

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**DESENVOLVIMENTO DE UM SOFTWARE
GERENCIADOR DE AGÊNCIAS DIGITAIS
UTILIZANDO PHP ORIENTADO A OBJETOS E AJAX**

TRABALHO DE GRADUAÇÃO

Paulo Gilberto Roehrs Filho

**Santa Maria, RS, Brasil
2010**

DESENVOLVIMENTO DE UM SOFTWARE GERENCIADOR DE AGÊNCIAS DIGITAIS UTILIZANDO PHP ORIENTADO A OBJETOS E AJAX

por

Paulo Gilberto Roehrs Filho

Trabalho de Graduação apresentado ao curso de Ciência da
Computação da Universidade Federal de Santa Maria (UFSM, RS)
Como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

Orientador: Prof.Dr. Eduardo Kessler Piveta

Trabalho de Graduação N. 313

Santa Maria, RS, Brasil
2010

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**DESENVOLVIMENTO DE UM SOFTWARE GERENCIADOR DE
AGÊNCIAS DIGITAIS UTILIZANDO PHP ORIENTADO A OBJETOS E
AJAX**

elaborado por
Paulo Gilberto Roehrs Filho

Como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

Comissão Examinadora:

**Prof. Dr. Eduardo Kessler Piveta
(Presidente/Orientador)**

Prof. Dr^a. Deise de Brum Saccol

Prof. Dr. Giovanni Rubert Librelotto

Santa Maria, 09 de Dezembro de 2010

AGRADECIMENTOS

Agradeço inicialmente a Deus, por me proporcionar saúde, paz e alegria, dádivas estas muito importantes para meu desenvolvimento.

Agradeço ao amor e apoio dos meus pais e irmã, em especial a minha mãe, que sempre confiou no meu potencial e sempre esteve ao meu lado me orientando e participando ativamente no meu desenvolvimento.

Obrigado a todos os professores que me ajudaram, passando seus conhecimentos e me mostrando novas perspectivas nas mais diversas áreas.

Não posso deixar agradecer a todos aqueles colegas que me auxiliaram durante o curso, ao Fábio Lorenzi e Jonathan Alves, e especialmente ao colega Luciano Minuzzi que me auxiliou durante toda elaboração deste trabalho. Sou muito grato pela amizade de vocês.

Por último, não menos importante, à Daniele Feiffer, minha namorada, sempre disposta a me auxiliar e que contribuiu com muitas sugestões positivas para este trabalho.

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

DESENVOLVIMENTO DE UM SOFTWARE GERENCIADOR DE AGÊNCIAS DIGITAIS UTILIZANDO PHP ORIENTADO A OBJETOS E AJAX

Autor: Paulo Gilberto Roehrs Filho
Orientador: Prof. Dr. Eduardo Kessler Piveta

Agências Digitais são um novo tipo de agência de comunicação que desenvolve *websites*, campanhas de marketing e produção de conteúdo voltado à internet e outros meios digitais. Como estas possuem geralmente uma grande equipe de trabalho, se faz necessário a aplicação de muita organização para que todo o serviço seja realizado com qualidade e agilidade.

No mercado de *softwares* existem diversas soluções para o gerenciamento de projetos, mas nenhuma dando o enfoque às tarefas cotidianas. Para suprir esta necessidade este trabalho propõe o desenvolvimento de um software capaz de centralizar em um só lugar todas as tarefas e informações para gerenciar o *workflow* de uma agência digital.

Este trabalho apresenta as facilidades proporcionadas no desenvolvimento de software com a utilização do paradigma de orientação a objetos aplicado à linguagem PHP 5, juntamente com AJAX. São discutidos os principais conceitos da Programação orientada a objetos, da linguagem de programação PHP, os conceitos de *workflow* e as características de sistemas que utilizam AJAX.

Palavras-chave: Agências Digitais, PHP, Orientação a Objetos, Workflow, AJAX.

ABSTRACT

Final Work
Computer Science Majoring
Federal University of Santa Maria

DEVELOPMENT OF A SOFTWARE MANAGER DIGITAL AGENCIES USING OBJECT-ORIENTED PHP AND AJAX

Author: Paulo Gilberto Roehrs Filho
Advisor: Prof. Eduardo Kessler Piveta, PhD

Digital agencies are a new kind of communications agency that develops websites, marketing campaigns and producing content geared to the Internet and other digital media. As these generally have a great team, it is necessary to apply plenty of organization for any service is done with quality and speed.

In the software market there are several solutions for managing projects, but none giving the approach everyday tasks. To meet this need this estudy proposes the development of software capable of centralizing in one place all the tasks and information to manage the workflow of a digital agency.

This paper presents the facilities provided in software development using the object oriented paradigm applied to PHP 5, along with AJAX. The most important concepts of object oriented programming, PHP programming language, the concepts of workflow and features of systems that use AJAX.

Keywords: Digital Agencies, PHP, Object-Oriented Programming, Workflow, AJAX.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de casos de uso.....	22
Figura 2 – Modelo do banco de dados	30
Figura 3 – Wireframe do sistema	32
Figura 4 – Tela de autenticação	34
Figura 5 – Tela de um usuário supervisor	36
Figura 6 – Tela Clientes	37
Figura 7 – Tela contratos.....	38
Figura 8 – Tela Projetos	40
Figura 9 – Tela inserção de um novo projeto	40
Figura 10 – Exemplo de gráficos gerados.....	41

LISTA DE SIGLAS

AJAX	Asynchronous Javascript And XML
HTML	HyperText Markup Language
MVC	Model View Controller
ODBC	Open Data Base Connectivity
PHP	HyperText Preprocessor
POO	Programação Orientada a Objetos
UML	Unified Modeling Language

SUMÁRIO

Sumário	9
1 Introdução	10
1.1 Objetivos Gerais e Específicos.....	11
1.1.1 Objetivo Geral	11
1.1.2 Objetivos Específicos	11
1.2 Estrutura da Monografia.....	12
2 Revisão de Literatura	13
2.1 Agências Digitais.....	13
2.2 Orientação a Objetos.....	15
2.2.1 Conceitos essenciais.....	15
2.2.2 Vantagens e desvantagens	16
2.3 PHP	17
2.3.1 Histórico	17
2.3.2 Características	18
2.4 Padrão de arquitetura MVC.....	19
2.5 AJAX	19
2.6 Workflow.....	20
3 Desenvolvimento do Sistema	22
3.1 Diagrama de casos de uso.....	22
3.3 Modelagem e implementação do banco de dados	29
3.4 Wireframes	31
3.4 Implementação	33
3.5 Telas do sistema	33
3.5.1 Tela de autenticação	33
3.5.1.1 Código de autenticação.....	34
3.5.2 Tela inicial	35
3.5.3 Tela Clientes	36
3.5.4 Tela Contratos.....	37
3.5.7 Tela Serviços.....	39
3.5.9 Tela Projetos	39
3.5.10 Tela Estatísticas	41
4 Considerações Finais	42
5 Referências	43

1 INTRODUÇÃO

A crescente demanda por *software* para a Web trouxe uma grande demanda de profissionais aptos no desenvolvimento de sistemas em linguagens para a internet. Dentro deste contexto está a linguagem de programação PHP, amplamente utilizada. Apesar da grande quantidade de programadores adeptos, muitos desses desconhecem as facilidades da utilização de técnicas de programação orientada a objetos e de AJAX, os quais promovem a redução no tempo de programação.

Software para a Internet pode ser disponibilizado gratuitamente, bem como um serviço pago. Em tal modelo não é necessário adquirir licença de uso do software para instalação, nem a compra de servidores para a execução. Paga-se uma assinatura pelo tempo de uso, ou o número de usuários.

Agências especializadas na criação de *websites* demandam cada vez mais de profissionais aptos a criar soluções que se integrem aos *sites*. Com a maior demanda de trabalhos, a maior dificuldade encontrada nestas agências é a gerência de todos os projetos em andamento, seus prazos e as necessidades de cada equipe no decorrer do projeto. Os profissionais encontrados nestas agências são provenientes das mais diversas áreas (Desenho Industrial, Administração, Comunicação Social, Informática, etc.) e nem sempre conhecem os fundamentos para o desenvolvimento de software e o seu ciclo de vida.

