
10	CNA Urupema	Gerencial	Julian Dalia Corte Matuchak	Video Conferência	2015-07-02	17:00:00	
9	CNA Urupema	Pedagógico	Julian Dalia Corte Matuchak	Video Conferência	2015-06-30	13:00:00	

Exibir 10 registros

© copyright 2015 Usuário: Julian - Nivel: Consultor

Figura 15 - Grid de agendamentos – visão consultor

Cada consultor terá acesso apenas aos seus agendamentos não sendo possível visualizar os registros de outros consultores.

Acionando o *link* “Novo Agendamento” o usuário será direcionado para a tela de cadastro de um novo agendamento exibido na Figura 17.

'Novo' - Agendamento

Registros Cadastrados

Cliente:
 Tipo do Treinamento:
 Consultor:
 Módulo:

Data: a
 Horário: as

Situação

Confirmado
 Pendente
 Cancelado

Participantes do Treinamento

Conteúdo do Treinamento

© copyright 2015 Usuário: Julian - Nivel: Consultor

Figura 16 - Cadastro de agendamentos – visão consultor

As informações contidas no cadastro de agendamentos serão de comum acordo com o cliente. Elas serão preenchidas pelo consultor no ato do agendamento e estarão

visíveis ao cliente por meio do seu acesso, porém não poderão ser editadas ou excluídas. A Figura 18 exibe a visão do cliente quanto aos seus agendamentos.

Agendamentos - Sair

Agendamentos

Orientações para realização dos treinamentos

Pesquisar

Cliente	Módulo	Consultor	Tipo Treinamento	Data Inicial	Horário Inicial	Ação
Wizard	Gerencial	Julian Dalla Corte Matuchak	OnLine via Skype	2015-07-02	20:00:00	
Wizard	Pedagógico	Julian Dalla Corte Matuchak	Visita inLoco	2015-07-08	16:00:00	

1/1

10

© copyright 2015

Cliente: Wizard

Figura 17 – Agendamentos – visão cliente

Ao acessar a imagem da lupa na coluna ação do *grid*, o cliente poderá visualizar os dados do agendamento.

Agendamento

Tipo do Treinamento: OnLine via Skype

Consultor: Julian Dalla Corte Matucha

Módulo: Gerencial

Data: 02/07/2015 a 03/07/2015

Horario: 20:00 as 21:00

Situação

Confirmado Pendente Cancelado

Paulo

Permissões

Voltar

© copyright 2015

Cliente: Wizard

Figura 18 – Visualização agendamento – visão cliente

Para evitar qualquer ação imprópria do cliente que não seja a visualização dos agendamentos, quando o agendamento é editado os campos são exibidos bloqueados para edição.

4.4 IMPLEMENTAÇÃO DO SISTEMA

Como apresentado na Seção 4.3, o leiaute do sistema é composto por três setores, isso é feito por meio do uso do HTML5 e da utilização do *framework front-end* Bootstrap, no qual é possível desenvolver todo um leiaute sem a necessidade de digitar código CSS.

Para isso é necessário a importação do arquivo que contém todas as informações do Bootstrap para dentro do projeto e informar nas páginas quais o mesmo será utilizado o caminho do arquivo “bootstrap.css” que contém as classes que serão utilizadas.

Para facilitar o desenvolvimento, foi utilizada uma página modelo, chamada “home.php”. Essa página contém o layout básico do sistema, que será utilizado por todas as outras páginas. O código dessa página é apresentado na Listagem 1.

```

<?php
session_start();
if (isset($_SESSION["Nome"])) {
    $nomeCompleto = $_SESSION["Nome"];
    $primeironome = explode(" ", $nomeCompleto);
    $nome = $primeironome[0];
    $oneNome = $nome;
}
?>
<!DOCTYPE html>
<html>
<script language = "JavaScript">
    function confirmar()
    {
        if (!confirm("Deseja Realmente sair do sistema?")) {
            history.go(-1);
            return " "
            document.write(confirmar())
        }
        else
            return true;
    }
</script>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title></title>
<link rel="stylesheet" type="text/css"
href="bootstrap/css/bootstrap.css">
<link rel="stylesheet" type="text/css" href="css/estilo.css">
<script src="js/JQuery 1.11.1.js"></script>
<script src="bootstrap/js/bootstrap.js"></script>
</head>
<body>
<header>
<nav class="navbar navbar-default">
<div class="navbar-header">
<a class="navbar-brand" href="Home.php"></a>
</div>
<?php
if ($_SESSION["Nivel"] == "Administrador") {
?>

```

```

<div class="collapse navbar-collapse">
  <ul class="nav navbar-nav">

    <!--Primeiro Menu-->
    <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Gerencial/Adm
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
    <li><a href="ListagemUsuarios.php" title="Cadastro de Usuários do
Sistema">Usuários</a></li>
    <li><a href="ListagemConsultores.php" title="Cadastro de
Consultores">Consultores</a></li>
    <li><a href="ListagemClientes.php" title="Cadastro de
Clientes">Clientes</a></li>
    <li><a href="ListagemCidades.php" title="Cadastro de
Cidades">Cidades</a></li>
    <li><a href="ListagemEstados.php" title="Cadastro de
Estados">Estados</a></li>
    <li><a href="ListagemMailling.php"
title="Mailling">Mailling</a></li>
  </ul>
</li>
    <!--Segundo Menu-->
    <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Geral
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
    <li><a href="ListagemTiposTreinamento.php" title="Cadastro de Tipos
de Treinamento">Tipos de Treinamento</a></li>
    <li><a href="ListagemTiposContato.php" title="Cadastro de Tipos de
Contato">Tipos de Contato</a></li>
    <!--<li><a href="CadastroTiposTreinamento.php">Novo Tipo
Treinamento</a></li-->
    <li><a href="ListagemModulos.php" title="Cadastro de Módulos de
Treinamento">Módulos de Treinamento</a></li>
  </ul>
</li>
    <!--Terceiro Menu-->
    <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Agendamentos
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
    <li><a href="ListagemAgendamentos.php" title="Agendamento de
Treinamentos">Treinamento</a></li>
  </ul>
</li>

    <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Relatorios
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
    <li><a href="Relatorios/RelatorioClientes.php">Rel. Clientes
</a></li>
    <li><a href="Relatorios/RelatorioGeralAgendamentos.php">Rel.
Agendamentos </a></li>
    <li><a href="Relatorios/RelatorioModulos.php">Rel. Modulos
</a></li>
  </ul>
</li>

    <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Sobre
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
    <li><a href="SobreOSistema.php">O Sistema </a></li>
  </ul>
</li>
    <li><a onclick='confirmar()' href="Login/index.php">Sair</a></li>

