

**ANÁLISE DE UM SISTEMA DE INFORMAÇÃO WEB APLICADO À GESTÃO DE
ESCOLA DE INFORMÁTICA**

**ANALYSIS OF A WEB INFORMATION SYSTEM APPLIED MANAGEMENT
SCHOOL OF COMPUTING**

**ANÁLISIS DE UN SISTEMA DE INFORMACIÓN WEB APLICADO A LA GESTIÓN
DE ESCUELA DE INFORMÁTICA**

ROGER CRISTHIAN GOMES ¹
OSVALDO CÉSAR PINHEIRO DE ALMEIDA ²

Recebido em janeiro de 2010. Aprovado em fevereiro de 2010.

¹ Tecnólogo em Informática para a Gestão de Negócios pela Faculdade de Tecnologia de Botucatu.

² Bacharel em Ciências da Computação pela USP. Mestre em Ciências da Computação e Matemática Computacional pela USP. Professor da Faculdade de Tecnologia de Botucatu.

ANÁLISE DE UM SISTEMA DE INFORMAÇÃO WEB APLICADO À GESTÃO DE ESCOLA DE INFORMÁTICA

RESUMO

Uma das tarefas de um empreendedor é escolher um sistema de informação computadorizado para a gestão de seu negócio, independente do porte e do ramo de atuação. Ter que definir se o sistema de informação será modelado para uso local, também conhecido como *stand-alone*, ou desenvolvido para *web*, está se tornando cada vez mais comum, já que a Internet, com suas características, facilita muito o trabalho do gestor. Porém, não se pode simplesmente deduzir, ou levar em conta apenas tendências tecnológicas e de mercado, para resolver uma questão que implicará na forma de operação, administração e gestão da empresa. Para escolher entre um ou outro modelo de sistema é necessário uma análise das vantagens e desvantagens, de cada modelo em relação ao negócio em questão. Este trabalho teve como objetivo elencar as principais características intrínsecas aos aplicativos *web* e dos aplicativos *stand-alone*. O estudo desses dois modelos de aplicativos foi baseado na análise de um sistema de informação aplicado a uma empresa de prestação de serviços em treinamento de informática. Para a análise desse sistema de informação, foram realizados o levantamento dos principais requisitos e a modelagem de um protótipo. Foi proposto o desenvolvimento do sistema em ambiente *web*, utilizando a plataforma JAVA em conjunto com o gerenciador de banco de dados MySQL, por serem ferramentas completas, bem documentadas, gratuitas, e com recursos que contribuem para garantir a funcionalidade e a qualidade do sistema de informação *web*.

PALAVRAS-CHAVE: Engenharia de *Software*. JAVA. Modelagem. Sistemas WEB.

**ANALYSIS OF A WEB INFORMATION SYSTEM APPLIED MANAGEMENT
SCHOOL OF COMPUTING**

ABSTRACT

One of the tasks of an entrepreneur is choose a computerized information system for the management of your business, regardless of their size and field of expertise. Having to determine if the information system will be modeling for local use, also known as stand-alone, or developed for the web, is becoming increasingly common, as the Internet, with its characteristics, greatly facilitates the work of the manager. However, can not simply deduct or take into account only the technological trends and market to resolve an issue that will require in the form of operation, administration and management. To choose between one or another type of system is necessary to examine the advantages and disadvantages of each model in relation to the business in question. This study aimed to list the main features intrinsic to web and stand-alone applications. The study of these two types of applications was based on analysis of an information system applied to a company to provide services in computer training. For the analysis of the information system were carried out a survey of the main requirements and modeling of a prototype. It was proposed to develop the system in a web environment, using the JAVA platform with the database manager MySQL, because the tools are complete, well documented, free, and with features that help to ensure the functionality and quality of the information system web.

KEYWORDS: Software Engineering. JAVA. Modeling. WEB Systems.

ANÁLISIS DE UN SISTEMA DE INFORMACIÓN WEB APLICADO A LA GESTIÓN DE ESCUELA DE INFORMÁTICA

RESUMEN

Una de las tareas de un empresario es elegir un sistema de información computadorizado para la gestión de su negocio, independiente de su tamaño y del campo de experiencia. Definir si el sistema de información será adaptado para uso local, también conocido como *stand-alone*, o desarrollado para *web*, está cada vez más común, ya que la Internet, con sus características, facilita mucho el trabajo del gestor. Sin embargo, no se puede simplemente deducir, o llevar en cuenta apenas tendencias tecnológicas y de mercado, para resolver una cuestión que implicará en la forma de operación, administración e gestión de la empresa. Para elegir entre uno u otro modelo de sistema es necesario un análisis de las ventajas y desventajas, de cada modelo en relación al negocio en cuestión. Este trabajo tuvo como objetivo reunir las principales características intrínsecas a los aplicativos *web* y a los aplicativos *stand-alone*. El estudio de esos dos modelos de aplicativos se basó en el análisis de un sistema de información aplicado a una empresa de prestación de servicios en entrenamiento de informática. Para el análisis de ese sistema de información fueron realizados la recopilación de los principales requisitos y el modelaje de un prototipo. Se propuso el desenvolvimiento del sistema en ambiente *web*, utilizando la plataforma *JAVA* en conjunto con la gerencia de banco de datos *MySQL*, por ser herramientas completas, bien documentadas, gratuitas, y con recursos que contribuyen para garantizar la funcionalidad y la calidad del sistema de información *web*.

PALABRAS CLAVE: Ingeniería de *Software*. *JAVA*. Modelaje. Sistemas *WEB*.

1 INTRODUCAO

O acesso à informação em tempo real e em qualquer lugar é uma necessidade dos empresários e gestores de empresas de todos os portes e ramos de atividade. Poder acompanhar o desenvolvimento dos negócios e gerenciá-los sem a necessidade de estar presente fisicamente nas sedes das empresas vem se tornando, mais que uma tendência, uma exigência no dia a dia dos empresários. Com o surgimento da Internet, sua disseminação e popularização, além do advento e avanço das técnicas de criptografia de dados e segurança de rede, isso se tornou possível.