O sistema de *software* proposto será disponibilizado como um serviço *online* e mesclará algumas funcionalidades encontradas em um CRM (gestão de relacionamento com o cliente), *intranets* e em gerenciadores de projetos. Será um sistema de software especializado em controlar todos os processos e peculiaridades de produção de uma Agência Digital, o fluxo de trabalho e a interdependência entre as tarefas (*workflow*), a comunicação interna e o atendimento ao cliente.

O sistema *software* proposto neste trabalho de graduação visa auxiliar o serviço realizado por estas agências, para que a produtividade e o relacionamento com os clientes melhorem, assim como auxiliie no planejamento estratégico através da análise dos relatórios estatísticos gerados pelo *software*.

Para isso, ele fornece o controle de usuários, o cadastro de clientes e contatos, a adição de um contrato, criação de novos projetos relacionados a um contrato e juntamente a determinação das tarefas necessárias para cada projeto em andamento. Além disso é possível consultar o histórico de cada cliente e visualizar alguns dados estatísticos dos clientes da agência.

Neste sentido, a elaboração de um sistema de gerência, através da linguagem PHP foi escolhida por se tratar de uma linguagem livre de custos, amplamente divulgada e em suas últimas versões possibilitando a programação orientada a objetos de forma concisa e precisa, o que facilita a organização de código e a futura implantação de novos módulos ao sistema.

1.1 Objetivos Gerais e Específicos

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é a implementação de um sistema gerenciador de Agências Digitais, dando ênfase ao uso de técnicas de engenharia software e a utilização da linguagem PHP orientada a objetos juntamente com scripts AJAX.

1.1.2 Objetivos Específicos

Como objetivos específicos, teremos:

- Estudo e aprofundamento em técnicas de engenharia de software.
- Estudo e aprofundamento em técnicas de programação orientada a objetos.
- Levantamento de requisitos de uma Agência Digital.

- Implementação de diversos tipos de relatórios, obtidos de forma dinâmica.

1.2 Estrutura da Monografia

O projeto foi executado em diversas etapas, estas que são listadas a seguir:

Revisão bibliográfica: Esta etapa visa realizar o levantamento teórico e construir uma base referente às técnicas de engenharia de software. Durante esta etapa foi levantadas técnicas que ajudarão a programação do próprio trabalho, assim como no auxílio para o estabelecimento das etapas necessárias no sistema de controle de projetos da Agência.

Levantamento de Requisitos: Nesta etapa, foram realizadas entrevistas com a equipe de trabalho de uma Agência Digital, para definir os requisitos necessários para atender suas necessidades.

Modelagem/Implementação da Aplicação: O objetivo desta etapa é a criação dos fluxogramas de cada módulo previsto, prevendo assim cada funcionalidade específica. A partir dos dados modelados foi dado início ao processo de programação e implementação.

Testes: O objetivo é realizar testes diversos para encontrar erros possíveis no sistema, e sua compatibilidade em diversas plataformas (navegadores).

Elaboração da Monografia: Compreende o período estabelecido para desenvolver a parte escrita, baseando-se nos estudos e na bibliografia pesquisada.

2 REVISÃO DE LITERATURA

De acordo com Taurion(2009), a internet passa a ser o repositório de arquivos digitais e as pessoas podem criar seus documentos, fotos e arquivos sem a necessidade de instalar nenhum software em seu computador.

Para o desenvolvimento e a modelagem de um software que auxilie no processo de controle das etapas da modelagem de sistemas web é importante seguir algumas métricas abordadas em engenharia de software.

De acordo com Gustafson (2003) planejamento é essencial, e o desenvolvimento de software não é exceção. Alcançar sucesso no desenvolvimento de software requer planejamento, o qual, neste caso, envolve decidir quais tarefas precisam ser feitas, em que ordem fazê-las e que recursos são necessários para cumpri-las.

Segundo Dall'Oglio(2009) o processo de programação utilizando orientação a objetos juntamente com boas práticas de programação possibilita manter um ritmo sustentável no desenvolvimento de aplicações. Para isso, são utilizados alguns padrões de projeto e algumas técnicas de mapeamento objeto-relacional, além do uso de vários componentes que auxiliam a criações de aplicações complexas com PHP.

A grande dificuldade encontrada em Agências Digitais é a manutenção e controle de seus clientes e trabalhos realizados. Muitas empresas possuem um ciclo de vida curto, ou seja, extinguem, não por falta competência no que fazem, mas sim pelo despreparo e desconhecimento de métricas que auxiliem na realização de suas atividades.

2.1 Agências Digitais

Historicamente, as agências digitais são originárias de empresas de publicidade e propaganda, as quais realizam trabalhos de comunicação através de mídias tradicionais como rádio, TV, jornais, revistas, etc. Com o crescimento e avanço da utilização da internet, a necessidade para o desenvolvimento de ações de comunicação na rede tornou-se cada vez mais imprescindível. Um maior

investimento em mídias eletrônicas ocorreu a partir de 2002, como estratégia para estreitar o relacionamento de empresas com seu público e, assim, reforçar a presença da marca e identidade da empresa, segundo Ferrari (2004).

Neste contexto, inserem-se as agências digitais; através da contratação dos serviços dela é possível nortear ações estratégicas para inserção *online*, principalmente de pessoas jurídicas (empresas). Para desenvolvimento de aplicativos *web*, projeto digital, várias ações podem ser realizadas, desde o desenvolvimento, a hospedagem do *website*, o projeto gráfico, a programação e implementação de sistemas, etc.

Uma agência digital, em sua essência, promove a aproximação da empresa com vários públicos que utilizam o meio digital. Assim, o trabalho de comunicação na rede torna-se uma porta de entrada para os negócios que são divulgados na internet. Uma constatação característica dos clientes de agências digitais é que em sua grande maioria são organizações que têm apenas como objetivo “marcar” presença digital, sem pensar estrategicamente que essa ferramenta, com um bom planejamento, pode agregar valor a suas estratégias de marketing e gerar muita oportunidade de negócio.

Ao realizar uma pesquisa de empresas que realizam serviços focados em comunicação digital, Ferrari (2004) observou-se que ação inicial, para estruturar a estratégia ideal para cada cliente, engloba a análise de várias informações. Essas informações envolvem desde a identificação do objetivo da inserção digital, como será o posicionamento da marca, identidade visual pretendida, conteúdo informativo, entre outros, mas principalmente o público-alvo que o cliente pretende alcançar, mas é preciso salientar que a *web* atinge também públicos que não estão previstos pela organização.

Neste contexto é fundamental que a empresa tenha clareza da dimensão de mercado que pode atingir, sendo assim necessário que as agências trabalhem em conjunto com seus clientes e proporcionem *feedbacks* constantes. Deste modo ocorrerá um alinhamento evolutivo nas questões de atualizações de conteúdo no *website*, geração de informações relevantes aos usuários, realização de assistência aos contatos oriundos da internet e acompanhamento dos resultados de acessos, através de ferramentas de análise estatística.

Neste processo de gerenciamento de projetos, muitas atividades demandam organização, planejamento, fluxo de informações, determinação de prioridades e processos.

Desta maneira, o desenvolvimento de um software que auxilie as agências digitais neste trabalho se faz de extrema importância, pois assim ganha-se em produtividade, consegue-se atender melhor e mais clientes, além de obter dados verdadeiramente úteis para traçar nichos de mercado.

2.2 Orientação a Objetos

A orientação a objetos é um paradigma de programação que utiliza uma ótica mais próxima do mundo real. Conjuntos de procedimentos são organizados como objetos que interagem entre si.

Não há dúvida que modelar e projetar com orientação a objetos é fundamentalmente diferente do que com o enfoque tradicional: requer uma maneira diferente de idealizar a decomposição, e produz arquiteturas de *software* grandemente à parte da cultura de modelagem e projeto estruturados (Booch, 1994)

O foco principal é o objeto. A construção deste agrega comportamento, e também pode ser atribuído conhecimento, regras, ciclo de vida. Depois de modelado, o objeto é tratado como realmente um módulo, que se encaixa em um sistema sem a necessidade de nenhuma modificação.

2.2.1 Conceitos essenciais

Para o trabalho com Orientação a Objetos, adaptado de acordo com Dall'Oglio (2009), é essencial conhecer suas principais características, que são entre elas: objetos, classes, atributos, métodos, mensagens, herança, associação, encapsulamento, abstração e polimorfismo.

