

ROMILDO FERREIRA DOS SANTOS

**ESTUDO DA VIABILIDADE DE IMPLANTAÇÃO DE UM SISTEMA DE
CONTROLE DE ENTRADA E SAÍDA DE VISITANTES NA EMEX EM
TEÓFILO OTONI - MG**

FACULDADES UNIFICADAS DE TEÓFILO OTONI
TEÓFILO OTONI – MG

2017

ROMILDO FERREIRA DOS SANTOS

**ESTUDO DA VIABILIDADE DE IMPLANTAÇÃO DE UM SISTEMA DE
CONTROLE DE ENTRADA E SAÍDA DE VISITANTES NA EMEX EM
TEÓFILO OTONI - MG**

Monografia apresentada ao Curso de Sistemas de Informação das
Faculdades Unificadas de Teófilo Otoni, como requisito parcial à obtenção do
título de Bacharel em Sistemas de Informação.
Área de Concentração: Desenvolvimento Web.
Orientador: Prof. Luiz Fernando Alves

TEÓFILO OTONI – MG
FACULDADES UNIFICADAS DE TEÓFILO OTONI
2017



FACULDADES UNIFICADAS DE TEÓFILO OTONI
NÚCLEO DE TCC / SISTEMAS DE INFORMAÇÃO
Autorizado pela Portaria 4.012 de 06/123/2004 – MEC

FOLHA DE APROVAÇÃO

A monografia intitulada: *Desenvolvimento e implantação de um Sistema de Controle de Visitas de uma empresa do ramo de explosivos em Teófilo Otoni,*

elaborada pelo aluno Romildo Ferreira dos Santos,

foi aprovada por todos os membros da Banca Examinadora e aceita pelo curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial da obtenção do título de

BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Teófilo Otoni, 20 de novembro de 2017


Professor Orientador: Luiz Fernando Alves Souza


Professor Examinador: Wilbert Viana Barbosa


Professor Examinador: Salim Ziad Pereira Aouar

A DEUS – razão de tudo
nas nossas vidas – razão do nosso existir.

AGRADECIMENTOS

Agradeço primeiramente a Deus por tudo que ele tem proporcionado em minha vida, pelo que ele tem me dado. Agradeço a minha família por todo o apoio e força que eles me oferecem, em especial a minha mãe que sem ela eu não estaria realizando este sonho, a minha companheira, esposa e amiga Rosenilda, meus queridos filhos Tárik e Mariana pelo incentivo que eles me deram. Obrigado a todos que acreditaram na minha vitória.

Não fiz o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser.

Mas graças a Deus, não sou o que eu era.

Martin Luther King

LISTA DE ABREVIATURAS E SIGLAS

BD – Banco de Dados

BSD - Berkley Standard Distribution

CSS - Cascading Style Sheets

FSF - Free Software Foundation

GNU - General Public License

GPL – General Public License

HTML - HyperText Markup Language

JSP - Java Server Pages

LGPL - Lesser General Public License

OO – Orientado Objeto

PHP - Hypertext Preprocessor

RDBMS - Relational Database Management System

SGBD – Sistema Gerenciador de Banco de Dados

SQL - Structured Query Language (Linguagem de Consulta Estruturada)

TI – Tecnologia da Informação

UML - Unified Modeling Language – Linguagem de Modelagem Unificada

RESUMO

Para que as organizações possam manter-se competitivas em um mercado cada vez mais acirrado, são necessários investimentos em alternativas que otimizem os processos e meios que melhorem o atendimento ao cliente, a partir da aquisição de ferramentas que visam melhorar a agilidade, a qualidade e a confiabilidade nos serviços prestados. Essa pesquisa tem como objetivo verificar a viabilidade de implantação de um sistema de controle de acesso na Emex em Teófilo Otoni/MG. Para realização do estudo valeu-se de pesquisa bibliográfica segundo os procedimentos de coleta, adotando a dedução como a base da investigação, com abordagem ao problema de natureza qualitativo e quantitativo, onde o estudo de caso objetiva auxiliar na tomada de decisão sobre o problema estudado. Em conclusão, afirma-se que a proposição de um sistema de informação para otimização do processo logístico além de viável economicamente, traz benefícios como a melhoria nos processos organizacionais.

Palavras-chave: Otimização, Viabilidade, Processos Organizacionais,

SUMÁRIO

INTRODUÇÃO	3
1 REFERENCIALTEÓRICO	7
1.1 FASES DE UM PROJETO.....	7
1.1.1 Projeto básico.....	7
1.1.2 Planejamento do projeto.....	8
1.1.3 Projeto de viabilidade.....	8
1.2 BANCO DE DADOS.....	8
1.2.1 Linguagem SQL.....	10
1.2.2 Sistema gerenciador de banco de dados - SGBD.....	10
1.2.3 MySQL.....	11
1.2.3.1 Capacidades do Mysql.....	12
1.2.4 Modelagem banco de dados.....	13
1.2.5 Modelo Conceitual.....	14
1.2.6 Modelo Lógico.....	14
1.2.7 Mysql Workbench.....	14
1.3 DESENVOLVIMENTO WEB.....	16
1.3.1 Plataforma de desenvolvimento.....	16
1.3.2 Linguagem de programação.....	16
1.3.3 Lógica de Programação.....	16
1.3.4 Php.....	17
1.3.5 Programação orientada a objetos.....	17
1.3.6 Programação Estruturada.....	18
1.3.7 Html.....	19
1.4 JAVASCRIPT.....	20

1.4.1 Cascading style sheets - CSS	20
1.4.1.1 Regras CSS.....	21
1.4.1.2 Estilos externos.....	21
1.4.1.3 Folhas de estilo importadas.....	21
1.4.2 Bootstrap	21
1.4.2.1 Suporte.....	22
1.5 ENGENHARIA DE SOFTWARE.....	22
1.5.1 Engenharia de Requisitos	23
1.5.2 Processos de Software	24
1.5.3 Modelos de Processos de Software	25
1.5.4 Objetivos	25
1.5.5 Modelos de Ciclo de Vida	26
1.6 SOFTWARE LIVRE.....	26
1.7 OPENSOURCE.....	27
1.8 UML.....	28
1.9 DIAGRAMA DE CASO DE USO.....	29
1.10 FRAMEWORK.....	29
1.10.1 Symfony	30
1.10.1.1 Framework PHP.....	30
1.10.2 Filosofia	31
1.10.3 Comunidade	31
1.10.4 Componentes Symfony	32
2 DESENVOLVIMENTO DO SISTEMA	33
2.1.1 Diagrama de Caso de Uso	33
2.1.2 Requisitos do sistema	34
2.1.3 Descrição dos casos de uso	34
2.1.4 Tabela de Relacionamento do sistema	36
2.1.5 Sistema (Controle de Acesso)	37
2.1.6 Tela de acesso ao sistema	37
2.1.7 Tela de pesquisa do sistema controle de acesso	38

2.1.8 Tela de cadastro de visitantes do sistema controle de acesso.....	39
2.1.9 Tela de cadastro de veículos do sistema controle de acesso.....	39
2.1.10 Tela de listagem de visitantes do sistema controle de acesso.....	40
CONSIDERAÇÕES FINAIS.....	42
REFERÊNCIAS.....	45

INTRODUÇÃO

A presente monografia concentrada na área de desenvolvimento web, tem como objetivo desenvolver um sistema capaz de auxiliar a Emex empresa do ramo de explosivos de Teófilo Otoni – MG, no controle das frequentes entradas e saídas de visitantes do seu parque industrial.

O Exército brasileiro, sob o decreto nº 3.665, de 20 de novembro de 2000, classifica um produto como controlado aquele que, por premissa básica, possui poder de destruição ou outra propriedade de risco. Esses produtos exigem a restrição de sua comercialização a pessoas físicas e jurídicas legalmente habilitadas e capacitadas técnica, moral e psicologicamente com o intuito de garantir segurança à sociedade. Nesse sentido, além de restringir a comercialização, é necessário que se restrinja também o acesso às áreas onde esses produtos são produzidos.

A Emex, produz e comercializa produtos da natureza citada acima. Como a necessidade do controle de acesso aos locais de produção é notória, o controle das pessoas que entram e saem das dependências da empresa deve ser organizado e eficiente.

Para resolução do problema citado, foi proposto o desenvolvimento de um sistema web que satisfaça as necessidades da empresa. Este sistema permitirá que se faça um controle mais detalhado, mais preciso, mais seguro das frequentes visitas na empresa. Além do mais o sistema trará alguns benefícios como um conhecimento a mais para seus usuários.

Vale lembrar que a área de concentração está redirecionada ao desenvolvimento web, especificamente para linguagem PHP, uma vez que hoje é uma das mais utilizadas mundialmente, por ser inovadora, versátil e eficaz, garantindo a todos os diversos serviços que ela dispõe.

Portanto, a presente monografia parte da seguinte questão investigativa:

Quais impactos o desenvolvimento de uma ferramenta automatizada para controlar as frequentes entradas e saídas de visitantes na Emex, empresa do ramo de explosivos de Teófilo Otoni – MG?

No decorrer da pesquisa, foram levantadas hipóteses a fim de obter respostas para questão investigativa, sendo elas:

H0: Não haveria a possibilidade de implantação deste sistema pois o mesmo não agregará valores à empresa e seus usuários.

H1: O desenvolvimento e a implantação deste sistema na empresa, possibilitaria um maior controle nas constantes entradas e saídas de seus visitantes.

H2: Poderia haver por parte dos usuários uma recusa pela implantação deste sistema na empresa.

H3: Haveria com a implantação deste sistema na empresa uma maior contribuição ao meio ambiente.

H4: Poderia este sistema agregar algum conhecimento a seus usuários.

O presente projeto visando atingir o seu objetivo geral que consiste em um estudo da viabilidade de implementação de um sistema de controle de entrada e saída de visitantes da Emex – Empresa Mineira de Explosivos de Teófilo Otoni, foram apresentados os objetivos específicos, sendo eles:

- Fazer o levantamento de requisitos para o sistema;
- Projetar os principais diagramas do sistema;
- Pesquisar sobre as tecnologias a serem usadas nesse projeto;
- Criar o banco de dados do sistema;
- Cadastrar as informações;
- Implementar o sistema e testar;

Quanto aos fins, a presente pesquisa do tipo descritiva, que tem característica a descrição do que está sendo observado. Para Vergara (1998, p.45), esse tipo de

pesquisa “expõe característica de determinada população ou de determinado fenômeno. Não tem o compromisso de explicar os fenômenos que descreve”.

Trata – se, ainda de uma pesquisa exploratória, pois caracteriza – se por ter como principal finalidade desenvolver, esclarecer e modificar conceitos e ideias, tendo em vista, a formulação de problemas mais precisos ou hipóteses pesquisáveis para estudos posteriores (Gil, 1999, p.43).

