

ERIVELTON DA SILVA

SISTEMA GERADOR DE FORMULÁRIOS E SCRIPTS PHP

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do título de Bacharel em Ciência da Computação.

Orientador

Prof. Reginaldo Ferreira de Souza

LAVRAS

MINAS GERAIS – BRASIL

2003

ERIVELTON DA SILVA

SISTEMA GERADOR DE FORMULÁRIOS E SCRIPTS PHP

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do título de Bacharel em Ciência da Computação.

Aprovada em 12 de Dezembro de 2003.

Prof. André Luiz Zambalde

Prof. Reginaldo Ferreira de Souza

UFLA

(Orientador)

LAVRAS

MINAS GERAIS – BRASIL

SUMÁRIO

1. Introdução.....	1
1.1. Definição.....	1
1.2. Objetivos.....	1
1.3. Funcionamento.....	2
2. Referencial teórico.....	4
2.1. Ciclo de Vida Clássico.....	4
2.2. Linguagem PHP (Personal Home Page).....	5
2.2.1. Definição.....	6
2.2.2. CGI Vesus PHP.....	6
2.2.3. Histórico.....	7
2.3. MySQL.....	8
2.4. UML.....	9
2.5. Borland Delphi.....	10
3. Metodologia.....	12
3.1 Descrição do projeto.....	12
3.1.1 Características das páginas.....	12
3.1.2 Formulário de entrada de dados.....	14
3.1.3. Arquivos gerados pelo SGFE.....	16
3.1.4. Requerimentos do SGFE.....	17
3.1.5. Linguagens.....	18
3.1.5.1. Desenvolvimento do software.....	18
3.1.5.2. Scripts.....	19
3.2. Análise e projeto.....	19
3.2.1. Entidades.....	20
3.2.2. Requisitos funcionais	20
3.2.3. Requisitos não funcionais.....	21
3.2.4. Diagramas de use case.....	22
4. Implementação.....	25
4.1. Classes.....	25
4.1.1. Classe página.....	26
4.1.2. Classe formulário.....	27
4.1.3. Classe questionário.....	28
4.1.4. Classe saída.....	28
4.1.5. Classe configurações.....	29
4.1.6. Classe geradora.....	30
4.1.7. Classe projeto.....	31
4.2. Diagrama de classe.....	32
4.3. Operações implementadas.....	33
4.3.1. Operações de criação do banco de dados.....	37
4.3.2. Geração do formulário de entrada de dados.....	37

4.3.3. Geração dos scripts de saída.....	38
5. Resultados.....	38
5.1. Interface do SGFE.....	38
5.1.1. Janela de configuração do desenho das páginas e formulário	40
5.1.2. janela de configuração e navegação das questões contidas no formulário	41
5.1.3. Janela de configuração dos scripts de saída.....	46
5.1.4. Janela de geração dos códigos.....	47
5.2. Exemplos de saídas do software.....	43
5.2.1. Uma simples enquete.....	44
5.2.2. Um formulário de cadastro de currículos.....	53
5.3. Melhorias do software.....	54
7. Conclusões.....	55
6. Bibliográfica.....	56

LISTA DE FIGURAS

Figura 1 - Representação do ciclo de vida clássico.....	5
Figura 2 - Server-side scripts.....	7
Figura 3 - Janela de configuração de página e formulário.....	15
Figura 4 - Formulário de entrada de dados.....	17
Figura 5 - Janela de configuração do SGFE.....	20
Figura 6 - Diagrama de use case.....	25
Figura 7 - Diagrama de caso de uso.....	26
Figura 8 - Classe pagina.....	27
Figura 9 - Classe formulário.....	28
Figura 10 - Classe questionário e classe questão.....	29
Figura 11 - Classe saída.....	30
Figura 12 - Classe configurações.....	31
Figura 13 - Classe geradora.....	32
Figura 14 - Classe projeto.....	33
Figura 15 - Diagrama de classe.....	34
Figura 16 - Esquema das tabelas de banco de dados.....	35
Figura 17 - Janela do SGFE.....	36
Figura 18 - Janela de configuração das páginas e formulários.....	4
Figura 19 - Janela de navegação das questões.....	45
Figura 20 - Janela de configuração dos scripts de saída.....	46
Figura 21 - Janela de geração dos códigos.....	47
Figura 22 - Criando uma enquête.....	49
Figura 23 - Formulário de instalação da base de dados.....	50
Figura 24 - Formulário de confirmação da instalação da base de dados	51
Figura 25 – Enquête	52
Figura 26 - Página resposta.	53

Figura 27 - gráfico estatístico da enquête.....	54
Figura 28 - Formulário de cadastro de informações.....	56
Figura 29 - Janela de relatório.....	57

SISTEMA GERADOR DE FORMULÁRIOS E SCRIPTS PHP

RESUMO

Essa monografia apresenta o desenvolvimento de um aplicativo, que facilitará o processo de modelagem e codificação de *scripts* em linguagem PHP. A partir de uma interface simples e direta o Sistema Gerador de Formulários e *Scripts* PHP (SGFE) auxilia na criação de *homepages* que requerem objetos de interação com o internauta, como formulários de entrada de dados.

GENERATOR SYSTEM OF FORMS AND PHP SCRIPTS

ABSTRACT

This project has a objective to introduce the application development for web use. This software generate interactive pages for use in internet. The SGFE (Electronic Form System Generator) is a useful and easy tool to web development able to design, modeling, codify and manage the scripts and web pages. The software interface allow the automatic development of codes, the user don't need to know program in PHP or HTML language, the code will be generate using the user decisions get from software interface.

1. Introdução

Essa monografia apresenta o desenvolvimento de um aplicativo cujo objetivo é auxiliar e facilitar o desenvolvimento de páginas web, no desenvolvimento deste material esse aplicativo é nomeado com a sigla SGFE, que significa sistema gerador de formulários eletrônicos e *scripts* PHP.

1.1. Definição

A criação de formulários de pesquisa *web* usando *scripts* ou CGI (*common gateway interface*) com acesso a um sistema de banco de dados é um processo que engloba as seguintes fases: projeto, implementação, testes e instalação, assim como o ciclo de desenvolvimento de um software. Diante disso, existe uma necessidade de ferramentas que auxiliem o projeto de formulários eletrônicos assim como existem para o desenvolvimento de softwares, como o Borland Delphi™ ou MS Visual Studio™, por exemplo.

O SGFE é um aplicativo que auxilia o programador ou *webmaster* no ciclo de desenvolvimento de páginas e formulários eletrônicos, o qual poderá ser definido como uma interface aplicada à geração de códigos para a internet a partir de decisões e variáveis definidos pelo usuário.

O processo de criação dos códigos e a modelagem do banco de dados são totalmente automatizados, o usuário do *software* deve apenas definir o desenho e o conteúdo das páginas, desprezando então a necessidade de conceitos de programação por parte do usuário, podendo-se então definir o SGFE como uma ferramenta RAD (*Rapid Application Development*) especializada em aplicações *web*.

1.2. Objetivos

Os principais objetivos são facilitar e otimizar a produção de páginas *web* iterativas a partir do desenvolvimento de um software gerador de códigos.

Várias linguagens de programação comuns na internet podem ser usadas nesse processo de codificação automática, como PHP (*Personal Home Page*), Java, Delphi CGI, Perl, entre outras. Nessa primeira versão apresentada, os códigos dos *scripts* serão gerados usando a linguagem PHP, visto que essa linguagem é bastante difundida nos servidores *web*.

Em primeira instancia, a base de dados utilizada será o MySQL, visto que esse SGBD (Sistema Gerenciador de Banco de Dados) é gratuito e livre, mas o aplicativo SGFE poderá tratar também outros mecanismos como BDE ou MySQL Server.

A geração produz *scripts* e páginas otimizadas para a *internet*, tentando sempre equilibrar conteúdo com interface, filtrando os inconvenientes causados por páginas rebuscadas de componentes gráficos, multimídia e\ou *scripts* sem utilidade. A partir disso é possível salientar que uma das metas principais do SGFE é criar páginas *web* iterativas otimizadas, cujo tempo de carregamento pelo *browser* seja o mínimo possível, mas sem prejudicar a qualidade da interface gráfica.

1.3. Funcionamento

O SGFE fará a geração dos códigos a partir de entradas de informações feitas pelo usuário. Esses conjuntos de informações serão configurados pelo

usuário a partir de uma interface de janelas que são compostas de componentes de entrada de dados comuns no ambiente *Windows*, como campos de edição ou campos de escolha.

A partir dessas decisões tomadas pelo usuário o SGFE fará a geração dos códigos, esse conjunto de decisões está relacionado ao tipo de formulário ou página que será criado, por exemplo, se este for um formulário de pesquisa, o usuário deverá entrar com o cabeçalho das questões e as alternativas que formarão essas questões, essas características definem o conteúdo do formulário. O usuário também poderá decidir o *designer* (desenho) da página, como: cores, fontes e alinhamento.

Essa característica de gerar códigos sem a necessidade do usuário ter que saber modificar os códigos é requisito principal do SGFE, o qual faz: a geração automática dos formulários de entrada de dados, os *scripts* de modelagem e instalação da base de dados, os *scripts* que submetem os dados do formulário para as tabelas do banco de dados, e *scripts* que permitem que usuário administre a base de dados, como por exemplo, um *script* que gere um gráfico estatístico.

Como se trata de códigos para internet, estes deverão usar um servidor local ou não para os testes, este servidor poderá ser o *Apache Web Server* ou *Personal Web Server*, ou outro que seja compatível com os scripts gerados pelo SGFE (linguagem PHP). Para fins de testes foi instalado um servidor local, *Apache Web Server*.

2. Referencial teórico

Nesse capítulo é apresentado os métodos e materiais utilizados como base no desenvolvimento do software SGFE., bem como palavras chaves que são referencia para o conteúdo dessa monografia.

2.1. Ciclo de Vida Clássico

O ciclo de vida clássico é um paradigma da engenharia de software que propõe seqüencialmente etapas no desenvolvimento de um software. Nesse ciclo o nível de criação de software cresce progressivamente através da análise, projeto codificação, testes e manutenção.

Esse ciclo está representado na figura 1.

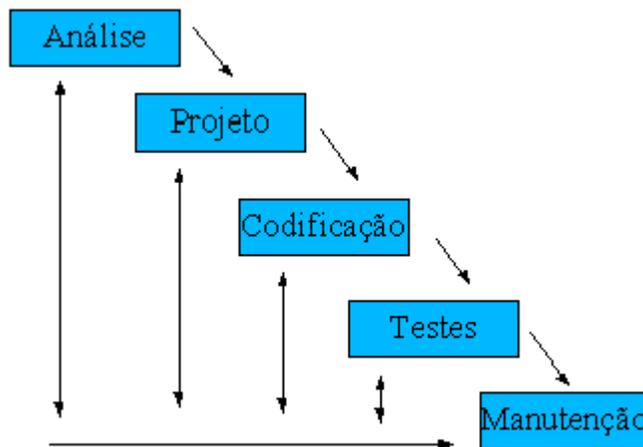


Figura 1 – Representação do ciclo de vida clássico.

Fonte: PRESSMAN(1992)

A seguir é explicado sucintamente cada passo desse paradigma:

Análise : nessa fase é feito um levantamento de todos requerimentos que o sistema necessita. Essa fase é fundamental para se obter informações como software e hardware que serão necessários para o funcionamento do sistema, bem como a análise das pessoas que utilizaram esse sistema.