Neste contexto, as características essenciais para objetos são capazes de reagir a mensagens enviadas a eles, de acordo com os atributos armazenados

nestes. Classes representam um conjunto de objetos com características afins. Métodos são as ações tomadas pelas classes, suas habilidades. Mensagens são as chamadas enviadas a um objeto para invocar seus métodos. Herança é a capacidade de extensão dos comportamentos e atributos de uma classe a uma sub-classe. Associação é a capacidade que um objeto tem de utilizar os recursos de outro. O encapsulamento é a possibilidade que existe de esconder aspectos não relevantes (atributos) existentes em um objeto. Abstração é a capacidade de focar somente no que é essencial, ignorando detalhes de pouco interesse. O fundamento do poliformismo, segundo Dall'Oglio (2009, p.101) é em orientação a objeto “o princípio que permite que classes derivadas de uma mesma superclasse tenham métodos iguais, mas comportamentos diferentes, redefinidos em casa uma das classes-filha”.

2.2.2 Vantagens e desvantagens.

A programação orientada a objetos geralmente possui uma organização mais lógica quando comparada a programas tradicionais. Esta característica traz agilidade em situações de manutenção de código, além do menor risco a erros. Esta técnica é muito conhecida pela grande capacidade de reuso de código.

No desenvolvimento de grandes projetos, a divisão de tarefa entre equipes numerosas se torna mais fácil, e, obedecendo aos conceitos essenciais, o trabalho realizado por diferentes equipes se comunicam de forma correta. Esta característica possibilita o desenvolvimento ágil.

Como desvantagem temos a maior dificuldade de aprendizado, se comparada a linguagens procedurais. Entender todas as características é importante para o desenvolvimento orientado a objetos de forma correta.

E por fim, em sistemas que necessitam de processamento mais veloz a programação orientada a objetos quase sempre perde espaço para as linguagens tradicionais.

2.3 PHP

2.3.1 Histórico

A linguagem PHP foi criada em 1994 pelo programador Rasmus Lerdorf. Inicialmente ela nada mais era que um conjunto de Scripts criados pelo autor para, de forma dinâmica, controlar o acesso ao seu currículo on-line.

Com o crescente desenvolvimento do projeto, seu criador resolveu escrever uma implementação em linguagem C, para que as pessoas pudessem escrever seus códigos de uma forma mais simples. Surge assim, em 1995 a primeira versão oficial da linguagem, denominada ainda como PHP/FI (Personal Home Pages/Form Interpreter), com seu código fonte disponibilizado na internet para que demais interessados ajuda-se no desenvolvimento da mesma.

Em novembro de 1997 foi lançada a segunda versão do PHP. Nesta mesma época dois estudantes, Andi Gutmans e Zeev Suraski começaram a cooperar com Rasmus. O fruto de seu trabalho foi chamado de PHP 3, este sendo a reescrita de todo o código do PHP2 juntamente com diversas novidades, como a extensibilidade, novos protocolos, a conexão com diversos bancos de dados, sintaxe mais robusta, suporte (mesmo que rudimentar) à orientação a objetos. O que mais atraiu vários desenvolvedores para esta linguagem foi a nova API, que possibilitava a criação de novos módulos.

No final de 1998, o PHP já estava presente em cerca de 10% dos domínios da internet. Nesta época o significado da sigla PHP mudou para: PHP: Hypertext Preprocessor, retratando assim a nova realidade de uma linguagem com propósitos mais amplos (Dall'Oglio 2009)

No ano de 2000, foi lançado a versão 4 da linguagem. Ela oferece alguns recursos adicionais em relação às versões anteriores, como, por exemplo, o suporte ao gerenciamento de sessões. (Niederauer, 2001). Esta foi desenvolvida juntamente com um novo mecanismo batizado de Zend, que possibilitava a manipulação de computações mais complexa, assim como o aumento da performance. Nesta

mesma data, a linguagem já atingia 20% dos domínios da internet era utilizado por milhares de desenvolvedores.

Apesar de todos os esforços, o PHP ainda necessitava maior suporte à orientação a objetos, tal qual existe em linguagens como C++ e Java. Tais recursos estão finalmente presentes no PHP 5, após um longo período de desenvolvimento que culminou com sua disponibilização oficial em julho de 2004. (Dall'Oglio 2009)

2.3.2 Características

PHP é uma linguagem de programação estruturada e com suporte a orientação a objetos, com tipagem fraca. Ela é interpretada, ou seja, uma linguagem de computador que é executada em um programa chamado interpretador, e logo em seguida é executado pelo sistema operacional. Sua principal utilização se dá no desenvolvimento web, apesar de algumas variantes que propõem a construção de aplicações desktop.

Uma vantagem desta linguagem é que ela é gratuita, sendo possível no site do desenvolvedor copiar inclusive o código fonte para alterações. PHP é uma linguagem que pode ser embutida, e é comumente utilizada juntamente com a linguagem HTML (HyperText Markup Language). É baseada em servidor, ou seja, toda sua execução ocorre no lado do servidor. Os clientes, que acessam comumente o conteúdo através de um Navegador somente irão receber o resultado da interpretação do algoritmo criado nesta linguagem. Desta forma fica impossível a cópia de rotinas criadas em PHP, o que gera segurança na disponibilização de dados de forma eletrônica.

Diversos bancos de dados são disponibilizados para uso na linguagem, utilizando-se assim módulos de acesso. Caso algum tipo de banco não possua módulo compatível, ainda assim é possível a comunicação com o mesmo utilizando acesso via ODBC (*Open Data Base Connectivity*).

PHP é uma linguagem portátil, sendo possível sua utilização nas mais diversas plataformas, garantindo assim que não seja necessária a adaptação em uma eventual troca de servidores.

2.3.3 Suporte a Orientação a Objeto

O suporte a orientação a objetos iniciou na versão PHP 3, onde o programador podia alcançar alguns pacotes, tinha herança, mas só possibilitava a criação de métodos e propriedades.

Na versão 4, a linguagem ficou mais robusta, possibilitando a criação de cópias de objetos, mesmo que de forma precária, pois a linguagem ainda não suportava apontadores. Com o advento da versão 5, esta característica foi adicionada ao sistema, o que consolidou a utilização destas técnicas de forma adequada. Quando um objeto é copiado, na verdade o que se copia é o apontador para o objeto original. Assim, caso o objeto original mude, muda-se automaticamente a cópia.

2.4 Padrão de arquitetura MVC

O sistema foi desenvolvido seguindo os conceitos de MVC (Modelo-Visualização-Controlador). Sua utilização proporciona a separação lógica de negócio da lógica de apresentação, o que permite o desenvolvimento, testes e manutenção de forma isolada.

Com a crescente complexidade dos softwares criados, o uso do MVC se faz interessante devido a esta característica. Desta forma uma futura alteração de layout não irá afetar a manipulação dos dados.

2.5 AJAX

Atualmente o uso do recurso de AJAX, passou a ser pré-requisito no desenvolvimento de *websites* para Web 2.0. Seus recursos são diversos e o autor

Maroni (2007) elucida tal solução como sendo um conjunto de tecnologias AJAX definida da seguinte maneira:

AJAX é a abreviação de Asynchronous JavaScript And XML. Não é uma tecnologia, e sim uma metodologia, que utiliza vários recursos e tecnologias como Javascript, DOM e XML. O objetivo do AJAX é evitar que a cada solicitação do servidor à página precise ser completamente recarregada. Para isso, usamos o objeto XMLHttpRequest por meio do Javascript para intermediar todas as solicitações do servidor. (MARONI, 2007, p. 6).

Neste contexto, a criação de aplicativos web se torna mais interessante, já que as ações não necessitam de constantes atualizações de página. Esta é uma das mais populares vantagens das aplicações que utilizam AJAX, trazendo assim economia dos dados trafegados pela rede, já que a comunicação pode ocorrer de forma assíncrona e somente dos dados, não do conteúdo. Além disso, parte da lógica da aplicação desenvolvida na plataforma web é repassada ao navegador, assim ele tendo um papel mais importante do que simplesmente um visualizador de conteúdo repassado pelos servidores.

Outra vantagem de Ajax é a aplicação de conceitos mais avançados de navegação, como arrastar e soltar. Esta característica traz para mais próximo do usuário a perspectiva de utilização de aplicações web em pé de igualdade com aplicações desktop.