A pesquisa se apresenta como um estudo de caso, proporcionando uma avaliação mais abrangente, no entender de Gil (1999, p.78), “é caracterizado pelo estudo profundo e exaustivo de um ou de poucos objetos, de maneira a permitir conhecimento amplo e detalhado do mesmo”.

A investigação da pesquisa foi realizada na forma de pesquisa bibliográfica em livros, artigos científicos, monografias, sites confiáveis e vídeos, com o objetivo de verificar a possível implementação do sistema. Ao observar a forma de pesquisa, pôde considera-la como qualitativa, pois pretende verificar o problema e trazer ao mesmo uma proposta de solução através de um sistema simples e de fácil acesso. E também como quantitativa, pois foi realizado uma coleta de resultados da utilização do sistema para que os mesmos possam ser analisados e validados.

Quanto aos meios, o projeto proposto utilizou a pesquisa de campo e laboratorial, pois os dados levantados da investigação foram buscados no próprio ambiente, através de observações e entrevistas com o principal usuário, para que o mesmo possa dar opiniões e sugestões sobre o desenvolvimento da aplicação.

O método a ser utilizado será de característica indutiva, que de acordo com Gil (1999, p.28), “parte do particular e coloca a generalização como um produto posterior do trabalho de coleta de dados particulares”. É um processo mental que, para chegar ao conhecimento ou demonstração da verdade, parte de fatos particulares, comprovados, e tirar uma conclusão genérica.

Contudo foi proposto uma mudança no meio do projeto para se alcançar o objetivo, onde foi proposto que o desenvolvimento de implantação, fosse realizado a implementação de uma ferramenta, um framework gratuito e OpenSource, capaz de atender aos requisitos para implementação deste software.

Após várias pesquisas optou-se pelo framework que melhor atenderia à implementação deste projeto, a Symfony um framework de Aplicações Web, o que significa que ele permite que aplicativos web sejam criados em PHP!

Este trabalho está dividido em etapas, sendo a primeira nomeada como referencial teórico, onde é contemplado teoricamente as tecnologias utilizadas no desenvolvimento da aplicação, como Banco de Dados, Desenvolvimento Web, e Engenharia de Software. O segundo capítulo está relacionado com o desenvolvimento da aplicação, explicando cada fase do processo de desenvolvimento desde a coleta de requisitos até os testes realizados.

1 REFERENCIAL TEÓRICO

Afirma Tomanik (1994), esta etapa pretende aprofundar os conhecimentos teóricos sobre o tema, detalhar e expor as várias exposições sobre o assunto. Além de levantar os pontos de concordância e discordância das várias posições, colocando a confrontação de vários autores entre si, com as próprias conclusões do pesquisador, estabelecendo desta maneira, a base teórica para continuação do projeto.

Este capítulo irá aprofundar conhecimentos teóricos dos temas pertinentes a um estudo de viabilidade econômico-financeira. Desta forma, se determinam os pontos iniciais para cada uma das etapas do trabalho, com o intuito de atingir os objetivos propostos por este estudo.

1.1 FASES DE UM PROJETO

Apresentamos a seguir um resumo da sequência de fases porque passa o desenvolvimento de um projeto. Tais fases não são sequenciais e estanques entre si; elas são executadas com a simultaneidade possível, como veremos ao longo da exposição do método (MADUREIRA, 2010, p.19).

1.1.1 Projeto básico

Ainda segundo o autor, escolhida a melhor entre as soluções viáveis o planejamento do projeto é feito e, a partir de agora, passa a ser um compromisso formal da empresa. A seguir, modelos do produto são submetidos a estudos e análises técnicas por meio das quais são quantificadas as principais características do produto. A otimização dessas características consolida o projeto básico do produto posição do método (MADUREIRA, 2010, p.19).

1.1.2 Planejamento do projeto

Nessa primeira fase segundo Madureira (2010), são colocados os objetivos para o programa de projeto: produto (necessidades, funções e atributos), mercado a que se destina, prazo para a implantação, ciclo de vida, recursos para o desenvolvimento, investimentos na implantação, custos de fabricação e lucratividade global desejada para o programa.

Segundo Madureira (2010, p.19), estabelecidos os objetivos e os requisitos técnicos, são gerados soluções possíveis e, mediante análises, selecionadas as que são técnicas, econômica e financeiramente viáveis. Essa seleção baseia-se também na viabilidade de projeto, de fabricação e de fornecimento. Assegurada a viabilidade com confiança suficiente, a empresa poderá aprovar o programa e autorizar a sua condução.

1.1.3 Projeto de viabilidade

Segundo Madureira (2010, p. 19), estabelecidos os objetivos e os requisitos técnicos, são gerados soluções possíveis e, mediante análises, selecionadas as que são técnicas, econômica financeiramente viáveis. Essa seleção baseia – se também na viabilidade de projeto, de fabricação e de fornecimento. Assegurada a viabilidade com confiança suficiente, a empresa poderá aprovar o programa e autorizar a sua condução.

1.2 BANCO DE DADOS

“Banco de dados e sua tecnologia tem um impacto importante sobre o uso crescente dos computadores” (ELMASRI; NAVATHE, 2011, p. 2).

Conforme os autores citados acima, os bancos de dados desempenham um papel crítico em quase todas as áreas em que os computadores são usados, incluindo negócios, comércio eletrônico, engenharia, medicina, genética, direito, educação e biblioteconomia.

Um banco de dados é uma coleção de dados relacionados. Com dados, queremos dizer fatos conhecidos que podem ser registrados e possuem significado implícito.

Ainda conforme Elmasri e Navathe (2011, p. 4), definir um banco de dados, envolve especificar os tipos, estruturas e restrições dos dados a serem armazenados.

Segundo Gudwin (2015, p. 4), bancos de dados são aplicativos que tem por finalidade o armazenamento de dados de diversas espécies (comerciais ou industriais) e seu acesso e manipulação. As aplicações de bancos de dados permitem que esses dados sejam criados, alterados, substituídos e apagados, de tal forma a dar suporte a algum negócio comercial ou industrial. São o tipo de sistema de software que talvez mais tenham mercado em ambientes comerciais e também em ambientes industriais.

Banco de dados (database), é um conjunto de registros dispostos em estrutura regular que possibilita a reorganização dos mesmos e produção de informação. Um banco de dados normalmente agrupa registros utilizáveis para um mesmo fim (DIAS, 2013, p. 6).

Ainda segundo Dias (2013, p. 6).

Um banco de dados é usualmente mantido e acessado por meio de um software conhecido como Sistema Gerenciador de Banco de Dados (SGBD). Normalmente um SGBD adota um modelo de dados, de forma pura, reduzida ou estendida. Muitas vezes o termo banco de dados é usado, de forma errônea, como sinônimo de SGDB.

Segundo Heuser (1998, p. 2), “o desenvolvimento de sistemas de informação ocorre hoje quase que exclusivamente sobre banco de dados, com uso de SGBD relacional”.

Além do SGBD relacional, segundo heuser (1998,), as pesquisas na área de BD resultaram também em um conjunto de técnicas, processos e notações para o projeto de BD.

O autor mostra ainda que inicialmente era feito com técnicas empíricas por alguns poucos especialistas no SGBD específico, é executado hoje com auxílio de técnicas padronizadas e suportadas por ferramentas CASE.

Figura 1: Modelo de um Banco de Dados



Fonte: www.Google.com/imagens

1.2.1 Linguagem SQL

Segundo Elmasri e Navathe (2011), a linguagem SQL (Structured Query Language), pode ser considerada um dos principais motivos para o sucesso de banco de dados relacionais comerciais.

A Linguagem SQL tem como grandes virtudes sua capacidade de gerenciar índices, sem a necessidade de controle individualizado de índice corrente, algo muito comum nas linguagens de manipulação de dados do tipo registro a registro. Outra característica muito importante disponível em SQL é sua capacidade de construção de visões, que são formas de visualizarmos os dados na forma de listagens independente das tabelas e organização lógica dos dados.

Outra característica interessante na linguagem SQL é a capacidade que dispomos de cancelar uma série de atualizações ou de as gravarmos, depois de iniciarmos uma sequência de atualizações. Os comandos Commit e Rollback são responsáveis por estas facilidades.

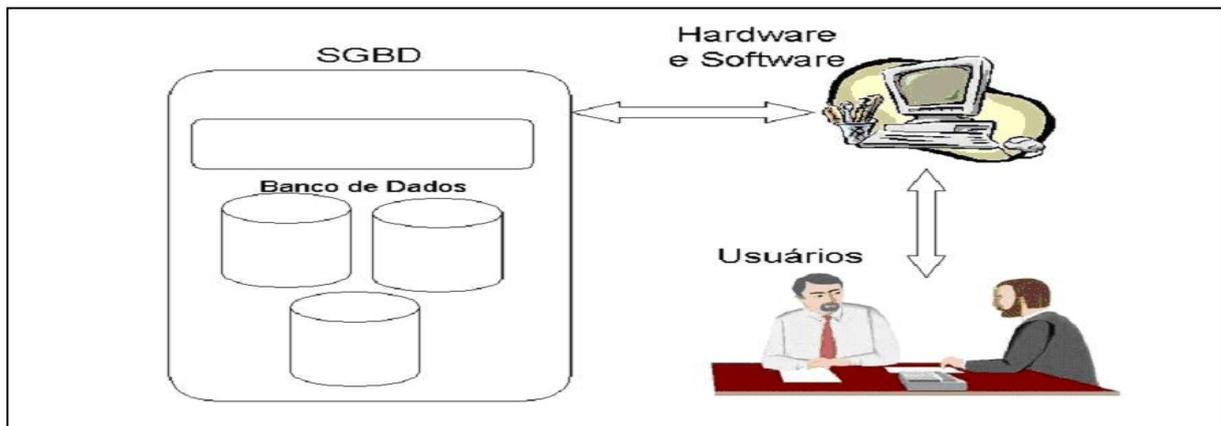
1.2.2 Sistema gerenciador de banco de dados – SGBD

Conforme Heuser (1998, p.2), sistemas de gerência de banco de dados (SGBD), surgiram no início da década de 70 com o objetivo de facilitar a programação de aplicações de banco de dados (BD). Os primeiros sistemas eram caros e difíceis de usar, requerendo especialistas treinados para usar o SGBD específico.

Segundo Silberschartz e Korth (2006, p.1):

Um sistema de gerenciamento de banco de dados (DBMS) é uma coleção de dados inter-relacionados em um conjunto de programas para acessar esses dados. A coleção de dados normalmente é chamada de banco de dados, contém informações relevantes de uma empresa. O principal objetivo de um é fornecer uma maneira de recuperar informações de banco de dados que seja tanto conveniente quanto eficiente.