```

```

    </ul>
  </div>
  <?php
  } else if ($_SESSION["Nivel"] == "Consultor") {
  ?>
  <div class="collapse navbar-collapse">
  <ul class="nav navbar-nav">

  <!--Terceiro Menu-->
  <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Agendamentos
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
  <li><a href="ListagemAgendamentos.php" title="Listagem de
Treinamentos">Treinamento</a></li>
  </ul>
  </li>

  <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Relatorios
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
  <li><a href="Relatorios/RelatorioClientes.php">Rel. Clientes
</a></li>
  </ul>
  </li>

  <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Sobre
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
  <li><a href="SobreOSistema.php">O Sistema </a></li>
  </ul>
  </li>
  <li><a onclick='confirmar()' ref="Login/index.php">Sair</a></li>
  </ul>
  </div>
  <?php
  } else if ($_SESSION["Nivel"] == "Gerente") {
  ?>
  <div class="collapse navbar-collapse">
  <ul class="nav navbar-nav">

  <!--Primeiro Menu-->
  <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Gerencial/Adm
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
  <li><a href="ListagemConsultores.php" title="Cadastro de
Consultores">Consultores</a></li>
  <li><a href="ListagemClientes.php" title="Cadastro de
Clientes">Clientes</a></li>
  <li><a href="ListagemCidades.php" title="Cadastro de
Cidades">Cidades</a></li>
  <li><a href="ListagemEstados.php" title="Cadastro de
Estados">Estados</a></li>
  </ul>
  </li>

  <!--Segundo Menu-->
  <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Geral
  <span class="caret"></span></a>
  <ul class="dropdown-menu">
  <li><a href="ListagemTiposTreinamento.php" title="Cadastro de
Tipos de Treinamento">Tipos de Treinamento</a></li>
  <li><a href="ListagemTiposContato.php" title="Cadastro de Tipos
de Contato">Tipos de Contato</a></li>

```

```

        <!--<li><a href="CadastroTiposTreinamento.php">Novo Tipo
Treinamento</a></li>-->
        <li><a href="ListagemModulos.php" title="Cadastro de Módulos de
Treinamento">Módulos de Treinamento</a></li>
        </ul>
    /li>
    <!--Terceiro Menu-->
    <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Agendamentos
    <span class="caret"></span></a>
    <ul class="dropdown-menu">
        <li><a href="CadastroAgendamentos.php" title="Agendamento de
Treinamentos">Treinamento</a></li>
    </ul>
    </li>

    <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Relatorios
    <span class="caret"></span></a>
    <ul class="dropdown-menu">
        <li><a href="Relatorios/RelatorioClientes.php">Rel. Clientes
</a></li>
    </ul>
    </li>

    <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="">Sobre
    <span class="caret"></span></a>
    <ul class="dropdown-menu">
        <li><a href="SobreOSistema.php">O Sistema </a></li>
    </ul>
    </li>
    <li><a onclick='confirmar()' href="Login/index.php">Sair</a></li>
</ul>
</div>
<?php
}
?>
</nav>
</header>
</body>
<footer class="panel-footer">
    © copyrigh 2015 <label>Usuário: <b><?php echo $oneNome . " -
Nivel: " . $_SESSION["Nivel"] ?></b></label>
</footer>
</html>

```

Listagem 1 – Código da página inicial (home)

Nas aplicações *web*, desenvolvidas em PHP, são frequentemente utilizados métodos para transporte de variáveis entre as páginas, são eles o `$_GET` e `$_POST`.

O método `$_GET` é utilizado para receber variáveis por meio da *Uniform Resource Locator* (URL) que é o link que insere no *browser*, pode-se evidenciar o uso deste método quando a URL estiver assim, por exemplo: `http://localhost/TCC_Projeto2/Function_EditarUsuario.php?Usuario_ID=15`. O ‘?’ presente no link é a forma de declarar que a seguir são variáveis. Podem ser declaradas mais de uma variável apenas separando-as com ‘&’ antes de cada uma delas. Esse é um

método limitado e não é adequado para grandes valores de variáveis, tendo um limite de até 2000 caracteres.

Já o método `$_POST` transfere esses dados de forma invisível entre os formulários, tendo como vantagem a não exibição das variáveis na URL, além de não ter limite no tamanho das mesmas.

Para cada operação básica de um *Create, Read, Update, Delete* (CRUD), existirá um arquivo PHP que contém todo o código responsável por executar a ação no sistema. Para que isso aconteça cada PHP é referenciado na página na qual a ação será realizada, ou seja, o arquivo que contém a ação de “gravar”, o registro é chamado ao submeter o formulário na tela de cadastro. Os arquivos de “Update” e “Delete” são referenciados em seus respectivos *links* localizados no *grid* que lista os registros.

