

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**Módulo de análise estatística
de dados de resposta do Polvo Survey**

Paulo Gustavo Magella de Faria Quinan

Florianópolis – SC

2011/1

UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE INFORMÁTICA E
ESTATÍSTICA

CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Módulo de análise estatística
de dados de resposta do Polvo Survey

Paulo Gustavo Magella de Faria Quinan

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do
grau de Bacharel em Ciências da Computação

Florianópolis – SC

2011/1

Trabalho de conclusão de curso sob o título “*Módulo de análise estatística de dados de resposta do Polvo Survey*”, defendida por Paulo Gustavo Magella de Faria Quinan e aprovada em 18 de Julho de 2011, em Florianópolis, Santa Catarina, pela banca examinadora constituída por:

Prof. Dr. Julibio David Ardigo
Universidade do Estado de Santa Catarina
Orientador

Prof. Dr. Mário Antônio Ribeiro Dantas
Universidade Federal de Santa Catarina
Responsável

Prof. Dr. Júlio da Silva Dias
Universidade do Estado De Santa Catarina

Me. Fábio dos Santos
Superintendência de Governança Eletrônica e Tecnologia da Informação e
Comunicação – Universidade Federal de Santa Catarina

Sumário

Sumário.....	i
Índice de Figuras.....	iii
1 Introdução.....	1
1.1 Objetivos.....	2
1.1.1 Objetivo Geral.....	2
1.1.2 Objetivos Específicos.....	2
1.2 Justificativa.....	3
1.3 Metodologia.....	4
2 Referencial Teórico.....	6
2.1 Polvo.....	6
2.2 Zend Framework.....	10
2.3 Model View Controller.....	14
2.4 Polvo Survey.....	15
2.5 Línguaem R.....	19
2.5.1 Exemplos de uso da línguaem R.....	20
2.5.2 Motivação para a escolha de R.....	23
3 Trabalhos Relacionados.....	25
3.1 Google Docs – Form.....	26
3.2 LimeSurvey.....	27
3.3 Free Online Surveys.....	29

3.4 SurveyMonkey.....	30
3.5 QuestionPro.....	32
4 Desenvolvimento.....	35
4.1 Levantamento de Requisitos.....	35
4.2 Implementação de funções em R.....	39
4.3 Desenvolvimento do Módulo Estatístico.....	44
4.3.1 Método de Execução de Funções R.....	45
4.3.2 Funcionalidade de Análise.....	47
4.3.3 Funcionalidade de Geração de Relatórios.....	52
5 Considerações Finais.....	57
6 Referências.....	60

Índice de Figuras

Figura 1 Edição de Formulários no Polvo Survey.....	17
Figura 2 Diagrama de Dispersão em R.....	22
Figura 3 Gráfico Suavização com Bandas de Confidência com biblioteca GGPLOT em R.....	23
Figura 4 Relatório Google Docs – Form.....	27
Figura 5 Análise Quantitativa LimeSurvey.....	28
Figura 6 Análise Quantitativa Free Online Surveys.....	29
Figura 7 Gráfico Qualitativo Free Online Surveys.....	30
Figura 8 Análise Qualitativa SurveyMonkey.....	31
Figura 9 Análise Qualitativa QuestionPro.....	32
Figura 10 Relatório Correlação QuestionPro.....	33
Figura 11 Exemplo Gráfico Gerado.....	40
Figura 12 Gráfico com Junção de Categorias.....	43
Figura 13 Gráfico Correlação.....	44
Figura 14 Visão de Resultado de Análise Gerada.....	50
Figura 15 Visão Edição de Relatórios.....	54

1 Introdução

Pesquisas do tipo *survey* são muito utilizadas por cientistas sociais tanto na iniciativa pública quanto na privada onde se faz um recorte quantitativo do objeto de estudo através de instrumentos de coleta de dados, como questionários e entrevistas pessoais.

Nessas pesquisas requisitos de segurança como integridade, sigilo e não repúdio não são tratadas adequadamente comprometendo a confiabilidade dos estudos realizados. Somado a isso o acesso cada vez maior à internet por parte de todas as camadas da população, ferramentas de pesquisa online (onde é possível ter um maior controle nos requisitos de segurança listados anteriormente) são cada vez mais usadas.

Existem hoje diversas ferramentas online de *survey*. Algumas de código aberto como o LimeSurvey e o Polvo Survey, outras proprietárias, mas gratuitas, como o Google Docs – Form e ainda algumas proprietárias e pagas como o SurveyMonkey e o QuestionPro. Todas essas ferramentas facilitam a criação e aplicação de *surveys* e provêm melhoras na segurança e confiabilidade das pesquisas.

Porém, depois da aplicação dos questionários o cientista social se vê com dificuldades de entender e analisar os dados obtidos. Os dados são mostrados em relatórios pouco úteis para estudos científicos. Fora isso podem ser exportados em tabelas de onde é difícil extrair análises válidas sem ferramentas adequadas.

Geralmente tais ferramentas possuem uma curva de aprendizado alta, o que pode dificultar, ou até mesmo impedir, a análise dos dados por parte dos pesquisados, que possuem uma formação em áreas humanas onde não há ênfase no aprendizado do uso de ferramentas de cálculos e análises estatísticas.

1.1 Objetivos

Os objetivos do presente projeto foram subdivididos em objetivo geral e objetivos específicos, como apresentados a seguir.

1.1.1 Objetivo Geral

Desenvolver uma ferramenta que facilite a análise estatística de respostas de *surveys* online e composição de relatórios de tais análises.

1.1.2 Objetivos Específicos

Para alcançar o objetivo geral é necessário que se atinja os seguintes objetivos específicos:

- Analisar ferramentas de *survey* online existentes com foco em suas funcionalidades de análise estatística;
- Definir as funcionalidades a serem implementadas;

- Definir as tecnologias a serem utilizadas no desenvolvimento da ferramenta;
- Desenvolver a solução;
- Validar a solução.

1.2 Justificativa

Existem muitas ferramentas online de aplicação de pesquisas tipo *survey*. Tais ferramentas facilitam a criação e aplicação de questionários em relação aos *surveys* em papel ou pelo telefone, além de fornecer alguns uma maior confiabilidade nas respostas em relação às outras mídias de aplicação de *surveys*.

Entretanto, as ferramentas não costumam oferecer suporte a análises estatísticas avançadas das respostas e tampouco permitir a composição de gráficos onde o usuário tenha total liberdade de escolha entre todos os parâmetros do gráfico. Quando uma ferramenta permite a criação de gráficos não é possível, pela própria ferramenta, a criação de relatórios com eles, sendo necessário o uso de softwares de edição de texto, gerando assim uma maior complexidade na criação de relatórios.