Projeto : o projeto de software é um processo formado por diversos passos que visam distinguir quatro atributos principais de um programa: estrutura dos dados, arquitetura do software, detalhes dos procedimentos e características da interface. Nessa fase os requerimentos do sistema são traduzidos em um modelo ou representação do software.

Codificação ou implementação : o projeto ou modelo é traduzido para o formato de máquina, isto é, a partir do modelo os códigos serão implementados.
Manutenção : depois de instalado o sistema, este pode sofrer alterações ou manutenção, se este vier a sofrer algum erro. A manutenção é uma fase que visa atualizar o sistema a partir de erros encontrados no mesmo. Porém, muitas vezes quando um erro é consertado, outros podem aparecer a partir do concerto do mesmo, e isto gera graves problemas, como por exemplo, insatisfação do cliente ou usuário.

A desvantagem principal de se usar esse paradigma está no fato de que o usuário só terá seu produto após um alto tempo de espera, uma das maneiras de se resolver isso é utilizar o paradigma dos protótipos.

Porém, como o SGFE se tratará de um software projeto, este poderá ser desenvolvido usando o ciclo de vida clássico, visto que não haverá nenhum cliente ou usuário impaciente esperando pelo software. PRESSMAN(1992).

2.2. Linguagem PHP (Personal Home Page)

PHP é uma linguagem que permite criar páginas eletrônicas (*web sites*) dinâmicas, possibilitando uma interação com o usuário através de formulários, e *links*. A diferença de PHP com relação a linguagens semelhantes a Javascript é que o código PHP é executado no servidor (*server-side scripts*), sendo enviado para o cliente apenas o código HTML puro, desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente, a figura 2 representa o funcionamento de *server-side script*. Essa vantagem pode ser útil quando o programa está lidando com senhas ou qualquer tipo de informação confidencial.

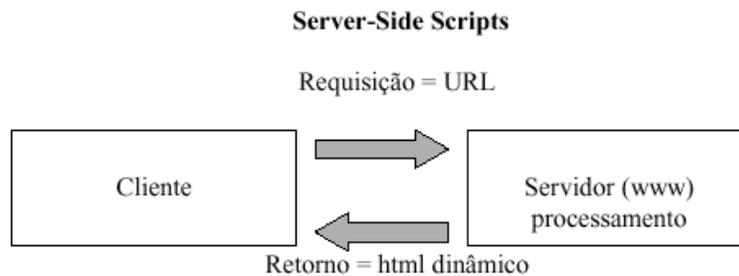


Figura 2– Server-side scripts.

Fonte: LERDORF(1999)

2.2.2. Comparação com CGI (Common Gateway Interface)

O que diferencia PHP de um *script* CGI escrito em C ou Perl é que o código PHP fica embutido no próprio HTML, enquanto no outro caso é necessário que o *script* CGI gere todo o código HTML, ou leia de um outro arquivo.

Portanto, qualquer aplicativo *web* que pode ser feita por algum programa CGI pode ser feito também com PHP, como coletar dados de um formulário, gerar páginas dinamicamente ou enviar e receber *cookies*.

PHP também tem como uma das características mais importantes o suporte a um grande número de bancos de dados, como dBase, Interbase, mSQL, MySQL, Oracle, Sybase, PostgreSQL e vários outros. Construir uma página baseada em um banco de dados torna-se uma tarefa extremamente simples com PHP.

2.2.3. Histórico

A linguagem PHP foi concebida durante o outono de 1994 por **Rasmus Lerdorf**. As primeiras versões não foram disponibilizadas, tendo sido utilizadas em sua *home-page* apenas para que ele pudesse ter formações sobre as visitas que estavam sendo feitas. A primeira versão utilizada por outras pessoas foi disponibilizada em 1995, e ficou conhecida como “*Personal Home Page Tools*” (ferramentas para página pessoal). Era composta por um sistema bastante simples que interpretava algumas *macros* e alguns utilitários que rodavam “por trás” das *home-pages*: um livro de visitas, um contador e algumas outras coisas.

Em meados de 1995 o interpretador foi reescrito, e ganhou o nome de PHP/FI, o “FI” veio de um outro pacote escrito por Rasmus que interpretava dados de formulários HTML (**F**orm **I**nterpreter). Ele combinou os scripts do pacote *Personal Home Page Tools* com o FI e adicionou suporte a mSQL, nascendo assim o PHP/FI, que cresceu bastante, e as pessoas passaram a contribuir com o projeto.

Estima-se que em 1996 PHP/FI estava sendo usado por cerca de 15.000 *sites* pelo mundo, e em meados de 1997 esse número subiu para mais de 50.000. Nessa época houve uma mudança no desenvolvimento do PHP. Ele deixou de

ser um projeto de Rasmus com contribuições de outras pessoas para ter uma equipe de desenvolvimento mais organizada. O interpretador foi reescrito por **Zeev Suraski** e **Andi Gutmans**, e esse novo interpretador foi a base para a versão 3.

Atualmente o uso do PHP3 vem crescendo numa velocidade incrível, e já está sendo desenvolvida a versão 4 do PHP. LERDORF(1999)

2.3. MySQL

O MySQL é servidor de banco de dados multiusuário, multitarefa que trabalha com uma das linguagens de manipulação de dados mais comuns do mundo.

SQL é uma linguagem simples, em que você facilmente pode gravar, alterar e recuperar informações num web site com segurança e rapidez. Ela foi desenvolvida pelo Departamento de Pesquisas da IBM como forma de interface para o Sistema de Banco de Dados Relacionais **SYSTEM R**, no início dos anos 70; em 1996, a *American National Institute (ANSI)* publicou um padrão SQL. A SQL estabeleceu-se como linguagem padrão de Banco de Dados Relacional. A linguagem SQL tem como grande virtude sua capacidade de gerenciar índices sem a necessidade de controle individualizado de índice corrente, algo muito comum nos Sistemas Gerenciadores de Arquivos, o Dbase por exemplo.

O MySQL foi originalmente desenvolvido pela empresa sueca **TCX**, que necessitava de um servidor de banco de dados que operasse com grandes escalas de dados rapidamente sem exigir caríssimas plataformas de hardware. A **TCX** opera desde 1996 com 40 bancos de dados, contendo 10.000 tabelas, sendo 500 delas com mais de 10 milhões de linhas.

Características do MySQL:

- Suporta diferentes plataformas: Win32, Linux, FreeBSD, Unix, etc...
- Suporte às APIs das seguintes linguagens: PHP, Perl, C, C++, Java, Python, etc... Suporte a múltiplos processadores
- Um sofisticado sistema de senhas criptografadas flexível e seguro.
- Suporte à ODBC, você pode facilmente conectar o Access a um banco de dados do MySQL
- Suporta até 16 índices por tabela
- Código fonte escrito em C e C++ e testado com uma variedade de diferentes compiladores
- O Cliente conecta no MySQL através de conexões TCP/IP.
- Nenhum problema com o Y2K, visto que o MySQL usa o relógio do Unix que não apresentará problemas até 2069. [MySQL(2003)]

2.4. UML(Unified Modeling Language)

O desenvolvimento de sistemas de software de grande porte são suportados por métodos de análise e projeto que modelam esse sistema de modo a fornecer para toda a equipe envolvida (cliente, analista, programador, etc.) uma compreensão única do projeto.

A UML (Unified Modeling Language) é o sucessor de um conjunto de métodos de análise e projeto orientados a objeto (OOA&D). A UML está, atualmente, em processo de padronização pela OMG (Object Management Group).

A UML é um modelo de linguagem, não um método. Um método pressupõe um modelo de linguagem e um processo. O modelo de linguagem é a notação que o método usa para descrever o projeto. O processo são os passos que devem ser seguidos para se construir o projeto.

O modelo de linguagem é uma parte muito importante do método. Corresponde ao ponto principal da comunicação. Se uma pessoa quer conversar sobre o projeto, como outra pessoa, é através do modelo de linguagem que elas se entendem. Nessa hora, o processo não é utilizado.

A UML define uma notação e um meta-modelo. A notação são todos os elementos de representação gráfica vistos no modelo (retângulo, setas, o texto, etc.), é a sintaxe do modelo de linguagem. A notação do diagrama de classe define a representação de itens e conceitos tais como: classe, associação e multiplicidade. Um meta-modelo é um diagrama de classe que define de maneira mais rigorosa a notação. [QUATRANI(200)]

2.5. Borland Delphi

Object (ou Delphi) Pascal, é um conjunto de extensões orientadas a objeto para o antigo Pascal, está é a linguagem da ferramenta Delphi.

Delphi pascal é uma linguagem de alto nível, compilável, além disso é uma linguagem robusta que suporta estruturas e orientação a objeto. Os principais benefícios do delphi são: implementação de códigos fáceis de ler, rápida compilação, e o uso de diversas unidades (*units*) para a programação linear.

Borland Delphi é um sofisticado ambiente de programação Windows, adaptável para programadores iniciantes e profissionais. Usando Delphi é possível facilmente produzir aplicativos de alto nível para Windows e atualmente para linux (Delphi versão 6 e posteriores). Uma das grandes razões da difusão de seu uso está na larga coleção de componente gráficos padrões e a possibilidade de se criar componentes novos. Além disso os componentes permitem a criação de aplicativos em pouco tempo, pois remove a necessidade de se codificar um a um cada *gadget* contida em uma janela.

A partir disso pode-se então definir que: Borland Delphi (1,2,3,4,5,6,7) é uma ferramenta de desenvolvimento para aplicações Microsoft Windows.

Delphi é uma ferramenta poderosa e fácil de usar para a criação de programas stand-alone graphical user interface (GUI).

Delphi é a primeira linguagem de programação que quebra a barreira entre alto-nível, facilidade e agilidade no desenvolvimento de aplicativos.

Delphi pode acessar muitos tipos de banco de dados através do BDE (Borland Database Engine). Com o BDE é possível acessar banco de dados locais como Paradox e Dbase, e é também possível acessar banco de dados em rede, como InterBase e SysBase. DELPHI(2003)

3. Metodologia

O desenvolvimento desse software englobou um conjunto de fases, baseando-se no ciclo de desenvolvimento clássico. A partir de uma análise de requisitos foi obtida a modelagem desse sistema.

Nesse capítulo é apresentado todo o processo de desenvolvimento desse software.

3.1 Descrição do projeto

O SGFE é um *software* capaz de gerar códigos para uso na *internet*, páginas *web* e *scripts*. O usuário do SGFE através de uma interface desenvolvida em ambiente *windows* decidirá as características e conteúdo do formulário de entrada, páginas de saída e páginas de controle dos dados. A partir dessas decisões os códigos serão gerados automaticamente dentro do computador.

3.1.1 Características das páginas

Defini-se como características das páginas o conjunto de decisões relacionadas com o desenho (*design*) das páginas geradas, como :

- Cores de fundo,
- Tamanho, tipo, e cor da fonte,
- Alinhamento de página e texto,
- Margens,
- Títulos, cabeçalhos,
- Separadores (quebras horizontais).

Essas características são de livre escolha do usuário, e é feita usando uma interface de janelas contendo campos de entrada de dados, nos quais o usuário poderá decidir as características visuais das páginas.