Na Listagem 2 é possível observar a estrutura do código responsável por inserir o registro no banco de dados, quando o usuário efetuar a ação de salvar.

```
<?php
$Nome = $_POST["txtNome"];
$Login = $_POST["txtLogin"];
$Senha = $_POST["txtSenha"];
$Email = $_POST["txtEmail"];
$Situacao = $_POST["chkSituacao"];

$con = mysqli_connect("localhost", "root", "", "bancoestagio") or
die("Erro: " . mysqli_connect_error());
$sql = "INSERT INTO usuarios (Usuario_ID, Nome, Login, Senha, Email,
Ativo )
VALUES ( NULL , '$Nome', '$Login', '$Senha', '$Email',
'$Situacao')";
mysqli_query($con, $sql) or die("Erro: " . mysqli_error($con));

?>
<?php
sleep(2);
?>
<?php
mysqli_close($con);

header("Location: ListagemUsuarios.php");

?>
```

Listagem 2 – Insert usuários

O código da Listagem 2 exibe a estrutura do *insert* realizado para a inclusão de um novo registro de usuários. Todos os dados de entrada informados pelo usuário serão primeiramente armazenados em uma variável, que geralmente possui o mesmo nome do campo, por meio da função `$_POST` do PHP. Assim torna-se mais fácil a manipulação desses dados na sequência do código.

Com os dados armazenados, a próxima etapa é realizar a conexão com o banco de dados da aplicação e para isto é utilizado a função `MySqli_Connect`, presente na classe `MySqli`. Essa classe é uma extensão do PHP que visa aproveitar os recursos avançados do MySQL.

Como se trata de uma conexão local o endereço do banco será a própria máquina, sendo referenciada pelo nome 'localhost'. Na sequência é passado o nome de usuário, criado no início da configuração do MySQL por meio do Xampp que por padrão será 'root' e a senha, caso não exista pode ser deixado em branco. Por fim, é informado o nome do banco de dados a ser utilizado, todas estas informações são armazenadas na variável `$con`.

Após realizar a conexão com o banco de dados, será armazenado na variável `$sql` o comando *insert* responsável por inserir os dados no banco. Logo em seguida é utilizada a função `mysqli_query`, que recebe como primeiro parâmetro o ponteiro de conexão e em seguida vem a *query* (comando SQL). Por fim, a conexão é fechada com o uso da função `MySqli_Close` que recebe como parâmetro o ponteiro de conexão criada anteriormente.

Na Listagem 3 é exibido o código responsável pela exclusão de um registro do banco de dados. A estrutura desse código é semelhante ao *insert*, porém com alguns parâmetros a menos. Nele o valor armazenado no campo 'Usuario_ID', que é chave primária da tabela de usuários, é armazenado em uma variável chamada `$id` por meio do método `$_GET`, que passa o valor de suas variáveis pela URL dinâmica ou parâmetros de URL.

```
<?php
$id = $_GET["Usuario_ID"];
$con = mysqli_connect("localhost", "root", "", "bancoestagio") or die("Erro: " . mysqli_connect_error());
mysqli_query($con, "delete from Usuarios where Usuario_ID = $id") or die("Erro: " . mysqli_error($con));

mysqli_close($con);
header("Location: ListagemUsuarios.php");
?>
```

Listagem 3 – Código para exclusão de usuários

Para edição dos registros, ao acionar o link 'Editar' localizado no *grid* de exibição dos registros, o usuário é direcionado para um novo formulário que será responsável por realizar as ações de edição. Essas ações estão implementadas no arquivo `Function_SalvarEdicaoUsuario.php` e são disparadas pela submissão do formulário. A Listagem 4 exibe o código de edição.

```

<?php
$id = $_POST["id"];
$Nome = $_POST["txtNome"];
$Login = $_POST["txtLogin"];
$Senha = $_POST["txtSenha"];
$email = $_POST["txtEmail"];
/*$Situacao = $_POST["chkSituacao"];*/

$con = mysqli_connect("localhost", "root", "", "bancoestagio") or die("Erro: " . mysqli_connect_error());
$sql = "UPDATE Usuarios SET Nome = '$Nome', Login = '$Login', Senha = '$Senha', Email = '$Email' WHERE Usuario_ID = $id";
mysqli_query($con, $sql) or die("Erro: " . mysqli_error($con));

mysqli_close($con);

header("Location: ListagemUsuarios.php");
?>

```

Listagem 4 - Edição de usuários

A página de gerenciamento do cadastro conterá alguns recursos que complementam todos os processos do CRUD que representa as operações de inclusão, consulta, atualização e exclusão realizadas em banco de dados, geralmente realizadas por meio de formulários de cadastro. É nela que serão exibidos e manipulados os dados inseridos pelo usuário. A linguagem JavaScript também é utilizada na aplicação, principalmente em processos de validação e confirmação.

Como exemplo, na Listagem 5 está o código da função responsável por exibir ao usuário uma mensagem de confirmação quando o mesmo tentar excluir um registro. O recurso também serve como um esquema de segurança evitando que registros sejam excluídos indevidamente.

```

<script language = "JavaScript">
function confirmacao(id) {
    var resposta = confirm("Deseja remover esse registro?");
    if (resposta == true) {
        window.location.href =
"Function_ExcluirUsuario.php?Usuario_ID=" + id
    }
}
</script>

```

Listagem 5 – Código para mensagem de confirmação de exclusão

Na Listagem 6 está o trecho de código que é referenciado nas páginas que contêm os métodos de edição e exclusão e a chamada da função de confirmação de exclusão através do onClick do link 'Excluir'.

```

<td><a
href=\"Function_EditarUsuario.php?Usuario_ID=$codigo\">[Editar]</a>
| "
. "<a onClick='confirmacao($codigo)'
href='javascript:func();'>[Excluir]</a></td>
</tr>\n";