Assim, se faz necessário o uso de softwares estatísticos para a criação de análises avançadas. Tais softwares possuem uma curva de aprendizado alta o que, portanto, dificulta ainda mais sua pesquisa. Não existe uma integração

entre as ferramentas de *survey* e as de análise estatística, que simplifique a análise dos dados pelos cientistas sociais.

A necessidade de uso de tantas ferramentas exige do cientista social conhecimentos que não são o foco do seu trabalho, nem são os pontos fortes de sua formação acadêmica. Isso dificulta a sua concentração no que é o foco do seu trabalho, o estudo dos resultados das pesquisas.

Também é válido ressaltar que as ferramentas mais avançadas são proprietárias o que acarreta em dificuldade para diversas esferas que fazem uso delas. Pesquisadores acadêmicos incorrem em custos muitas vezes além da realidade de suas pesquisas. Empresas de pequeno porte não possuem orçamento suficiente para comprá-las. Por fim no poder público, atualmente, existe um direcionamento, para a contenção de custos, ao uso de softwares livres, o que impede o uso de ferramentas proprietárias.

Por isso, acredita-se que o sistema proposto possa facilitar a análise das respostas geradas por ferramentas online de pesquisa tipo *survey*, bem como a criação de relatórios envolvendo as análises desejadas por seus usuários. Permitindo assim que o cientista social possa se concentrar no estudo dos dados de sua pesquisa e não no aprendizado de diversas ferramentas.

1.3 Metodologia

O projeto iniciará com o levantamento dos requisitos funcionais e não funcionais do sistema. Tais requisitos serão levantados através de entrevistas com o público alvo do sistema, que são professores da UDESC, e de um

estudo comparativo do tratamento estatístico, criação de gráficos e geração de relatórios de algumas ferramentas online de pesquisas tipo *survey*.

Em seguida serão estudadas as tecnologias envolvidas no desenvolvimento do projeto, construindo-se o referencial teórico.

O sistema será desenvolvido sobre a plataforma Polvo 5, ambiente de ensino virtual desenvolvido pelo Laboratório de Tecnologia de Informação e Comunicação (LabTIC) da ESAG. O Polvo é totalmente modular, sendo assim, a ferramenta proposta será um novo módulo da plataforma. Além disso, o Polvo já possui um módulo de *survey* chamado de Polvo Survey com o qual o sistema a ser desenvolvido será integrado. Tal integração se dará a fim de demonstração do uso do sistema, não sendo uma imposição.

Para fins de descrição das tecnologias usadas no desenvolvimento do projeto podemos dividir o sistema em três partes: O banco de dados, a aplicação (servidor) e a interface com o usuário (cliente). O banco de dados usado será o MySQL. A aplicação será desenvolvida em PHP, usando o Zend Framework como *framework* de aplicações web, além de códigos na linguagem estatística R. Já a interface com o usuário será feita usando HTML, CSS, PHP e JavaScript com o auxílio da biblioteca jQuery. A validação dos requisitos se dará ao longo do desenvolvimento.

Ao final do projeto pretende-se apresentar um sistema que auxilie a criação de análises e gráficos estatísticos, bem como a preparação de relatórios das respostas de *surveys*.

2 Referencial Teórico

Para a concretização da primeira etapa do presente projeto realizou-se um estudo teórico a cerca dos assuntos mais relevantes dentro de seu contexto.

O estudo teórico começa com a plataforma sobre a qual o sistema será desenvolvido, a plataforma Polvo. Segue-se o estudo com o *framework* para aplicações web usado pela plataforma Polvo e sua arquitetura, o Model View Controller. Depois, será explanado sobre o Polvo Survey, módulo de aplicação de pesquisas tipo *survey* do Polvo. Por último será estudado o motor estatístico escolhido para auxiliar os cálculos estatísticos e a criação de gráficos de acordo com as necessidades do sistema, a linguagem R. Cada assunto encontra-se em uma das seguintes seções

2.1 Polvo

O [Polvo](#) é um ambiente de ensino virtual ou *virtual learning environment* (VLE) em inglês. É desenvolvido e mantido em Florianópolis pelo Laboratório de Tecnologia de Informação e Comunicação (LabTIC) da Escola de Administração e Gerência (ESAG) hoje centro de ensino de ciências da administração e socioeconômicas da Universidade do Estado de Santa Catarina (UDESC).

O Polvo hoje está em sua versão número 5 e é usado por vários centros de ensino da UDESC com planos de expansão para todos os centros da UDESC e também para outras universidades.

Suas principais funcionalidades são a postagem e manutenção de materiais de apoio ao ensino por parte dos professores que podem ser acessados pelos alunos; Envio de mensagens entre membros (alunos e professores) de uma turma; Controle de atividades, como trabalhos ou exercícios, individuais ou em grupo, com total controle e gerência de tais grupos; Calendários de turmas e individuais onde usuários podem controlar datas de provas, trabalhos, reuniões, aulas extras e afins.

As turmas, doravante chamadas de entidades, são organizadas de forma hierárquica em uma árvore. De fato, chamar as entidades de turmas pode ser enganoso pois entidades tanto podem ser turmas, como cursos, centros ou até mesmo grupos em geral como centros acadêmicos ou grupos de estudos.

Uma entidade que não esteja numa folha da árvore de entidades, como por exemplo a entidade ESAG, possui, obviamente, filhas - no caso cursos, como Administração Empresarial ou Administração Pública. Essas filhas também podem ter suas filhas, como por exemplo matérias como Administração em Marketing I ou Teorias de Apoio a Decisão, que, no caso, são folhas.

A vantagem de tal estrutura é a facilitação de uso das funcionalidades do Polvo para todos os usuários pertencentes a todas as entidades filhas. Um exemplo simples de tal facilitação é a disponibilização dos horários das turmas

durante um semestre. Só é necessário postar os horários como um material de apoio do curso do qual se deseja postar os horários.

Todos os alunos e professores pertencentes a qualquer matéria do referido curso tem então acesso rápido e fácil aos horários do seu curso. Não é necessário portanto que o usuário quando queira descobrir um horário tenha que procurar num site, muitas vezes de usabilidade ruim, ou mesmo deixar salvo em seu computador.

Cada indivíduo de cada entidade possui um papel específico. Isso significa que o mesmo usuário pode ter papéis diferentes dentro de cada entidade que participa. Portanto um usuário pode ter o papel de professor em uma entidade e de aluno em outra. Por padrão o papel de um usuário em uma entidade se estende para todas as entidades filhas, mas não para as entidades superiores. Um usuário sem papel em uma entidade é recebe o papel de visitante.

Não existe hierarquia entre os papéis, apesar de ser intuitivo achar que um professor seria “superior” a um aluno. Cada papel pode ter um conjunto de ações que não está contido no conjunto de ações de outros papéis.

Uma ação é uma relação de permissão entre um papel e uma determinada tarefa dentro do sistema. São as ações que definem a possibilidade da execução de uma atividade dentro do sistema. O papel simplesmente possui um conjunto de ações que permitem a execução de suas respectivas atividades.