É possível sempre visualizar a página durante o processo de desenvolvimento da mesma, auxiliando as decisões do usuário, apresentando a ele uma visão parcial da página que será criada pelo SGFE.

Nesse processo de decisões, em nenhum momento o usuário deve escrever ou alterar nenhum tipo de código, pois o processo de criação é automático.

A Figura 3 apresenta a janela de configuração das páginas, nota-se que nessa janela encontram-se componentes padrões do sistema operacional Windows, que qualquer usuário com conhecimento básico sobre o sistema operacional *Windows* consegue manipular.

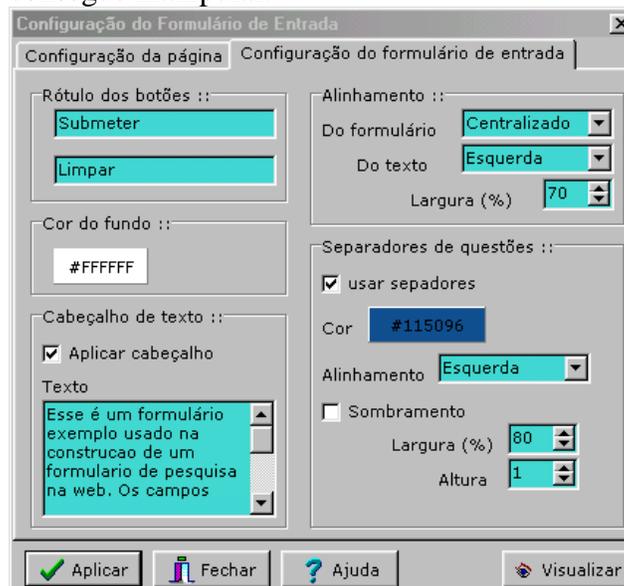


Figura 3 – Janela de configuração de página e formulário.

Fonte: Tela extraída do software desenvolvido.

3.1.2 Formulário de entrada de dados

Um formulário de entrada de dados é uma página de *internet* que é composta por um conjunto de componentes que permitem que o internauta interaja com o *website*. Os principais componentes de um formulário são:

1. Botões de rádio (radio button), esse componente permite que o internauta escolha uma única alternativa dentre várias,
2. Botões de verificação (check button), ou caixa de verificação, esse componente permite a escolha de várias alternativas em um conjunto, cada alternativa possui dois estados, verificado ou não,
3. Caixa de seleção de texto ou lista de seleção de texto (select list), esse componente contém um conjunto de palavras, no qual o internauta deve fazer uma única escolha,
4. Caixa de entrada de texto (edit box), esse componente permite a entrada de texto,
5. Área de texto (text area), esse componente semelhante a caixa de entrada de texto, porém permite a entrada de múltiplas linhas de texto.

A Figura 4 mostra um formulário gerado pelo SGFE contendo todos os tipos de campos descritos. A partir dos componentes apresentados acima é possível desenvolver formulários para diversas funções, como por exemplo pesquisas e cadastros de informações.

Escolha várias alternativas abaixo :

Alternativa 1

Alternativa 2

Alternativa 3

Escolha apenas uma alternativa no campo abaixo:

alternativa 1

alternativa 2

Escolha na caixa de seleção a seguir um texto qualquer :

Digite um texto

Escreva um texto na caixa abaixo

Figura 4 – Formulário de entrada de dados.

Fonte: Formulário *web* gerado pelo software.

O SGFE faz uma padronização dos campos que compõem o formulário de entrada, definindo-os como questões. Uma questão pode ser dita aqui como um cabeçalho (pergunta) e alternativas (resposta), a partir disso foi definido que: para cada tipo de componente existe um tipo de questão, e esse conjunto de questões forma o formulário de entrada de dados.

O SGFE é capaz de criar formulários de entrada de dados contendo os seguintes tipos de questões:

- Questão de rádio,
- Questão de verificação

- Questão de seleção
- Questão de entrada de texto
- Questão de caixa de texto.

3.1.3. Arquivos gerados pelo SGFE

A partir das decisões feitas pelo usuário o SGFE faz a geração dos seguintes arquivos (saída):

Arquivo de projeto, esse arquivo é onde o projeto em execução é salvo, para posterior uso, esse arquivo guarda as decisões tomadas pelo usuário relacionados aos campos do formulário de entrada e os atributos relacionados ao desenho das páginas. Foi usado como padrão a extensão .eri para o arquivo de projeto criado pelo SGFE.

Página de entrada de dados ou formulário, o nome desse arquivo deve ser definido pelo usuário, esse arquivo é gerado usando unicamente HTML, e contém uma chamada ao *script* PHP responsável pelo funcionamento do formulário(“executar.php”).

Arquivo “executar.php”, esse *script* é o código que insere os dados do formulário em um Banco de Dados.

Página de instalação do banco de dados, essa página (“instalar.htm”) executa o arquivo “instalar.php”, esse *script* executa a instalação das tabelas em um Banco de Dados MySQL.

Arquivos de saída, são arquivos que manipulam os dados das tabelas de banco de dados, esses *scripts* são gerados a partir de decisões do usuário, e a quantidade de *scripts* é dinâmica, permitindo diversos tipos de recuperação das informações no banco de dados, por exemplo um gráfico apresentando um cálculo estatístico sobre determinada questão ou uma página contendo todos (ou

um conjunto de) dados sobre uma determinada questão (ou várias) como em um cadastro de mala-direta.

Todos os arquivos descritos aqui são gerados automaticamente a partir de decisões tomadas do usuário a partir de um conjunto de janelas, portanto o usuário não precisa possuir nenhum tipo de conhecimento sobre a linguagem que está sendo gerada.

3.1.4. Requerimentos do SGFE

O SGFE requer que o computador no qual ele foi instalado tenha um servidor de *internet* local para testes, para que os códigos sejam testados em tempo real. No entanto esse requerimento é desnecessário caso o usuário não queira visualizar o funcionamento dos *scripts* de seu projeto em tempo real. Esse requerimento é decorrente do fato de que todos *scripts* são *server-side*, ou seja funcionam somente quando estão alojados em um servidor. Como existem servidores locais, como o *Apache Web Server* é possível então simular o funcionamento desses *scripts*, antes de que esses sejam colocados em um servidor público de *internet*.

O local (diretório) de geração dos arquivos deve ser definido pelo usuário, como se trata de códigos para *internet (client-server)*, o local de instalação desses códigos tem que ser dentro da pasta usada pelo servidor, para total funcionamento dos *scripts* (testes em tempo real).

A Figura 5 mostra a janela de configuração do SGFE. O campo *IP* ou *Server Name* é o campo no qual o usuário deve configurar o nome do servidor no qual os códigos funcionaram. O campo *HTTPDoc path* é o caminho onde fica o diretório em que os códigos deverão ser instalados.

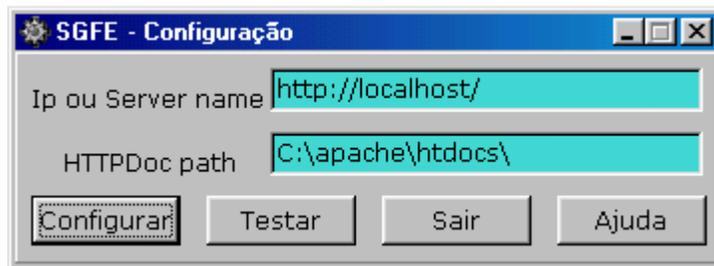


Figura 5 – Janela de configuração do SGFE.

Fonte: Janela extraída do software.

Os valores apresentados pela Figura 5 representa a configuração de testes locais usando o servidor *Apache Web Server*. Essa configuração é necessária para se testar os códigos gerados, durante o processo de desenvolvimento do mesmo. No entanto, os códigos gerados podem ser instalados em qualquer tipo de servidor, desde que o serviço suporte as tecnologias usadas pelos códigos.

3.1.5. Linguagens

Nesse tópico são apresentadas as linguagens que foram usadas para desenvolver o software SGFE e suas saídas (*scripts*).

3.1.5.1. Desenvolvimento do software

O SGFE é um software que foi desenvolvido usando a ferramenta *Borland Delphi 5*®. Foi escolhido essa ferramenta devido a sua capacidade de equilibrar qualidade e velocidade de desenvolvimento, principalmente quando o problema requer o desenvolvimento de uma interface gráfica.

O *Delphi* possui um SGBD (sistema gerenciador de banco de dados), denominado BDE, de alto nível, porém seu uso foi descartado nesse projeto, pelo fato de que a linguagem de *scripts* gerados (saída do SGFE) usa um outro tipo de SGBD, o *MySQL*, que é um SGBD multiplataforma e desenvolvido especialmente para uso na *internet*.

3.1.5.2. Scripts

A linguagem PHP foi escolhida para a geração dos *scripts*, dado que essa linguagem é amplamente difundida na *internet* e de fácil configuração. Outras características que a torna a melhor linguagem de *script* são: portabilidade e segurança.

A versão usada no desenvolvimento desse projeto foi a versão mais recente do PHP: 4.3.3.

Deve-se destacar ainda que os *script(s)* são códigos que permitem a troca de informações entre o servidor e o navegador usando a linguagem de hipertexto HTML. Por exemplo, o formulário de entrada de dados é constituído pela linguagem HTML, que usando um *script* PHP consegue interagir os dados entrados no formulário pelo internauta com o Banco de Dados no servidor.

3.2. Análise e projeto

Na fase de análise são identificadas as classes, objetos e os mecanismos que estarão presentes no domínio do problema. As classes são modeladas e interligadas através de relacionamentos utilizando o Diagrama de Classe. Na análise, só serão modeladas classes que pertençam ao domínio do problema, classes que gerenciem banco de dados, comunicação, interface e outros não estarão presentes nesse modelo.ESMIM

O resultado da análise é expandido na fase de projeto em soluções técnicas. Novas classes serão adicionadas para prover uma infra-estrutura técnica: a interface do usuário e de periféricos, gerenciamento de banco de dados, comunicação com outros sistemas, entre outros.ESMIN

3.2.1. Entidades

A partir da descrição apresentada pode se definir as seguintes entidades para a modelagem do sistema:

- SGFE: essa entidade representa o software Sistema gerador de formulários eletrônicos;
- Usuário: entidade física que representa a pessoa que usará o SGFE para criar as páginas eletrônicas.
- Internauta: representa a pessoa que pelo navegador acessa a página criada pelo SGFE;
- *Webpage*: representa a página ou conjunto de páginas geradas pelo SGFE.
- Servidor: onde as páginas e *scripts* são instalados para prover o acesso as mesmas.
- SGBD: sistema gerenciador de banco de dados, é usado pelo sistema o MySQL, como já apresentado.
- BD: Banco de dados.

3.2.2. Requisitos funcionais

Os requisitos funcionais para um sistema descrevem a funcionalidade ou os serviços que se espera que o sistema forneça. Eles dependem do tipo de

software que está sendo desenvolvido, dos usuários de *software* que se espera verificar e do tipo de sistema que está sendo desenvolvido.