```

Listagem 6 – Chamada dos métodos/funções

O trecho de código apresentado na Listagem 7 verifica se há algum registro cadastrado para exibição no *grid* por meio de um *select* que é armazenado na variável “\$Resultado”. Em seguida é verificado se o número de linhas contido em “\$Resultado” é maior que 1. Se for é exibido o *grid* com o registro; caso não seja, uma mensagem informativa é apresentada ao usuário, juntamente com o *link* para acesso a tela de cadastro.

```

<?php
include './php/AbreConnexao.php';
$resultado = mysqli_query($con, "select * from usuarios") or
die("Erro: " . mysqli_error($con));
mysqli_close($con);

if (mysqli_num_rows($resultado) < 1) {

    echo "<fieldset> Nenhum registro cadastrado. Para inclusão acesse
o link: <br> "
    . "<a href='CadastroUsuarios.php'>Novo Usuario</a>"
    . "</fieldset>";
    exit;
}
?>

```

Listagem 7 – Verifica se há registros no banco de dados

O código da Listagem 8 é responsável pelo carregamento dos registros no *grid*, por meio de um *loop* que armazena em uma variável “\$Registro” o *array* de itens armazenado em “\$Resultado” e incrementa nas variáveis \$Codigo e \$descricao o seu conteúdo. Em seguida, a função `mysqli_free_result()` libera a memória associada ao resultado.

```

<?php
while ($registro = mysqli_fetch_array($resultado)) {
    $codigo = $registro["Usuario_ID"];
    $Nome = $registro["Nome"];
    $Login = $registro["Login"];
    $Senha = $registro["Senha"];
    $Email = $registro["Email"];

    echo "<tr>"
?>
<td><input type="checkbox" value="1" name="marcar[]" /></td>
<?php
echo "<td>$codigo</td>"

```

```

<td>$Nome</td>
<td>$Login</td>
<td>$Email</td>

    <td><a
href=\"Function_EditarUsuario.php?Usuario_ID=$codigo\">[Editar]</a>
| "
    . "<a onClick='confirmacao($codigo)'
href='javascript:func();'>[Excluir]</a></td>
    </tr>\n";
}
mysqli_free_result($resultado);
?>

```

Listagem 8 – Código para carregamento de dados no *grid*

A estrutura deste cadastro segue a mesma forma de implementação utilizada para o cadastro de usuários, passando pelo armazenamento dos dados informados pelo usuário em variáveis, abrindo a conexão com o banco de dados, criando o *insert* dos dados no banco, fechando a conexão e direcionando para a tela de exibição dos registros. A Listagem 9 apresenta o código de inserção de módulos.

```

<?php

$NumModulo = $_POST["txtNumModulo"];
$Nome = $_POST["txtNomeModulo"];
$Conteudo = $_POST["txtConteudo"];

$con = mysqli_connect("localhost", "root", "", "bancoestagio") or die("Erro: "
. mysqli_connect_error());
$sql = "INSERT INTO modulos (Modulo_ID, NomeModulo, Conteudo, NumeroModulo)
VALUES ( NULL, '$Nome', '$Conteudo', '$NumModulo)";
mysqli_query($con, $sql) or die("Erro: " . mysqli_error($con));
?>

<?php

sleep(2);
?>

<?php

mysqli_close($con);

header("Location: ListagemModulos.php");
?>

```

Listagem 9 – Insert módulos

FPDF é uma classe PHP que facilita a geração de documentos PDF de forma dinâmica. Além disso, é possível realizar toda a configuração do documento como, formato das páginas, cabeçalhos e rodapé, suportando também imagens no formato JPEG e PNG. A listagem 10 exibe a configuração do arquivo responsável pela geração do relatório de agendamentos.

```

<?php

//inclusão da biblioteca

```

```

define('FPDF_FONTPATH', 'fpdf/font');
require('fpdf/fpdf.php');
require('./HeaderAgendamentosGeral.php');

//Novo Objeto setando a formatação da página.
$pdf = new PDF('P', 'cm', 'A4');
$pdf->AliasNbPages();
$pdf->Open();

//Insere uma nova página
$pdf->AddPage();

//Seta a formatação da fonte
$pdf->SetFont('Arial', '', 9);

$con = mysql_connect("localhost", "root", "");
$db = mysql_select_db("bancoestagio");
$data = $_POST["txtDataInicio"];
$data2 = $_POST["txtDataTermino"];
$idCliente = $_POST["txtIDCliente"];
$filtroData = "";
$filtroSituacao = "";
if ($data != "" && $data2 != "") {
    $filtroData = " where Data_Inicio between '$data' and '$data2' ";
}
if ($idCliente != "") {
    if ($filtroData == "") {
        $filtroSituacao = " where Cli.Cliente_ID = '$idCliente' ";
    } else {
        $filtroSituacao = " and Cli.Cliente_ID = '$idCliente' ";
    }
}

$sql = "select * from agendamentos A
inner join consultores C on A.Funcionario_ID = C.Funcionario_ID
inner join clientes Cli on A.Cliente_ID = Cli.Cliente_ID
inner join modulos M on A.Modulo_ID = M.Modulo_ID
$filtroData $filtroSituacao";

$result = mysql_query($sql) or die(mysql_error());

while ($row = mysql_fetch_array($result)) {
    $pdf->Cell(5, 1, $row['Nome'], 1, 0, 'L');
    $pdf->Cell(5, 1, $row['NomeCliente'], 1, 0, 'L');
    $pdf->Cell(3, 1, $row['NomeModulo'], 1, 0, 'L');
    $pdf->Cell(2, 1, $row['Data_Inicio'], 1, 0, 'C');
    $pdf->Cell(4, 1, $row['Hora_Inicial'] . ' as ' . $row['Hora_Final'], 1,
1, 'C');
//    $pdf->Cell(2, 1, $row['Hora_Final'], 1, 1, 'L');
}
mysql_close();
$pdf->Output();
?>