A aplicação de AJAX é muito recomendada, mas não é algo trivial. Quase não existe software para o controle de erros durante sua programação, assim como o programador deve se adaptar às peculiaridades que cada navegador têm em interpretar seu código. Uma solução prática é a utilização de uma biblioteca AJAX, de código aberto e prontas para utilização. Seu uso traz o aumento da produtividade e da qualidade do que se é desenvolvido.

2.6 Workflow

Os serviços realizados em uma agência digital devem seguir um fluxo de desenvolvimento. Esta tarefa pode ser realizada com a utilização do modelo *Workflow*, que tem como objetivo o suporte ao trabalho cooperativo, onde

informações ou tarefas são passadas de um participante a outro seguindo um conjunto de regras e procedimentos pré-estabelecidos.

O modelo *workflow* é centrado no processo. Ele não atua de forma passiva onde informações como no modelo *workgroup*. Ele atua ativamente entre os usuários de forma que estes cumpram o fluxo de trabalho que está determinado (USINORO, 2003).

Sistemas workflow possibilitam a visualização em tempo real do andamento de um processo e cada integrante sabe exatamente quais tarefas deve desempenhar no projeto. Isto facilita o gerente da equipe no controle dos prazos e de prioridades, na localização de informações e os responsáveis por tarefas passadas ou atuais.

Com tudo isso, podemos notar que este sistema é o melhor para a necessidade deste tipo de negócio.

3 DESENVOLVIMENTO DO SISTEMA

Vindo ao encontro da proposta de implementação, o desenvolvimento do sistema software gerenciador de agências digitais seguiu basicamente três etapas. Inicialmente foi projetado o diagrama de casos de uso, para posterior modelagem do banco de dados, e logo após o esboço do layout. Por fim o sistema escrito e implementado visualmente.

3.1 Diagrama de casos de uso

O diagrama de caso de uso é utilizado para descrever o cenário com as principais funcionalidades previstas no software contratado. A figura 1 demonstra o diagrama de uso para o sistema gerenciador proposto.

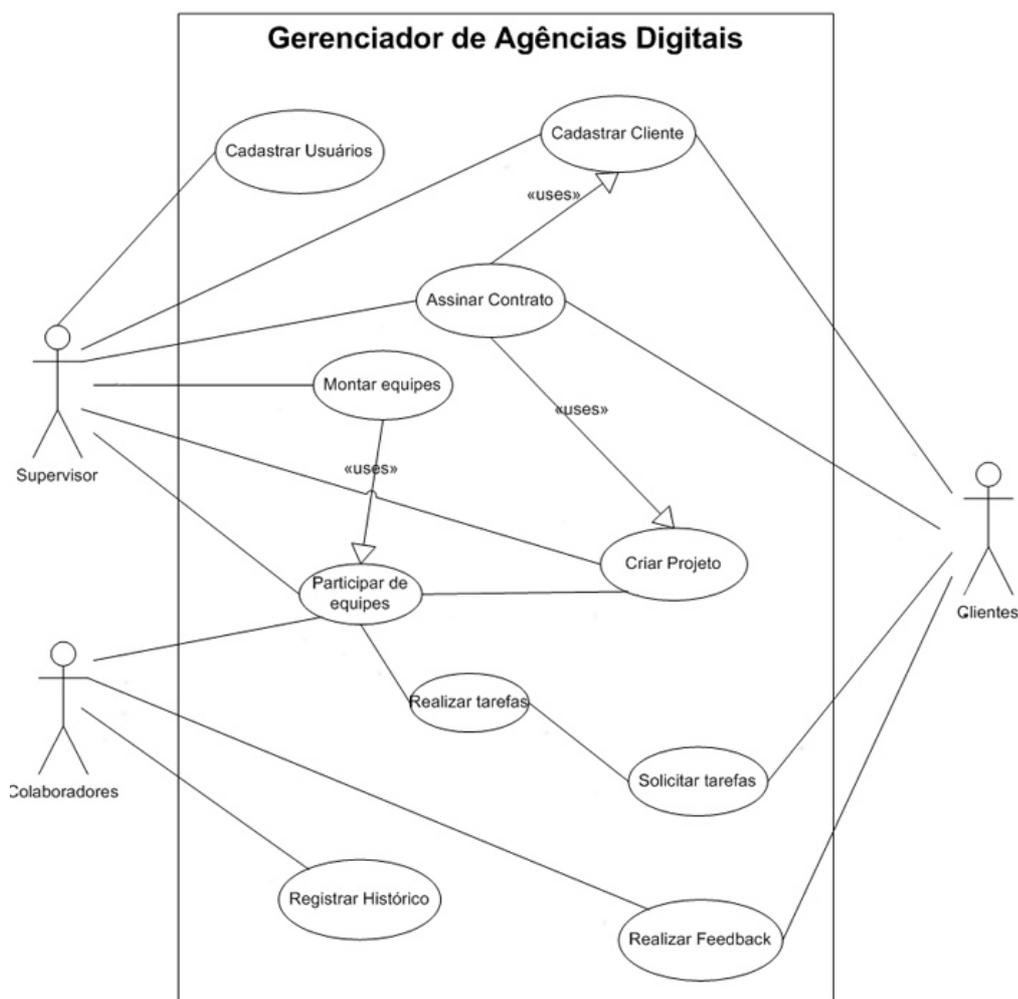


Figura 1 – Diagrama de casos de uso

Casos de uso:**a) Cadastrar Usuários**

Atores:

Supervisor.

Pré-condição:

O supervisor deve ter feito *log-in* e obtido autorização do sistema.

Fluxo de Eventos:

1. O caso de uso começa quando o supervisor seleciona “Colaboradores”
2. O supervisor clica no ícone “Adicionar Colaboradores”
3. O supervisor preenche os campos com o nome, sexo e o cargo do novo colaborador. Caso o cargo não exista, o mesmo pode ser criado ao clicar no ícone “Adicionar novo Cargo”.
4. O supervisor submete os dados ao sistema.
5. O sistema verifica se as informações estão corretas ou são suficientes para o cadastro, e em caso afirmativo, realiza o mesmo.

Pós-condição:

O usuário colaborador deve ter sido gravado no sistema e aparecido na lista de usuários.

b) Cadastrar Cliente

Atores:

Supervisor.

Pré-condição:

O usuário deve ter feito *log-in* e obtido autorização do sistema.

Fluxo de Eventos:

1. O caso de uso começa quando o supervisor seleciona “Clientes”
2. O supervisor clica no ícone “Adicionar Cliente”

3. O supervisor preenche os campos com o nome, telefone, e-mail, endereço, complemento, bairro, CEP, cidade, se é pessoa física ou jurídica.
4. O supervisor submete os dados ao sistema.
5. O sistema verifica se as informações estão corretas ou são suficientes para o cadastro, e em caso afirmativo, realiza o mesmo.

Pós-condição:

O cliente deve ter sido gravado no sistema e aparecido na lista de clientes.

c) Assinar Contrato

Atores:

Supervisor e Cliente.

Pré-condição:

O supervisor deve ter feito *log-in* e obtido autorização do sistema e ter realizado o fluxo denominado Cadastrar Cliente.

Fluxo de Eventos:

1. O caso de uso começa quando o supervisor seleciona “Contratos”
2. O supervisor clica no ícone “Adicionar Contrato”
3. O supervisor preenche os campos com a descrição, data de assinatura, data de término, o status assinado ou não, seleciona o cliente atrelado responsável pelo contrato, e seleciona a modalidade do contrato. A modalidade remete ao tipo de contrato, de acordo com os serviços pré-estabelecidos pela agência.
4. O supervisor submete os dados ao sistema.
5. O sistema verifica se as informações estão corretas ou são suficientes para o cadastro, e em caso afirmativo, realiza o mesmo.

Pós-condição:

O contrato deve ter sido gravado no sistema e aparecido na lista de contratos.

d) Criar Projeto

Atores:

Supervisor.

Pré-condição:

O supervisor deve ter feito *log-in* e obtido autorização do sistema, assim como ter realizado as etapas anteriores.

Fluxo de Eventos:

1. O caso de uso começa quando o supervisor seleciona “Projeto”
2. O supervisor clica no ícone “Adicionar Projeto”
3. O supervisor preenche os campos com o título, descrição, data de início, data de término, status do evento, seleciona um colaborador responsável e a qual contrato o projeto pertence.
4. O supervisor submete os dados ao sistema.
5. O sistema verifica se as informações estão corretas ou são suficientes para o cadastro, e em caso afirmativo, realiza o mesmo.