Figura 2: componentes de um sistema de Banco de dados



Fonte: DATE C.J, 2000, p. 37

1.2.3 Mysql

O mysql é um sistema de gerenciamento de banco de dados relacional (relational database management system – RDBMS) poderoso e muito rápido (WELLING; THOMSON, 2005, p. 26).

Neste sentido, afirmam Welling e Thomson (2005, p. 27), “que o servidor mysql controla o acesso aos dados para assegurar que vários usuários possam trabalhar com os dados ao mesmo tempo, fornecem acesso rápido aos dados e assegurar que somente usuários autorizados obtenham acesso”. Portanto para os autores, “o mysql é um servidor multiusuário (ou multithreaded). Ele utiliza SQL (Structured Query Language), a linguagem de consulta padrão de banco de dados em todo o mundo”.

O Servidor MySQL destina-se a sistemas de produção de carga pesada e de missão crítica, bem como a incorporação em software implementado em massa. Oracle é uma marca registrada da Oracle Corporation e / ou de suas afiliadas. O MySQL é uma marca registrada da Oracle Corporation e / ou de suas afiliadas e não

deve ser usado pelo Cliente sem a autorização expressa por escrito da Oracle. Outros nomes podem ser marcas registradas de seus respectivos proprietários.

O mysql está publicamente disponível desde 1996, mas tem uma história de desenvolvimento que remota a 1979.

Segundo, afirmam Welling e Thomson (2005, p. 27):

O mysql está disponível sob um esquema de licença dupla. Você pode usá-lo sob a licença Open Source (GPL – General Public License) gratuitamente contanto que cumpra os termos da licença. No entanto se distribuir uma aplicação não – GPL que inclua o Mysql, você pode comprar uma licença comercial.

Figura 3: SGBD MySQL



Fonte: www.Google.com/imagens

1.2.3.1 Capacidades do Mysql

Segundo Welling e Thomson (2005), alguns dos principais concorrentes do MySQL são PostgreSQL, Microsoft SQL Server e Oracle.

Mysql tem muitas capacidades, incluindo:

- ✓ Alto desempenho;
- ✓ Baixo custo;
- ✓ Fácil configuração e aprendizado;
- ✓ Portabilidade;
- ✓ Disponibilidade de código fonte;
- ✓ Disponibilidade de suporte;

De acordo com a Oracle, o MySQL Community Edition é uma versão livre para download do banco de dados de código aberto mais popular do mundo, que é suportado por uma comunidade ativa de desenvolvedores de código aberto e entusiastas.

1.2.4 Modelagem banco de dados

Um modelo de banco de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados (HEUSER, 1998, p. 5).

Segundo Heuser (1998), Linguagens de modelagem de dados podem ser classificadas de acordo com a forma de apresentar modelos, em linguagens textuais ou linguagens gráficas. Um mesmo modelo de dados pode ser apresentado de várias formas. Cada apresentação do modelo recebe a denominação esquema de banco de dados.

De acordo com a intenção do modelador, um banco de dados pode ser modelado (descrito) há vários níveis de abstração. Um modelo de dados que servirá para explicar a um usuário qual é a organização de um banco de dados provavelmente não conterá detalhes sobre a representação em meio físico das informações.

No projeto de banco de dados, normalmente são considerados dois níveis de abstração de modelo de dados, o do modelo conceitual e o do modelo lógico.

Linguagens de modelagem de dados podem ser classificadas de acordo com a forma de apresentar modelos, em linguagens textuais ou linguagens gráficas.

A modelagem de dados é a criação de uma estrutura de dados eletrônica (banco de dados) que representa um conjunto de informações. Esta estrutura permite ao usuário recuperar dados de forma rápida e eficiente. O objetivo é incluir dados em uma estrutura que possibilite transformar os dados originais em vários tipos de saídas como formulários, relatórios, etiquetas ou gráficos.

Essa capacidade de transformar informações caracteriza as operações de banco de dados e é a chave de sua utilidade. Um Banco de Dados – BD, representa uma coleção de dados que possui algum significado e objetiva atender a um conjunto de usuários. Por exemplo, um catálogo telefônico pode ser considerado um BD. Sendo assim, um BD não necessariamente está informatizado.

Quando resolvemos informatizar um BD, utilizamos um programa especial para realizar essa tarefa, o SGBD – Sistema Gerenciador de Banco de Dados. Em um SGBD relacional, enxergamos os dados armazenados em uma estrutura chamada tabela. Neste modelo, as tabelas de um BD são relacionadas, permitindo assim que possamos recuperar informações envolvendo várias delas.

1.2.5 Modelo Conceitual

Um modelo conceitual é uma descrição do banco de dados de forma independente de implementação em um SGBD. O modelo conceitual registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados a nível de SGBD (HEUSER, 1998).

Conforme Heuser (1998), um modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD. Assim, o modelo lógico é dependente do tipo particular de SGBD que está sendo usado.

1.2.6 Modelo Lógico

Conforme Heuser (1998), um modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD. Assim, o modelo lógico é dependente do tipo particular de SGBD que está sendo usado.

1.2.7 Mysql Workbench

Esta ferramenta, bastante completa, permite o desenvolvimento de scripts, modelagem de bases de dados, desenvolvimento de SQL, administração, configuração e backups do servidor. Tudo isto e muito mais numa aplicação totalmente gratuita e disponível para Windows, Mac OS, Fedora e Linux. O aplicativo permite criar e desenhar o modelo e funcionamento da base de dados, de modo a que a criação de banco de dados seja rápida e intuitiva. Todos os processos de desenvolvimento e criação de base de dados tornam-se visivelmente mais simples e menos trabalhosos. Para além disso possui uma panóplia de recursos para a criação, execução e

otimização das consultas SQL. O editor incorporado possui a capacidade de enfatizar sintaxe, auto-completar, memória de utilização, entre outros recursos.

O MySQL Workbench é uma ferramenta gráfica para trabalhar com servidores MySQL e bancos de dados. O MySQL Workbench suporta completamente o servidor MySQL versões 5.5 e superiores. Também é compatível com versões anteriores do servidor MySQL 5.x, exceto em determinadas situações (como exibir a lista de processos) devido a tabelas de sistema alteradas. Não suporta versões do servidor MySQL 4.x.

A funcionalidade do MySQL Workbench abrange cinco tópicos principais:

- **Desenvolvimento SQL:** permite que você crie e gerencie conexões para servidores de banco de dados. Além de permitir que você configure parâmetros de conexão, o MySQL Workbench fornece a capacidade de executar consultas SQL nas conexões do banco de dados usando o SQL Editor incorporado.
- **Modelagem de dados (Design):** permite que você crie modelos de seu esquema de banco de dados graficamente, engasgar reverso e encaminhar entre um esquema e um banco de dados ao vivo e editar todos os aspectos do seu banco de dados usando o extenso Editor de tabelas. O Table Editor fornece facilidades fáceis de usar para edição de tabelas, colunas, índices, disparadores, particionamento, opções, inserções e privilégios, rotinas e visualizações.
- **Administração do servidor:** permite que você administra instâncias do servidor MySQL administrando usuários, executando backup e recuperação, inspecionando dados de auditoria, visualizando a saúde do banco de dados e monitorando o desempenho do servidor MySQL.

- **Migração de dados:** permite que você migre do Microsoft SQL Server, Microsoft Access, Sybase ASE, SQLite, SQL Anywhere, PostgreSQL e outras tabelas RDBMS, objetos e dados para o MySQL. A migração também suporta a migração de versões anteriores do MySQL para os lançamentos mais recentes.
- **Suporte empresarial do MySQL:** Suporte para produtos corporativos, como MySQL Enterprise Backup, MySQL Firewall e MySQL Audit.

1.3 DESENVOLVIMENTO WEB

1.3.1 Plataforma de desenvolvimento

Normalmente está associada a várias linguagens de programação como PHP, JAVA ou (JSP) e a uma linguagem de marcação de texto como: HTML. As linguagens de marcação são as que rodam do lado cliente, ou seja, são elas que rodam no browser (navegador), enquanto as linguagens de programação rodam no lado servidor.

1.3.2 Linguagem de programação

É um método padrão de comunicação de instruções para o computador. É onde o programador tem permissão para especificar em cima de quais dados o computador pode trabalhar, e pode ser usada para expressar algoritmos.

1.3.3 Lógica de Programação

Segundo Forbellone (2005), lógica de programação é o uso correto das leis do pensamento, da 'ordem da razão' e de processos de raciocínio e simbolização formais na programação de computadores, objetivando a racionalidade e o desenvolvimento de técnicas que cooperem para a produção de soluções logicamente válidas e coerentes, que resolvam com qualidade os problemas que se deseja programar.

1.3.4 Php

PHP é uma linguagem de criação de scripts do lado servidor que foi projetada especificamente para a web. (WELLING; THOMSON, 2005, p.XXVI).

Dentro de uma página HTML, pode - se embutir código de PHP que será executado toda vez que a página for visitada. O código PHP é interpretado no servidor web e gera HTML ou outra saída que o visitante verá.

O PHP foi concebido em 1994 como resultado de trabalho de uma única pessoa, Ramus Lerdorf.

Segundo Pablo Dall' Oglío (2015), a linguagem de programação PHP, foi criada no outono de 1994 por Rasmus Lerdorf. No início era formada por um conjunto de scripts voltados à criação de páginas dinâmicas que Rasmus utilizava para monitorar o acesso ao seu currículo na internet. À medida que essa ferramenta foi crescendo em funcionalidades, Rasmus teve de escrever uma implementação em C, a qual permitia às pessoas desenvolverem de forma muito simples suas aplicações para web.

O PHP tem conexões nativas disponíveis para muitos sistemas de banco de dados. Além do Mysql, você pode conectar – se diretamente a banco de dados PostgreSQL, Mysql, Oracle, dbm, FilePRO, HiperWave, Informix, InterBase e Sybase, entre outros. O PHP também tem uma interface integrada SQL para um arquivo simples, chamada SQLLite (WELLING; THOMSON, 2005).

1.3.5 Programação orientada a objetos

Segundo Sebesta (2003, p. 96):

A essência da programação orientada a objeto é resolver problemas, identificando os objetos do mundo real do problema e processamento exigido desses objetos, e depois criar simulações desses objetos, de seus processos e da comunicação requerida sobre eles.

Ainda segundo Sebesta (2003, p. 96), “tipos de dados abstratos, vinculação dinâmica e herança são os conceitos que tornam a solução de problemas orientado a objeto não somente possível, mas também conveniente e efetiva”.

A orientação a objetos é um paradigma que representa uma filosofia para construção de sistemas, conforme esboça Dall’oglio (2015, p.88). O autor acrescenta que, na orientação a objetos, utiliza-se uma ótica mais próxima do mundo real. Lida-se com objetos: estruturas que carregam dados e comportamento próprio, além de trocarem mensagens entre si com o objetivo de formar algo maior, um sistema.