Optar por desenvolver um sistema de informação (SI) voltado para *web*, simplesmente denominado como SIW por Zaneti Junior (2003), ou um sistema de informação *stand-alone* é uma tarefa que tem se tornado cada vez mais comum. Desenvolver sistemas *web* demanda conhecimento específico, investimento em *hardware* e *software* diferenciado em relação aos aplicados em sistemas locais, além da contratação de serviços que não são comuns em sistemas sem essa característica. A escolha das ferramentas de desenvolvimento, da forma de hospedagem e da maneira como será tratada a base de dados são pontos que merecem uma atenção especial nessa categoria de sistema.

Algumas características dos sistemas de informação voltados para *web*, tais como: disponibilidade e mobilidade, entre outras, levam as organizações a optarem, sem um devido estudo de caso, por esse tipo de aplicativo. Apesar dos fatores citados soarem como vantagens, é preciso, antes da contratação de serviços para desenvolver e implantar o sistema, analisar os requisitos do sistema para o negócio em questão e a viabilidade do investimento, já que existe a opção de trabalhar com sistemas locais *stand-alone*.

Empresas que prestam serviços no ramo de treinamento em informática possuem como algumas de suas rotinas: cadastro de alunos, geração de contrato e de parcelas, distribuição de turmas e professores, entre outros. Executar e gerenciar essas operações, bem como outras rotinas administrativas, seria muito mais rápido e confiável com o auxílio de um sistema de informação que utilizasse recursos dos computadores e das ferramentas de informática e tecnologias da informação disponíveis.

A aquisição ou contratação do desenvolvimento de um SI específico de uma empresa geram custos com produtos, serviços e infraestrutura. Empenhar esses recursos de maneira eficiente é de extrema importância para as empresas, principalmente para aquelas que estão sendo concebidas. O SI escolhido deve ser

utilizado pela empresa por um longo período de tempo, e se não for bem escolhido e definido, pode ser necessário adaptações e até mesmo sua substituição por completo, gerando aumento de custos para a empresa. Para evitar que isso ocorra, é necessário realizar uma análise prévia dos recursos demandados e dos esforços necessários para aquisição e implantação do SI.

Para minimizar os custos de implementação do SI, é possível optar por ferramentas gratuitas, sem abrir mão da qualidade. A plataforma JAVA, que utiliza a linguagem de programação JAVA, foi desenvolvida para trabalhar em diversas plataformas, conferindo aos aplicativos, com ela desenvolvidos, portabilidade e total integração com a Internet, além de ser gratuita. O SGBD (sistema de gerenciamento de banco de dados) MySQL é um dos principais SGBDs gratuitos existentes e tem como uma de suas principais características a fácil integração com o ambiente da Internet.

Este trabalho teve como objetivo elencar as principais características intrínsecas aos aplicativos *web* e dos aplicativos *stand-alone*. O estudo desses dois modelos de aplicativos foi baseado na análise de um sistema de informação aplicado a uma empresa de prestação de serviços em treinamento de informática. Com isso, foi possível gerar a modelagem

lógica da porção essencial do sistema de informação a ser implementado.

Como objetivo secundário, este trabalho oferece subsídios para que empresas de ramos de atuação semelhantes à empresa aqui abordada consigam escolher, de maneira consciente e consistente, entre um ou outro tipo de aplicativo para gestão de suas organizações.

A necessidade de um sistema de informação computadorizado eficaz para a gestão de uma empresa é indiscutível. A questão é definir se é melhor optar por um SI local (*stand-alone*) ou um SI voltado para *web*. Para responder a esta questão, é necessário ponderar muitos fatores. A possibilidade de acessar as informações de qualquer localidade, sendo necessário apenas um computador conectado à Internet, é uma característica de grande peso nessa decisão.

Considerando as características primordiais de uma empresa de prestação de serviços de treinamento em informática, é possível que o uso de um SI *web* seja o mais indicado. Porém, não é possível afirmar categoricamente qual SI é mais indicado sem antes realizar o levantamento dos requisitos do sistema e estudar a viabilidade do investimento. Por esta razão, foram elencadas e analisadas as necessidades de serviços, *hardware* e *software*, além da demonstração das

principais vantagens e, por ventura, desvantagens, do uso de um sistema *web*.

2 ANÁLISE DE *SOFTWARES*

O conceito de engenharia de *software* evolui paralelamente ao conceito de sistema de informação. Os dois temas destinam-se a auxiliar as organizações a tomarem decisões sobre o foco e o escopo das mesmas. Os princípios da engenharia de *software* são usados tanto no projeto quanto no produto final do desenvolvimento de sistemas, com uma relação entre um e outro, muito íntima, sendo que, quando o processo está correto, a qualidade do produto é garantida (REZENDE, 2005).

A engenharia de *software*, segundo Pressman (2006), proporciona detalhes de como construir um *software*. Isso abrange uma gama de elementos e métodos definidos e classificados como fundamentais. Além disso, oferece aos desenvolvedores subsídios para elaboração de *softwares* com grande qualidade produtiva. Permite também, ao gerente do projeto de *software*, um controle pleno sobre o processo de implementação.

Entre os métodos que a engenharia de *software* proporciona, algumas tarefas são primordiais, como um projeto que mostre a estrutura dos dados, um planejamento completo que demonstre sua estimativa incluindo a análise de requisitos

Tékhnē e Lógos, Botucatu, SP, v.1, n.2, fev. 2010.

do sistema a ser implementado. Incluem ainda, algoritmo, codificação, testes e manutenção constante. Atualmente existem ferramentas para auxílio aos métodos da engenharia de *software*. Essas ferramentas oferecem apoio automatizado para cada um dos métodos já mencionados. Tais métodos sustentam um desenvolvimento racional e adequado de sistemas de computador (PRESSMAN, 2006).

Nas seções 2.1, 2.2 e 2.3 há uma breve descrição de alguns dos principais métodos e ferramentas da engenharia de *software*.

2.1 Levantamento e análise de requisitos

O levantamento e a análise de requisitos de *software*, de maneira geral, consistem em identificar todos os processos e dados envolvidos no problema analisado, descobrir suas necessidades e objetivos, para, de forma adequada, documentá-los para análise, comunicação e implementação (TOGNERI, 2002).