Vale notar que as ações são independentes dos papéis, portanto uma ação pode ser adicionada a inúmeros papéis sem estar atrelada a nenhum

deles. Essa independência entre ações e papéis, junto com a facilidade de adição e remoção de ações em papéis cria uma grande versatilidade aos papéis, que podem ganhar ou perder funcionalidade rapidamente, sem a necessidade de mudanças no código, somente com pequenas alterações de configurações.

Essa versatilidade garante ao Polvo uma grande configurabilidade. Papéis podem ser criados ou deletados facilmente, e suas atribuições e permissões também adicionadas ou removidas facilmente por administradores do sistema, sem a intervenção de programadores.

Outro grande ponto básico do Polvo são os módulos. Módulos são ferramentas, algumas já citadas anteriormente, como Mala Direta e Material de Apoio. Tais módulos são adicionados às entidades e provêm a elas e seus usuários - caso eles tenham papéis com ações para tal - funcionalidades extras que não são oferecidas pelo Polvo básico.

A base do Polvo é chamada de Core, que provê as funcionalidades mais básicas, necessárias para o funcionamento mínimo do sistema como controle de usuários, entidades, papéis, ações e módulos. Os módulos então podem ser adicionados individualmente às entidades e assim aumentam a gama de funções disponíveis naquela entidade.

Mais uma vez o desacoplamento de partes ou funcionalidades do sistema confere ao Polvo uma maior versatilidade já que cada entidade pode ser ajustada com módulos que sirvam exatamente para as suas necessidades. Não se tem então entidades com funcionalidades alheias as suas competências.

2.2 Zend Framework

[Zend Framework](#) é um *framework* de código aberto para aplicações web desenvolvido em PHP otimizado para PHP5, que é a versão com melhor suporte a orientação a objetos (OO). Sua implementação faz uso avançado do padrão de arquitetura de software Model View Controller (MVC) o que facilita aos desenvolvedores a criação da estrutura básica de suas aplicações web.

Apesar de ser uma facilidade, e geralmente de grande ajuda para o desenvolvimento de aplicações de grande porte, a estruturação baseada em MVC não é obrigatória. O Zend Framework provê vários componentes fracamente acoplados que permitem a sua utilização de forma independente e sem a necessidade do uso do *framework* completo. Por isso também pode ser considerado como uma biblioteca de componentes, o que o torna utilizável a aplicações de grande e pequeno porte (ZEND TECHNOLOGIES, 2010a).

Seguindo o padrão MVC, cada aplicação possui um conjunto de modelos, controladores e visões. Uma particularidade do Zend Framework é o uso de ações para fazer as conexões do controlador com suas visões. Cada ação é responsável por uma parte das funcionalidades do controlador. A ação pode ou não possuir uma visão. Em caso positivo a ação também é responsável por mapear diretamente as requisições de sua visão para os modelos.

Geralmente uma aplicação possui inúmeras visões. Cada visão, normalmente, tem diversos elementos de interface comuns às diversas visões

tais como menus, barras de navegação, campos de busca, etc. Assim caso cada visão precisasse implementar todos esses elementos de interface existiria em cada visão um código redundante.

Neste modelo de desenvolvimento, qualquer mudança no *layout* exigiria a mesma modificação em cada visão presente na aplicação. Para facilitar a manutenção da consistência do *look-and-feel* em toda a aplicação o Zend Framework disponibiliza dois padrões de projeto chamados de Two Step View e Composite View.

Basicamente os dois padrões são usados da seguinte forma: Cada visão foca somente na sua responsabilidade, sem se importar com assuntos globais. Em tempo de execução essa visão é inserida no layout principal que é responsável pelo conteúdo redundante da aplicação.

Por fim, os modelos que contêm a lógica da aplicação se conectam aos controladores e as bases de dados. No Zend Framework, a conexão do modelo com o banco de dados é feita de forma completamente transparente. São utilizados adaptadores, que criam uma interface padrão para as aplicações conectarem-se aos diferentes Sistemas de Gerenciamento de Banco de Dados Relacionais – RDBMS (ZEND TECHNOLOGIES, 2010b).

Cada RDBMS precisa de um adaptador específico, mas como todos usam a mesma interface tornam-se transparente para a aplicação as diferenças entre os bancos de dados. Isso facilita o uso de diferentes RDBMS e a substituição não exige alterações no código da aplicação. A interface dos adaptadores é semelhante à interface da extensão PHP Data Objects (PDO) e

quando existentes o Zend Framework encapsula os *drivers* PDO em seus respectivos adaptadores.

Outro ponto importante na arquitetura do Zend Framework é a separação em módulos. Um módulo é como uma mini-aplicação, sua estrutura em disco é igual a estrutura da aplicação principal. A aplicação principal pode ser considerada como o módulo padrão (*default*). É fácil notar aqui uma analogia com o Polvo, onde o Core é o módulo padrão do sistema e os demais módulos são mini-aplicações agregadas a este.

Os módulos não têm dependência de outros módulos, apenas do módulo padrão. Esse baixo acoplamento permite, por exemplo, que somente alguns módulos sejam inseridos conforme a necessidade dos usuários do sistema.

Sendo estruturalmente iguais à parte principal da aplicação os módulos têm suas próprias visões, controladores e modelos. Cada um pode ter um layout diferente para as suas visões, pode usar um RDBMS diferente através de um adaptador diferente. Isto permite à aplicação o uso de bancos de dados diferentes inclusive em servidores diferentes, possibilitando o balanceamento de carga da aplicação entre os diversos servidores, de maneira transparente.

Outra grande facilidade do Zend Framework é o suporte a formulários. O grande problema dos formulários na internet é a validação de dados. O formulário do *framework* permite a adição de elementos para a entrada de dados com suporte a validadores, diversos deles providos pelo Zend Framework, como validadores de e-mail, de data, limitadores de tamanho de *strings* e números, e até mesmo de CAPTCHA.

Para a definição de estilos personalizados, os elementos dos formulários possibilitam a adição de classes CSS como um atributo do elemento ou a adição de decoradores, que melhora a orientação a objetos.

Os decoradores seguem o padrão de projeto Decorator, que permitem mudanças no comportamento de um objeto sem a necessidade de criar uma subclasse que mude esse comportamento. A grande vantagem desse padrão é que os decoradores podem ser empilhados, i.e., é possível fazer múltiplas mudanças em um objeto adicionando um decorador para cada mudança desejada.

Para esta mesma solução utilizando subclasses seriam necessárias subclasses para cada modificação desejada e para cada combinação possível entre tais modificações. É fácil notar que a quantidade de subclasses necessárias para cobrir todas as possíveis variações de comportamento cresce exponencialmente com cada nova mudança de comportamento adicionada, portanto o uso do padrão Decorator é justificável.