Esses requisitos funcionais de usuário definem recursos específicos que devem ser fornecidos pelo sistema. SOMMERVILLE

Os requisitos funcionais do SGFE estão relacionados diretamente a geração de páginas e *scripts*, como é mostrado a seguir:

- O usuário deve construir uma página (formulário de entrada de dados) e os *scripts* a partir de características definidas pelo usuário. Todas as características (conteúdo, desenho e saídas) devem ser decididas pelo usuário usando a interface do programa SGFE,
- O sistema deve gerar *scripts* sem requerer que o usuário conheça a linguagem usada para a criação desses *scripts*,
- Para cada questão contida no projeto deve existir um *script* que recupere e manipule os dados do banco de dados, apresentado como saída um conjunto de relatórios ao usuário,
- A inserção de dados no banco de dados (servidor) deve ser realizada por um *script* que deverá ser criado pelo SGFE,
- O sistema deve possuir uma interface gráfica simples, de forma a facilitar o uso do *software* pelo usuário.

3.2.3. Requisitos não funcionais

Os requisitos não funcionais, como o nome sugere, são aqueles que não dizem respeito diretamente às funções específicas fornecidas pelo sistema. Eles

podem estar relacionados a propriedade de sistemas emergentes, como confiabilidade, tempo de resposta e espaço em disco. SOMMERVILLE

Requisitos não funcionais do SGFE:

- O SGFE deverá fazer a geração dos *scripts* usando a linguagem PHP, compatível com a versão mais atual do interpretados.
- A partir dos campos existentes no formulário (questões) o SGFE fará a modelagem do banco de dados e inicializado dos dados do formulário (conteúdo).
- O SGFE não armazenará informações no arquivo de projeto relacionados a dados sigilos, como senha do servidor de banco de dados. Esses dados deverão ser inseridos pelo usuário no momento de criação dos códigos.
- Todos códigos gerados deverão ser endentados e comentados, facilitando seu uso durante todo ciclo de vida do SGFE.
- Gravar os arquivos diretamente na pasta pública do servidor, facilitando o manuseio dos arquivos.

3.2.4. Diagramas de *use case*

O diagrama “*Use case*” é um diagrama usado para se identificar como o sistema o sistema se comporta em várias situações que podem ocorrer durante sua operação Esmín.

O diagrama a apresentado na Figura 6 representa o comportamento dos atores: internauta, webpage e servidor.

As páginas (autor webpage) devem ser instaladas na pasta compartilhada na *web* pelo servidor, que por sua vez libera as páginas para que sejam acessadas pelo internauta. Deve-se enfatizar que o procedimento de instalação

das páginas e suas respectivas configurações devem ser realizadas pelo usuário do sistema. Para testes, usando um servidor local, o SGFE pode fazer a geração dos códigos diretamente na pasta compartilhada pelo servidor, fazendo a instalação da Webpage automaticamente.

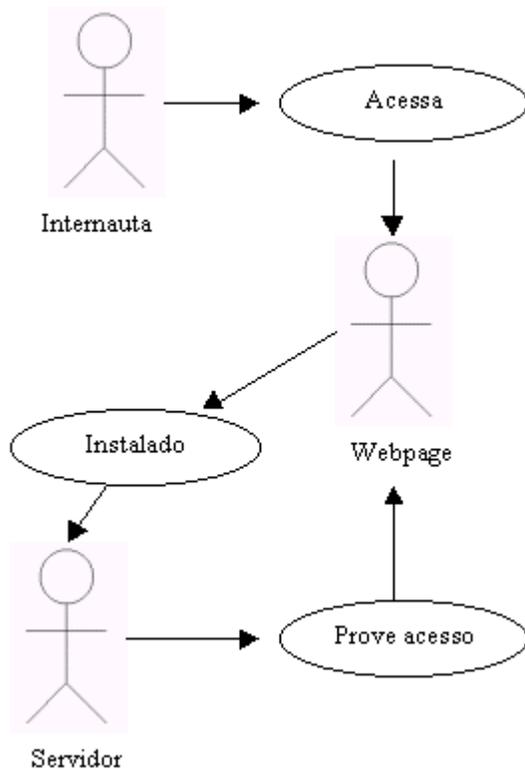


Figura 6 – Diagrama de use case. Fonte:

Elaborado pelo autor.

O diagrama apresentado pela Figura 7 mostra o diagrama de use case dos atores usuário, SGFE, webpage e servidor.

Esse diagrama representa o processo de criação das páginas. O usuário defini o conteúdo e o desenho das páginas do projeto, o SGFE a partir desses

dados faz a geração dos códigos e os instala no servidor, o qual permite que o usuário acesse as páginas e desenvolva seu projeto simultaneamente.

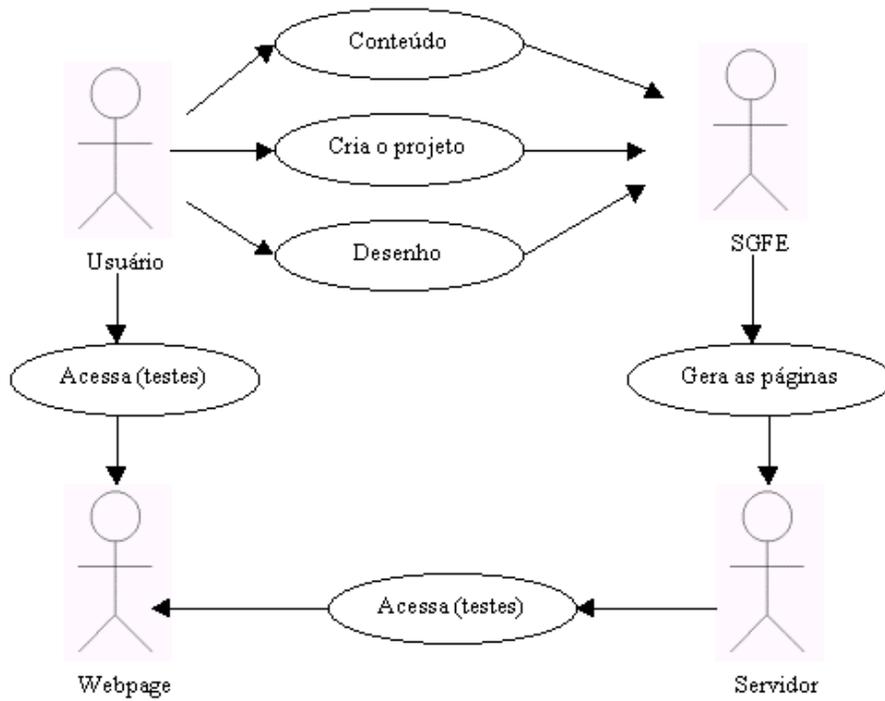


Figura 7 – Diagrama de caso de uso.

Fonte: Elaborado pelo autor.

4. Implementação

Todo processo de implementação do software foi feito a partir de um modelo. Nesse capítulo é apresentada a modelagem e a implementação desse software, visto que a implementação é consequência de uma modelagem.

4.1.Classes

A classe é uma descrição de um grupo de objetos com atributos, relacionamentos com outros objetos e semântica comum. Uma classe é uma abstração que enfatiza características relevantes dos objetos, suprimindo outras características relevantes dos objetos, suprimindo outras características. ESMIN

4.1.1. Classe página

A classe página contém atributos relacionados as características visuais das páginas geradas, o gráfico a seguir (Figura 8) mostra os atributos e operações dessa classe.

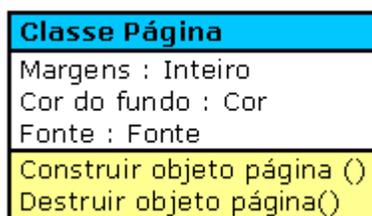


Figura 8 – Classe pagina.

Fonte: Elaborada pelo autor.

Os atributos definem as características visuais das páginas como as margens da página, tipo da fonte usada nos textos contidos nas páginas, e cor do

fundo da página. Esses atributos são usados para definir as características de todas páginas geradas (incluindo a página resposta dos *script*).

As operações executadas nessa classe se restringem a apenas construtores e destrutores de objeto, os quais podem ser usados em outras classes.

4.1.2. Classe formulário

Essa classe contém atributos que definem as características visuais do formulário, possui ainda um atributo com o texto que será usado como um cabeçalho para o formulário, esse campo é facultativo. O atributo rótulo dos botões é uma configuração feita pelo usuário que defini o texto contido nos botões de ação do formulário.

O gráfico a seguir (Figura 9) mostra os atributos e operações dessa classe.

Classe formulário
Alinhamento : Inteiro
Cor do fundo : Cor
Cabeçalho : Texto
Rótulo dos botões : Texto
Construir objeto formulário()
Destruir objeto formulário ()

Figura 9– Classe formulário. Fonte: Elaborada pelo autor.

As operações executadas nessa classe se restringem a apenas construtores e destrutores de objeto, os quais podem ser usados em outras classes.

4.1.3. Classe questionário

A classe questionário é um conjunto de objetos da classe questão.

A classe questão contém o atributos relacionados aos campos inseridos no formulário de pesquisa, cada questão se relaciona ao conteúdo de cada questão contida no formulário como pode ser observado na Figura 10.

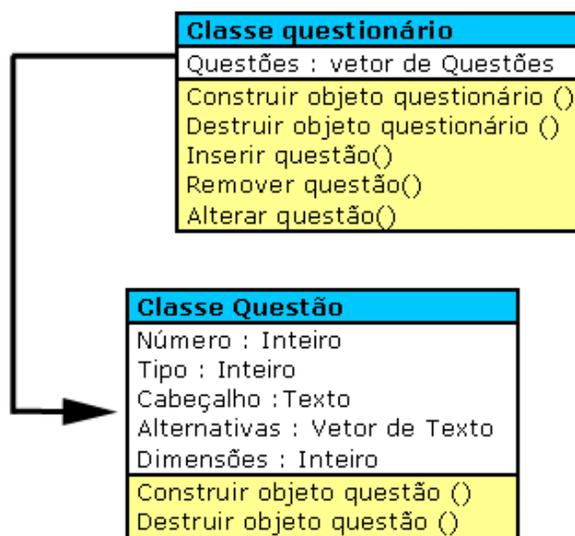


Figura 10 – Classe questionário e classe questão. Fonte:

Elaborado pelo autor.

A classe questão representa cada questão inserida no formulário, o atributo tipo que um valor inteiro que defini qual o tipo de questão (qual componente será usado nessa questão), o atributo cabeçalho é um texto que é apresentado antes das alternativas de cada questão, assim como as alternativas é um conjunto de textos (vetor) que correspondem a cada resposta da questão, isto

se o tipo da questão for de escolha (componente de rádio, verificação e seleção). Em casos em que o tipo da questão é de entrada de dados (componentes de edição de texto) o atributo dimensões representa o tamanho do componente (número de linhas e colunas).

As operações executadas na classe questão se restringem a apenas construtores e destrutores de objeto, os quais são usados na classe questionaria que além de possuir operações de construção e destruição de objetos possui operações que envolvem inserir, remover e alterar alguma questão do vetor.

4.1.4. Classe saída

A classe página contém atributos relacionados as características visuais das páginas geradas, o gráfico a seguir (figura 11) mostra os atributos e funções dessa classe.

Classe saída
Cor relatório gráfico : Cor
Construir objeto saída ()
Destruir objeto saída ()

Figura 11 – Classe saída. Fonte: Elaborada pelo autor.

O único atributo dessa classe é usado na criação do *script* que é usado na criação de um gráfico estatístico a partir de uma questão de escolha.

As operações executadas nessa classe se restringem a apenas construtores e destrutores de objeto, os quais podem ser usados em outras classes.