Os requisitos para um sistema são as descrições das funções que este sistema deve incorporar, e das restrições que devem ser satisfeitas, ou seja, os requisitos do sistema definem as circunstâncias nas quais ele deve operar, e o que ele deverá fazer. (SOMMERVILLE, 2003).

Segundo Sommerville (2003), efetuar o levantamento de requisitos não se

restringe apenas a perguntar às pessoas o que elas desejam e necessitam. Devem ser utilizadas várias técnicas, para que o resultado seja satisfatório e completo. Algumas técnicas como questionários, prototipagem, entrevistas, observação de processos, dinâmicas de grupo, são utilizadas por analistas de sistemas e engenheiros de requisitos.

2.2 Diagrama de Fluxo de Dados

Um DFD (Diagrama de Fluxo de Dados) é um esquema que facilita a visualização do tráfego dos dados entre as entidades externas (que são o destino ou a origem dos mesmos), dos processos que transformam esses dados, e ainda os depósitos de dados que representam o armazenamento das informações necessárias para o funcionamento do sistema. Assim, o DFD é composto por processos, fluxos de dados, entidades externas e depósito de dados. Os processos são operações responsáveis por transformar os dados lógica ou fisicamente. Um fluxo de dados se compara a uma via expressa por onde passam os pacotes de dados, os quais identificam os processos. Já os depósitos de dados representam o local onde o sistema armazenará suas informações (GANE e SARSON, 1983).

2.3 Modelo Entidade Relacionamento

Conforme definido por Setzer (1987), um Modelo Entidade Relacionamento (MER) representa graficamente as entidades existentes no mundo real e os relacionamentos que existem entre elas. Essa representação de entidades e de seus relacionamentos facilita a visualização de como o sistema de informação precisa simular isso internamente. Além das entidades e dos relacionamentos entre elas, ambos podem ter atributos, que são usados para descrever propriedades e características dos mesmos.

Uma entidade pode ser entendida como sendo uma representação abstrata de algo do mundo real que deverá ser monitorada. Uma entidade pode representar um objeto, um organismo social, um ser, entre outros. Como exemplo, o conjunto de funcionários de uma empresa é uma entidade. Uma entidade no MER é representada por um retângulo (CHEN, 1990).

Já um relacionamento é a representação abstrata do relacionamento entre duas ou mais entidades. Da mesma maneira que as entidades, os relacionamentos são modelos em conjuntos, sendo um conjunto de relacionamentos, o produto cartesiano dos conjuntos de entidades envolvidas. Os relacionamentos são representados por um losango (na sua

forma mais usual) contendo um verbo para indicar a ação (CHEN, 1990).

3 DESENVOLVIMENTO WEB

3.1 JAVA

A plataforma JAVA é constituída de uma linguagem e uma série de recursos (bibliotecas) nativos do JAVA. A linguagem JAVA foi criada como parte de um projeto chamado “Green Project”, em 1991, pela empresa americana Sun Microsystems. A princípio se chamava OAK e foi originalmente concebida para ser aplicada no controle de eletrodomésticos como TV, CD players, microcomputadores, entre outros, como simples ferramenta de programação. Como se mostrou mais poderosa do que era a intenção, passou a ser amplamente utilizada (LEITE, 2006).

Entre as várias vantagens de JAVA em relação a outras linguagens de programação destaca-se a independência de plataforma, o que para Internet é muito importante, pois não se sabe sob qual plataforma de *software* e *hardware* o usuário está navegando. O fato de ser uma linguagem orientada a objetos facilita a reutilização de código, favorecendo assim o desenvolvedor e diminuindo o tempo de implementação, além de ter um conjunto vasto e bem documentado de bibliotecas (FIGUEIRA, 2005).

3.2 MySQL

O MySQL é um gerenciador de banco de dados que utiliza a linguagem padrão SQL. É muito utilizado em aplicações para a Internet. Entre os bancos de dados com código-fonte aberto é o mais popular. Existem milhares de instalações do MySQL em todo o mundo, entre as quais de alguns sites como Google, Nasa e Suzuki com alto volume de armazenamento e tráfego de dados (PRATES e NIEDERAUER, 2006).

Apesar de possuir uma tecnologia complexa de banco de dados o MySQL tem um custo baixo o que o torna uma alternativa bastante atrativa. Vem sendo adotado por desenvolvedores *web* e departamentos de TI (Tecnologia da Informação) devido a suas características, entre as quais se destacam: velocidade, escalabilidade e confiabilidade (PRATES; NIEDERAUER, 2006).

O gerenciador de banco de dados MySQL utiliza o SQL como base, sua implementação do tipo cliente-servidor, é um servidor de banco de dados multiusuário, imprime rapidez e flexibilidade nas transações com a base de dados (CALDEIRA, 2006).

4 SISTEMAS WEB

Pela ótica da engenharia de *software*, que incluem procedimentos como o gerenciamento, o suporte, o desenvolvimento, a análise, entre outros, aplicativos *stand-alone* e aplicativos *web* são, em sua essência, equivalentes. Entretanto, algumas diferenças com relação ao acesso concorrente às informações e aos tratamentos das mesmas, a multiplicidade de perfis e contextos, entre outras, são marcantes, demonstrando a preocupação dos aplicativos *web* em relação à navegabilidade por meio do sistema (PRESSMAN, 2006).

Devido ao fato de que os computadores não são mais componentes isolados como eram há pouco tempo, os conceitos de sistema de informação se expandiram e são mais complexos. Os computadores interagem entre si através das redes, podendo se comunicar, trocar informações e trabalhar um mesmo processo distribuídamente. Sendo a Internet a maior rede de computadores do mundo, e tendo qualquer computador, devidamente configurado, acesso a ela, esse avanço nos sistemas de informação se concretizou (LAUDON e LAUDON, 1998).

Com o avanço das tecnologias da Internet e o advento da *web*, as empresas passaram a enxergar a Internet como um meio de divulgar seus negócios no mundo.

Tékhnē e Lógos, Botucatu, SP, v.1, n.2, fev. 2010.