Cada decorador então pode adicionar uma pequena mudança no elemento do formulário como uma mudança de posição, tamanho ou mesmo o uso de uma lista, ao invés de uma tabela na representação de dados.

Com todas essas facilidades, como adaptadores de banco dados, formulários com decoradores, fácil balanceamento de carga, separação em módulos independentes (sua separação em módulos se encaixa perfeitamente com a filosofia do Polvo de separação e independência de ferramentas, que não tem sua instalação obrigatória para o uso do sistema), fácil integração com o MVC ou mesmo uso como uma simples biblioteca de componentes, como

servidor SOAP ou SMD, o Zend Framework é uma grande ferramenta na criação de aplicações web de pequeno, médio ou grande porte, como o Polvo.

2.3 Model View Controller

O Model View Controller foi primeiramente descrito em 1979 por Trygve Reenskaug enquanto trabalhava com Smalltalk na Xerox PARC. Inicialmente chamado de Thing-Model-View-Editor ou Coisa-Modelo-Visão-Editor em português (tradução nossa) onde Coisa é o que se deseja modelar, como um projeto de uma ponte ou de uma usina e Editor é responsável por coordenar um conjunto de visões sendo mostradas na tela em um determinado momento e também um meio de entrada de dados (REENSKAUG, 1979a).

Nota-se que o controlador ainda não estava formalmente definido. Isso aconteceu ainda em 1979 em um segundo artigo (REENSKAUG, 1979b) que descreve de forma mais semelhante ao que é hoje o Model View Controller, não contanto mais com a Coisa e o Editor sendo retirado do nome do padrão e perdendo a responsabilidade de coordenar diversas visões. O Editor, então, foi descrito somente como um meio de entrada de dados criado por uma visão para edição de dados e destruído assim que a edição estiver terminada.

Ao longo dos anos o MVC perdeu completamente o conceito do Editor, que foi concatenado à visão mas fora isso não se afastou muito do descrito por Reenskaug. O conceito atual dos seus componentes é geralmente o seguinte:

Os modelos implementam as regras de negócio gerando a lógica da aplicação e controlam o acesso às bases de dados;

Os controladores formam a ponte entre os modelos e as visões compondo o fluxo da aplicação determinando o mapeamento correto das requisições para os modelos e visões além de decidir quais dados cada visão precisa ou mesmo passando o controle para outro controlador caso necessário;

As visões definem o que será apresentado, e como isso será apresentado ao usuário, além de coletar dados. Desta forma, as visões representam a interface do sistema com o usuário.

Dessa forma, o MVC consegue isolar a lógica da aplicação de sua interface. Essa separação permite o desenvolvimento independente de cada camada e ainda possibilita a criação de interfaces diferentes para o mesmo modelo.

O uso de MVC em é quase um padrão no desenvolvimento de *frameworks* de aplicações web (SHAN; HUA, 2006) por suas vantagens de separação de preocupações e por conseguir se encaixar no modelo cliente-servidor da web, ficando as visões no cliente e os modelos e controladores no servidor.

2.4 Polvo Survey

Polvo Survey é um módulo do Polvo para criação de pesquisas do tipo *survey*. O objetivo de sua criação foi disponibilizar um sistema seguro, utilizando métodos criptográficos, que possibilite a coleta online de informações para pesquisas do tipo *survey* tendo como escopo a simplicidade de uso.

Nos sistemas existentes geralmente os requisitos de segurança, tais como integridade, sigilo e não repúdio não são tratados de forma adequada comprometendo a confiabilidade dos estudos realizados.

Outro aspecto a ser considerado é que em certas instituições pesquisas desenvolvidas devem ser analisadas pelo comitê de ética da instituição responsável pelo projeto de pesquisa. Nesta situação são necessários controles para que possa ser garantida a origem e que o questionário foi aprovado pelo comitê para sua efetiva aplicação.

Assim, percebeu-se a necessidade de um sistema que se preocupe com todos estes aspectos indo além da simples automação do processo de montagem, envio e análise básica dos questionários.

Para a implementação do Polvo Survey algumas ferramentas online de *survey* foram avaliadas sob o ponto de vista de suas usabilidades, funcionalidades, restrições e limites. Não foi considerado o tratamento de dados feitos por essas ferramentas, já que o objetivo do Polvo Survey é somente a criação e aplicação dos formulários. Tal tratamento é o objeto deste projeto, havendo assim uma divisão de tarefas entre módulos do Polvo.

Levando em consideração os erros e os acertos das ferramentas estudadas a modelagem foi terminada. Uma das principais restrições encontradas em todas as ferramentas foi a imposição da posição e tamanho das perguntas e demais elementos gráficos e textuais que formam um formulário de pesquisa.

Para dar mais liberdade ao criador do *survey* foi criado um editor de formulários que permite a completa edição da disposição dos elementos do

formulário. A figura Edição de Formulários no Polvo Survey mostra o ambiente de edição desenvolvido.

AVALIAÇÃO INSTITUCIONAL DA UNIVERSIDADE DO ESTADO DE SANTA CATARINA

PARTE I - O DOCENTE AVALIA AS AÇÕES DO CURSO

Prezado(a) Professor(a)

Este formulário integra o projeto de Autoavaliação Institucional da UDESC e tem como objetivo coletar dados para a avaliação dos cursos, com foco principal nas disciplinas e nos respectivos processos de ensino-aprendizagem.

Solicitamos o preenchimento do instrumento com postura crítica e consciente, pois seus resultados permitirão a reflexão sobre a qualidade do ensino ministrado e a sistematização de informações para subsidiar o planejamento e as decisões institucionais.

Para responder às questões utilize a escala numérica de 1 até 10, considerando que o valor 1 corresponde ao pior desempenho e o 10 ao melhor desempenho.

* Assinale NC naquelas questões sobre as quais julga não ter conhecimento suficiente para responder.

O professor deverá responder a Parte I do questionário para cada uma das disciplinas que estiver ministrando, na graduação e na pós-graduação

Centro:

Curso:

Disciplina:

Departamento:

1. Autoavaliação do desempenho docente e da disciplina

Figura 1 Edição de Formulários no Polvo Survey

Os tipos de campos de entrada de dados também foram bastante avaliados na concepção do projeto. Campos de única ou múltipla escolha, caixa e campo de texto, além de textos e imagens foram adotadas. Entradas de dados mais complexas não foram adotadas por fugirem do escopo de simplicidade do Polvo Survey.

Pesquisas diferentes requerem níveis de segurança diferentes, por isso, no Polvo Survey, existem três níveis de controle de acesso, chamados de visões: A visão aberta, a mais permissiva é, que permite que qualquer pessoa, sem nenhum controle, responda o *survey*; A visão semi-aberta limita as respostas às pessoas escolhidas pelo criador da pesquisa. Para cada pessoa é

gerado um *link* de uso único e só usando esse *link* é que é possível responder o *survey*; Por fim, a visão mais restritiva, a visão fechada. Ela limita as respostas aos usuários da entidade em que o *survey* foi criado.