Conforme Welling e Thomson (2005, p.117), “as linguagens de programação modernas normalmente suportam ou até mesmo exigem uma abordagem orientada a objetos para desenvolvimento de software”.

Para este autor, o desenvolvimento orientado a objetos (Object-Oriented-OO) tenta utilizar as classificações, os relacionamentos e as propriedades dos objetos no sistema para ajudar no desenvolvimento de programa.

Segundo Sommerville (2011, p. 125)

Um sistema orientado a objetos é composto de objetos interativos que mantêm seu próprio estado local e oferecem operações nesse estado. A representação do estado é privada e não pode ser acessada diretamente, de fora do objeto. Processos de projeto orientado a objetos envolvem projetar as classes de objetos e os relacionamentos entre essas classes. Essas classes definem os objetos no sistema e suas interações. Quando o projeto é concebido como um programa em execução, os objetos são criados dinamicamente a partir dessas definições de classe.

Ainda segundo Sommerville (2011), Sistemas orientados a objetos são mais fáceis de mudar do que os sistemas desenvolvidos com abordagens funcionais. Os objetos incluem os dados e as operações para manipulá-los. Portanto, eles podem ser entendidos e modificados como entidades autônomas.

Para este autor, desenvolver um projeto de um sistema desde o conceito até o projeto detalhado orientado a objeto, existem várias atitudes que você precisa tomar:

- Compreender e definir o contexto e as interações externas com o sistema.
- Projetar a arquitetura do sistema.
- Identificar os principais objetos do sistema.
- Desenvolver modelos de projeto.
- Especificar interfaces.

1.3.6 Programação Estruturada

Conforme Pablo Dall'Oglio (2007, p. 86), “programação estruturada é um paradigma de programação que introduziu uma série de conceitos importantes na época em que foi criada e dominou a área da engenharia de software durante algumas décadas”.

Segundo Sommerville (2011, p.130), descrevem a estrutura estática do sistema, usando as classes de objetos e seus relacionamentos. Relacionamentos importantes que podem ser documentados nesse estágio são os de generalização (herança), usa/usado por, e composição.

1.3.7 Html

Segundo Freeman e Freeman (2008), a linguagem de marcação de hipertexto, HTML é o que dá base para o funcionamento do Browser, pois o mesmo interpreta as *tags* no código e as mostra de maneira que o usuário consiga entender o que o código pretende mostrar.

Pedro Neves (2004, p. 13) explica que:

HTML significa HyperText Markup Language, e consiste numa linguagem de formatação de texto que possibilita a construção (programação/formatação) de ficheiros de texto, com a extensão htm ou html, que contêm pequenas marcas (tags) que indicam ao Web browser como apresentar o conteúdo incluído ou referenciado no documento, sob a forma de texto, imagens ou suportes multimedia.

E acrescenta que, “a estrutura de um documento HTML é muito simples: consiste numa marca (tag) exterior <html> que engloba o cabeçalho e o corpo do documento” (NERES, 2004, p.15).

Figura 4: Código-fonte HTML

```
<! DOCTYPE html>
< html lang = "en">
< head >
< title > Story </ title >
</ head >
< corpo >
< h1 > Minha história </ h1 >
< p > Era uma vez,
... </ p >
```

```
</ body >  
</ html >
```

Fonte: www.w3.org/html/

1.4 JAVASCRIPT

Samy Silva (2010, p. 22), define: “Java Script é uma linguagem desenvolvida para rodar no lado do cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário”.

Ainda segundo Samy Silva (2010, p.25):

JavaScript é capaz de definir, alterar e controlar de forma dinâmica a apresentação de um documento HTML, como os aspectos relacionados à cor de fundo, de textos e de links, ou mesmo interferir no posicionamento dos elementos HTML de um documento. É possível manipular a folha de estilos associada ao documento criando novas regras CSS ou anulando regras existentes.

De acordo com Miyagusku (2007, p. 86), “O Javascript é uma linguagem interpretada, isto é, não é previamente compilada para gerar um executável, mas seu próprio código consiste de linhas executáveis”.

Miyagusku (2007, p. 22) ainda diz que.

Ao contrário do que muitos imaginam, o Javascript não tem relação com o Java, que é uma linguagem de programação poderosíssima que se enquadra no mesmo nível do C++, e não é só voltada para a Internet como o Javascript, que em geral, é aplicado embutido ao código HTML.

1.4.1 Cascading style sheets – CSS

Samy Silva (2012, p. 24), diz que “CSS é a abreviação para o termo em inglês Cascading Style Sheet, traduzido para o português como folhas de estilo em cascata”.

E continua:

A definição mais precisa e simples para folha de estilo encontra-se na homepage das CSS no site do W3C e diz:
Folha de estilo em cascata é um mecanismo simples para adicionar estilos (por exemplo: fontes, cores, espaçamentos) aos documentos web. As CSS

têm por finalidade devolver à marcação HTML/XML o propósito inicial da linguagem (SAMY SILVA,2012, p. 24 - 25).

1.4.1.1 Regras CSS

Samy Silva (2012, p. 26), diz que, “regra CSS é a unidade básica de uma folha de estilo. Nessa definição, o termo unidade básica significa a menor porção de código capaz de produzir efeito de estilização”. E acrescenta que, uma regra CSS é composta de duas partes: o seletor e a declaração”. Uma regra CSS poderá conter várias declarações separadas por ponto e vírgula.

1.4.1.2 Estilos externos

Folha de estilo externa é aquela que não foi escrita no documento HTML. Trata-se de um arquivo de texto contendo as regras de estilo e os comentários CSS. Um arquivo de folha de estilo deve ser gravado com a extensão .css e pode ser vinculado a um documento HTML de duas maneiras distintas (SAMY SILVA, 2012, p. 41).

1.4.1.3 Folhas de estilo importadas

Conforme Samy Silva (2012), “nesse método, você vincula uma folha de estilo externa a um documento usando a diretiva @import dentro do elemento style”.

1.4.2 Bootstrap

Samy Silva (2015), apresenta o bootstrap como um poderoso framework front-end voltado para o desenvolvimento rápido e fácil de sites e aplicações web responsivos e alinhados com a filosofia mobile-first”. Ainda segundo Samy Silva (2015, p. 20), “bootstrap é indicado para dispositivos e projetos de todos os tamanhos e destinado a desenvolvedores com qualquer nível de conhecimento, dispositivos de todos os tipos e projetos de todos os tamanhos”.

Para ele, o bootstrap é definido como um poderoso, elegante e intuitivo framework front-end que possibilita um desenvolvimento web de modo ágil e fácil.

O Bootstrap foi desenvolvido e otimizado para ser usado com marcação HTML5.

1.4.2.1 Suporte

O Bootstrap foi criado para funcionar em navegadores modernos tanto em desktop como em dispositivos móveis. Navegadores antigos falharão na estilização, mas a parte funcional da página será preservada.

Conforme atesta Samy Silva (2015, p.32):

Quanto ao Internet Explorer, o suporte é consistente nas versões IE9 e superiores. Se você pretende servir sua aplicação para o IE8, deve incluir os scripts shim e Respond.js que se destinam a possibilitar que aquele navegador reconheça os elementos da HTML5 e as regras media queries respectivamente, pois ele não oferece suporte nativo para tais funcionalidades.

1.5 ENGENHARIA DE SOFTWARE

A engenharia de software é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação (SOMMERVILLE, 2003).

Engenharia de software é metodologia de desenvolvimento e manutenção de sistemas modulares, com as seguintes características: processo (roteiro) dinâmico, integrado e inteligente de soluções tecnológicas; adequação aos requisitos funcionais do negócio do cliente e seus respectivos procedimentos pertinentes; efetivação de padrões de qualidade, produtividade e efetividade em suas atividades e produtos; fundamentação na Tecnologia da informação disponível, viável, oportuna e personalizada; planejamento e gestão de atividades, recursos, custos e datas (REZENDE, 2005).

Maffeo (1992 apud REZENDE, 2005, p.5) considera que os objetivos primários da Engenharia de Software são o aprimoramento da qualidade dos produtos

de software e o aumento da produtividade dos engenheiros de software, além do atendimento aos requisitos de eficácia e eficiência, ou seja, efetividade.

1.5.1 Engenharia de Requisitos

Antes de iniciar qualquer trabalho técnico, é uma boa idéia aplicar um conjunto de tarefas de engenharia de requisitos. Estas levam a um entendimento de qual será o impacto do software sobre o negócio, o que o cliente quer e como os usuários finais irão interagir com o software. Quem realiza a engenharia de requisitos são os engenheiros de software (algumas vezes conhecidos no mundo da TI como engenheiros de sistemas ou “analistas”), e outros interessados no projeto (gerentes, clientes, usuários finais), todos participam da engenharia de requisitos (PRESSMAN, 2011).

O processo de engenharia de requisitos leva à produção de uma documentação de requisitos, que é a especificação para o sistema. Os requisitos geralmente são apresentados em dois níveis de detalhes nesse documento (SOMMERVILLE, 2005).

Projetar e construir um programa de computador elegante que resolva o problema errado não atende às necessidades de ninguém. Por isso a importância de entender o que o cliente quer antes de começar a projetar e construir um sistema baseado em computador (PRESSMAN, 2011, p. 126).

Engenharia de requisitos é um estágio particularmente importante do processo de software, uma vez que erros nesse estágio inevitavelmente produzem problemas posteriores no projeto e na implementação do sistema (SOMMERVILLE, 2005).

Existem quatro fases principais no processo de engenharia de requisitos:

- Estudo de Viabilidade É feita uma estimativa para verificar se as necessidades dos usuários que foram identificadas podem ser satisfeitas com a utilização das atuais tecnologias de software e hardware. O estudo decidirá se o sistema proposto será viável, do ponto de vista comercial, e se poderá ser desenvolvido certas restrições orçamentárias existentes. Um estudo de viabilidade deve ser relativamente barato e rápido.

- Requisitos do sistema pela observação de sistemas existentes, pela conversa com usuários e compradores em potencial, pela análise de tarefas assim por diante.
- Especificação de requisitos é a atividade de traduzir as informações coletadas durante a atividade de análise em um documento que defina um conjunto de requisitos. Dois tipos de requisitos podem ser incluídos nesse documento. Os requisitos dos usuários são declarações abstratas dos requisitos de sistema para o cliente e os usuários finais do sistema; os requisitos do sistema são uma descrição mais detalhada da funcionalidade a ser fornecida.
- Validação de requisitos essa atividade verifica os requisitos quanto a sua pertinência, consistência e integridade. Durante esse processo, inevitavelmente são descobertos erros na documentação de requisitos. Os requisitos devem então ser modificados, a fim de corrigir esses problemas.

1.5.2 Processos de Software

É um conjunto de atividades cuja meta é o desenvolvimento ou evolução do software (SOMMERVILLE, 2003, p. 6).