Esses avanços tecnológicos permitiram também que clientes fizessem transações *online* por meio de sistemas de informação *web*. A grande contribuição da *web* para as organizações não parou na área comercial, mas se expandiu para área gerencial, permitindo que os gestores de redes de empresas administrem suas filiais, em outras cidades, estados e países. Isso transformou a *web* em um apoio para áreas organizacionais de empresas (RÉ, 2002).

Mesmo com as declarações insistentes de que a *web* representa novas regras que definem um novo paradigma, os desenvolvedores percebem, a cada dia, que todos os conceitos de desenvolvimento aprendido anteriormente, com as aplicações tradicionais, ainda se aplicam. Fazendo um paralelo, escrever código HTML é como programar, criar aplicações para navegador é desenvolver sistemas de *software*, e tudo isso se beneficiando dos princípios básicos da engenharia de *software* (CONSTANTINE e LOCKWOOD, 1999).

5 METODOLOGIA

Este trabalho iniciou com o levantamento de informações sobre o desenvolvimento de aplicações voltadas para *web*. Analisar essas informações e compará-las com as informações sobre sistemas de informação *stand-alone* foi parte importante do trabalho desenvolvido.

Para atingir os objetivos deste trabalho, foi necessário, além do levantamento e revisão de literatura, uma análise do sistema de informação a ser proposto e futuramente implementado.

O caso proposto neste trabalho é o desenvolvimento de um sistema de informação, para o gerenciamento de uma empresa de prestação de serviços em treinamento de informática.

A análise deste caso inclui o levantamento de requisitos, a análise dos usuários e seus respectivos perfis, entre outras etapas da metodologia de engenharia de *software*, tais como: modelagem dos dados usando o modelo de entidade relacionamento (MER), a modelagem dos processos usando o diagrama de fluxo de dados (DFD, ou fluxograma de dados), e a criação dos modelos de navegação.

Com a análise e modelagem do sistema de informação definido, foi possível observar que o desenvolvimento desse SI seria mais adequado caso fosse aplicado à plataforma *web*. Desta maneira, o trabalho de criação de modelos de navegação foi baseado no desenvolvimento *web*.

5.1 Levantamento e análise de requisitos

Nesta fase foram levantados os requisitos do sistema. A partir das necessidades da empresa, de suas rotinas de

Tékhnē e Lógos, Botucatu, SP, v.1, n.2, fev. 2010.

trabalho, e de suas operações. Os dados necessários para isso foram coletados a partir de entrevistas com funcionários e administradores de uma empresa, em seus respectivos ambientes de trabalho. Em seguida, esses requisitos foram analisados e implementados na documentação básica do SI proposto.

5.2 Modelagem de dados

A partir das informações coletadas e dos requisitos definidos, foram modelados os dados conforme especificações e as necessidades da empresa. Foi utilizada a modelagem de dados usando MER (Modelo Entidade Relacionamento), o que gerou o diagrama de entidade relacionamento (DER) do sistema. Foi realizada a modelagem dos processos usando DFD (diagrama de fluxo de dados).

5.3 Modelos de navegação

Os modelos de navegação são telas de interface com o usuário, que são apresentadas no navegador (*browser*), criadas previamente para definir, com uma melhor perspectiva visual, como o sistema interage com o usuário.

Nesta fase, foram criadas as telas principais do sistema de informação *web*, as quais serão usadas futuramente no *software* implementado.

5.4 Comparação entre *stand-alone* e *web*

Como o objetivo deste trabalho é definir qual o modelo de SI mais adequado ao negócio e a empresa do caso estudado, foi necessário comparar todas as características de cada modelo de sistema de informação, suas vantagens e desvantagens em relação ao problema proposto.

Tendo em mãos as características de cada modelo, os requisitos do sistema e a modelagem de dados, foi possível gerar subsídios para comparar os dois modelos e então concluir qual o mais indicado.

6 RESULTADOS E DISCUSSÃO

6.1 Análise de Requisitos

A proposta do trabalho estava focada no desenvolvimento de um sistema de informação *web*. Desta maneira, foi proposta a criação de um site onde, a partir da tela inicial será realizado o *logon* (acesso ao aplicativo *web*), com a utilização de assinatura digital e protocolo *https* (usando o nome de usuário e a senha). Baseado nas informações colhidas, foram relacionados os perfis dos usuários que terão acesso ao sistema, com os respectivos tipos e níveis de acesso e atribuições.

Foi proposta a criação de uma tabela para armazenar os dados referentes aos

usuários do SI. Esses usuários são cadastrados na base de dados e gerenciados pelo administrador do SI e/ou, pelos diretores da instituição com conhecimento para isso.

A partir da digitação das informações para *logon*, o módulo de programação responsável por autenticar o usuário, verifica na tabela de usuários a existência e autenticidade das informações digitadas (usando criptografia e página segura *https*). Em seguida, confirmada a autenticidade dos dados, o usuário é direcionado ao módulo do aplicativo que é pertinente ao seu perfil.

6.1.1 Módulos usuário

A proposta é que cada usuário tenha acesso apenas aos módulos pertencentes ao seu perfil, permanecendo transparente a existência dos outros módulos do aplicativo. Com isso, cada usuário tem conhecimento somente do que lhe cabe como função e atribuições. Como exemplo, pode-se citar os usuários do setor financeiro que terão acesso apenas a esse módulo no aplicativo, desconhecendo o aplicativo como um todo. Para qualquer outra informação sobre a instituição, o usuário pode acessar o site institucional, que é totalmente aberto e irrestrito. Isso ajuda a manter o aplicativo e o usuário focado nas funcionalidades administrativas, enquanto o

site se incumbe da divulgação da instituição.

6.1.2 Área do aluno

Uma funcionalidade implementada no site institucional é a “Área do aluno”, onde os alunos matriculados terão acesso (por meio da inserção de usuário e senha) a algumas informações de âmbito acadêmico. Assim, os alunos poderão verificar notas, frequência, trabalhos a desenvolver, mudanças de datas de aula e provas (por feriados ou falta de professores), mural de recados e alteração de dados, como telefone e endereço para contato. Apesar da alteração de dados poder ser realizada pelo usuário (aluno), a base de dados do sistema mantém os dados antigos, para resguardar as informações fornecidas por responsáveis durante o processo de matrícula.