```

Listagem 10 – Relatório

Na parte inicial do código apresentado na Listagem 10, primeiramente é realizada a inclusão da biblioteca por meio da linha “require('fpdf/fpdf.php');”, o arquivo responsável pelo Header (cabeçalho) do relatório” require('./HeaderAgendamentosGeral.php);” é incluído em sequência. Após isso, é criado um novo objeto PDF passando como parâmetro a formatação da página e finalizando a primeira parte é inserido uma página por meio da função \$pdf-

>AddPage()); e definido a formatação da fonte que será utilizada \$pdf->SetFont('Arial',
", 9);

Configurado a primeira parte do relatório, a próxima etapa é realizar a conexão com o banco de dados e para isso são utilizadas duas variáveis \$con e \$db, que respectivamente recebem a conexão e o nome do banco de dados a ser acessado.

De acordo com a necessidade do relatório, ele poderá ou não conter filtros para exibição dos resultados. Para o relatório de agendamentos visando facilitar o acesso do usuário a determinadas informações foram implementados dois tipos de filtros, por data e por código.

A funcionalidade destes filtros é determinada pelas seguintes etapas: declaração das variáveis que vão receber os dados informados pelo usuário na tela de filtros; verificação lógica de condições que podem vir a ocorrer, isso possibilita o trabalho em conjunto dos filtros; e, por último, a *query* que será responsável por buscar no banco de dados as informações com base no filtro especificado.

Em seguida é necessário armazenar o resultado da *query* dentro da variável \$Result e por meio de uma estrutura de repetição *while* percorrê-la, buscando os dados para o preenchimento do relatório por meio do comando \$pdf->Cell(5, 1, \$row['Nome'], 1, 0, 'L');. Com esse comando, cada dado é armazenado em uma célula e possui suas configurações.

O PhpMailer é uma classe do PHP que possui funções mais avançadas que a classe mail() padrão. Seu diferencial é permitir um envio seguro, pois faz a autenticação do usuário e senha do *Simple Mail Transfer Protocol* (SMTP) no servidor e garante que o e-mail seja enviado realmente pela conta autenticada.

O código da listagem 11 voltado ao envio do *e-mail* pode ser dividido em três partes principais: Seção, Query e E-mail.

Antes de qualquer codificação é preciso fazer um *start* na seção para que o usuário possa manipular os dados. Em seguida é realizada a conexão com o banco e são realizadas as consultas (*queries*) responsáveis por trazer a informação. Essas por sua vez serão armazenadas em variáveis que posteriormente serão utilizadas na configuração do envio e no conteúdo do *e-mail*.

```
<?php
session_start();
if (isset($_SESSION["Nome"])) {
    $nomeCompleto = $_SESSION["Nome"];
}
$teste = $_SESSION["Usuario_ID"];
```

```

$con = mysqli_connect("localhost", "root", "", "bancoestagio") or die("Erro: "
. mysqli_connect_error());
$resultado = mysqli_query($con, "select * from opcoesemail") or die("Erro: "
. mysqli_error($con));
$dados = mysqli_fetch_array($resultado);

$idAgendamento = $_GET["Agendamento_ID"];

$resultadoAgendamento = mysqli_query($con, "select * from agendamentos "
. "A inner join clientes C on A.Cliente_ID = C.Cliente_ID "
. "inner join modulos M on A.Modulo_ID = M.Modulo_ID "
. "inner join consultores S on A.Funcionario_ID = S.Funcionario_ID "
. "inner join tipos_treinamento T on A.Tipo_Treinamento_ID =
T.Tipo_Treinamento_ID "
. "where S.Usuario_ID = '$teste' and A.Agendamento_ID =
'$idAgendamento'" ) or die("Erro: " . mysqli_error($con));
//mysqli_close($con);
$dadosAgendamento = mysqli_fetch_array($resultadoAgendamento);

$data = date('d/m/Y H:i:s', mktime(gmdate("H") - 3));
$dataEmail = date('Y/m/d H:i:s', mktime(gmdate("H") - 3));
$dataInicio = $dadosAgendamento["Data_Inicio"];
$horaInicio = $dadosAgendamento["Hora_Inicial"];
$modulo = $dadosAgendamento["NomeModulo"];
$consultor = $dadosAgendamento["Nome"];
$codigoCliente = $dadosAgendamento["Cliente_ID"];

//$conexao = mysqli_connect("localhost", "root", "", "bancoestagio") or
die("Erro: " . mysqli_connect_error());
$sql = "INSERT INTO logenvioemails (LogEnvioEmailID, DataEnvio, Cliente_ID,
Agendamento_ID) "
. "VALUES ( NULL , '$dataEmail', '$CodigoCliente',
'$idAgendamento')";
mysqli_query($con, $sql) or die("Erro: " . mysqli_error($con));

$resultadoCliente = mysqli_query($con, "select * from clientes where
Cliente_ID = '$CodigoCliente'" ) or die("Erro: " . mysqli_error($con));
$dadosCliente = mysqli_fetch_array($resultadoCliente);

// Inclui o arquivo class.phpmailer.php localizado na pasta phpmailer
require("../email/class.phpmailer.php");

$mail = new PHPMailer();
$body = " Prezado cliente, <br><br>

Conforme contato, o seu próximo treinamento esta agendado para:<br><br>

Data: $DataInicio<br>
Horário: $HoraInicio<br>
Módulo: $Modulo<br>
Consultor: $Consultor<br><br><br>

Para que possamos cumprir nosso cronograma de treinamentos pré-estabelecido,
contamos com sua colaboração para o cumprimento das orientações abaixo:<br>
1) Os treinamentos serão agendados conforme horário de Brasília;<br>
2) Para a realização dos treinamentos é necessário que o Cliente possua conta
de Skype e Headphone;<br>
3) Para otimizar o tempo de treinamento, pedimos que a escola esteja
disponível no Skype alguns minutos antes do horário agendado;<br>
4) Havendo a necessidade de cancelar ou remarcar treinamentos, pedimos que
nos informe com 24 horas de antecedência através do e-mail
implantacao@gerenciamento.com.br; <br>
5) A tolerância de atraso é de apenas 15 minutos após o horário agendado,
diferente disso, não conseguiremos cumprir o cronograma programado e o
treinamento é cancelado sem possibilidade de reposição;<br>
6) Caso não consiga nos informar com antecedência sobre o cancelamento, não
conseguiremos repor este horário, sendo necessário contratar este ou mais
treinamentos separadamente ao término do período de implantação; <br>