Conforme Sommerville (2003, p. 7), há quatro atividades fundamentais comuns a todos os processos de software:

Especificação do software: Funcionalidade do software e as restrições em sua operação devem ser definidas;

Desenvolvimento do software: O software deve ser produzido de modo que atenda às suas especificações;

Validação do software: O software deve ser validado para garantir que ele faz o que o cliente deseja;

Evolução do software: O software deve evoluir para atender às necessidades mutáveis do cliente;

Sommerville (2005) demonstra que diferentes processos de software organizam essas atividades de maneiras diversas e são descritos em diferentes níveis de detalhes. Os prazos das atividades variam, do mesmo modo que os resultados de

cada atividade. Diferentes organizações podem utilizar processos diferentes para produzir o mesmo tipo de produto.

Os processos de software são complexos e, como todos os processos intelectuais, dependem de julgamento humano. Por causa da necessidade de utilizar o julgamento e a criatividade, tentativas de automatizar processos de software têm tido sucesso limitado.

1.5.3 Modelos de Processos de Software

É uma representação simplificada de um processo de software, apresentada a partir de uma perspectiva específica (SOMMERVILLE, 2003, p. 6). Ainda segundo este autor, os modelos, por sua natureza, são simplificações; e, assim, um modelo de processo de software é uma abstração do processo real que está sendo descrito.

Exemplos dos tipos de modelos de processo de software que podem ser reproduzidos (SOMMERVILLE, 2003, p. 8):

Um modelo de workflow: Mostra a sequência de atividades no processo, juntamente com suas entradas, saídas e dependências. As atividades, nesse modelo, representam ações humanas.

Um modelo de fluxo de dados ou de atividade: Representa o processo como um conjunto de atividades, cada uma das quais realiza alguma transformação de dados. Esse modelo mostra como a entrada para o processo, tal como uma especificação, é transformada em uma saída, como um projeto.

Um modelo de papel/ação: Esse modelo representa os papéis das pessoas envolvidas no processo de software e as atividades pelas quais elas são responsáveis.

Conforme afirma Pressman (2011, p. 52), quando se trabalha na elaboração de um produto ou sistema, é importante seguir uma série de passos previsíveis, um roteiro que ajude a criar um resultado de alta qualidade e dentro do prazo estabelecido. O roteiro é denominado “processo de software”.

1.5.4 Objetivos

Fornece todas as partes um entendimento escrito do problema. Isso pode ser alcançado por meio de uma série de artefatos: cenários de uso, listas de funções e características, modelos de análises ou uma especificação (PRESSMAN, 2011, p. 126).

1.5.5 Modelos de Ciclo de Vida

Segundo Filho (2000), a Engenharia de Software se preocupa com o software enquanto produto. Estão fora de seu escopo programas que são feitos unicamente para diversão do programador. Estão fora de seu escopo também pequenos programas descartáveis, feitos por alguém exclusivamente como meio para resolver um problema, e que não serão utilizados por outros

Em engenharia de software, processos podem ser definidos para atividades como desenvolvimento, manutenção, aquisição e contratação de software. Podem se definir subprocessos para cada um desses, um processo de desenvolvimento abrange subprocessos de determinação dos requisitos, análise, desenho, implementação e testes (FILHO, 2009).

Conforme Filho (2000, p. 12), como todo produto industrial, o software tem um ciclo de vida:

- ✓ Ele é concebido a partir da percepção de uma necessidade;
- ✓ Desenvolvido, transformando-se em um conjunto de itens entregue a um cliente;
- ✓ Entra em operação, sendo usados dentro de um algum processo de negócio, e sujeito a atividades de manutenção, quando necessário;
- ✓ É retirado de operação, ao final de sua vida útil;

1.6 SOFTWARE LIVRE

Software livre é software que vem com permissão para qualquer um copiar, usar e distribuir, com ou sem modificações, gratuitamente ou por um preço. Em particular, isso significa que o código-fonte deve estar disponível.

Se um programa é livre, ele pode potencialmente ser incluído em um sistema operacional livre, como o GNU, ou versões livres como o sistema GNU/Linux.

O GNU é um sistema operacional que é um software 100% livre. Foi lançado em 1983 por Richard Stallman (rms) e foi desenvolvido por muitas pessoas trabalhando juntas por causa da liberdade de todos os usuários de software para controlar sua computação. Tecnicamente, o GNU é geralmente como o Unix. Mas, ao contrário do Unix, o GNU dá liberdade a seus usuários.

O sistema GNU contém todos os pacotes de software GNU oficiais (que estão listados abaixo), e também inclui software livre que não é GNU, notadamente TeX e X Window System. Além disso, o sistema GNU não é um único conjunto estático de programas; os usuários e distribuidores podem selecionar diferentes pacotes de acordo com suas necessidades e desejos. O resultado ainda é uma variante do sistema GNU. Segundo Silveira (2004), programas de computador que possuem o código-fonte aberto – é um movimento pelo compartilhamento do conhecimento tecnológico. Começou nos anos 1980 e se espalhou pelo planeta levado pelas teias da rede mundial de computadores. Seus maiores defensores são os apaixonados por programação e sistemas de informática, os acadêmicos, os cientistas, os mais diferentes combatentes pela causa da liberdade e, mais recentemente, as forças político-culturais que apóiam a distribuição mais equitativa dos benefícios da era da informação.

As quatro liberdades que caracterizam o software livre são as de uso, cópia, modificações e redistribuição (SILVEIRA, 2004, p. 9). Ainda segundo o autor, a licença do software livre é uma licença não-proprietária de uso. O software livre possui um autor ou vários autores, mas não possui donos. Dessa forma, o usuário do software livre também tem o direito de ser desenvolvedor, caso queira. Quem o adquire pode usá-lo para todo e qualquer fim, inclusive tem a permissão de alterá-lo completamente.

1.7 OPENSOURCE

“Suas raízes estão no Free Software Foundation (<<http://www.fsf.org>>), que defende que o código-fonte não deve ser proprietário, mas deve estar sempre disponível para os usuários analisarem e modificarem como quiserem” (SOMMERVILLE, 2011, p. 139).

Pelo menos segundo Sommerville (2011, p.139) em princípio, qualquer contribuinte para um projeto open source pode relatar e corrigir bugs e propor novas características e funcionalidade. No entanto, na prática, sistemas open source de sucesso ainda contam com um grupo de desenvolvedores que controlam as mudanças no software.

Ainda conforme explica Sommerville (2011), O produto open source mais conhecido é, naturalmente, o sistema operacional Linux, amplamente usado como um sistema de servidor e, cada vez mais, como um ambiente de desktop. Outros importantes produtos open source são o Java, o servidor Web Apache e o sistema de gerenciamento de banco de dados MySQL.

Ainda segundo o autor, a aquisição de software open source costuma ser bastante barata ou gratuita, pois geralmente é possível baixar esses softwares sem custos. No entanto, se você precisar de documentação e suporte, você pode ter de pagar por isso, embora os custos sejam usualmente bastante baixos.

A maioria das licenças open source segundo Sommerville (2011, p. 140) derivam de um dos três modelos gerais:

1. A GNU General Public License (GPL). Essa é a chamada licença 'recíproca'; de forma simplista, significa que se você usar um software open source que esteja licenciado sob a licença GPL, você deve fazer um software open source.
2. A GNU Lesser General Public License (LGPL). Essa é uma variação da licença GPL, segundo a qual você pode escrever componentes que se ligam com código open source sem publicar a fonte desses componentes. No entanto, se você alterar o componente licenciado, você deve publicá-lo como open source.
3. A Berkley Standard Distribution (BSD) License. Essa é uma licença não recíproca, o que significa que você não é obrigado a republicar quaisquer alterações ou modificações feitas no código open source. Você pode incluir o código em sistemas proprietários que sejam vendidos. Se você usar componentes open source, deve reconhecer o criador original do código.

1.8 UML

A UML (Unified Modeling Language – linguagem de modelagem unificada), é “uma linguagem padrão para descrever/documentar projeto de software. A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de software-intensivo” (PRESSMAN, 2011, p. 727). Ainda segundo o autor em outras palavras, assim como os arquitetos criam plantas e projetos para ser usados por uma empresa de construção, os arquitetos de software criam diagramas UML para ajudar os desenvolvedores de software a construir o software.

1.9 DIAGRAMA DE CASO DE USO

Segundo Pressman (2011, p. 731), casos de uso e o diagrama de caso de uso ajudam a determinar a funcionalidade e as características do software sob o ponto de vista do usuário. O autor afirma ainda que, um caso de uso descreve como um usuário interage com o sistema definindo os passos necessários para atingir um objetivo específico.

O diagrama de caso de uso é representado por atores, casos de uso e relacionamentos. Os relacionamentos podem ser associações entre atores e casos de uso; generalizações entre os atores e generalizações, extends e includes.

1.10 FRAMEWORK

Como o nome sugere, um framework é uma estrutura genérica estendida para se criar uma aplicação ou subsistema mais específico (SOMMERVILLE, 2011, p.300).

Ainda segundo Sommerville (2011)

Os frameworks fornecem suporte para recursos genéricos, suscetíveis de serem usados em todas as aplicações de tipos semelhantes. Por exemplo, um framework de interface de usuário fornecerá suporte para tratamento de eventos de interface e incluirá um conjunto de recursos que possam ser usados para construir displays. Então, o desenvolvedor fica responsável por especializá-los, adicionando funcionalidade específica para uma aplicação específica. Por exemplo, em um framework de interface de usuário, o desenvolvedor define layouts de display apropriados para a aplicação que está sendo implementada.

Conforme Sommerville (2011), os frameworks são implementados como uma coleção de classes de objetos concretos e abstratos em uma linguagem de

programação orientada a objetos. Por conseguinte, frameworks são específicos da linguagem. Existem frameworks disponíveis em todas as linguagens de programação orientadas a objetos mais comumente usadas (por exemplo, Java, C# e C++, assim como linguagens dinâmicas como Ruby e Python). Ainda segundo o autor, um framework pode incorporar vários outros, de forma que cada um deles é projetado para suportar o desenvolvimento de parte da aplicação. Você pode usar um framework para criar uma aplicação completa ou para implementar partes de uma aplicação, como a interface gráfica de usuário.

O framework segundo afirma Sommerville (2011, p. 302), são uma abordagem eficaz de reuso, mas são caros para serem introduzidos em processos de desenvolvimento de software. Eles são inerentemente complexos e pode demorar meses para alguém aprender a usá-los. Pode ser difícil e caro avaliar frameworks disponíveis para a escolha do framework mais adequado.

1.10.1 Symfony

O Symfony é um conjunto de componentes PHP, um framework de Aplicações Web, uma Filosofia e uma Comunidade - todos juntos em harmonia.