A área de aluno não possibilita o acesso nem visualização dos módulos do aplicativo. Para isso, o processo de *logon* no sistema é totalmente separado do processo de *logon* da área de aluno.

6.1.3 Informações e procedimentos

O fluxo de entrada de informações no SI segue a sequência dos procedimentos que são efetuados nos casos de visita à escola, por interessados em informações sobre os cursos e serviços prestados, *Tékhne e Lógos*, Botucatu, SP, v.1, n.2, fev. 2010.

matrícula de alunos, por si próprios, ou por seus responsáveis (quando menores de idade), e também nos casos de pagamentos de parcelas dos cursos em andamento.

Um exemplo desse processo seria o caso em que um interessado chega à unidade escolar para saber sobre um dos cursos e efetuar a matrícula de um dependente. Esse procedimento corriqueiro será efetuado nos seguintes passos: primeiramente o usuário do sistema deve estar “logado”, ou efetuar o *login* no sistema.

Na tela do aplicativo o usuário escolhe a opção “Cursos e Turmas”, que se encontra no lado esquerdo da tela. Essa opção abre a tela de informações sobre os cursos, com seus respectivos preços, formas de pagamento, turmas disponíveis e horários.

Na sequência, se o interessado não desejar efetuar a matrícula por algum motivo, ele pode ser cadastrado na relação de “visitantes”, para que futuramente, quando preenchidas suas aspirações de preço, horários, ou cursos, o visitante possa ser contatado. A proposta é cadastrar esses dados de interesse para que o próprio SI proponha o contato quando os requisitos forem preenchidos. Para esse procedimento usa-se a opção “Cadastro” da mesma tela. Devido à construção da base de dados utilizando o conceito de especialização de tabelas, o cadastro de visitantes, alunos,

clientes, funcionários e todos os dados que utilizam campos iguais, que nos exemplos são: nome, endereço, telefone, cidade e CEP, usam a mesma tela inicial de cadastro, e a mesma tabela de dados base.

Havendo o interesse do visitante em efetuar a matrícula os procedimentos são os mesmos descritos acima. Porém, na caixa de opções “Tipo de cadastro”, ao invés de visitante pode ser escolhido a opção “Cliente”, seguindo com a inserção de dados específicos como CPF e RG, para a geração do contrato de prestação de serviços. Em seguida, é possível optar pela opção “Cliente Aluno” (quando o próprio cliente irá cursar) ou “Aluno Dependente” (quando o Cliente está efetuando a matrícula de um dependente, menor de idade ou não). Caso seja “Cliente Aluno”, o cadastro será encerrado, prosseguindo com a inserção dos dados referentes ao curso e a turma em que o aluno será matriculado. Caso seja “Aluno Dependente”, será necessário especificar outros dados específicos do aluno, como nome, data de nascimento e sexo para, em seguida, entrar com os dados referentes ao curso escolhido.

6.2 Diagrama de Fluxo de Dados

A Figura 1 (Anexo) ilustra parte do Diagrama de Fluxo de Dados (DFD) abordada no trabalho para o desenvolvimento do sistema analisado.

6.3 Modelo Entidade Relacionamento

A Figura 2 (Anexo) ilustra parte do Diagrama de Entidade Relacionamento (DER) gerado a partir dos requisitos levantados aplicando o Modelo de Entidade Relacionamento (MER).

6.4 Modelos de navegação *web*

Como o sistema de informação deste trabalho é proposto para a plataforma *web*, todas as telas possuem um *layout* voltado para uso via *browser*. As telas de navegação oferecem uma boa percepção de como o SI irá interagir com o usuário, e como ele usará o sistema para atingir seus objetivos de uso.

Caso fosse um sistema de informação *stand-alone*, o *layout* seria bem diferente dos propostos neste trabalho. A plataforma *desktop* exige uma formulação de telas com uma distribuição e formas diferentes.

Todo o *layout* do sistema proposto foi desenvolvido a partir de um *layout* base. O uso de uma tela base ajuda, pois a construção gráfica de quase todos os outros módulos do aplicativo parte desse *layout* proposto inicialmente, mantendo um padrão visual em todo o sistema. A tela base do sistema serve tanto como tela inicial do *site* institucional da empresa, como tela inicial do aplicativo *web*. Esta é uma das

vantagens do desenvolvimento de um sistema de informação voltado para *web*, pois todos os seus componentes visuais atendem aos dois propósitos: como o *site* e como o aplicativo de gestão.

6.4.1 Layout de miniaturas

O conceito de miniaturas está implementado de maneira a permitir a exibição dos vários módulos do sistema de uma só vez. Com algumas características desse novo conceito, a tela, ilustrada na Figura 3, permite observar como é possível aproveitar o conceito de miniaturas no processo de navegação para o aplicativo *web* proposto.

6.4.2 Layout para usuário

Aproveitando o conceito de miniaturas de tela, que permite uma visão mais ampla dos módulos utilizados, foi proposto o desenvolvimento de um mecanismo para destacar os módulos mais utilizados nos últimos acessos do usuário. Assim, quando o usuário acessa o sistema a partir do seu *login*, são exibidas as miniaturas dos últimos seis módulos acessados. Isso permite um acesso mais rápido e eficiente para os módulos mais utilizados do sistema.