Algumas vezes alguns dados (geralmente de cunho pessoal) de pesquisas são sigilosos. Para garantir o sigilo dos mesmos, o Polvo Survey permite ao criador do *survey* a criação de perguntas com sigilo de dados. Ao ativar a opção de segurança de dados na pergunta as respostas de tais perguntas passam a ser cifradas.

Para garantir tal segurança, é criado – e usado – de forma transparente ao usuário um par de chaves pública e privada no sistema usando RSA. A chave pública é usada para cifrar as respostas e, obviamente, a chave privada para decifrá-las.

Além desse par de chaves é criada uma senha alfanumérica aleatória que cifra, usando AES-256, a chave privada no servidor. Essa senha é enviada ao usuário através de um túnel de dados seguro sem ser salva no servidor. Isso permite que só o usuário criador do *survey* possa decifrar as respostas do *survey* garantindo o sigilo dos dados sem que a ele seja necessário o conhecimento do uso de chaves pública e privada.

Com todas essas funcionalidade e seguranças o Polvo Survey é uma ferramenta inovadora, com características não encontradas mesmo em soluções comerciais disponíveis.

2.5 Linguagem R

R é uma linguagem de programação e um ambiente computacional de código aberto para computação de dados e gráficos estatísticos inicialmente desenvolvido em 1993 por Ross Ihaka e Robert Gentleman ambos da Universidade de Auckland na Nova Zelândia.

Devido ao grande sucesso o projeto se tornou muito grande para ser cuidado só pelos dois criadores, então em 1997 foi criado um grupo de desenvolvimento do núcleo da linguagem, chamado de R Core Team, que o mantém até hoje.

Esse núcleo conta com capacidades para um grande número de procedimentos estatísticos para a maioria das distribuições estatísticas mais comuns, gráficos 2D e 3D diversos, amostragem, distribuição cumulativa, modelos linear, linear generalizado, regressão não linear, análises de séries temporais, testes paramétricos e não paramétricos clássicos, análises de agrupamento e suavização de dados.

Além disso, a linguagem permite o uso de bibliotecas desenvolvidas externamente. Essas bibliotecas são chamadas de *add-on packages* e dão uma grande flexibilidade à linguagem. Devido ao grande sucesso de R inúmeras bibliotecas foram surgindo para desempenhar as mais diversas funções estatísticas de manipulação, análise e geração de gráficos estatísticos.

As funcionalidades dessas bibliotecas vão desde distribuições pouco usuais até pacotes de estudo do genoma humano passando por bibliotecas de análise financeira e análise de padrões de voz.

Essa separação de pacotes permite a manutenção de um núcleo relativamente pequeno, com as principais e mais básicas funções de análise estatística sem que limite a sua extensão com novas funções.

O fato de ser uma linguagem simples, mas ao mesmo tempo poderosa (através de suas inúmeras bibliotecas), de código aberto e com suporte para os mais diversos sistemas operacionais fez com que R se tornasse, na prática, um padrão no desenvolvimento de softwares estatísticos.

Estima-se que seja usado por dezenas de milhares de pessoas de diversos tipos de entidades diariamente (TEETOR, 2011) indo desde universidades, para o ensino de estatística e análises de dados de pesquisas acadêmicas, até grandes empresas como Google, Pfizer, Bank of America, e Shell (ASHLEE, 2009).

2.5.1 Exemplos de uso da linguagem R

A simplicidade da linguagem pode ser percebida com alguns exemplos. O cálculo de diversas funções estatísticas são feitas com uma só linha, sem que o usuário precise saber exatamente como é calculada. No exemplo abaixo são calculadas a média, a mediana, o desvio padrão e variância de um conjunto de dados chamado x , mostrado na primeira linha, além da correlação

e covariância de Pearson deste conjunto com outro conjunto de dados chamado *y*, mostrado após a linha em branco:

```
> x <- c(5, 6, 7, 18, 19, 27, 29, 36, 42, 47);
> mean(x);
[1] 23.6
> median(x);
[1] 23
> sd(x);
[1] 15.14522
> var(x);
[1] 229.3778

> y <- c(4, 5, 23, 27, 30, 38, 39, 46, 48, 49);
> cor(x,y);
[1] 0.9392178
> cov(x,y);
[1] 234.0667
```

A correlação usando o método de Spearman também é simples:

```
> cor(x,y, method='spearman') ;
[1] 1
```

Cálculos estatísticos mais complexos também são simples de serem feitos como os mais básicos. Por exemplo em uma amostra que se ajusta a distribuição normal com média igual a 72 e desvio padrão de 15,2 qual a porcentagem de valores maiores de 84?

```
> pnorm(84, mean=72, sd=15.2, lower.tail=FALSE);
[1] 0.214917
```

Além da facilidade no cálculo estatístico, R também permite a criação de gráficos com simplicidade. Para o uso em exemplos e testes o R possui um conjunto de dados chamado *faithful*, que relaciona o tempo de duração de uma

erupção e o tempo desde a erupção anterior do géiser Old Faithful do parque Yellowstone. Usando esses dados o exemplo mostra como criar um gráfico de dispersão, o mais simples possível. O gráfico gerado é mostrado na figura Diagrama de Dispersão em R:

```
> plot(faithful$eruptions, faithful$waiting);
```