1.10.1.1 Framework PHP

Para entender melhor o que é um quadro, deixe o mundo da ciência da computação por um momento e imagine uma aventura de montanhismo. O desenvolvimento de uma aplicação é muito parecido com a escalada de um muro de rocha: você está no final (você tem um aplicativo a ser criado) e você precisa alcançar a cúpula (e ficar satisfeito com a realização de projetar um aplicativo que funcione perfeitamente).

Se ninguém já escalou a parede em questão, você terá que passar por conta própria: rotas de teste, ocasionalmente retrocedendo para que você não fique preso em um canto, dirigindo em pitons, etc. Em contraste, se a parede já foi conquistada, aqueles que foram antes de você já terão feito este trabalho de teste e erro, abrindo possíveis caminhos (o quadro) e instalando as ferramentas que facilitarão a escalada (seu trabalho).

Basicamente, uma estrutura consiste em:

- **Uma caixa de ferramentas** - um conjunto de componentes de software pré-fabricados, integralmente integráveis. Isso significa que você terá que escrever menos código, com menos risco de erro. Isso também significa maior produtividade e a capacidade de dedicar mais tempo a fazer as coisas que proporcionam maior valor agregado, como a gestão de princípios orientadores, efeitos colaterais, etc.
- **Uma metodologia** - um "diagrama de montagem" para aplicações. Uma abordagem estruturada pode parecer constrangedor no início. Mas, na realidade, permite que os desenvolvedores trabalhem de forma eficiente e eficaz nos aspectos mais complexos de uma tarefa e o uso de Melhores Práticas garante a estabilidade, a manutenção e a capacidade de atualização dos aplicativos que você desenvolve.

E, embora possa ser óbvio, achamos melhor dizer de qualquer maneira – o Symfony é um framework PHP, o que significa que ele permite que aplicativos web sejam criados em PHP!

1.10.2 Filosofia

O que poderia ser mais útil do que um aplicativo desenvolvido pelos usuários para suas próprias necessidades? Esta é a história da gênese da Symfony - nascida da imaginação da web designers da SensioLabs, um desenvolvedor web por direito próprio. A Symfony foi disponibilizada para todos sob uma licença de código aberto. Isso beneficia outros desenvolvedores, que também têm a capacidade de aprimorá-lo, adicionando seus próprios módulos. Tudo isso foi feito em um ambiente de Melhores Práticas, padronização e interoperabilidade de aplicativos.

1.10.3 Comunidade

E, por trás de Symfony, existem pessoas. Nós somos pessoas de SensioLabs. Criamos a Symfony para resolver nossos próprios problemas de desenvolvimento web e estamos satisfeitos por promovê-lo. Nós também somos uma comunidade de desenvolvedores, usuários e colaboradores que se formaram em torno do Symfony nos últimos 5 anos.

1.10.4 Componentes Symfony

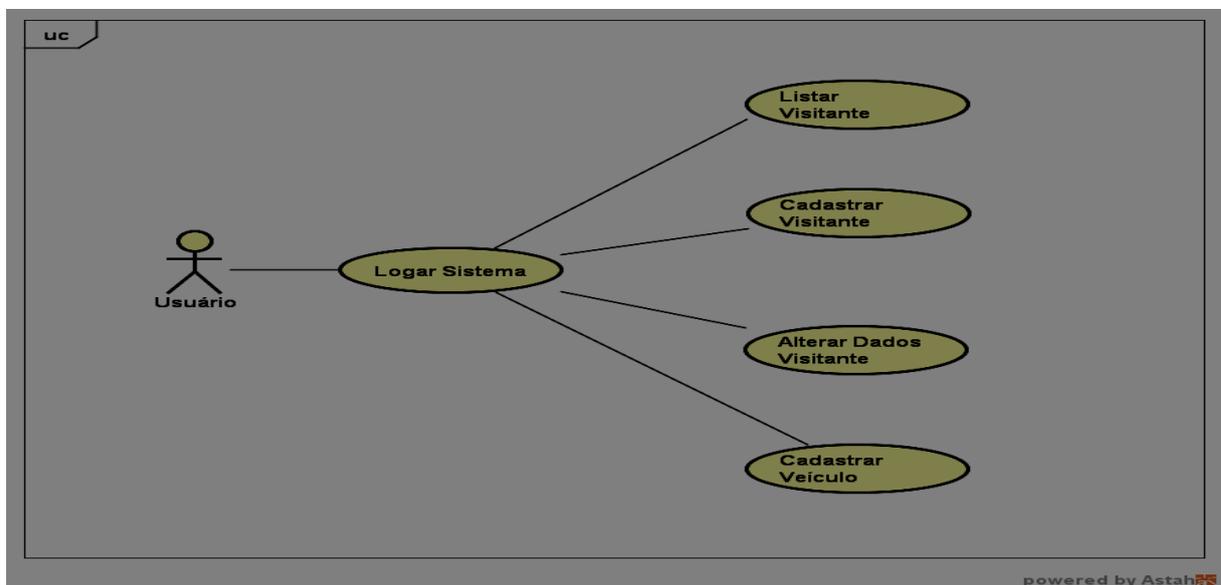
Os Symfony Components são um conjunto de bibliotecas PHP desacopladas e reutilizáveis. Eles estão se tornando a base padrão sobre a qual os melhores aplicativos PHP estão baseados. Você pode usar qualquer um desses componentes em seus próprios aplicativos independentemente do Symfony Framework.

2 DESENVOLVIMENTO DO SISTEMA

A partir de agora será dado início ao processo de desenvolvimento do projeto. Colocando em prática todos os conhecimentos adquiridos, através dos estudos realizados sobre as diversas ferramentas que serão utilizadas, bem como um maior aprofundamento do conteúdo no decorrer do desenvolvimento, onde serão mostrados os passos importantes para o desenvolvimento e funcionamento do mesmo.

2.1.1 Diagrama de Caso de Uso

Figura 5: Caso de uso Sistema Controle de Acesso



Fonte: Do próprio autor (2017)

2.1.2 Requisitos do sistema

RF01	Cadastrar Visitante
RF02	Alterar Visitante
RF03	Pesquisar Visitante
RF04	Cadastrar Veículo

2.1.3 Descrição dos casos de uso

[UC 01]: Logar no Sistema

Descrição: O usuário deverá entrar com seus dados: login e senha. Se o mesmo não for cadastrado deverá solicitar ao administrador o seu cadastro. O administrador realizará o cadastro do usuário por meio do prompt de comando (terminal) por motivo de segurança. O Sistema deverá permitir acesso ao conteúdo do software somente se os dados estiverem corretos.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário já deverá possuir seu cadastro no sistema.

Cenário Principal de Sucesso:

1. O caso de uso é iniciado com usuário fornecendo o login e senha.
2. O usuário deverá entrar com seus dados.
3. O sistema busca no banco de dados se os dados estão corretos.
4. O sistema é iniciado, liberando as funcionalidades.

Cenário Secundário de Falha:

- 4.1 O sistema retorna a mensagem de erro login inválido.

[UC 02] Cadastrar Visitante

Descrição: Antes de iniciar o cadastro do visitante, o usuário deverá usar a tela de pesquisa do sistema, podendo usar nome, cpf ou rg, para saber se o mesmo já possui cadastro na empresa, se o mesmo já possuir cadastro dar prosseguimento no cadastro do visitante.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E nome, cpf e rg do visitante ainda não cadastrado.

Pós-Condições: Retorno mensagem de visitante cadastrado.

Cenário Principal de Sucesso:

1. O usuário deve ir ao cadastro do visitante.
2. O usuário deverá informar os dados do visitante.
3. O usuário submete os dados necessários para armazenamento no banco de dados.
4. O sistema validará os dados e retorna mensagem “Cadastro com sucesso”.

Cenário Secundário de Falha:

- 4.1. O sistema aborta a validação dos dados e retorna mensagem de erro, “Cadastro não realizado”.

[UC 03] Alterar Visitante

Descrição: O usuário deverá fazer entrada dos dados referente à busca do visitante, nome, cpf ou rg do visitante.

Atores envolvidos: Usuário do sistema.

Pré-condição: O usuário deverá estar logado no sistema. E o usuário deve fornecer um dos dados para encontrar o visitante.

Pós-Condições: Retorno mensagem de alteração, realizada com sucesso.

Cenário Principal de Sucesso:

1. O usuário deve ir ao relatório do visitante.
2. O usuário deverá informar os dados do visitante referente à busca.
3. O usuário terá os dados do visitante, podendo alterá-los conforme sua necessidade, exceto código identificador do produto.
4. O usuário submete a alteração e o sistema validará os dados.
5. O sistema retornará mensagem de sucesso, “Visitante Atualizado”.

Cenário Secundário de Falha:

- 4.1. O sistema aborta a validação dos dados e retorna mensagem de erro, e mostra “Atualização Não Realizada”, “Faltando Dados”.

[UC 04] Cadastrar Veículo

Descrição: O usuário deverá fazer entrada dos dados referente a cada veículo.

Pré-condição: O usuário deverá estar logado no sistema. E modelo e placa do veículo ainda não cadastrado.

Pós-Condições: Retorno mensagem de veículo cadastrado.

Cenário Principal de Sucesso:

1. O usuário deve ir ao cadastro do veículo.
2. O usuário deverá informar os dados do veículo.
3. O usuário submete os dados necessários para armazenamento no banco de dados.
4. O sistema validará os dados e retorna mensagem “Cadastro com sucesso”.

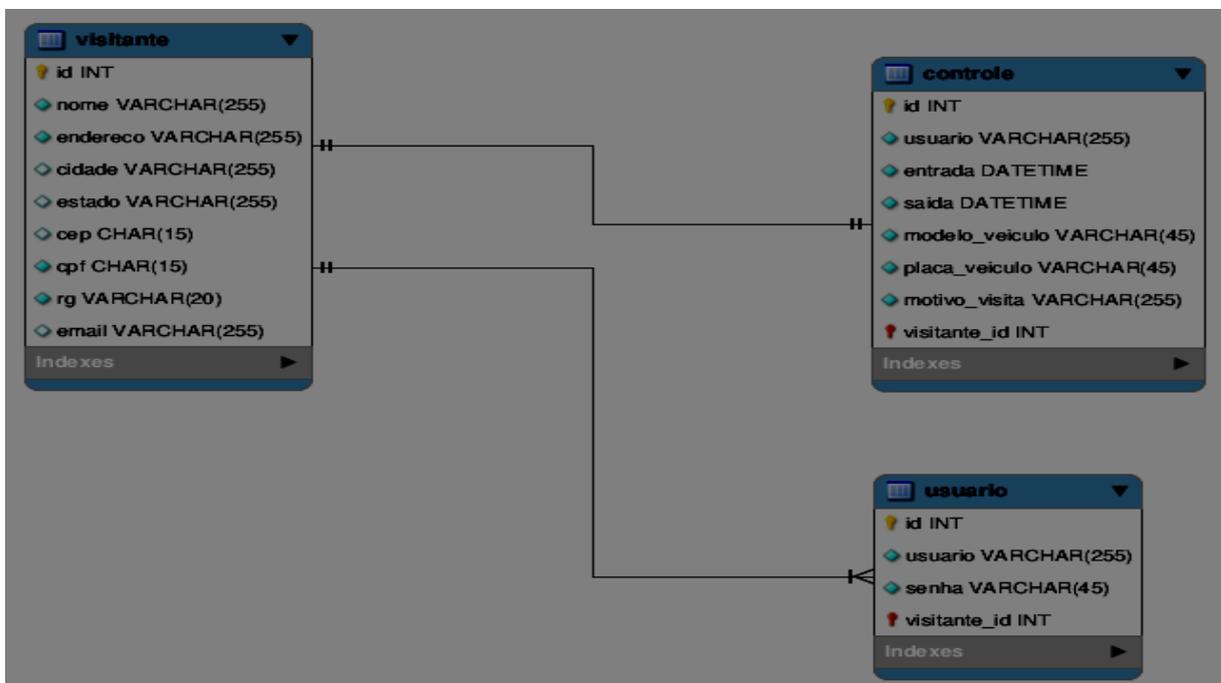
Cenário Secundário de Falha:

- 4.1. O sistema aborta a validação dos dados e retorna mensagem de erro, “Cadastro não realizado”.