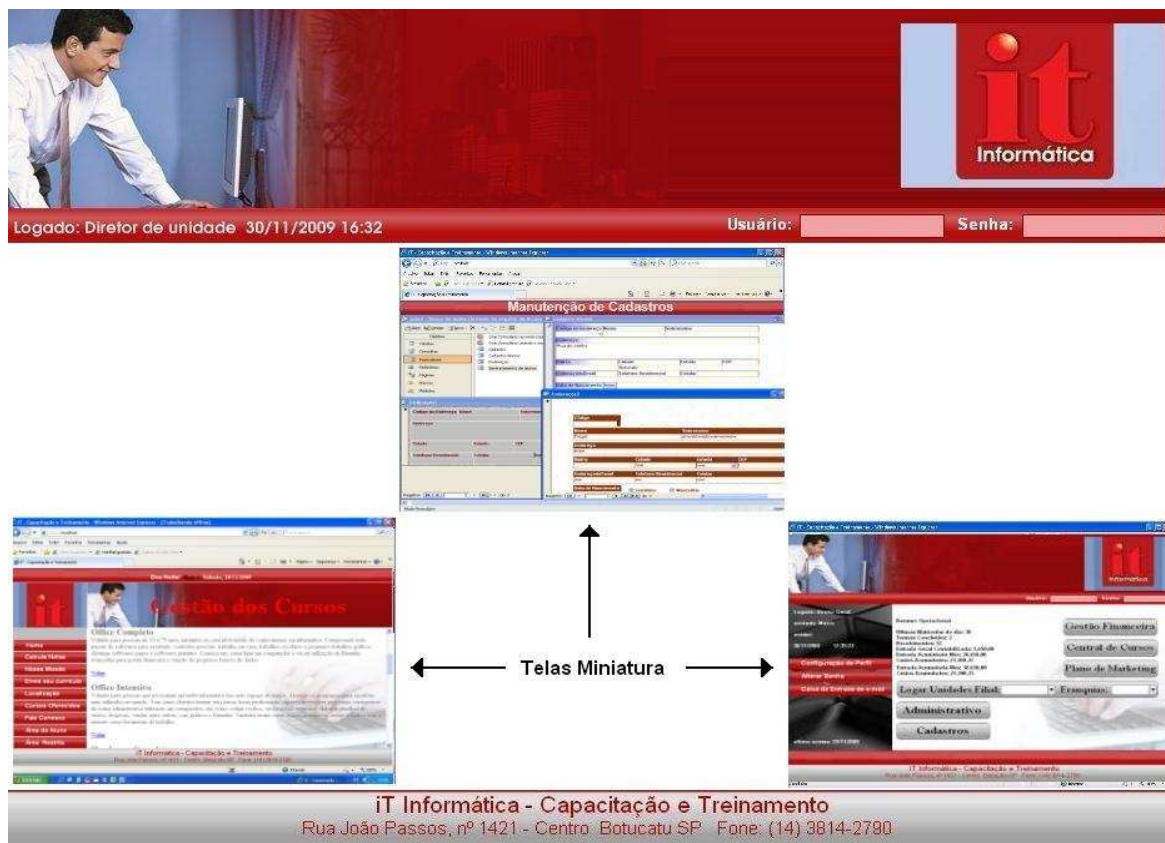


Figura 3. Ilustração do modelo de miniaturas utilizada no plano de navegação dos módulos do sistema.

Além disso, foi proposto um mecanismo que permitisse que o usuário tivesse uma visão mais ampla de cada miniatura, antes de acessá-la diretamente. Assim, ao posicionar o *mouse* em uma das miniaturas do sistema, a tela desta miniatura é ampliada, para que o usuário tenha uma melhor perspectiva de qual módulo ele irá utilizar. Após esta ampliação, o usuário poderá acessar o módulo que for de seu interesse. A Figura 4 ilustra esses mecanismos de miniatura e ampliação da miniatura.

6.5 Comparação entre *stand-alone* e *web*

Os aplicativos *stand-alone* são sistemas executados localmente e usam compartilhamento das bases de dados por meio de redes locais de computadores. Esse aplicativo tem como principais características sua plataforma de instalação, a forma de desenvolvimento, a necessidade de instalação de uma cópia do *software* em cada computador da rede local, entre outras. É importante ressaltar que existem tecnologias que proporcionam certa mobilidade aos usuários de sistemas *stand-alone*, porém são tecnologias complexas e

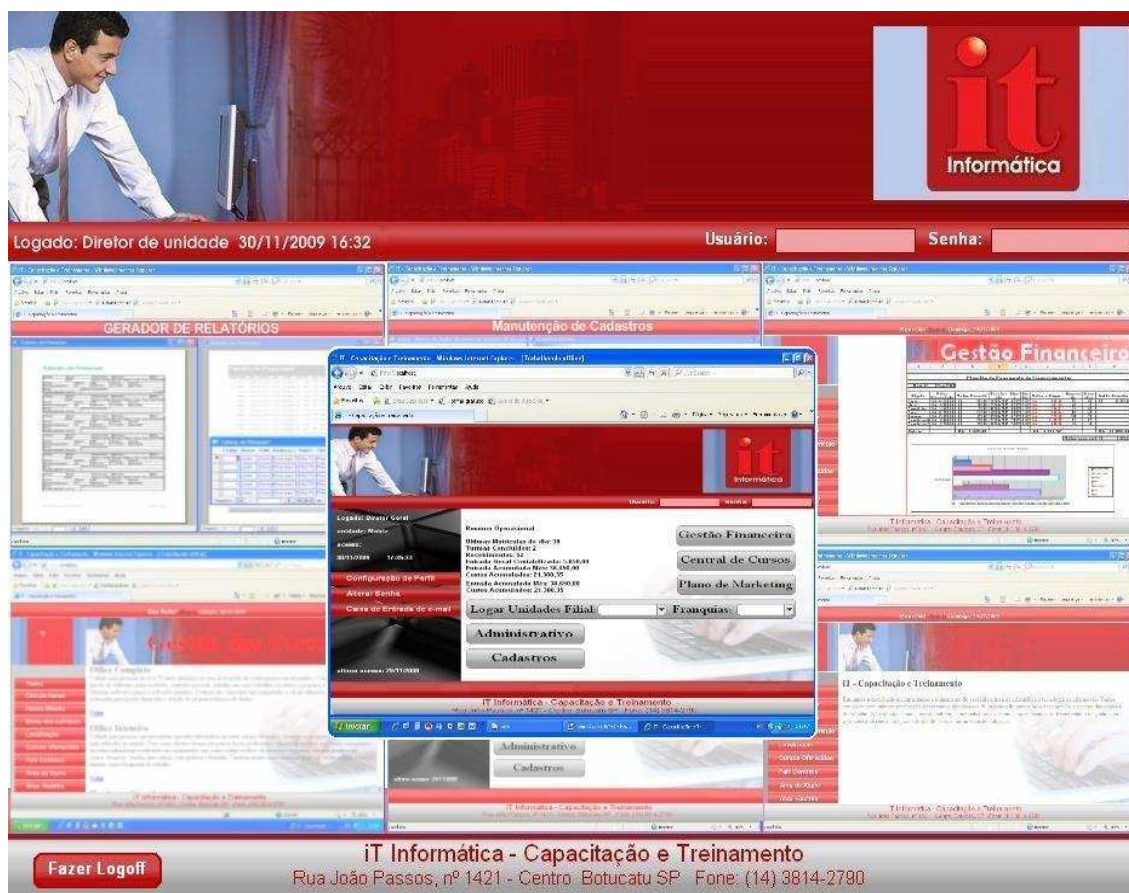


Figura 4. Ilustração dos mecanismos de miniatura e ampliação no plano de navegação do sistema.

com custo e risco que devem ser levados em consideração ao utilizá-las. Além disso, sem o uso da Internet, esta mobilidade é bem restrita.