Pós-condição:

O projeto deve ter sido gravado no sistema e aparecido na lista de projetos vigentes.

e) Montar Equipes

Atores:

Supervisor.

Pré-condição:

O supervisor deve ter feito *log-in* e obtido autorização do sistema, assim como ter realizado a primeira etapa deste fluxo.

Fluxo de Eventos:

1. O caso de uso começa quando o supervisor seleciona “Equipes”

2. O supervisor clica no ícone “Adicionar Equipes”
3. O supervisor preenche os campos com o nome da equipe, o número de participantes, seleciona um colaborador responsável, e aloca a mesma para determinado projeto.
4. O supervisor submete os dados ao sistema.
5. O sistema verifica se as informações estão corretas ou são suficientes para o cadastro, e em caso afirmativo, realiza o mesmo.
6. Caso seja necessário adicionar outro projeto para a mesma equipe, clica-se em “adicionar novo projeto à equipe” e seleciona-se outro projeto previamente cadastrado.

Pós-condição:

A equipe deve ter sido gravado no sistema e aparecido na lista de projetos vigentes.

f) Montar Equipes

Atores:

Supervisor.

Pré-condição:

O supervisor deve ter feito *log-in* e obtido autorização do sistema, assim como ter realizado a primeira etapa deste fluxo.

Fluxo de Eventos:

1. O caso de uso começa quando o supervisor seleciona “Equipes”
2. O supervisor clica no ícone “Adicionar Equipes”
3. O supervisor preenche os campos com o nome da equipe, o número de participantes, seleciona um colaborador responsável, e aloca a mesma para determinado projeto.
4. O supervisor submete os dados ao sistema.
5. O sistema verifica se as informações estão corretas ou são suficientes para o cadastro, e em caso afirmativo, realiza o mesmo.

6. A nova equipe deve aparecer na lista, e agora são adicionados todos os colaboradores que participarão da mesma, para tanto clica-se no link “colaboradores”.

7. Na tela “Adicionar Membros da Equipe” são selecionados os colaboradores.

8. O supervisor submete os dados ao sistema.

Pós-condição:

A equipe deve ter sido gravada no sistema e aparecido na lista de equipes.

g) Cadastrar Tarefas

Atores:

Supervisor e colaboradores.

Pré-condição:

O usuário deve ter feito *log-in* e obtido autorização do sistema.

Fluxo de Eventos:

1. O caso de uso começa quando o supervisor seleciona “Tarefas”
2. O usuário clica no ícone “Adicionar nova Tarefa”
3. O usuário preenche os campos com o nome da tarefa, descrição, seleciona qual o colaborador responsável, a qual projeto a mesma é relacionada, o status, data de início e de término. Usuários Supervisores podem atrelar a tarefa a qualquer projeto, enquanto que colaboradores poderão atrelar aos projetos que são responsáveis.
4. O supervisor submete os dados ao sistema.
5. O sistema verifica se as informações estão corretas ou são suficientes para o cadastro, e em caso afirmativo, realiza o mesmo.

Pós-condição:

A tarefa deve ter sido gravada no sistema e aparecido na lista de tarefas.

h) Registrar Histórico

Atores:

Supervisor e colaboradores.

Pré-condição:

O usuário deve ter feito *log-in* e obtido autorização do sistema.

Fluxo de Eventos:

1. O caso de uso começa quando o supervisor seleciona “Tarefas”
2. É então mostrada a listagem de todas as tarefas, e ao lado de cada tarefa existe um botão chamado “histórico”, e é nele que são adicionadas informações pertinentes a cada.
3. O usuário clica em “histórico” e abre uma nova janela com a lista dos cadastros prévios caso tenha ocorrido.
4. O usuário clica em “adicionar Histórico” e preenche os campos de título, descrição, a data que tal assunto ocorreu e seleciona um tipo previamente cadastrado de atendimento, e caso não exista é possível realizar o cadastro de novo tipo.
5. O usuário submete os dados ao sistema.
6. O sistema verifica se as informações estão corretas ou são suficientes para o cadastro, e em caso afirmativo, realiza o mesmo.

Pós-condição:

O histórico deve ter sido gravado no sistema e aparecido na lista de histórico da tarefa.

i) Realizar *feedback*

Atores:

Colaboradores e clientes.

Pré-condição:

O usuário deve ter feito *log-in* e obtido autorização do sistema.

Fluxo de Eventos:

1. O caso de uso começa quando o supervisor finaliza uma tarefa.
2. É então disparado um *e-mail* para o cliente com uma mensagem pré-determinada informando que tal tarefa foi concluída e com um *link* para que o cliente possa determinar uma nota para tal.
3. O cliente seleciona a nota.
4. O cliente submete os dados ao sistema e recebe uma mensagem de agradecimento.
5. O sistema verifica se as informações estão corretas ou são suficientes para o cadastro, e em caso afirmativo, realiza o mesmo.

Pós-condição:

O *feedback* deve ter sido gravado no sistema e aparecido nos dados estatísticos.

3.3 Modelagem e implementação do banco de dados

Conforme a figura 2 foi modelado um modelo do banco de dados previsto para o sistema, que demonstra as relações entre as tabelas que serão utilizadas durante a programação.

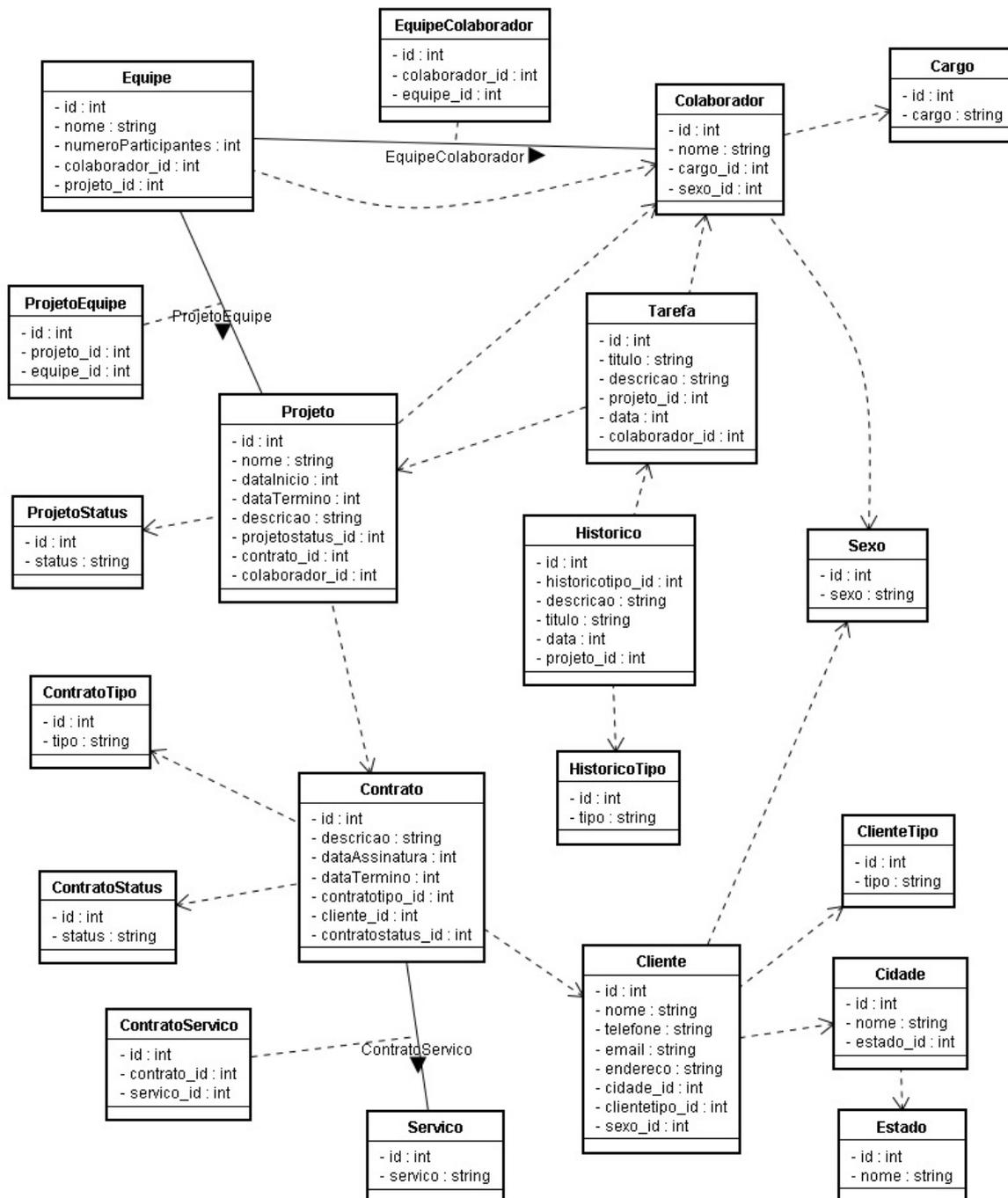


Figura 2 – Modelo do banco de dados

De acordo com o modelo apresentado é possível visualizar as relações existentes entre as tabelas, onde as principais são detalhadas abaixo:

- **Cliente:** Um cliente é cadastrado no sistema, e a ele poderá ser atribuído um ou mais contratos.
- **Contratos:** A assinatura de um contrato é uma maneira formal e prática para dar abertura a um ou mais projetos contratados pelo

cliente. Nele também são descritos os serviços contratados e detalhado o custo de cada.

- **Serviços:** Cada contrato possui um ou mais serviços atrelados.
- **Projeto:** Um projeto geralmente é dividido em etapas, aqui chamada de tarefas. Cada etapa tem como responsável por seu desempenho uma equipe previamente criada.
- **Tarefa:** Um projeto geralmente é formado por um conjunto de etapas, estas aqui designadas para a conclusão por cada equipe
- **Equipe:** É um conjunto de colaboradores (empregados da Agência Digital) organizados de acordo com suas funções dentro da empresa.
- **Colaboradores:** Existe um nível hierárquico de acesso a funções do sistema, na qual todas são acessíveis somente ao usuário que possuir um cargo de liderança.

3.4 Wireframes

Foi realizado um guia visual básico para a interface proposta. A esta etapa se dá o nome de Wireframe. Seu objetivo é ajudar o desenvolvedor de sistemas a entender o que se espera do sistema, sugerir a estrutura e o relacionamento entre as páginas, mostrando assim os requisitos funcionais básicos. A figura 3 é o esboço criado para o sistema.

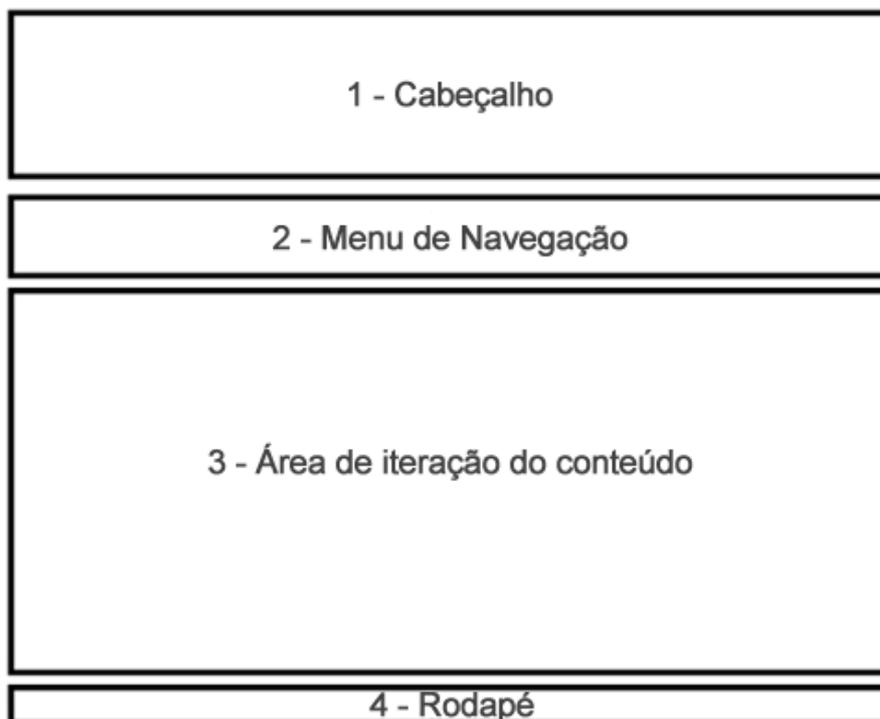


Figura 3 – Wireframe do sistema

A localização 1 é o cabeçalho, composto pelo nome do sistema, situado ao lado esquerdo, e pelo nome do usuário autenticado juntamente com um botão de saída do sistema. Na localização 2 temos o menu de navegação pelas principais funções do sistema gerenciador. O item 3 é o local previsto para a interação de todo o conteúdo previsto no sistema. O topo deste sempre mostrará um pequeno menu horizontal que oferece as funções possíveis para o conteúdo. E por último, o item 4 é um espaço para um rodapé com os créditos e nome do autor.

Excetuando a tela inicial de autenticação, onde não constará o item 2, todas as telas do sistema são como a figura 3.

3.4 Implementação

Para o desenvolvimento do sistema foi utilizado a plataforma Windows 7. Optou-se pela instalação separada de cada um dos softwares necessários, estes relacionados abaixo:

- Apache Server versão 2.2
- PHP versão 5.3.2
- Banco de dados MySQL 5.1.51
- Biblioteca AJAX PHPLiveX 2.6

Os softwares acima listados foram escolhidos por serem disponibilizados de forma gratuita em seus sites oficiais, além de serem amplamente utilizados e por sua facilidade de ajuda pela comunidade de adeptos de cada um deles.

3.5 Telas do sistema

3.5.1 Tela de autenticação

Com o banco de dados modelado e o esboço do layout criado, iniciou-se a implementação. Inicialmente foi criada a tela de autenticação, conforme a figura 4.

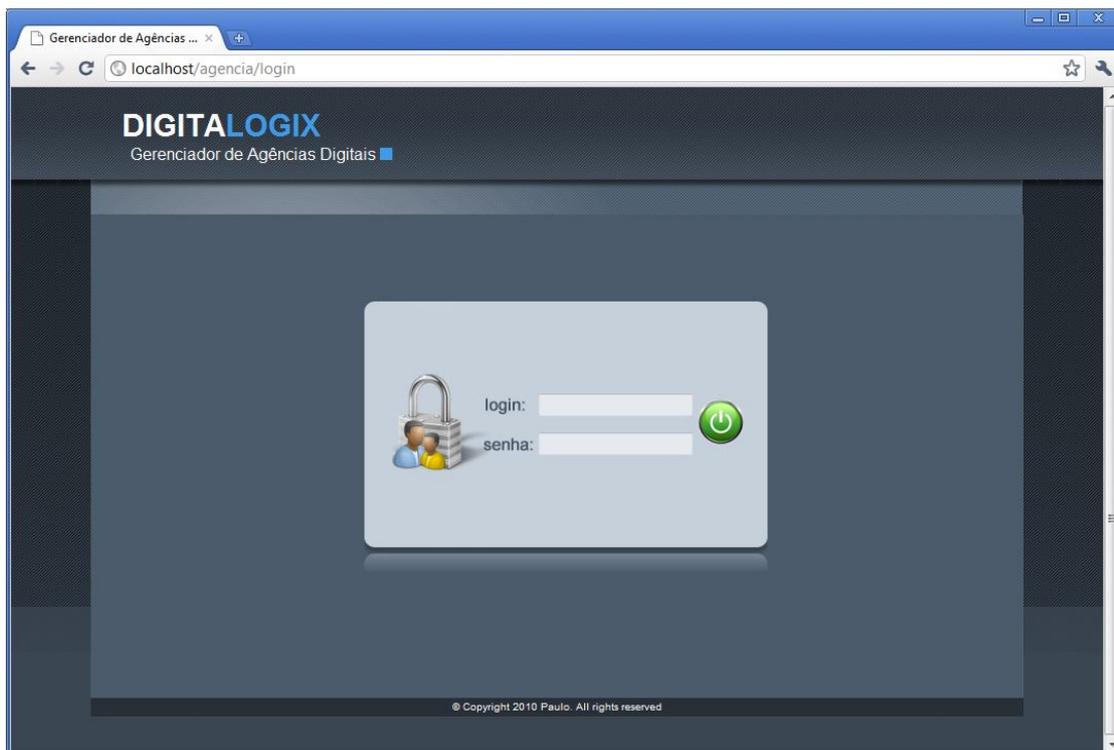


Figura 4 – Tela de autenticação

Nela o usuário deve inserir o seu nome de usuário, assim como a palavra-chave. Novos usuários devem solicitar a criação de um usuário (no sistema chamado de colaborador) ao usuário Supervisor. O controle é dado com a utilização de variáveis de sessão. Estas são variáveis globais, possibilitando assim a verificação se o usuário tem a permissão necessária para acessar determinado conteúdo.