2.1.4 Tabela de Relacionamento do sistema

Esta tela informa os relacionamentos feitos para criação do banco de dados.

Figura 6: Tabela de Relacionamento do Sistema



Fonte: O próprio autor (2017)

2.1.5 Sistema (Controle de Acesso)

O Sistema foi desenvolvido para a empresa Emex da cidade de Teófilo Otoni – MG, com a finalidade de controlar as entradas e saídas de visitantes no parque industrial da empresa, e também ter o cadastro de seus visitantes. Esse sistema teve uma mudança no decorrer de seu desenvolvimento com a implementação de um framework “Symfony”, conjunto de componentes PHP, um framework de Aplicações Web. Um framework que tem suporte para PHP, mas também suporte para banco de dados a exemplo do mariadb que foi o utilizado na realização deste trabalho e nos permite a realização de uma conexão e até mesmo a criação do banco, e tem como paradigma orientado a objeto um dos mais utilizados na linguagem PHP, nos próximos parágrafos será apresentada parte do sistema.

O sistema está dividido em:

1. Tela de autenticação com login e senha;
2. Tela de Pesquisa com opção de novo cadastro
3. Tela de cadastro de visitantes;
4. Tela de cadastro de veículos;

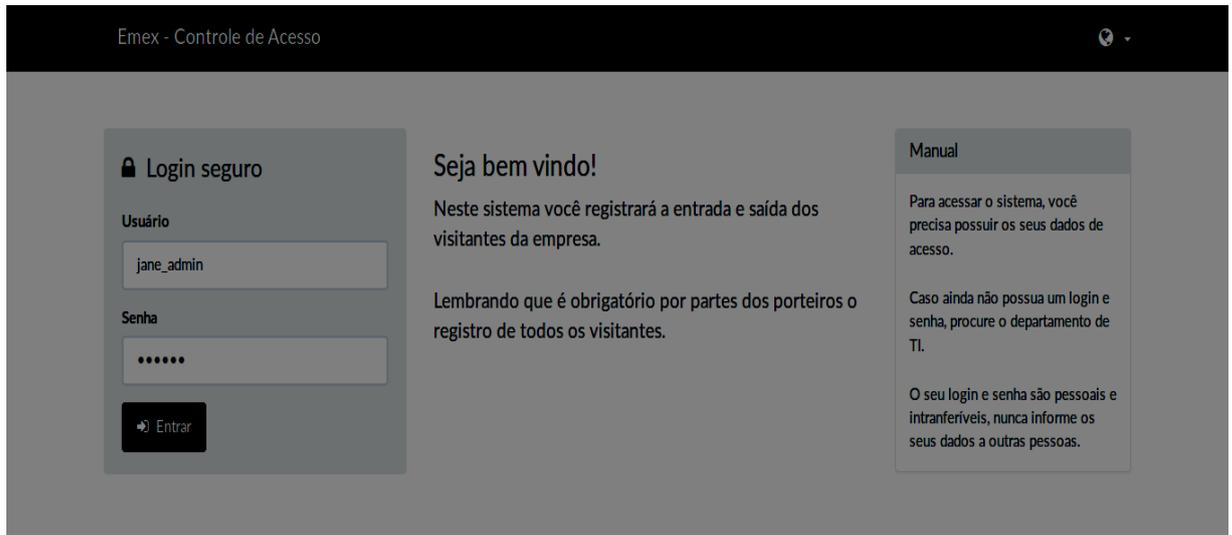
A partir das telas que o usuário fará toda a interação com o sistema.

2.1.6 Tela de acesso ao sistema

Para se fazer acesso ao sistema o usuário terá que informar um usuário e a senha corretamente para entrar no sistema. Se não for cadastrado será pedido para procurar o administrador para realizar o cadastro.

Todo cadastro de usuário será feito via prompt de comando (terminal) pelo administrador do sistema, por motivo de segurança.

Figura 7: Tela de Login do Sistema Controle de Acesso

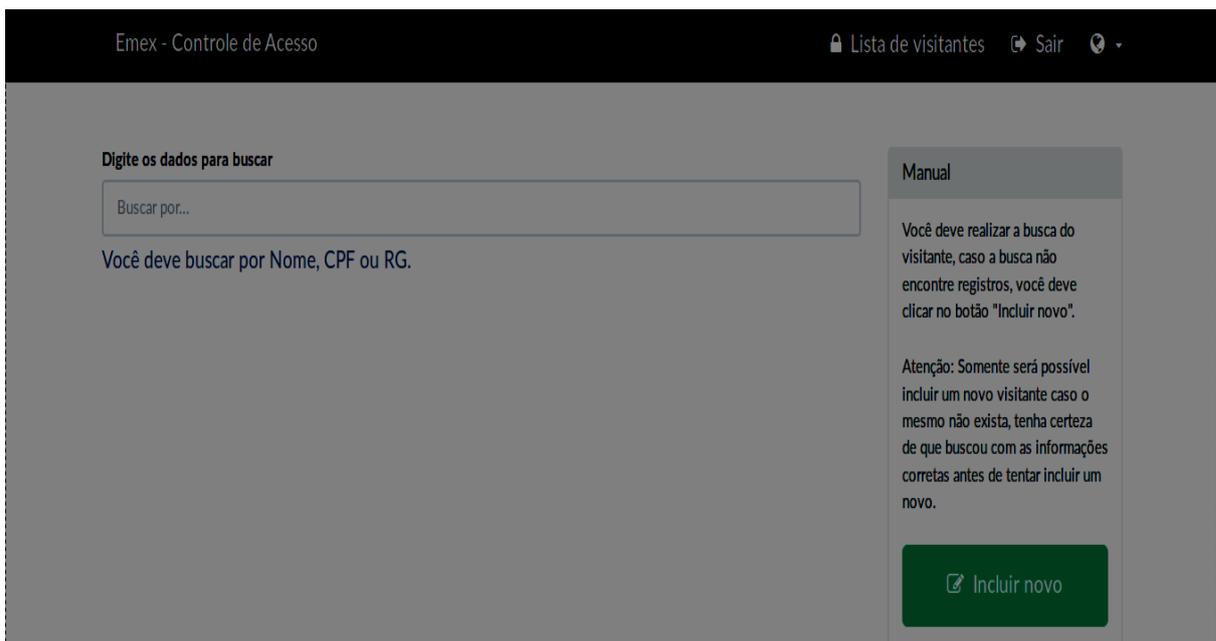


Fonte: O próprio autor (2017)

2.1.7 Tela de pesquisa do sistema controle de acesso

Essa tela permite ao usuário realizar a pesquisa de visitantes já cadastrados no sistema pelo nome, cpf ou rg apenas, não aceitando qualquer outro dado para pesquisa.

Figura 8: Tela de Pesquisa do Sistema controle de acesso



2.1.8 Tela de cadastro de visitantes do sistema controle de acesso

O cadastro de visitantes é onde o usuário informa todos os dados das visitas tais como o nome, endereço, cidade, estado, cep, cpf, rg e o email. Todos os itens do formulário são obrigatórios o preenchimento para validação do cadastro.

Figura 9: Tela Cadastro Visitante

Emex - Controle de Acesso

Lista de visitantes Sair

Adicionar contato

Nome

CPF

RG

CEP

Endereço

Cidade

UF

E-mail

[Salvar e iniciar visita](#) [Salvar e criar novo](#) [Voltar para lista de contatos](#)

© 2017 - Projeto TCC
Romildo Ferreira dos Santos

Fonte: O próprio autor (2017)

2.1.9 Tela de cadastro de veículos do sistema controle de acesso

O cadastro de veículo é onde o usuário informa os dados do veículo, tais como modelo e placa do veículo. São de preenchimento obrigatório. Essa tela inicia e encerra uma visita.

Figura 10: Tela de Cadastro de Veículos

Emex - Controle de Acesso

Lista de visitantes Sair

Romildo

CPF: 123
RG: 123
CEP: 123
Endereço: 123
Cidade: asdasd
UF: SP
E-mail:

Adicionar um comentário

Modelo veículo Placa veículo

Moto Honda CG 125 HX-2202

Motivo visita

Falar com Setor Laboratório.

Registrar entrada

Manual

Nesta tela você deve registrar a entrada e saída dos visitantes da empresa.

Preencha os dados da visita no formulário ao lado.

Caso exista alguma divergência do cadastro do visitante com os documentos informados, você deve alterar os dados do visitante clicando no botão "Alterar dados".

Alterar dados

Fonte: O próprio autor (2017)

2.1.10 Tela de listagem de visitantes do sistema controle de acesso

Esta tela apresenta todas as visitas já cadastradas no sistema no sistema controle de acesso, com opção de um novo registro e editar.

Figura 11: Tela de relação dos Visitantes Cadastrados

Emex - Controle de Acesso

Lista de visitantes Sair

Lista de visitantes

Nome	Ações
José da Silva	Registrar visita Editar
Romildo	Registrar visita Editar
Rosenilda Pereira Gomes	Registrar visita Editar
Tárik Gomes dos Santos	Registrar visita Editar
oio0ik0	Registrar visita Editar
josé da silva	Registrar visita Editar
Mariana Gomes dos Santos	Registrar visita Editar
José Rivaldo	Registrar visita Editar
Maria Stella dos Santos Ferreira	Registrar visita Editar
Mariana Gomes dos Santos	Registrar visita Editar
Rosane Ferreira dos Santos	Registrar visita Editar

Manual

Nesta tela são exibidos todos os cadastros de visitantes.

Você pode alterar os dados do visitante ou iniciar uma nova visita.