Como vantagem, aplicativos *stand-alone* apresentam um modelo de desenvolvimento já bem difundido, com muitas soluções já implementadas e vasta documentação. Contudo, não é possível explorar de maneira eficiente toda a mobilidade oferecida pelos avanços tecnológicos, característica que enriqueceria as aplicações das empresas. Problemas como a necessidade de atualizar o *software* em todos os computadores, onde as cópias foram instaladas, e a restrição de acesso aos SI apenas localmente, são alguns dos problemas que não acompanham os aplicativos *web*. Para atualizar uma versão do aplicativo *web*, basta implementar a modificação no servidor do aplicativo. A partir daquele momento, todos os usuários que acessarem o *site* do aplicativo estarão rodando a versão atualizada do *software*. Para empresas que dependem de acesso, a qualquer momento, a uma base de dados atualizada em tempo real e a partir de qualquer lugar, aplicativos *stand-alone* acabam sendo insuficientes e inviáveis.

Para a engenharia de *software*, os aplicativos *web* e *stand-alone* não se distinguem. Todos os conceitos e métodos aplicáveis aos aplicativos tradicionais *stand-alone* são também aplicáveis, em sua

totalidade, ao processo de desenvolvimento dos aplicativos *web*. Apesar da maior parte da metodologia de implementação ser comum aos dois tipos de aplicativos, uma característica particular dos aplicativos *web* é de extrema importância, a mobilidade. Essa característica deve ser considerada no momento da escolha do tipo de aplicativo que será desenvolvido para a empresa.

Os aplicativos *web*, em geral, são sistemas de informação criados para serem executados a partir de *browsers*, encontrados na maioria das instalações de computadores, como um *software* básico. Mas não é somente esta característica que determina se um aplicativo é *web* ou não, algumas outras são implícitas a esse tipo de aplicação. A princípio, um aplicativo pode ser implementado para *web* e ao mesmo tempo ser usado como um aplicativo *stand-alone*, bastando para isso que seja instalado no mesmo computador e executado a partir do seu navegador (sem o acesso a partir de outros computadores). Afirmar que um aplicativo é *web* implica, necessariamente, que este aplicativo esteja disponível a partir da Internet (ou outra rede), e que ele execute funcionalidades de um aplicativo, não somente um *site*, apesar de manter o *layout* de *site web*.

A necessidade e possibilidade de administrar a organização praticamente de qualquer lugar do mundo (principalmente quando a empresa gerencia filiais e

franquias), ou também administrar bases de dados distribuídas por regiões geográficas diferentes tem tornado a *web* a plataforma mais adequada para a implementação dos *softwares* e sistemas de informação.

7 CONCLUSÃO

Um sistema de informação desenvolvido para *web* pode ser empregado tanto em uma intranet, extranet ou Internet. Os custos de desenvolvimento do SI são determinados pela sua complexidade e não pela plataforma de implantação. Os investimentos em uma infraestrutura de plataforma *web* são, na maioria dos casos, maiores que os empenhados em plataforma *stand-alone*. Contudo, um aplicativo desenvolvido para *web* não precisa, necessariamente, ser implantado em uma plataforma *web*, é possível usá-lo em uma infraestrutura de rede local, como seria com uma implementação *stand-alone*. Isso consolida a ideia de que as necessidades e os requisitos, da estrutura organizacional do negócio, devem ser observados para o desenvolvimento do sistema de informação. Assim, caso seja observado que o desenvolvido do SI for beneficiado pelo paradigma de desenvolvimento *web*, com todas as suas características se transformando em vantagens para o negócio, o mais indicado é o investimento em um SI voltado para *web*.

Tékhnē e Lógos, Botucatu, SP, v.1, n.2, fev. 2010.

O caso estudado nesse trabalho, baseado no levantamento e análise dos seus requisitos, no desenvolvimento e modelagem do sistema proposto, e considerando ainda os resultados, gera argumentos suficientes que justifiquem a adoção da plataforma *web*, como base para o desenvolvimento de um sistema de informação para escolas de treinamento de informática.

Esse trabalho contribui para dirimir dúvidas de outros gestores que, por sua vez, venham a ter a mesma dúvida com relação à escolha de um sistema de informação adequado ao seu negócio (*stand-alone* ou *web*). Nesse trabalho, há informações suficientes que corroboram as conclusões obtidas e apóiam a implementação de sistemas de informação *web* para empresas de perfil semelhante ao perfil da empresa estudada. De maneira genérica, foram elencadas as vantagens dos sistemas *web* em relação aos sistemas implementados sobre a plataforma *stand-alone*.

8 REFERÊNCIAS

CALDEIRA, B.P. **Alta disponibilidade-replicação de dados via MySQL, com ênfase em identificação e recuperação de falhas**. Monografia (Pós-graduação Lato sensu-Rede Linux) Universidade Federal de Lavras, 2006.

CHEN, P. **Gerenciando Banco de Dados: A Abordagem Entidade-Relacionamento**

para Projeto Lógico. São Paulo: McGraw-Hill, 1990.

CONSTANTINE L. L.; LOCKWOOD L. A. D. **Software for use: a practical guide to the models and methods of usage.** 1999.

FIGUEIRA, J.S. **Easy Java simulations - Modelagem computacional para o ensino de Física.** Revista Brasileira de Ensino de Física, Pato Branco, PR, v.27, n.4, p.613-618, ago. 2005.