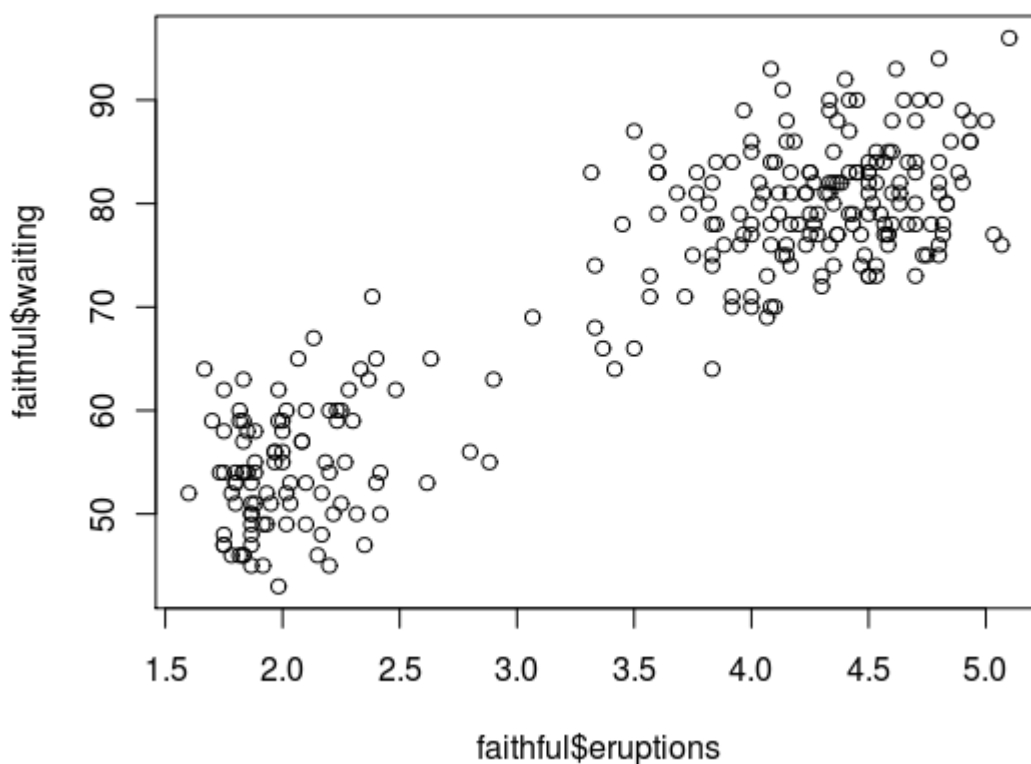


Figura 2 Diagrama de Dispersão em R

Conforme explanado anteriormente, usando pacotes de extensão é possível fazer cálculos e gráficos estatísticos muito mais avançados. Com a biblioteca GGLOT, por exemplo, é possível, com duas linhas de código, criar um gráfico de suavização de um conjunto de dados que relaciona o peso do carro e o tempo para percorrer um quarto de milha:

```
> c <- ggplot(mtcars, aes(qsec, wt));  
> c + stat_smooth() + geom_point();
```

O resultado é mostrado abaixo na figura Gráfico Suavização com Bandas de Confidência com biblioteca GGLOT em R. Os pontos pretos são os dados da tabela; a linha azul é função obtida pela suavização e as linhas cinza representam o grau de confiança da aproximação naquela posição.

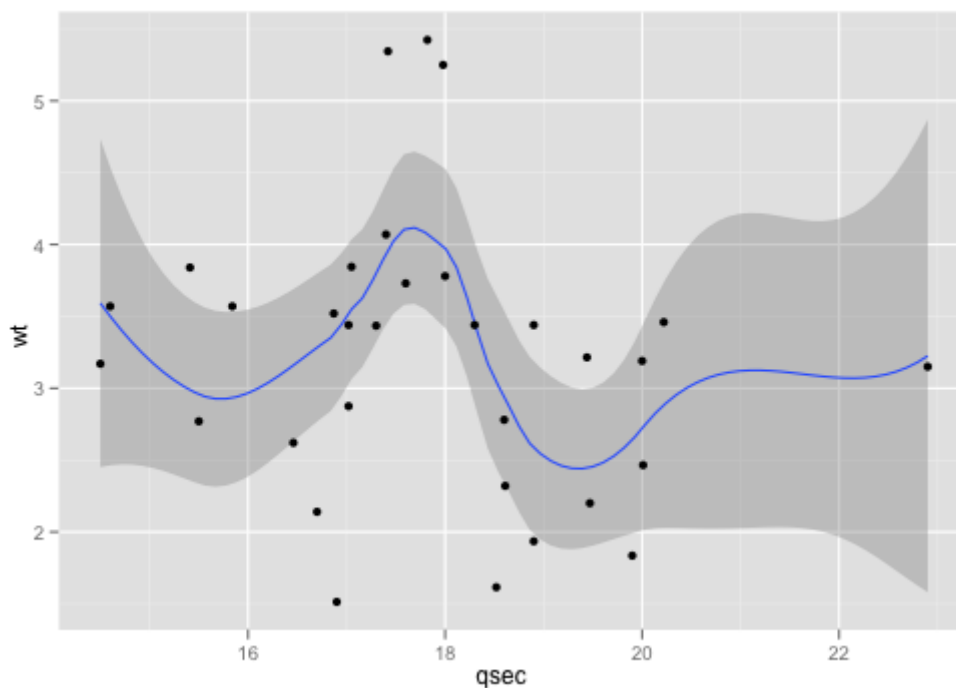


Figura 3 Gráfico Suavização com Bandas de Confidência com biblioteca GGLOT em R

2.5.2 Motivação para a escolha de R

Como já dito e demonstrado através de exemplos a linguagem R é muito poderosa, principalmente com o uso de bibliotecas externas, e ao mesmo

simples. É possível fazer cálculos e gerar gráficos extremamente complexos com poucas linhas de código. Ao criador de novas funções só é requerido o conhecimento da linguagem e, quando for o caso, a descoberta de uma biblioteca específica para o que ele deseja implementar.

Além disso, por ser primariamente uma linguagem e não um ambiente estatístico, é possível através de códigos usar todas as suas funcionalidades. Esses códigos, então, podem ser salvos e executados posteriormente, ao contrário de um ambiente estatístico que obriga o usuário a seguir (muitas vezes diversos) passos para realizar uma análise, sem a possibilidade de salvar esses passos para uso posterior.

O principal objetivo deste projeto de conclusão de curso é facilitar ao cientista social a análise estatística de dados e a criação de relatórios, livrando-os da necessidade de aprender novas ferramentas, com altas curvas de aprendizado. Assim, não será pedido ao usuário do sistema que crie funções em R, somente use funções previamente criadas.

Devido a facilidade do R, desenvolvedores, administradores do sistema ou mesmo usuários com experiência prévia em R (como peritos em estatística) poderão criar novas funções e adicioná-las ao sistema.

Por fim, R é um *software* de código aberto – fator fundamental para o uso na plataforma Polvo, que também é de código aberto – apoiado por uma grande comunidade e, na prática, o padrão para a análise estatística em meios acadêmicos e mesmo no meio privado.

3 Trabalhos Relacionados

Todos os trabalhos relacionados estudados são na verdade são ferramentas de criação e formulação de questionários de pesquisa, que é justamente onde focam a maior parte de seus esforços. Mas para fins comparativos com este trabalho de conclusão de curso foram estudados os tipos e quantidades de tratamentos de dados (análises estatísticas, criação de gráficos) possíveis de serem realizados pelas ferramentas, assim como a usabilidade de tais tratamentos.

A criação e postagem de respostas não foram estudadas, mas como para estudar os tratamentos de dados é antes necessário criar e responder questionários a criação e resposta podem ser mencionadas a fim de gerar uma compreensão mais geral sobre a ferramenta estudada.

As ferramentas analisadas foram:

- Google Docs – Form;
- LimeSurvey;
- Free Online Surveys;
- SurveyMonkey;
- QuestionPro.