```

```

7) Os Treinamentos deverão ser concluídos em no máximo 60 dias após a
contratação do sistema, por este motivo, para que sua instituição não seja
prejudicada, pedimos que toda a equipe esteja comprometida com a implantação
do sistema; <br>
8) Ao termino da implantação será possível realizar a contratação de novos
treinamentos junto ao departamento comercial.<br>
9) Para os treinamentos acima agendados, adicione no seu Skype previamente o
consultor<br>
Em paralelo aos treinamentos, gostaríamos de ressaltar, que sua instituição
terá acesso ao Suporte Técnico através dos canais de comunicação abaixo:<br>
Chat online (ferramenta de comunicação integrada ao sistema).<br>
• O manual e os vídeos de treinamento do sistema estão disponíveis através de
nossa ferramenta de Wiki, acessível através do sistema no menu Ajuda >
Central de Ajuda.<br>
<br>
Agradecemos a atenção e ficamos a disposição.<br><br>";

$mail->IsSMTP(); // telling the class to use SMTP
$mail->SMTPAuth = true; // enable SMTP authentication
$mail->SMTPSecure = "tls"; // sets the prefix to the server
$mail->Host = $dados["Servidor"]; // sets GMAIL as the SMTP server
$mail->Port = $dados["Porta"]; // set the SMTP port for the
GMAIL server
$mail->Username = $dados["Usuario"]; // GMAIL username
$mail->Password = $dados["Senha"]; // GMAIL password
$mail->SetFrom($mail->Username, 'Software para Gerenciamento de
Treinamentos');
$mail->AddReplyTo($mail->Username, "Software para Gerenciamento de
Treinamentos");
$mail->Subject = "Treinamento Agendado"; //Titulo do Email
$mail->AltBody = "$body";
$mail->MsgHTML($body);
$address = $dadosCliente["Email"];
$mail->AddAddress($address, $address);

if (!$mail->Send()) {
    echo "Erro ao enviar mensagem: " . $mail->ErrorInfo;
} else {
    echo "<script>alert('Email Enviado!!!');</script>";

    header("Location: ListagemAgendamentos.php");
}

// header("Location: ListagemAgendamentos.php");
?>

```

Listagem 11 – E-mail

A função principal do sistema passa pelo mesmo processo de inserção, exclusão e edição que os demais cadastros, ou seja, a estrutura dos arquivos que compõem estas funções é idêntica. A Listagem 12 nos mostra o código que compõe a listagem dos agendamentos.

```

<?php
include 'Home.php';
?>
<html>
    <script language = "JavaScript">
        function confirmacao(id) {
            var resposta = confirm("Deseja remover esse registro?");

            if (resposta == true) {
                window.location.href =
                "Function_ExcluirAgendamento.php?Agendamento_ID=" + id;
            }
        }
    </script>

```

```

    }
  }
</script>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Listagem de Agendamentos</title>
  <script src="jquery-1.8.3.min.js"></script>
  <script src="jquery.tablesorter.min.js"></script>
  <script src="jquery.tablesorter.pager.js"></script>
  <link rel="stylesheet" href="css/custom.css" media="screen" />
  <link href="bootstrap/css/bootstrap.css" rel="stylesheet" type="text/css" />
  <link rel="stylesheet" type="text/css" href="css/estilo.css"/>
</head>
<body>
  <?php
    include './php/AbreConnexao.php';
    $usuario = $_SESSION["Usuario_ID"];

    $resultado = mysqli_query($con, "select * from agendamentos "
      . "A inner join clientes C on A.Cliente_ID = C.Cliente_ID "
      . "inner join modulos M on A.Modulo_ID = M.Modulo_ID "
      . "inner join consultores S on A.Funcionario_ID =
S.Funcionario_ID "
      . "inner join tipos_treinamento T on A.Tipo_Treinamento_ID =
T.Tipo_Treinamento_ID "
      . "where S.Usuario_ID = $usuario ") or die("Erro: " .
mysqli_error($con));
    mysqli_close($con);

    if (mysqli_num_rows($resultado) < 1) {
      echo "<fieldset> Nenhum registro cadastrado. Para inclusão acesse
o link: <br> "
        . "<a href='CadastroAgendamentos.php'>Novo Agendamento</a>"
        . "</fieldset>";
      exit;
    }
  ?>
  <a class="Links" href="CadastroAgendamentos.php">Novo Agendamento</a>
  <a class="Links" href="Relatorios/RelatorioGeralAgendamentos.php">Relatório
de
Agendamentos</a> <a class="Links" href="RelatorioComFiltro.php">Filtros</a>
  <!--<input class=" btn btn-warning btn-lg" type="Button"
value="Enviar E-mail" />-->
  <!--Filtro-->
  <form method="post" action="exemplo.html" id="frm-filtro">
    <p>
      <input type="search" id="pesquisar" name="pesquisar"
size="30" placeholder="Pesquisar" />
    </p>
    <input class=" btn btn-success btn-lg" type="submit"
value="Visualizar Relatório"/><br/>-->
  </form>
  <table cellpadding="0" cellspacing="0" summary="Tabela de dados fictícios"
class="html">
    <thead>
      <tr>
        <th><input type="checkbox" value="1" id="marcar-todos"
name="marcar-todos" /></th>
        <th>ID</th>
        <th>Cliente</th>
        <th>Módulo</th>
        <th>Consultor</th>
        <th>Tipo Treinamento</th>
        <th>Data Inicial</th>
        <th>Horário Inicial</th>
        <th>Ação</th>
      </tr>
    </thead>
  </table>