4.1.5. Classe configurações

A classe configurações contém atributos relacionados as configurações do servidor de banco MySQL, o gráfico a seguir (figura12) mostra os atributos e funções dessa classe.

Classe Configurações
Nome : Texto
Senha : Texto
Endereço : Texto
Construir objeto configurações ()
Destruir objeto configurações ()

Figura 12 – Classe configurações. Fonte: Elaborada pelo autor.

Os atributos dessa classe estão relacionados aos requisitos do banco de dados MySQL. Esses atributos são usados na conexão com o banco de dados, e são inseridos nos códigos como variáveis de um *script*.

As operações executadas nessa classe se restringem a apenas construtores e destrutores de objeto, os quais podem ser usados em outras classes.

4.1.6. Classe geradora

Essa classe contém apenas operações que são realizadas usando os objetos das classes em que se relaciona, essas operações são responsáveis pela geração dos códigos e das páginas. A Figura 13 mostra o diagrama dessa classe.

Classe Geradora
Objeto: projeto
Objeto: Configurações
Gerar formulário instalar banco de dados()
Gerar formulário de questões()
Gerar script executar pesquisa()
Gerar script relatório()
Gerar script gráfico()

Figura 13 – Classe geradora. Fonte: Elaborada pelo autor.

A partir de um objeto da classe projeto e da classe configurações a classe geradora executa as seguintes operações de geração de códigos: gerar formulário que faz a instalação das tabelas, gera o formulário com as questões, o *script* que executa esse formulário e os *scripts* de saída gráfica e relatório.

4.1.7. Classe projeto

A classe página contém atributos relacionados ao conjunto de informações que podem ser manipuladas, de forma a permitir que determinadas informações sobre o conteúdo e desenho das páginas possam ser salvas e recuperadas a partir de um arquivo.

Essa classe se relaciona com as classes: página, formulário, questionário e saídas, guardando e reparando os dados em arquivo, além disso essa classe é relacionada com a classe geradora que usando os objetos da classe projeto faz a geração dos códigos. Essa relação é mais bem apresentada no diagrama de classes no próximo tópico.

A Figura 14 a seguir representa a classe projeto.

Classe Projeto
Objeto : Página Objeto : Formulário Objeto : Questionário
Construir objeto configurações () Destruir objeto configurações () Salvar projeto() Carregar projeto()

Figura 14– Classe projeto. Fonte:

Elaborada pelo autor.

As operações executadas são: construtores e destrutores de objeto, salvar os atributos em um arquivo e carregar os atributos dessa classe a partir de um arquivo.

4.2. Diagrama de classe

O diagrama de classes apresentado na Figura 15 mostra as relações entre todas as classes do SGFE.

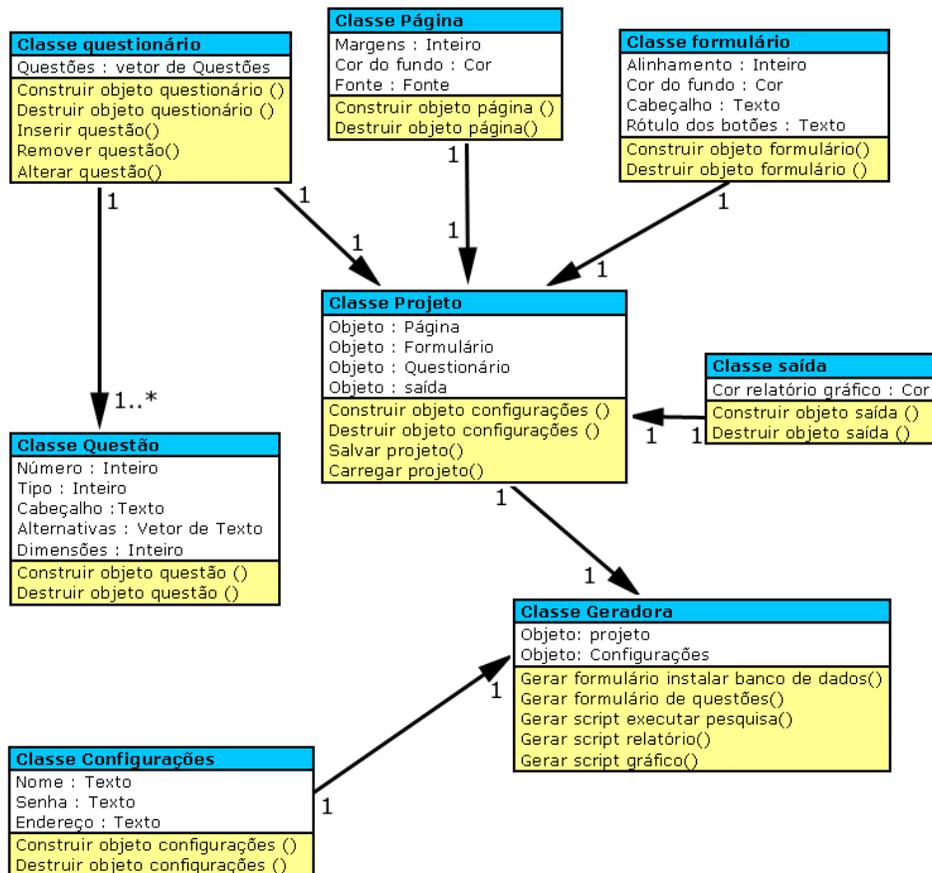


Figura 15 – Diagrama de classe. Fonte: Elaborada pelo autor.

4.3. Operações implementadas

Serão apresentadas nesse tópico as principais operações que o SGFE realiza para conseguir gerar os códigos em PHP e HTML, não serão mostrados os códigos fonte da implementação, visto que este é apresentado em anexo a essa monografia.

4.3.1. Operações de criação do banco de dados

Para se apresentar essas operações é necessário verificar o seguinte:

- O banco de dados deverá guardar as informações do formulário, que são os textos que representam as alternativas e o cabeçalho, de modo que ao se criar um *script* resposta, este possa usar esses dados na criação da resposta, por exemplo, num *script* relatório, o relatório deve conter o cabeçalho das questões e em seguida a resposta do internauta,
- Questões que envolvam entrada de informações (campo de edição de texto) necessitam um campo na tabela de dados para guardar essas informações, já questões de escolha (botões de radio) necessitam apenas guardar a escolha efetuada. Como para ambos os casos se trata de dados do tipo texto, então foi criado um campo que recebe ambas informações, ou seja, não houve uma separação entre os dois tipos de entrada,
- Quando a questão se tratar de campos de botões de verificação, o usuário deve escolher várias ou nenhuma alternativa, portanto, cada uma dessas escolhas deve ser guardado na tabela, e é adotado aqui que entradas em branco (não escolha) não são armazenados na tabela,
- É interessante guardar a data em que o internauta acessou o formulário.

A partir dessas notas foi elaborado um seguinte esquema de banco de dados, que pode ser dito universal, ou seja, independente do tipo de questão,

portanto qualquer esquema de formulário proposto poderá usar esse esquema para receber os dados. Esse esquema é apresentado na Figura 16.

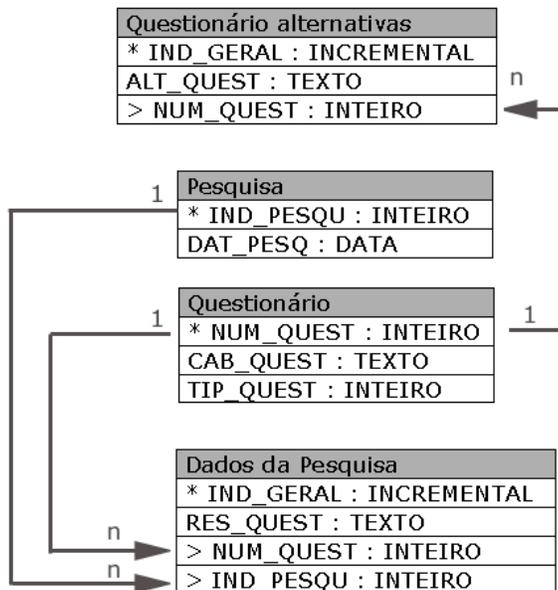


Figura 16 – Esquema das tabelas de banco de dados.

Fonte: Elaborada pelo autor.

Pelo esquema apresentado nota-se que a tabela de dados da pesquisa conterá todas respostas dos internautas (guardadas no campo RES_QUESTION), separadas pelo índice da pesquisa, pois cada índice representa uma dada pesquisa. E as questões são separadas por um número que a representa. Desse modo, independente do tipo de questão a resposta do internauta é armazenado e organizado nesse esquema.

A partir das questões inseridas no formulário, durante o processo de criação deste no software, essa base de dados é inicializada, ou seja, as alternativas e cabeçalhos são alocados.

O código SQL usado pelo MySQL é um código que requer bastante estudo e prática, e desenvolver um código desse nível requer um alto grau de conhecimento de SQL. Como a instalação do Banco de Dados e sua manipulação são requisitos do sistema, o SGFE cria um script que faz essa instalação e inicialização.

O código SQL a seguir apresenta o código comum a todos projetos, esse código resulta na criação de um conjunto de tabelas de banco de dados.

```
DROP TABLE PESQ_DATA;

CREATE TABLE QUESTIONARIO (NUM_QUEST INT NOT NULL,
                            CAB_QUEST BLOB,
                            TIP_QUEST INT NOT NULL,
                            PRIMARY KEY(NUM_QUEST)) TYPE=INNODB;

CREATE TABLE QUEST_ALT (IND_GERAL INT NOT NULL AUTO_INCREMENT,
                        ALT_QUEST CHAR(255),
                        NUM_QUEST INT NOT NULL,
                        PRIMARY KEY(IND_GERAL),
                        INDEX (NUM_QUEST),
                        FOREIGN KEY (NUM_QUEST)
                        REFERENCES QUESTIONARIO(NUM_QUEST)
                        ON UPDATE CASCADE ON DELETE CASCADE
                        ) TYPE=INNODB;

CREATE TABLE PESQUISA (IND_PESQU INT NOT NULL,
                       DAT_PESQU DATE,
                       PRIMARY KEY(IND_PESQU)
                       ) TYPE=INNODB;

CREATE TABLE PESQ_DATA (IND_GERAL INT NOT NULL AUTO_INCREMENT,
                        RES_QUEST CHAR(255),
                        NUM_QUEST INT NOT NULL,
                        IND_PESQU INT NOT NULL,
                        PRIMARY KEY(IND_GERAL),
                        INDEX (NUM_QUEST),
                        FOREIGN KEY (NUM_QUEST) REFERENCES
                        QUESTIONARIO(NUM_QUEST)
                        ON UPDATE CASCADE ON DELETE CASCADE,
                        INDEX (IND_PESQU),
                        FOREIGN KEY (IND_PESQU) REFERENCES
                        PESQUISA(IND_PESQU)
                        ON UPDATE CASCADE ON DELETE CASCADE
                        ) TYPE=INNODB;
```

Mesmo com o código pronto muitos usuários ainda não conseguem criar a base de dados em um servidor MySQL, para evitar que o usuário tenha que fazer isso, o SGFE gera um *script* que deve ser usado para executar o código SQL apresentado.

Como se trata de um *script*, este deve possuir variáveis que permita que o mesmo execute essas operações, dado que essas operações somente são realizadas por um usuário do MySQL que tenha esse privilégio, essas variáveis devem ser configuradas pelo usuário no SGFE, e recebem valores como senha, nome do usuário e endereço do servidor (*host*).