3.5.1.1 Código de autenticação

O arquivo abaixo é o conteúdo do arquivo usuário.php, que contém a classe.

```
< ?php
class Usuario {
    var $nome, $senha;

    function getNome() {
        return this->nome;
    }

    function Usuario($nome,$senha) {
        $this->nome = $nome;
```

```

//vamos encriptar a senha
$this->senha = md5($senha);
}

function autentica() {
    //Aqui está o metodo de acesso ao banco.
    $nome = $this->nome;
    $senha = $this->senha;
    $query = "SELECT nome, senha FROM usuarios WHERE nome=$nome
AND senha=$senha";
    $resultados = mysql_query($query) or die(mysql_error());
    if (mysql_num_rows($resultados)>0) {
        $this->geraSessao($this);
        return true;
    }
    else{
        return false;
    }
}

function geraSessao($usuario) {
    session_start();
    $_SESSION['usuario'] = $usuario;
}
}
?>

```

Os dados captados do formulário HTML de login e senha, são tratados através do código abaixo:

```

< ?php
include("usuario.php");
$nome = $_POST["nome"];
$senha = $_POST["senha"];
var usuario = new Usuario($nome,$senha);
if (usuario->autentica()) {
    header("location:principal.php");
}
else {
    header("location:login.php");
}
?>

```

3.5.2 Tela inicial

Após a autenticação ser feita com sucesso o usuário tem acesso a tela inicial do sistema. A figura 5 mostra o layout esperado para o acesso de um usuário Supervisor. No topo direito consta o nome do usuário autenticado, assim como um botão para efetuar o *logout*. No corpo do sistema é apresentada uma mensagem de

boas-vindas, assim como os próximos prazos de projetos. Inclui ainda um link de acesso rápido a *websites* e serviços úteis em Agências Digitais.



Figura 5 – Tela de um usuário supervisor

Na tela inicial também consta acesso a algumas estatísticas do software, assim com uma área livre, para inserção de conteúdo geral, como notícias internas ou recados.

3.5.3 Tela Clientes

A figura 6 exemplifica o layout da tela de clientes. Esta é a primeira etapa na inserção de informações no gerenciador, onde são cadastrados os principais dados do novo cliente.

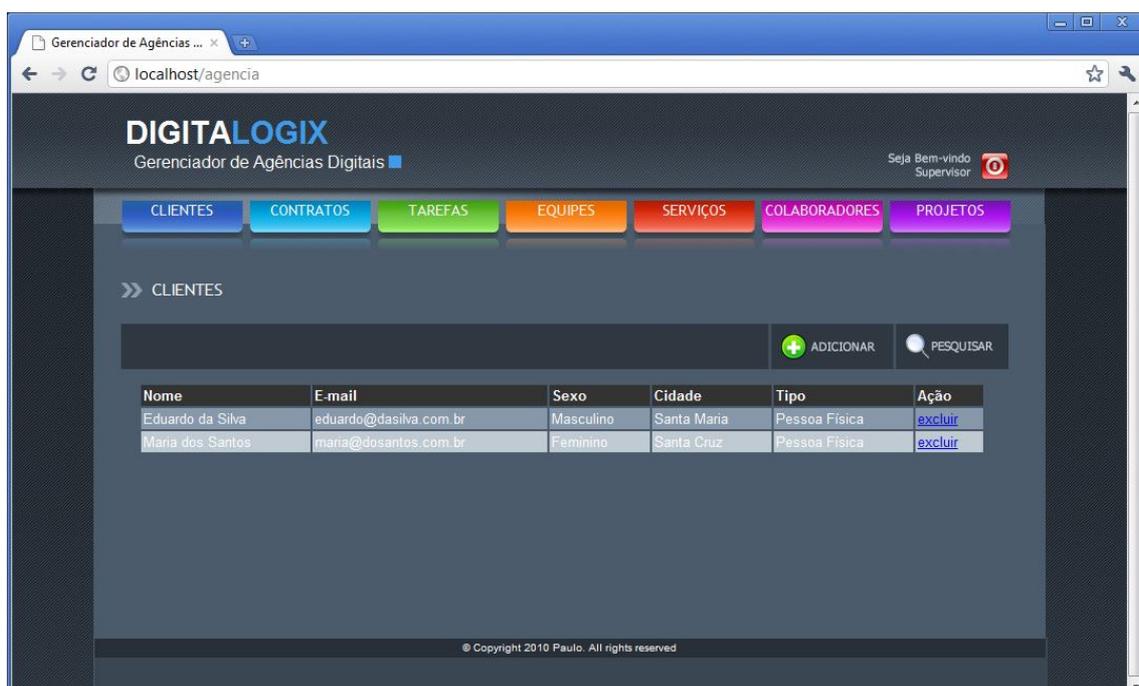


Figura 6 – Tela Clientes

Como nas demais telas, aqui é possível adicionar novo cliente, assim como fazer uma pesquisa na lista de clientes, função essa ótima quando se possui uma extensa lista de clientes. Ao clicar em qualquer dos clientes, é possível alterar os dados cadastrais ou excluir o mesmo.

3.5.4 Tela Contratos

A próxima etapa é a inclusão de um contrato. A figura 6 demonstra tal tela, onde é possível ver que um contrato é relacionado a um cliente. É cadastrado aqui a data de assinatura do contrato, assim como seu prazo de validade. É também relacionado uma lista de serviços, que discrimina os dados e custos de cada serviço prestado.

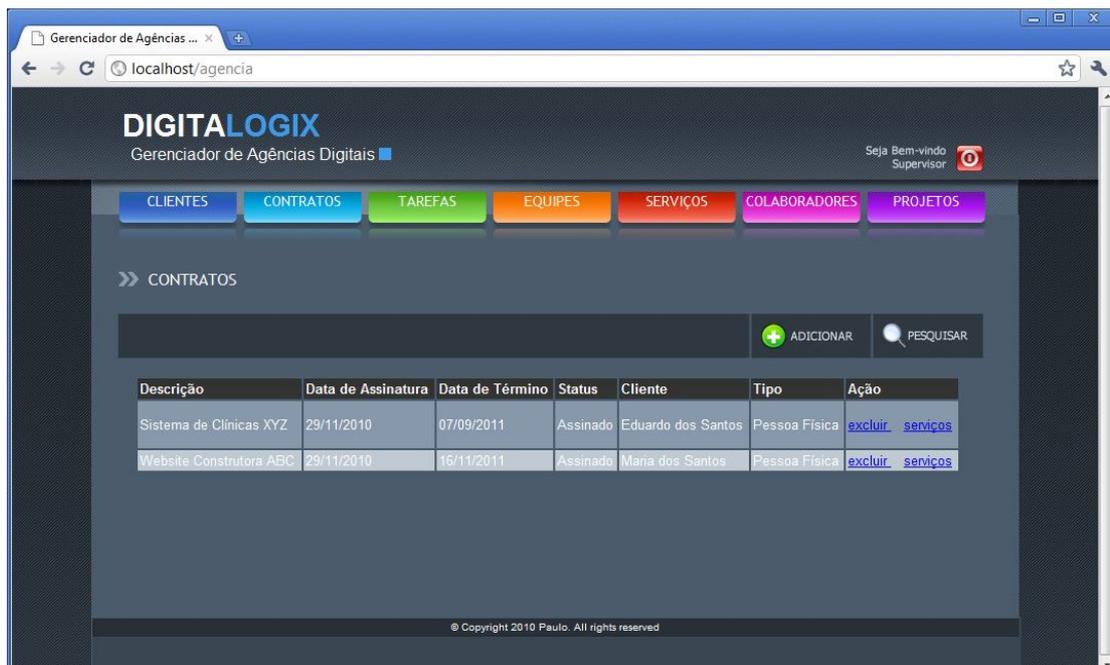


Figura 7 – Tela contratos

3.5.5 Tela Tarefas

A tela tarefas aglutina todas as pendências que o usuário autenticado tem com os projetos cadastrados. Ela apresenta uma visão geral das tarefas, assim como acesso ao histórico das tarefas já realizadas pelo usuário devidamente autenticado.