```

```

        </tr>
    </thead>
    <tbody>
        <?php
            while ($registro = mysqli_fetch_array($resultado)) {
                $codigo = $registro["Agendamento_ID"];
                $Cliente_ID = $registro["NomeCliente"];
                $Treinamento_ID = $registro["NomeModulo"];
                $Funcionario_ID = $registro["Nome"];
                $Tipo_Treinamento_ID = $registro["Descricao"];
                $Data_Inicio = $registro["Data_Inicio"];
                $Hora_Inicial = $registro["Hora_Inicial"];

                echo "<tr>"
                ?>
                <td><input type="checkbox" value="1" name="marcar[]" /></td>
                <?php
                    echo "<td>$codigo</td>"
                    <td>$Cliente_ID</td>
                    <td>$Treinamento_ID</td>
                    <td>$Funcionario_ID</td>
                    <td>$Tipo_Treinamento_ID</td>
                    <td>$Data_Inicio</td>
                    <td>$Hora_Inicial</td>
                    <td><a
href="\Function_EditarAgendamento.php?Agendamento_ID=$codigo\">[Editar]</a> |
"
                    .
                    " <a
href='javascript:func();'>[Excluir]</a> | "
                    .
                    "<a
href=\"EmailTeste.php?Agendamento_ID=$codigo\">[Enviar E-mail]</a></td>
                    </tr>\n";
            }

            mysqli_free_result($resultado);
            ?>
        <!--
            <tr>
            <td><input type="checkbox" value="1" name="marcar[]" /></td>
            <td>1</td>
            <td>Treinamento In-Loce</td>
            <td>
            <a href="#"></a>
            <a href="#"></a>
            </td>
            </tr>
            <tr>
            <td><input type="checkbox" value="1" name="marcar[]" /></td>
            <td>2</td>
            <td>Treinamento On-Line</td>
            <td>
            <a href="#"></a>
            <a href="#"></a>
            </td>
            </tr>-->
        </tbody>
    </table>
    <div id="pager" class="pager html2" >
        <form>
            <span>
                <select class="pagesize">
                    <option selected="selected" value="10">10</option>
                    <option value="20">20</option>
                    <option value="30">30</option>

                </select> registros
            </span>
        </form>
    </div>

```

```



<input type="text" class="pagedisplay"/>


</form>
</div>
<script>
    $(function () {

        $('table > tbody > tr:odd').addClass('odd');
        $('table > tbody > tr').hover(function () {
            $(this).toggleClass('hover');
        });
        $('#marcar-todos').click(function () {
            $('table > tbody > tr > td > :checkbox')
                .attr('checked', $(this).is(':checked'))
                .trigger('change');
        });
        $('table > tbody > tr > td > :checkbox').bind('click change',
function () {
            var tr = $(this).parent().parent();
            if ($(this).is(':checked'))
                $(tr).addClass('selected');
            else
                $(tr).removeClass('selected');
        });
        $('form').submit(function (e) {
            e.preventDefault();
        });
        $('#pesquisar').keydown(function () {
            var encontrou = false;
            var termo = $(this).val().toLowerCase();
            $('table > tbody > tr').each(function () {
                $(this).find('td').each(function () {
                    if ($(this).text().toLowerCase().indexOf(termo) > -1)
                        encontrou = true;
                });
            });
            if (!encontrou)
                $(this).hide();
            else
                $(this).show();
            encontrou = false;
        });
    });
    $("table")
        .tablesorter({
            dateFormat: 'uk',
            headers: {
                0: {
                    sorter: false
                },
                5: {
                    sorter: false
                }
            }
        })
        .tablesorterPager({container: $("#pager")})
        .bind('sortEnd', function () {
            $('table > tbody > tr').removeClass('odd');
            $('table > tbody > tr:odd').addClass('odd');
        });
    });
</script>
</body>
</html>

```

5 CONCLUSÃO

Além da modelagem do aplicativo, foi apresentada a implementação das suas principais funcionalidades. Todas as funcionalidades definidas para o sistema foram implementadas. É certo que ajustes serão necessários a partir de uma avaliação dos usuários do sistema e mesmo de outras pessoas.

Para gerenciar a execução foi utilizado o Apache por meio do XAMPP que é um servidor de aplicação independente (software livre). A implementação das funcionalidades foi utilizado PHP 5, JQuery, JavaScript, e para a implementação da interface, foi utilizado o *framework front end* Bootstrap, juntamente com HTML5 e CSS. Essas tecnologias apresentam recursos que facilitam o desenvolvimento da aplicação e permitem desenvolver interface com funcionalidades que visam auxiliar na interação do usuário.

Como sequência do trabalho será finalizada a implementação do mesmo e a colocação em testes para verificação do atendimento das funcionalidades. Mesmo identificação de erros de codificação e de regras de negócio que não tenham sido detectados nos testes unitários realizados durante a implementação.