Depois de criado o banco de dados, as questões (cabeçalho e alternativas) devem ser iniciadas, para se fazer isso o script executa diversos códigos SQL que inserem essas informações no banco de dados.

Deve se notar que toda vez que um dado projeto é alterado, a instalação da base de dados deve ser executada, de forma a garantir a integridade do modelo apresentado.

A operação de gerar o formulário de instalação do banco de dados requer objetos das classes página e formulário, a partir desses objetos é criado uma página HTML (“instalar.htm”) que contém as áreas nas quais o usuário deve entrar com as variáveis de acesso ao MySQL, nesse ponto não são usados as variáveis da classe configuração para evitar problemas de segurança, pois algum internauta poderia descobrir um *script* que apaga a base de dados e executa-lo, por isso o usuário deverá entrar sempre com essas variáveis antes de se fazer a execução do script (“instalar.php”).

4.3.2. Geração do formulário de entrada de dados

Essa operação da classe geradora requer objetos da classe página e formulário, pois a partir desses o código HTML do formulário e o *script* que insere os dados no formulário são gerados.

O *script* deve possuir funções de validação, que não permitem que o internauta entre com campos em branco (com exceção do tipo de questão de caixa de verificação).

4.3.3. Geração dos *scripts* de saída

Essa operação da classe geradora requer objetos da classe página e formulário, pois a partir desses o código PHP dos *scripts* são gerados. O SGFE cria dois *scripts*:

- Relatório, esse *script* pesquisa a base de dados e devolve uma página contendo todas as respostas dos internautas, agrupadas em campos na página. Para cada conjunto de respostas é retornada a data em que esses dados foram inseridos.
- Gráfico, esse tipo de *script* retorna um gráfico representativo das questões do tipo escolha, calculando a quantidade de escolhas de cada alternativa (porcentagem) para cada questão. Para cada questão de escolha existe um único gráfico.

5. Resultados

A implementação resultou em um software capaz de criar diversos tipos de formulários *web*, bem como a implementação automática de *scripts* relacionados à manipulação dos dados, sendo que durante o processo de criação dos códigos o usuário não teve que editar ou inserir linhas de código do *script*, nem mesmo modelar o banco de dados e criar as tabelas, pois essas funções são requisitos do sistema, como já foi apresentado.

Como todo processo é automatizado, o tempo de desenvolvimento de uma página com formulário se restringi à apenas decidir os campos que serão colocados no formulário e o conteúdo desses campos, além disso toda interface das páginas (desenho) são totalmente configuráveis, ou seja não existe um padrão para a criação dos formulários, que podem possuir o conteúdo e desenho que o usuário requerer.

5.1. Interface do SGFE

A Figura 17 apresenta a interface do software, observa-se que todos componentes contidos na interface são padrão *Windows*, portanto qualquer usuário do sistema operacional *Windows* não terá grandes dificuldades de se usar o software.

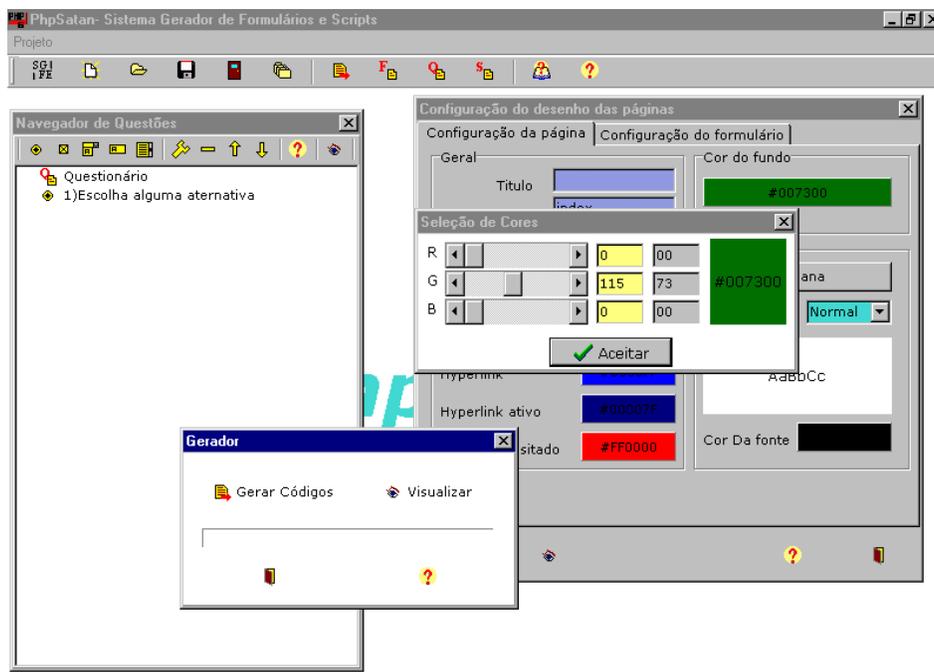


Figura 17 – Janela do SGFE.

Fonte : Janela do software implementado.

A barra de menu principal contém botões que devem ser usados para manipular os projetos, como criar um novo projeto, salvar e abrir um projeto existente. Nessa mesma barra estão dispostos botões que acessam as demais janelas de configuração do projeto.

A partir de janelas distintas o usuário configura os objetos do projeto, a partir dos quais os códigos são criados pela classe geradora. A interface compõe-se das seguintes janelas:

- Janela de configuração do desenho das páginas e formulário;
- janela de configuração e navegação das questões contidas no formulário;

- janela de configuração dos *scripts* de saída;
- janela de geração dos códigos, nessa janela o usuário faz a criação de todos os códigos a partir de suas decisões.

5.1.1. Janela de configuração do desenho das páginas e formulário

Nessa janela o usuário do software decide as características das páginas que serão criadas pelo SGFE, essa janela permite a configuração do desenho das páginas, observa-se na Figura 18 que todas as decisões são feitas usando componentes padrões do windows, como caixas de escolha, campo de entrada de dados, a única característica diferente é uma janela criada para manipular cores, que permite a visualização das cores no formato usado pela *web* (hexadecimal), que é um padrão diferente de cores adotado pelo *Delphi*.

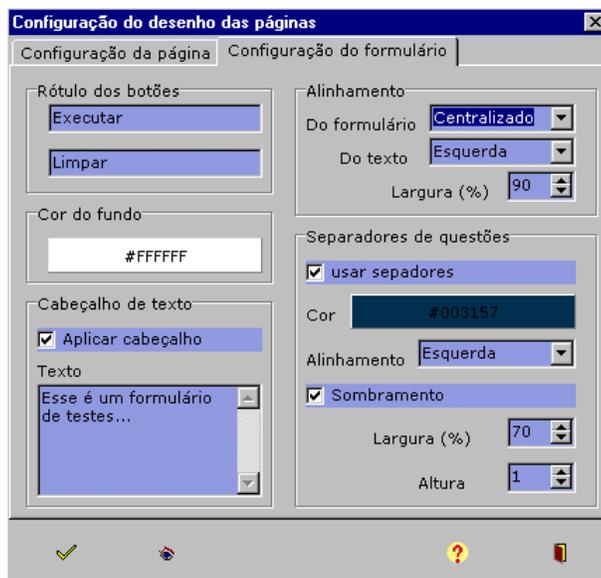


Figura 18 – Janela de configuração das páginas e formulários.

Fonte : Janela do software implementado.

5.1.2. Janela de configuração e navegação das questões contidas no formulário

Essa é janela (figura 19) que permite que o usuário crie e manipule as questões que compõe o formulário. As questões são disposta em uma lista e identificadas com um ícone, o qual permite que o usuário saiba qual é o tipo da questão. O texto após o ícone é o cabeçalho da questão. Clicando-se duas vezes sobre um objeto da lista é aberta uma janela que permite que se visualize e altere os dados da questão que foi selecionada pelo duplo clique.

A barra de ferramentas dessa janela contém botões identificados com ícones permitem a criação, alteração e remoção das questões na lista, bem como também alterar seu posicionamento em relação a lista.

O botão com um ícone em formato de olho permite visualizar o formulário com as questões contidas nele em tempo real. Essa saída é visualizada no navegador padrão do sistema operacional, e é um arquivo temporário, criado apenas para testes.

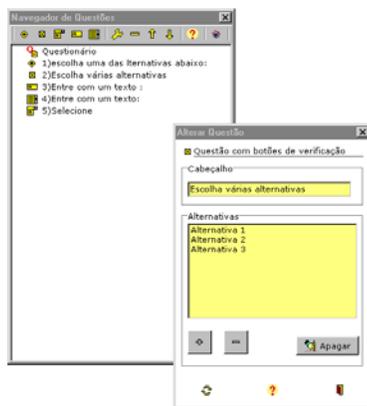


Figura 19 – Janela de navegação das questões.

Fonte : Janela do software implementado

5.1.3. Janela de configuração dos *scripts* de saída

Nessa janela os dados dos *scripts* são configurados pelo usuário, nessa janela é possível definir a cor das barras do relatório gráfico, e algumas características dos relatórios.

Os campos senha e nome do usuário devem ser configurados toda vez que um projeto é aberto, pois são informações sigilosas, e não podem ser salvas pelo projeto de forma a evitar que pessoas não autorizadas possam ter acesso a essas informações.

A figura 20 mostra a janela de configuração do software:

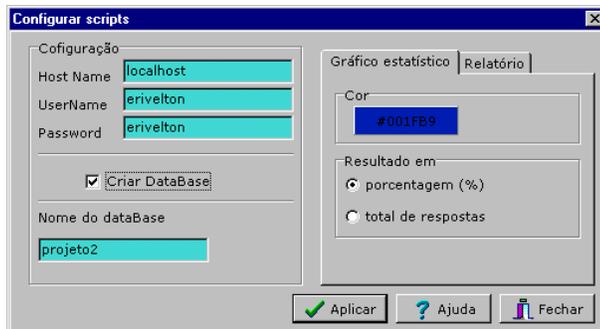


Figura 20 – Janela de configuração dos *scripts* de saída. Fonte : SGFE

5.1.4. Janela de geração dos códigos

Quando o usuário decide gerar os códigos ele deve clicar no botão com a legenda gerar código, contido na janela de geração de códigos, mostrado na Figura 21.

Essa operação gera automaticamente todas páginas saídas.

O Botão visualizar (com ícone em formato de olho) permite que se abra a página gerada usando o navegador padrão do sistema operacional, essa visualização requer que o servidor esteja iniciado e que os códigos tenham sido gerados na pasta onde as páginas são disponibilizadas pelo servidor.



Figura 21 – Janela de geração dos códigos.

Fonte : Janela do software implementado.

Quando se clica no botão gera código, o SGFE faz a geração dos códigos no caminho definido na janela de configuração do software. Esse caminho deve ser definido pelo usuário do software, e pode ser, por exemplo, o caminho em que se encontra a pasta do servidor, na qual os documentos *web* são disponibilizados.

5.2. Exemplos de saídas do software

Como todo formulário *web* segue um padrão em termos de componentes, é possível através desses desenvolver formulários para diversos fins, como por exemplo, uma enquête, um fórum ou um formulário de cadastramento de dados dos visitantes da página, a única diferença entre esses exemplos são os *scripts*, cada tipo de formulário possui um script personalizado para sua função, porém a interface (componentes) é a mesma.