3.1 Google Docs – Form

O [Google Docs](#) é uma ferramenta online de código fechado de uso gratuito. A sua criação e uso são simples e eficientes e as respostas são salvas numa tabela no próprio Google Docs.

Já o tratamento é muito básico ou quase inexistente. É possível gerar um relatório das respostas, mas não é possível definir nenhum parâmetro do relatório. Cada pergunta mostra um gráfico específico de acordo com o tipo de resposta que tal pergunta gera, conforme mostra a figura Relatório Google Docs – Form.

Não é possível mudar o tipo de gráfico; Escolher seus parâmetros, como cores e tamanho; Gerar análises estatísticas, e.g., desvio padrão, em cima dos resultados; Adicionar textos ou imagens externas ou mesmo selecionar quais perguntas serão mostradas no relatório.

Caso o usuário da ferramenta deseje qualquer outra coisa, deve usar a tabela de dados e com ela gerar outros gráficos ou análises que desejar com o Google Docs - Spreadsheets ou outra ferramenta de sua preferência.

22 [responses](#)

Summary [See complete responses](#)

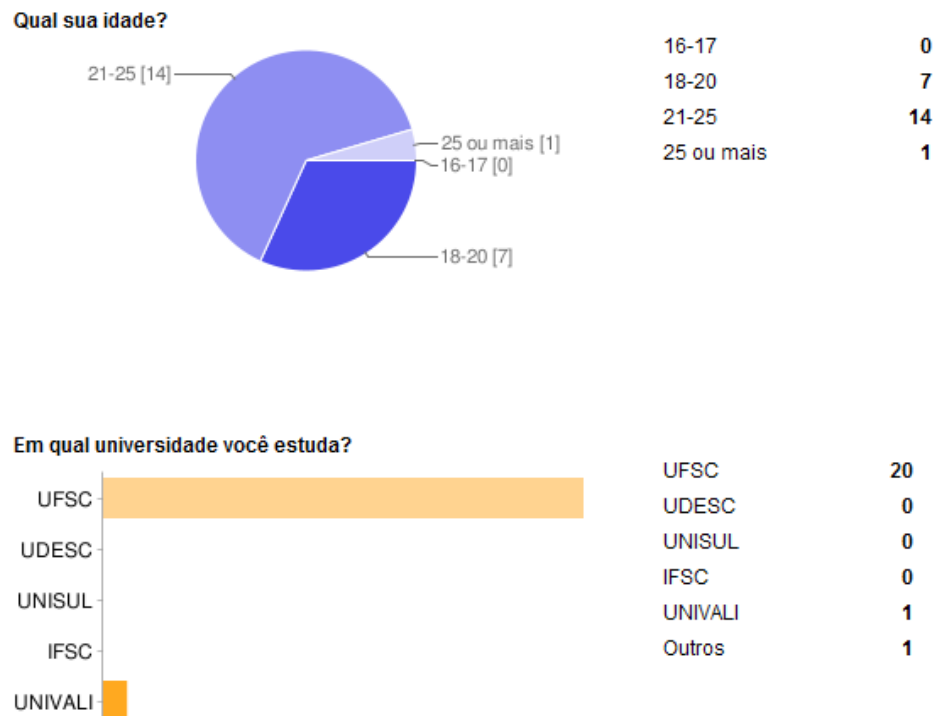


Figura 4 Relatório Google Docs – Form

3.2 LimeSurvey

O [LimeSurvey](#) é uma ferramenta online e open-source. É possível usá-la através dos servidores dos desenvolvedores da ferramenta, mas o foco é na instalação pelo cliente em seu próprio servidor.

Não tem uma interface muito intuitiva devido ao excesso de informações nas telas principal e de edição. Criar um *survey* pode ser confuso e demorado.

Por outro lado, é muito mais completo que o Google Docs - Form. Existem muito mais tipos de campos de entrada de dados, tais como campo de data e entrada numérico, que possuem validadores embutidos. Responder uma pesquisa é tão simples quanto o pelo Google Docs - Form. As respostas podem ser exportadas em CSV ou tabelas específicas para SPSS e R.

O tratamento de dados é um pouco mais completo que o do Google Docs - Form, mas também é relativamente básico. É possível gerar um relatório, selecionando as perguntas desejadas, que gera gráficos de pizza para as perguntas qualitativas sem a possibilidade de mudança de parâmetros ou tipo de gráfico. Sobre as perguntas quantitativas são feitos alguns cálculos estatísticos básicos, como por exemplo média, desvio padrão e quartis conforme mostra a figura Análise Quantitativa LimeSurvey.

Field summary for 4:	
Idade	
Calculation	Result
Count	10
Sum	315
Standard deviation	13.5
Average	28.6
Minimum	18
1st quartile (Q1)	21
2nd quartile (Median)	23
3rd quartile (Q3)	28.5
Maximum	56
Null values are ignored in calculations Q1 and Q3 calculated using minitab method	

Figura 5 Análise Quantitativa LimeSurvey

3.3 Free Online Surveys

[Free Online Surveys](#) é uma ferramenta online proprietária com uma versão paga e uma versão gratuita. A versão testada foi a gratuita, mas a versão paga não oferecia nenhuma funcionalidade extra relevante ao tema deste estudo. A criação de pesquisas não possui uma boa usabilidade pois exige seguidas mudanças de páginas em comparação às outras ferramentas apresentadas.

A ferramenta permite exportar os dados das respostas em um arquivo CSV e gera um relatório semelhante aos das demais ferramentas já apresentadas, com gráficos em barra para cada pergunta qualitativa. Para as perguntas quantitativas gera uma pequena tabela com análises estatísticas muito básicas conforme mostra a figura Análise Quantitativa Free Online Surveys.