Buscar visitante

Fonte: O próprio autor (2017)

CONSIDERAÇÕES FINAIS

Este trabalho apresentou o estudo das possíveis vantagens em desenvolver um sistema web para controle de entrada e saída de visitantes como forma de auxiliar os usuários do setor de portaria a agregar conhecimentos, sem contar o ganho social, que vai ser a inclusão dos usuários do sistema à uma nova tecnologia, o ganho pessoal, conhecimento de uma nova ferramenta, o symfony e o ganho acadêmico como conhecimento de novas teorias.

Segundo a métrica da pesquisa realizada para alcançar o objetivo geral desse trabalho de conclusão de curso, foram levantados alguns objetivos específicos que serão apresentados a seguir:

Fazer o levantamento de requisitos para o sistema:

Para o desenvolvimento de qualquer software é necessário realizar diálogos com o usuário do sistema, pois será através desses diálogos que o desenvolvedor irá colher as informações ou requisitos necessários para o desenvolvimento.

Projetar os principais diagramas do sistema;

Para se iniciar um projeto, os diagramas são de suma importância para o desenvolvimento, no caso deste projeto foi criado o diagrama de caso de uso, ajudam a determinar a funcionalidade e as características do software sob o ponto de vista do usuário.

Pesquisar sobre as tecnologias a serem usadas nesse projeto;

Foram feitas todas as pesquisas das tecnologias usadas neste projeto como consta no referencial teórico.

Criar o banco de dados do sistema:

De acordo com os requisitos levantados, foi iniciada a fase do desenvolvimento do sistema começando com a modelagem do banco de dados atribuindo os campos respectivos de acordo com os requisitos coletados utilizando o SGBD MySQL.

Cadastrar as informações;

Foi cadastrado todas as informações do sistema

Implementar o sistema e testar;

O sistema foi desenvolvido havendo uma mudança no meio da implementação usando um framework Symfony.

No decorrer da pesquisa, foram levantadas hipóteses a fim de obter respostas para questão investigativa, sendo elas:

H0: Não haveria a possibilidade de implantação deste sistema pois o mesmo não agregará valores à empresa e seus usuários.

Em relação a hipótese acima **não foi validada**, visto que o sistema não foi implantado e assim não sendo possível agregar valores para a empresa e usuários.

H1: O desenvolvimento e a implantação deste sistema na empresa, possibilitaria um maior controle nas constantes entradas e saídas de seus visitantes.

Hipótese **não validada**, pois o sistema não foi implantado, não possibilitando assim controle das constantes e frequentes entradas e saídas de visitantes da empresa.

H2: Poderia haver por parte dos usuários uma recusa pela implantação deste sistema na empresa.

Sim, se o sistema fosse implantado, como não houve a implantação ficando uma **hipótese inválida**.

H3: Haveria com implantação deste sistema na empresa uma maior contribuição ao meio ambiente.

Haveria se o sistema fosse implantado, como não houve a implantação do sistema **hipótese inválida**.

H4: Poderia este sistema agregar algum conhecimento a seus usuários.

Poderia com implantação do sistema, mas o mesmo não foi implantado tornando a **hipótese inválida**.

A conclusão desse trabalho resultou no desenvolvimento e implementação de um sistema para desenvolvimento web, em que o mesmo mostrou ser uma ferramenta vantajosa, auxiliando os usuários de portaria da Emex a um maior controle das entradas e saídas de visitantes da empresa além do mais, a aplicação poderá ser utilizada em qualquer outra empresa com necessidade para controlar seus acessos. Para trabalhos futuros pode-se adicionar novos módulos.

Contudo foi proposto uma mudança no meio do projeto para se alcançar o objetivo, onde foi proposto que o desenvolvimento de implantação, fosse realizado a implementação de uma ferramenta, um framework gratuito e OpenSource, capaz de atender aos requisitos para implementação deste software.

Após várias pesquisas optou-se pelo framework que melhor atenderia à implementação deste projeto, a Symfony um framework de Aplicações Web, o que significa que ele permite que aplicativos web sejam criados em PHP!

Com base em tudo que foi abordado, este trabalho apresenta-se de grande importância, pois o desenvolvimento desse sistema fez com que agregasse ainda mais conhecimento na área de desenvolvimento web, permitindo que futuramente seja criado mais sistemas web satisfazendo a necessidade no mercado de trabalho.

REFERÊNCIAS

BEZERRA, Eduardo. *Princípios de análise e projeto de sistemas UML*, Rio de Janeiro, Elsevier, 2007.

DALL' OGLIO, Pablo. *Programando com orientação a objetos*. Rio de Janeiro: Editora Novatec, 2007.

_____. *Programando com orientação a objetos*. Rio de Janeiro: Editora Novatec, 2015.

ELMASRI, Ramez; NAVATHE, Shamkant B. *Sistema de Banco de Dados*. São Paulo: Editora Ciência Moderna Ltda, 2011.

FILHO, W. *Engenharia de Software fundamentos, métodos e padrões*. São Paulo: LTC – Livros Técnicos e Científicos Editora S.A, 2000.

_____. *Engenharia de Software fundamentos, métodos e padrões*. São Paulo: LTC – Livros Técnicos e Científicos Editora S.A, 2009.

FREEMAN, Elisabeth; FREEMAN, Eric. *Use a Cabeça! HTML com CSS & XHTML*. Rio de Janeiro: Editora Alta Books Ltda, 2008.

FORBELLONE, André Luiz Villar. *Lógica de Programação, a construção de algoritmos e estruturas de dados*. São Paulo: Pearson, 2005.

GIL, Antônio Carlos. *Métodos e técnicas de pesquisa social*. 5.ed. São Paulo: Atlas, 1999.

HEUSER, Carlos Alberto. *Projeto de Banco de Dados*. Rio de Janeiro: Editora Pearson Education do Brasil, 1998.

MADUREIRA, Omar Moore. *Metodologia do Projeto, planejamento, execução e gerenciamento*. São Paulo: Editora Edgard Blucher, 2010.

MIYAGUSKU, Renata Hiromi Minami. *Crie sites arrasadores*. São Paulo: Digerati Books, 2007. 128 p.

M C NEVES, Pedro. *O Guia Prático da HTML*. Lisboa: Editora Centro Atlântico, Ltda, 2004.

PRESSMAN, Roger S. *Engenharia de Software, Uma Abordagem Profissional*. Porto Alegre: AMGH Editora LTDA, 2011.

REZENDE, D. *Engenharia de software e sistemas de informação*. Rio de Janeiro: BRASPORT livros e Multimídia Ltda, 2005.

ROBERT W, Sebesta. *Conceitos de Linguagem de Programação*. Porto Alegre: Editora Bookman. 2003.

SAMY SILVA, Maurício. *JavaScript: Guia do Programador*. São Paulo: Novatec Editora Ltda. 2010.

_____. *CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das css3*. São Paulo: Editora Novatec Ltda, 2012.

_____. *Aprenda a usar o framework Bootstrap 3.3.5*. Rio de Janeiro: Editora Novatec LTDA 2015.

SOMMERVILLE, Ian. *Engenharia de Software*. São Paulo: Pearson Education do Brasil, 2003.

_____. *Engenharia de Software*. São Paulo: Pearson Education do Brasil, 2005.

_____. *Engenharia de Software*. São Paulo: Pearson Education do Brasil, 2011.

TOMANIK, Eduardo Augusto. *O olhar no espelho: "conversas" sobre a pesquisa em ciências sociais*. Maringá: EDUEM, 1994. p. 133-139.

VERGARA, Sylvia C. *Projetos e relatórios de pesquisa em administração*. São Paulo: Atlas 1998.

WELLING, Luke; THOMSON, Laura. *PHP e MySQL: Desenvolvimento Web*. Rio de Janeiro: Elsevier, 2005.

Apostila de banco de dados, disponível em: <<http://www.regilan.com.br>> acessado em: 30.out.2017.

DECRETO Nº 3.665, DE 20 DE NOVEMBRO DE 2000, Disponível em: <http://www.planalto.gov.br/ccivil_03/decreto/d3665.htm> acessado em: 1º.Jun.2017.

Free Software Foundation, disponível em: <<http://www.fsf.org>> acessado em: 07.nov.2017

MySQL Workbench 6.0, disponível em: <<http://www.fciencias.com/2013/08/21/mysqlworkbench-6-0-construa-base-de-dados-facilmente/>> acessado em: 30.mai.2017.

Mysql, disponível em: <<https://dev.mysql.com/doc/refman/5.7/en/introduction.html>> acessado em: 30.out.2017.

Symfony, disponível em:< <https://symfony.com/what-is-symfony> > acessado em: 02.nov.2017

APÊNDICE

Instituto Ensinar Brasil

INSTITUTO ENSINAR BRASIL
FACULDADES UNIFICADAS DE TEÓFILO OTONI
PORTARIA MEC Nº 269 DE 18/04/2016

rede de ensino
DOCTUM

CARTA DE PEDIDO DE PERMISSÃO PARA REALIZAÇÃO DE PESQUISA CIENTÍFICA

Aos cuidados de,

EMEX – Empresa Mineira de Explosivos Ltda. CNPJ: 04.570.580/0001-03, situada na Rua Coronel Mário Cordeiro, 310, Jardim Serra Verde, Teófilo Otoni-MG, CEP: 39800-457.

As **Faculdades Unificadas de Teófilo Otoni**, Rede de Ensino Doctum, situadas na Rua Gustavo Leonardo, 1127 – São Jacinto, Teófilo Otoni-MG, CEP: 39801-260, têm tradição na seriedade com os trabalhos acadêmicos que buscam, acima de tudo, contribuir para o desenvolvimento social e econômico da região, seja com trabalhos mais abrangentes, seja com trabalhos internos em organizações que exercem fundamental importância para a nossa comunidade.

A coordenação do curso de **Bacharelado em Sistemas de Informação** destas Faculdades, vem por meio deste, na pessoa do Coordenador de Curso, **Wilbert Viana Barbosa**, solicitar permissão ao aluno deste curso, **Romildo Ferreira dos Santos**, sob matrícula nesta instituição nº **140500045**, para realizar pesquisa científica no interior de sua organização durante o 2º semestre de 2017.

Ressaltamos o compromisso com o sigilo e preservação da imagem e outras informações da empresa sob quaisquer aspectos. Ainda garantimos que a utilização dos dados coletados será exclusivamente para fins acadêmicos e não comerciais.

Sob assinatura abaixo, a Coordenação e o aluno supracitados firmam o compromisso.



Wilbert Viana Barbosa
Coordenador - Sistemas de informação
Faculdades Unificadas de Teófilo Otoni
Rede de Ensino Doctum



Romildo Ferreira dos Santos
Matrícula: 140500045

A empresa para a qual foi endereçada essa petição, deferi o pedido.

Teófilo Otoni, 20 de novembro de 2017.



Assinatura do Representante.