O SGFE gera 4 scripts de saída:

- Modela e instala o Banco de dados;
- Inserir os dados que o internauta entrou no formulário no banco de dados;
- Devolve um gráfico a partir dos campos do formulário em que o internauta deve escolher alternativas;
- Retorna um relatório com todas respostas que o internauta inseriu no formulário.

A partir desses scripts é possível usar o SGFE na geração de diversos tipos de formulários, como exemplo será citado uma enquête (pesquisa) e um formulário de cadastramento de currículos.

5.2.1. Uma simples enquête

Esse projeto requer que seja inserido uma questão do tipo botão de rádio no navegador, para se fazer isso basta clicar no ícone adicionar questão de rádio, e definir o cabeçalho e as alternativas, como é apresentado na figura 22, feito isso deve-se configurar os *scripts* usando a janela configurar *scripts* (Figura 21) e configurar a interface das páginas usando a janela configurar páginas e formulários (Figura 19). Com o projeto pronto clique no botão gerar código (Figura 22) e está criado o formulário de enquête.

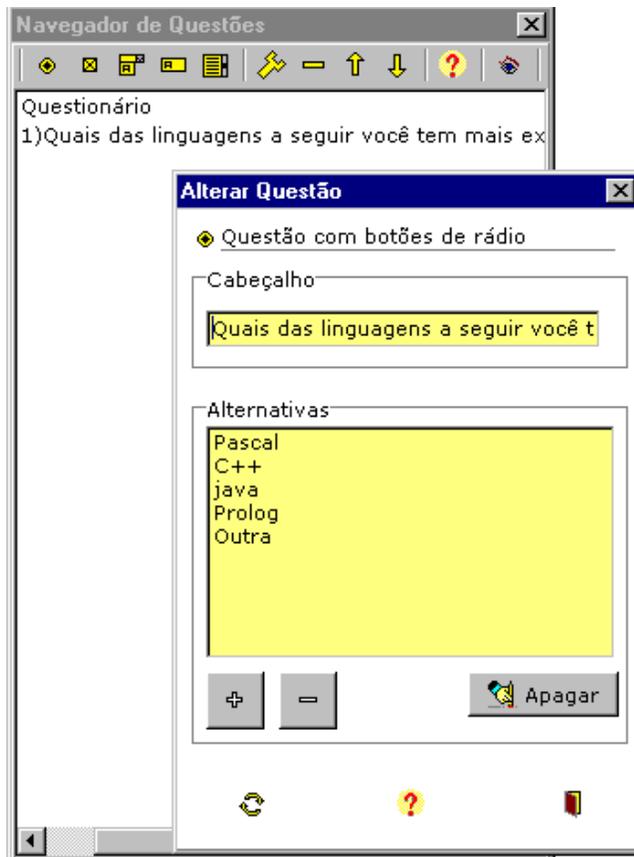


Figura 22 – Criando uma enquête.

Fonte : Janela do software implementado.

Antes de rodar o formulário deve-se instalar o banco de dados que será usado para guardar as informações que serão inseridas no formulário, basta acessar a página “instalar.htm” a partir de seu servidor (o servidor e o MySQL deve estar em execução). Será apresentado um formulário mostrado na Figura 33.

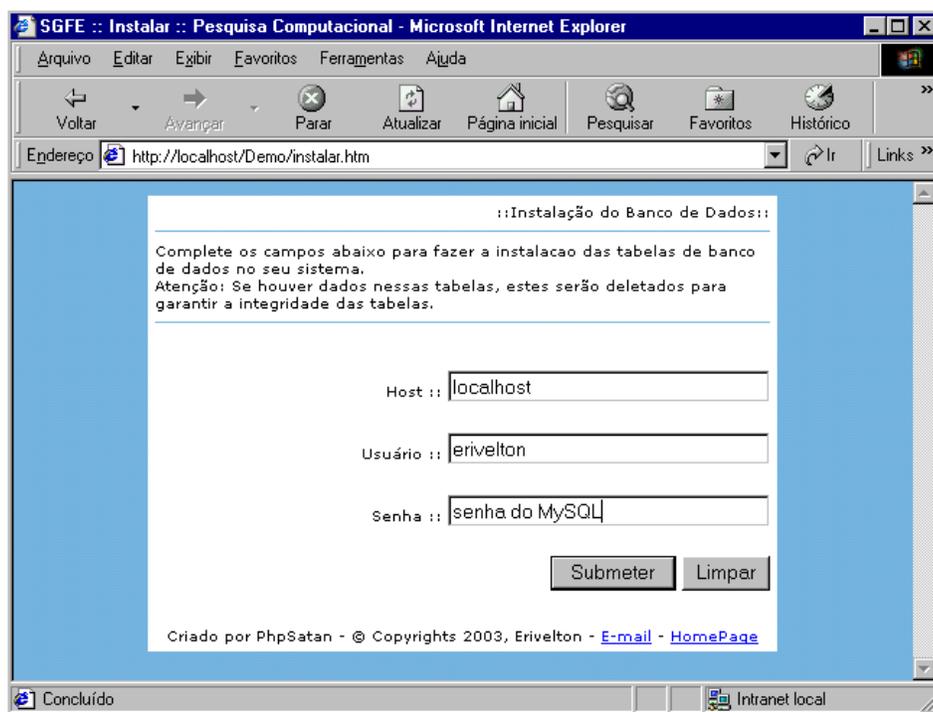


Figura 23 – Formulário de instalação da base de dados.

Fonte : Página web saída do software.

Apos inseridos os dados que permitem acesso ao MySQL deve-se clicar no botão submeter do formulário, para que um *script* (“instalar.php”), crie e inicie as tabelas de dados. Se a instalação for concluída com sucesso, será apresentado a página mostrada na Figura 24 caso contrário será apresentado uma página mostrando os erros que podem vir à acontecer ao se tentar acessar o MySQL.

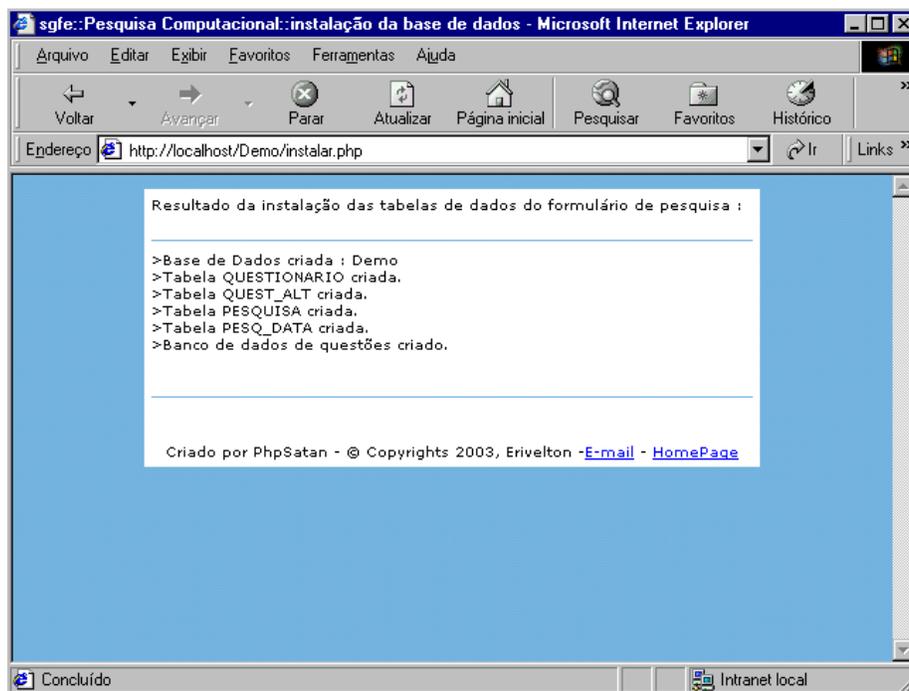


Figura 24 – Formulário de confirmação da instalação da base de dados. Fonte : *Script* criado pelo SGFE.

Nota-se que o desenho de ambas paginas apresentadas (Figuras 23 e 24) possuem o mesmo desenho, que é o mesmo do formulário de entrada de dados, como mostrado na Figura 25. Esse *script* retorna uma página confirmando a criação automática das tabelas de dados no MySQL.

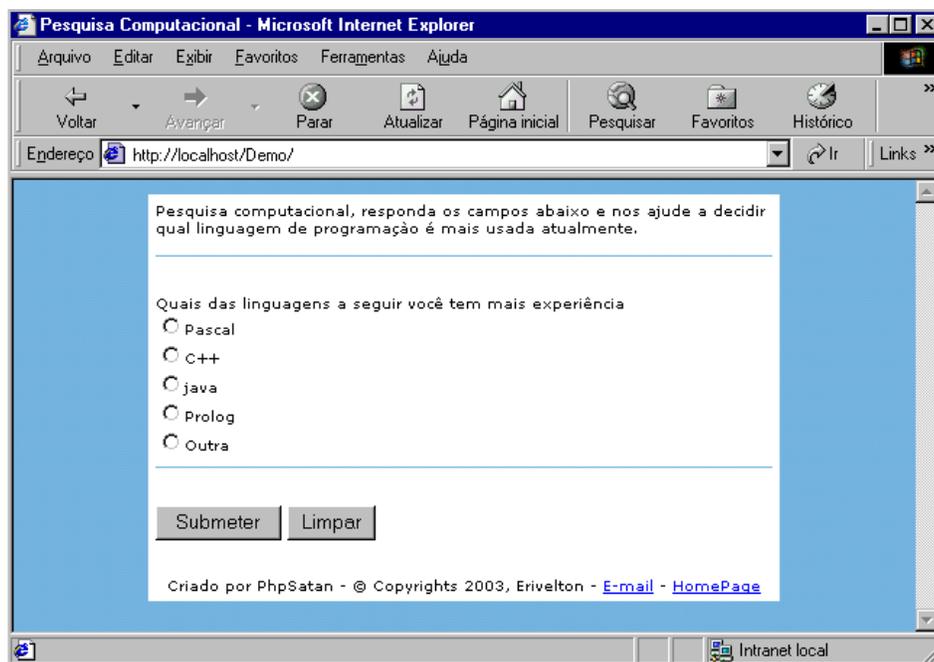


Figura 25 – Enquete.

Fonte : Página web saída do software.

Instalado a base de dados no MySQL, o formulário principal (enquete) está pronta para ser colocada on-line para que os internautas possam responde-la. Quando o internauta escolhe e clica no botão com rótulo submeter, os dados são passado para um script que os guarda no banco de dados e retorna uma página resposta que é mostrada na Figura 26.