3.5.6 Tela Equipes

Nesta tela o usuário com permissões de supervisor pode criar equipes de trabalho, e estas podem ser alocadas para um determinado projeto. É importante notar que um usuário poderá participar de uma ou mais equipes.

3.5.7 Tela Serviços

Conforme já explicado, aqui são incluídos os serviços realizados pela empresa. No caso de empresas que utilizam preço tabelado há um campo para a inserção do valor cobrado pelo serviço.

3.5.8 Tela Colaboradores

Usuários são cadastrados e categorizados através desta tela. As categorias também são cadastradas de forma dinâmica, cabendo ao usuário supervisor o pré-cadastro dos cargos. No momento da inserção de um novo colaborador será solicitado o cargo que o mesmo possui e se tem permissões de supervisor. Para aqueles que não possuem permissão é mostrado somente a lista dos colaboradores e seus respectivos dados.

3.5.9 Tela Projetos

Aqui são criadas os projetos e especificados as etapas que cada projeto possui. Durante este cadastro é definido os grupos responsáveis por cada tarefa, sendo que um colaborador será o líder.

Os prazos que foram inseridos ao sistema há mais tempo são mostrados no topo da tabela, assim como seu responsável e o status da mesma (vide figura 8).

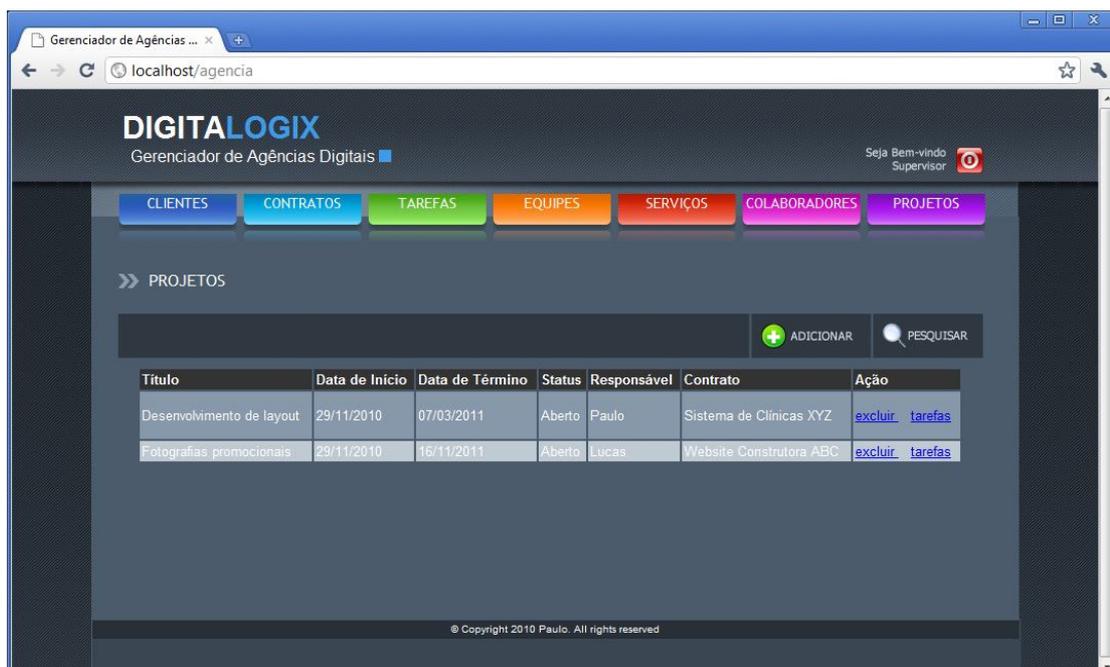


Figura 8 – Tela Projetos

A inserção de um novo projeto é exemplificada na figura 9. Nela deverá ser relacionado ao projeto um responsável e um contrato.

Título *

Descricao *

Data de Início *

Data de Término *

Status *

Responsável *

Contrato *

Figura 9 – Tela inserção de um novo projeto

3.5.10 Tela Estatísticas

Através desta tela é possível acompanhar alguns dados relevantes. Nesta primeira versão pode-se descobrir quantos projetos estão em andamento, quantos estão atrasados, quais usuários possuem mais tarefas pendentes e qual o colaborador que finalizou mais tarefas em um determinado prazo de tempo.

Os gráficos são gerados através da ferramenta JS Charts, uma ferramenta JavaScript que permite a criação de diversos tipos de gráficos.

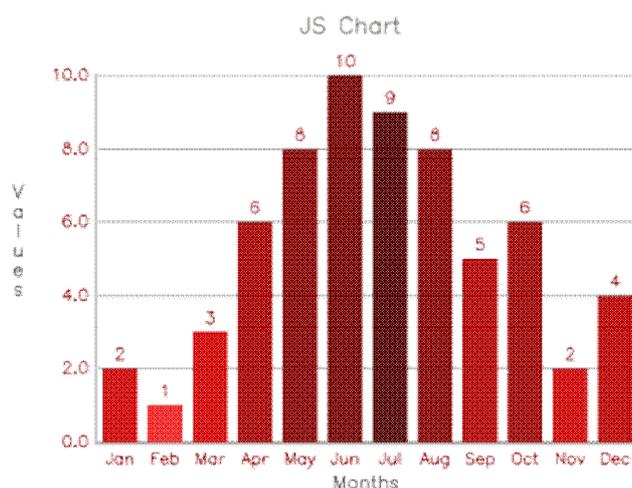


Figura 10 – Exemplo de gráficos gerados

Fonte: <http://www.jscharts.com/>

4 CONSIDERAÇÕES FINAIS

O presente trabalho apresentou uma proposta para o desenvolvimento de um software gerenciador de Agências Digitais. Para este fim foi utilizado a linguagem PHP com orientação a objetos, funções AJAX e o modelo MVC. Um sistema de Workflow foi respeitado, para organizar o fluxo de trabalho.

A principal contribuição na realização deste trabalho foi a concepção de um sistema de software capaz de trazer agilidade e organização para Agências Digitais. Seus processos estarão organizados e documentados, de forma centralizada e compartilhada. Assim, estas empresas podem focar nos serviços, tendo a tranquilidade de que seus processos estão sempre organizados e acessíveis.

Outro foco deste trabalho foi apresentar a utilização da linguagem PHP juntamente com o paradigma da orientação a objetos e a utilização de AJAX. Foi constatado que é uma alternativa muito eficaz para a programação de sistemas WEB, possibilitando a divisão de tarefa entre vários programadores e o reuso de código.

Sendo assim, se torna possível que seja futuramente implementada novas funcionalidades a este sistema de software, a fim de automatizar novas tarefas realizadas e as peculiaridades de determinada Agência Digital.

5 REFERÊNCIAS

BOOCH, Grady. **Object-Oriented Analysis and Design with Applications**. Redwood City, USA: Benjamin/Cummings, 1994.

CATAPAN, Araci H. e FIALHO, Francisco A. Pereira. **Pedagogia e Tecnologia: a comunicação digital no processo pedagógico**. Universidade Federal de Santa Catarina – UFSC, 2009. Disponível em: <<http://diaadiaeducacao.pr.gov.br/>>. Acessado em: setembro de 2010.

DALL’OGLIO, Pablo. **PHP: Programando com Orientação a Objetos**. 2ª edição. São Paulo: Novatec Editora, 2009.

FAYAD, Mohamed and SCHIMIDT, Douglas C. **Object-oriented application frameworks**. *Communications of the ACM*. New York: 1997.

FERRARI, Pollyana. **Jornalismo Digital**. 2ª edição. São Paulo: Contexto, 2004 - (Coleção Comunicação).

GUSTAFSON, David A. **Engenharia de Software**. Porto Alegre: Bookman, 2003.

MORONI, Herbert. **Criação de sites em AJAX**. São Paulo: Digerati Books, 2007.

USIONORO, C. H. **Tecnologia de workflow: O impacto de sua utilização nos processos de negócio**.: Dissertação (Mestrado em Administração) – Universidade de São Paulo, 2003.