GANE, C.; SARSON, T. **Análise Estruturada de Sistemas.** Rio de Janeiro: Editora LTC, 1983.

LAUDON, K. C.; LAUDON, J. P., **Sistemas de Informação: com Internet.** 4. ed. Rio de Janeiro: Editora LTC, 1998.

LEITE, M. **Técnicas de Programação: Uma abordagem moderna.** 1 ed. Rio de Janeiro: Editora Brasport, 2006.

PRATES, R.; NIEDERAUER, J. **MySQL 5 Guia de Consulta Rápida.** São Paulo: Editora Atlas, 2006.

PRESSMAN, R. **Engenharia de Software.** 6 ed. São Paulo: McGrawHill, 2006.

RÉ, R. **Um Processo para construção de Frameworks a partir da Engenharia Reversa de Sistemas de Informação baseados na Web: Aplicação ao Domínio de Leilões Virtuais.** Dissertação de Mestrado na Faculdade de Ciências de Computação e Matemática Computacional – Universidade de São Paulo, São Paulo, 2002.

REZENDE, D. A. **Engenharia de Software e Sistemas de Informação.** 3ed. Rio de Janeiro: Brasport, 2005.

SETZER, W. **Bancos de Dados.** 2 ed. Blücher: Editora Edgard, 1987.

SOMMERVILLE, I. **Engenharia de Software.** 6 ed. São Paulo: Addison-Wesley, 2003.

TOGNERI, D.F. **Apoio Automatizado à Engenharia de Requisitos Cooperativa.** Dissertação de Mestrado, Mestrado em Informática da UFES, 2002.

ZANETI JUNIOR, L. A. **Sistemas de informação baseados na tecnologia web: um estudo sobre seu desenvolvimento.** 1 ed. São Paulo: Editora Campus, 2003.

ANEXO

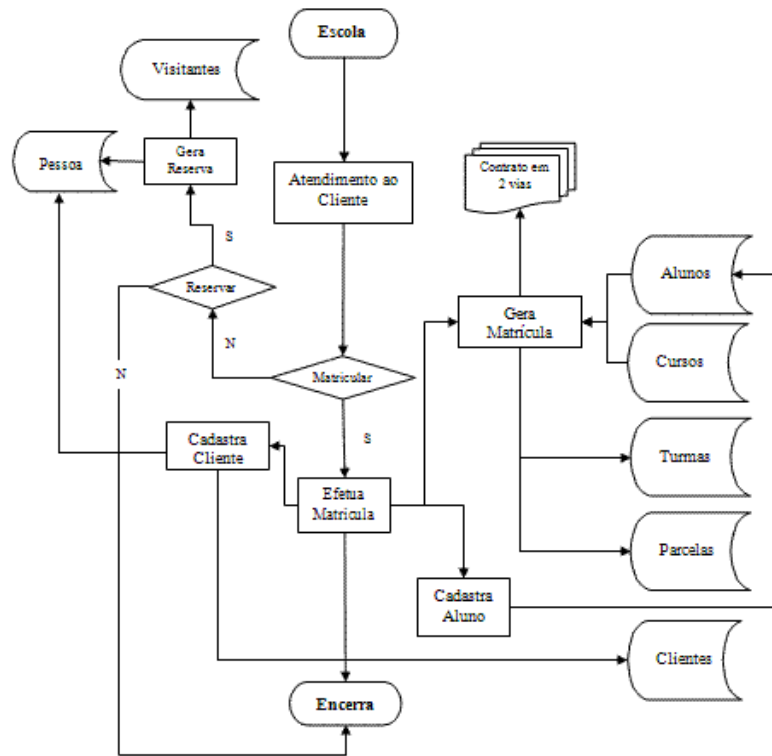


Figura 1. DFD (Diagrama de Fluxo de Dados) de parte do sistema proposto.

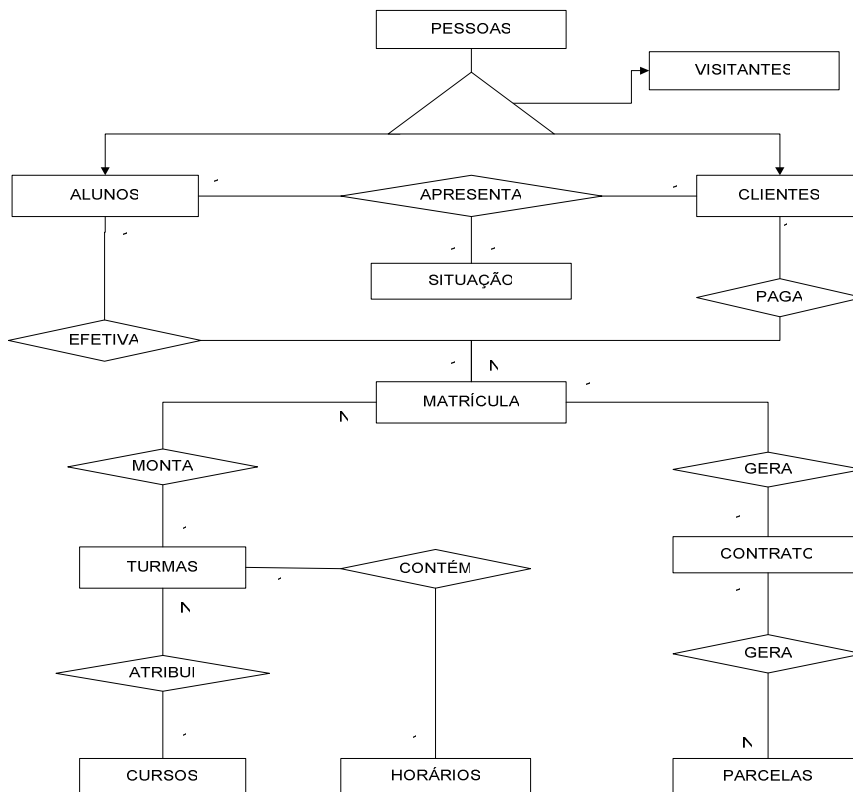


Figura 2. DER (Diagrama de Entidade Relacionamento) de parte do sistema proposto.