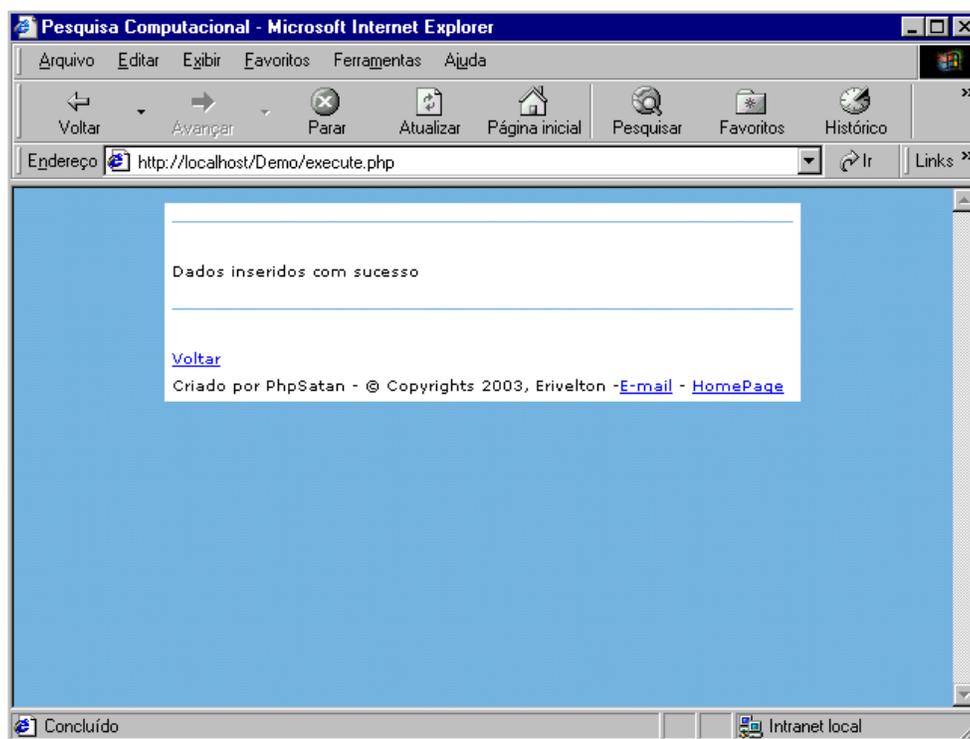


Figura 26 – Página resposta.

Fonte : Página web saída do software.

Após um tempo disponibilizado formulário na *web*, nossa enquete recebe vários visitantes e nossa base de dados cresce, para se acessar os dados da enquete chama-se o script “grafico.php” que retorna um gráfico, mostrando em porcentagem a quantidade de escolhas efetuadas pelos internautas para cada alternativa, a Figura 27 mostra um resultado parcial de nossa enquete.

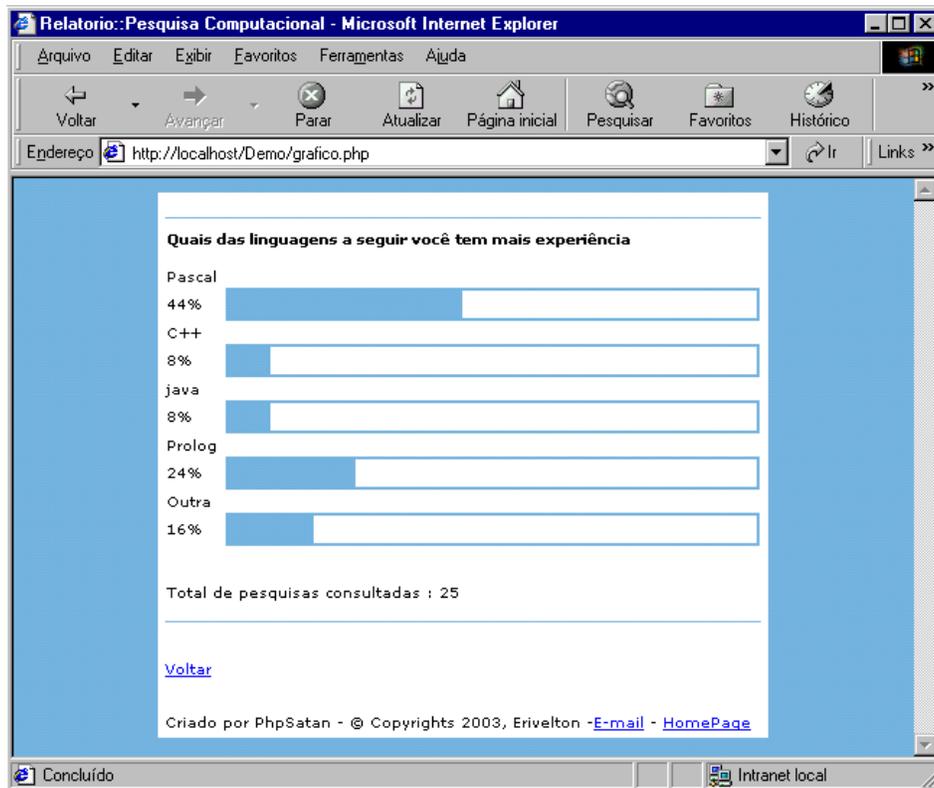


Figura 27 – gráfico estatístico da enquête.

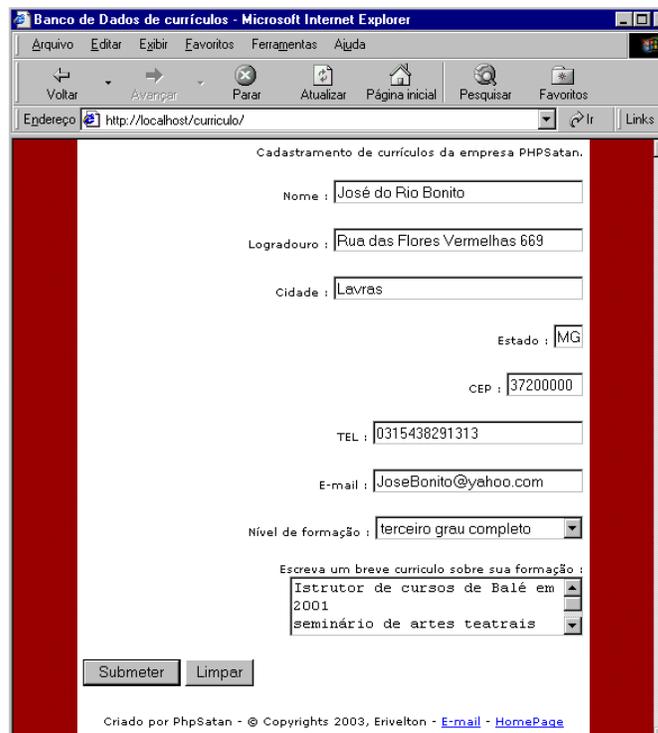
Fonte: Página web saída do software.

O script responsável pela resposta gráfica retorna o cabeçalho de todas questões e suas alternativas, mostrando a porcentagem de escolhas realizadas em cada uma dessas, e uma barra, criada usando intercalação de tabelas (HTML), esse gráfico não requer uso de imagens, pois a barra é constituída apenas de código HTML.

5.2.2. Um formulário de cadastro de currículos

Esse segundo exemplo apresenta um formulário no qual o internauta deve preenche-lo de forma a cadastrar um conjunto de dados, nesse exemplo esse conjunto se refere a um simples currículo, no qual o internauta cadastra seus dados de contato e deixa seus dados profissionais.

A Figura 28 mostra o formulário criado usando o SGFE. Observa-se que todos campos desse formulário são componentes de entrada de dados, com exceção do campo nível de formação que é uma lista de seleção.



The image shows a screenshot of a Microsoft Internet Explorer browser window. The title bar reads "Banco de Dados de currículos - Microsoft Internet Explorer". The address bar shows "http://localhost/curricula/". The main content area displays a registration form titled "Cadastramento de currículos da empresa PHPSatan." The form contains the following fields and values:

- Nome : José do Rio Bonito
- Logradouro : Rua das Flores Vermelhas 669
- Cidade : Lavras
- Estado : MG
- CEP : 37200000
- TEL : 0315438291313
- E-mail : JoseBonito@yahoo.com
- Nível de formação : terceiro grau completo (dropdown menu)
- Escreva um breve currículo sobre sua formação :
 - Istrutor de cursos de Balé em 2001
 - seminário de artes teatrais

At the bottom of the form are two buttons: "Submeter" and "Limpar". At the very bottom of the page, there is a footer: "Criado por PhpSatan - © Copyrights 2003, Erivelton - E-mail - HomePage".

Figura 28 – Formulário de cadastro de informações.

Fonte : Página web saída do software.

O *script* “relatorio.php” retorna uma página contendo as informações que foram inseridas pelo usuário, essas informações são dispostas em um formato de relatório, esses dados são separados usando linhas horizontais. A Figura 28 mostra um relatório gerado a partir do formulário de cadastro de informações (Figura 28).



Figura 29 – Janela de relatório.

Fonte : Página web saída do software.

Nota-se que o desenho da página relatório possui as mesmas características do formulário, portanto esse *script* gerado pelo SGFE assim como todos outros seguem as características definidas pelo usuário, eliminado a

possibilidade do usuário ter que editar o código para ajustar propriedades de interface.

5.3. Melhorias do software

Várias melhorias poderiam tornar esse software uma verdadeira ferramenta de desenvolvimento *web*, no entanto as funções aqui já implementadas, já auxiliam o desenvolvimento de páginas sem requerer programação. Mas alguns detalhes a mais no software poderiam torna-lo um software comercial, esses detalhes estão descritos a seguir.

Permitir que os scripts guardassem as informações do formulário e das pesquisas não somente num sistema de banco de dados, mas também em arquivos binários, visto que muitos servidores não se utilizam do MySQL, e as funções da linguagem PHP possuem otimizações para se criar banco de dados em arquivos com qualidade, segurança e velocidade.

Dinamizar a quantidade de formulários em um projeto, essa versão implementada permite a criação de um conjunto de questões (campos) porém contidas em um único formulário.

Desenvolver novas saídas, o SGFE poderá gerar além dos *scripts* relatório e gráfico, outros tipos de *scripts*, como por exemplo: pesquisa, fóruns, notícias, salas de bate-papo, entre outros, pois todos esses exemplos descritos usam os mesmos componentes padrões de formulário.

Adicionar ao próprio *software* mecanismos de visualização e de administração da base de dados em tempo real.

O SGFE é apenas o princípio de um futuro *software* capaz de criar diversos tipos diferentes de *websites*, desde uma formulário de pesquisa até um portal de informações, sem requerer implementação, pois todo código é criado

automaticamente a partir de objetos definidos pelo usuário via interface *windows*.

6. Conclusões

Como conclusão desse projeto foi implementado um software destinado ao desenvolvimento de *homepages*.

7. Bibliográfica

[LERDORF(1999)] - LERDORF, R. - PHP Manual, 1999 by PHP documentation group, 2003.

[RESENDE(2001)] - RESENDE, A. M. P. - Notas de aula Engenharia de Software, COM-114, 2001 - Universidade Federal de Lavras

[MySQL(2003)] - MySQL Manual - <http://www.mysql.com>, 2003, EUA.

[DELPHI(2003)] - DELPHI, Borland Webpage - <http://www.borland.com>, 2003

[PRESSMAN(1992)] - PRESSMAN, R. S. - Software engineering: a practitioner's approach, 1992, , Editora Makron Books.

[QUATRANI(200)] - QUATRANI, T - Visual modeling with Rational Rose 2000 and UML, 2000, Editora Campus.

[SOMMERVILLE(2002)] - SOMMERVILLE, I. - Engenharia de software , 2002, 6^o edição, Editora Saraiva.

[ESMIN] - ESMIN, A. A. A. - Modelando com UML - Unified Modeling Language - Revista InfoComp, DCC - Universidade Federal de Lavras, Ano 1 Volume 1, 1999.