Ao final deste trabalho foi possível perceber que as tecnologias voltadas ao desenvolvimento de aplicações *web* desde *frameworks* voltados à criação de sua interface até a linguagem de programação utilizada em suas efetivas funcionalidades tornam o processo de desenvolvimento mais dinâmico, com resultados mais rápidos e eficazes, proporcionando mais clareza aos desenvolvedores na hora de elaborar as soluções via código fonte.

No presente trabalho, algumas dessas tecnologias, como a linguagem de programação PHP e o *framework front-end* Bootstrap foram utilizadas.

A linguagem PHP, bastante utilizada no desenvolvimento de aplicações *web*, tem como vantagem ser de fácil aprendizado, pois é composta por elementos de outras linguagens como o C e o Java. Outra vantagem é a sua estrutura que permite facilmente a conexão com muitos bancos de dados que sejam compatíveis com o padrão *Open Database Connectivity* (ODBC), e por fim é uma linguagem multiplataforma.

Já o *framework* Bootstrap tem como principal vantagem, a possibilidade de criar interfaces sem a necessidade de utilização de CSS, apenas trabalhando com suas mais diversas classes é possível desenvolver uma interface com o usuário que atenda a requisitos e padrões de usabilidade.

No decorrer do desenvolvimento do presente trabalho, também ocorreram alguns problemas, como por exemplo, o não acesso aos menus das funcionalidades quando a aplicação não esteve conectada à Internet. Essa situação veio a ocorrer devido a um arquivo do *framework* de interface que direcionava diretamente a uma página web, sem internet as configurações deste arquivo não eram carregadas e com isso os menus não eram exibidos. A solução encontrada para tal problema foi direcionar os *links* da aplicação para um arquivo local dentro do próprio projeto, com isso a aplicação se comporta normalmente independente de acesso a rede ou não.

Outra dificuldade encontrada foi adaptar a interface das telas de listagem, devido ao *grid* possuir um arquivo de customização específico, isso ocasionava conflito com os arquivos do *framework* Bootstrap e a interface acabava sofrendo alterações de acordo com a ação do usuário. A solução encontrada foi realizar pequenos ajustes na chamada dos *links* responsáveis por buscar estes arquivos dentro do projeto, assim a interface passou a ficar estável nas ações do usuário.

O sistema como um todo, procura tratar as suas ações de forma simples, deixando em evidência o seu objetivo principal de controle e organização das informações relacionadas ao processo de implantação de sistemas.

REFERÊNCIAS

- BENHARREF, Abdelghani; SERHANI, Mohamed Adel; BOUKTIF, Salah. **Managerial community of web services for management of communities of web service**, 2010, p. 97-104.
- BOZZON, Alessandro; COMAI, Sara; FRATERNALI, Piero; CARUGHI, Giovanni T. **Conceptual modeling and code generation for rich internet applications**, In: ICWE '06 Proceedings of the 6th international conference on Web engineering, ACM Press, New York, 2006, p. 353–360.
- CARDOSO, Jorge; WINKLER, Matthias; VOIGT, Konrad; BERTHOLD, Henrike. **IoS-based services, platform services, sla and models for the internet of services**, In: Software and Data Technologies Communications in Computer and Information Science, v. 50, 2011, p. 3-17.
- COMAI, Sara; CARUGHI, Giovanni. **A behavioral model for rich internet applications**. In: Web Engineering, Lecture Notes in Computer Science, v. 4607, 2007, p.364-369.
- COOPER, Ezra; LINDLEY, Sam; WADLER, Philip; Yallop, Jeremy. **Links: web programming without tiers**. In: Formal Methods for Components and Objects, Lecture Notes in Computer Science, v. 4709, 2007, p.266-269.
- DUHL, Joshua. **Rich Internet Applications**. Technical report, IDC, November 2003. Disponível em:
<http://www.adobe.com/platform/whitepapers/idc_impact_of_rias.pdf>. Acesso em: 15 abr. 2015.
- INTERNET. **Internet usage statistic**, 2014. Disponível em:
<<http://www.internetworldstats.com/stats.htm>>. Acesso em: 10 fev. 2015.
- PANG, Zhen; WEN, Fuan; PAN, Xiwei; LUI, Cen. **Migration model for rich internet applications based on PureMVC Framework**. In: 2010 International Conference on Computer Design and Applications (ICCD), 2010, v. 5, p. V5-340 - V5-343.
- PAVLIÆ, Daniel; PAVLIÆ, Mile; JOVANOVIÆ, Vladan. **Future of Internet technologies**. MIPRO 2012, p. 1366-1371.
- POWELL, Courtney; NAKAMURA, Keisuke; AKAMA, Kiyoshi. **Towards a formal behavioral model for rich internet applications**. In: International Conference on Computational Intelligence and Software Engineering (CiSE 2009), 2009, p. 1-5.
- PRECIADO, Juan Carlos; LINAJE, Marino; COMAI, Sara; SANCHEZ-FIGUEROA, Fernando. **Designing rich internet applications with web engineering methodologies**. In: 9th IEEE Workshop Web Site Evolution (WSE 07), IEEE CS Press, 2007, p. 23–30.

REDING, Viviane. **The future of Internet: a compendium of european projects on ICT**, European Commission Report, 2008.

STEARNS, Brent. **XULRunner: a new approach for developing rich internet applications**, IEEE Internet Computing, v.11, n.3, 2007, p.67-73.

VAZ, Agostinho M.; MARTINS, Bruno M.; BRANDÃO, Rodrigo C.; ALBERTI, Antonio M. **Internet of information and services: a conceptual architecture for integrating services and contents on the future Internet**. In: IEEE Latin America Transactions, v. 10, n. 6, december 2012, p. 2292-2300.

W3C, **Web services architecture**, 2006. Disponível em:
<<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>>. Acesso em: 10 abr. 2015.
20040211/>. Acesso em: 10 abr. 2015.