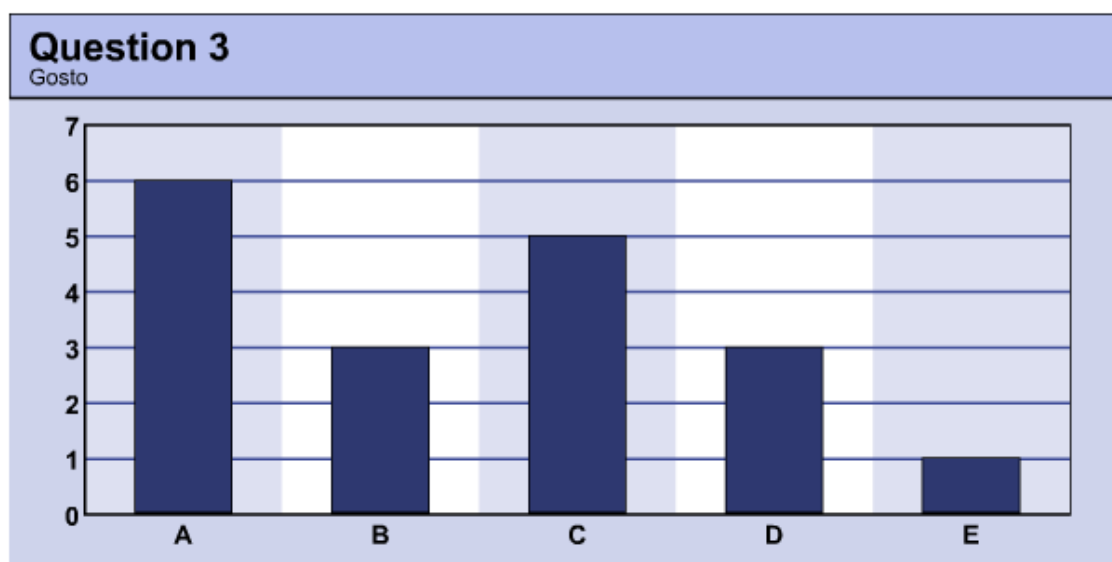
5) Idade

Average: 29.87
Range: 20<=>54
Median: 29
Total Responses: 8

Figura 6 Análise Quantitativa Free Online Surveys

O diferencial do Free Online Surveys em relação as ferramentas já apresentadas é que a ferramenta possibilita a criação de gráficos diferentes como barras, colunas, pizza e etc. para as perguntas qualitativas. Um exemplo de gráfico gerado é mostrado na Gráfico Qualitativo Free Online Surveys.

Esses gráficos são gerados individualmente em uma janela separada, portanto não ficam integrados ao relatório. Caso seja desejado criar um relatório usando tais gráficos é necessário salvá-los individualmente e criar um relatório usando um programa de edição de texto.



- A) Música
- B) Teatro
- C) Cinema
- D) Literatura
- E) Opera

Figura 7 Gráfico Qualitativo Free Online Surveys

3.4 SurveyMonkey

Assim como o Free Online Surveys o [SurveyMonkey](#) é uma ferramenta online e proprietária com versões pagas e uma gratuita. De novo, a versão gratuita foi testada mas, diferentemente do Free Online Surveys, algumas funcionalidades pertinentes ao tema do trabalho não puderam ser testadas.

A criação dos questionários é eficiente. Possui vários tipos de campos, inclusive com alguns validadores. Para a análise de resultados o SurveyMonkey cria um relatório simples com gráficos de barras para as perguntas qualitativas mas sem nenhum tipo de cálculo estatístico para as perguntas quantitativas, que, inclusive, são consideradas como perguntas textuais. Na versão paga existe uma ferramenta de análise textual que promete categorizar os textos para a realização de análises qualitativas sobre elas.

Para as perguntas qualitativas é possível também gerar gráficos definidos pelo usuário. Tem seu funcionamento semelhante ao Free Online Surveys, onde cada gráfico deve ser feito individualmente e não é possível gerar, na própria ferramenta, um relatório com esses gráficos. O diferencial do SurveyMonkey neste aspecto é a possibilidade de fazer algumas configurações nos gráficos, conforme mostra a figura Análise Qualitativa SurveyMonkey.

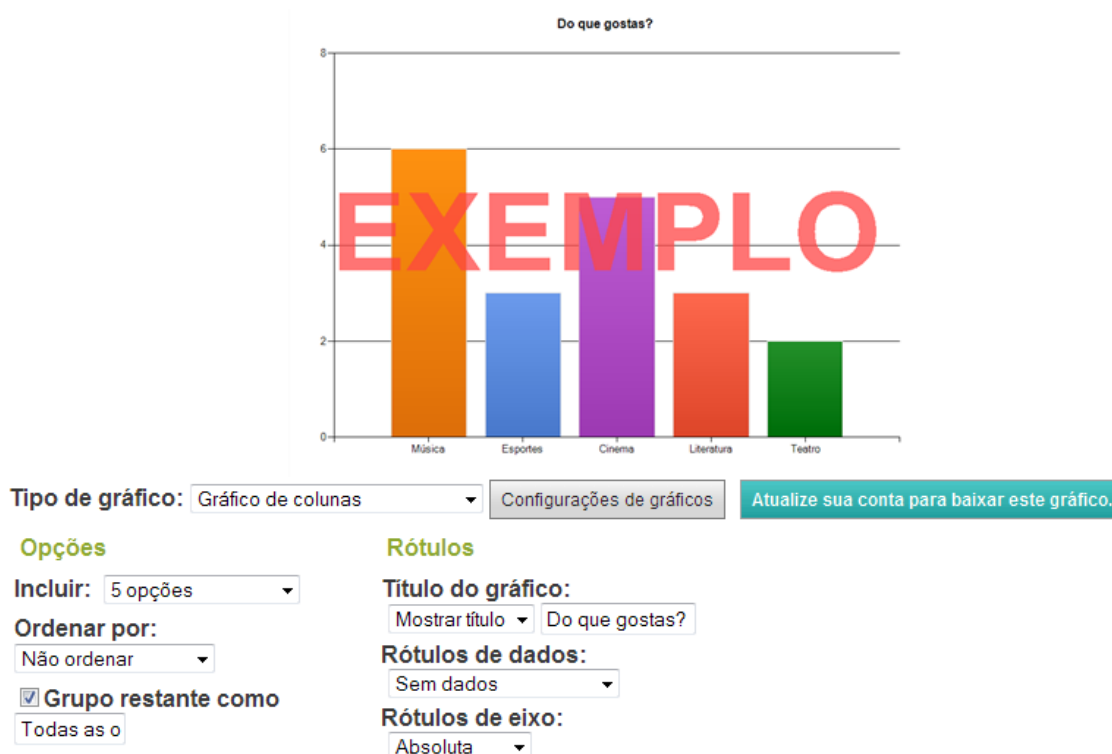


Figura 8 Análise Qualitativa SurveyMonkey

Além disso, na versão paga é possível fazer certas correlações de dados. A utilidade de tais correlações não puderam ser avaliadas.

3.5 QuestionPro

[QuestionPro](#) também é uma ferramenta proprietária com versões paga e gratuita, porém, oferece uma licença de avaliação gratuita, válida por quinze dias, com a qual é possível testar todas as suas funcionalidades.

Ela é, dentre as ferramentas testadas, a com mais funcionalidades e análises. A criação de formulários possui algumas facilidades como uso de *templates*. Com a versão paga é possível usar alguns tipos de perguntas além do encontrado em outras ferramentas. Esses tipos diferentes geram alguns gráficos e análises diferentes.

O relatório básico do QuestionPro segue a linha das demais ferramentas. Cada pergunta qualitativa é mostrada através de um gráfico de barras do percentual de cada resposta, conforme mostra a figura Análise Qualitativa QuestionPro. Na mesma figura pode-se notar uma inconsistência da ferramenta. São gerados alguns cálculos como a média e o desvio padrão para perguntas qualitativas.

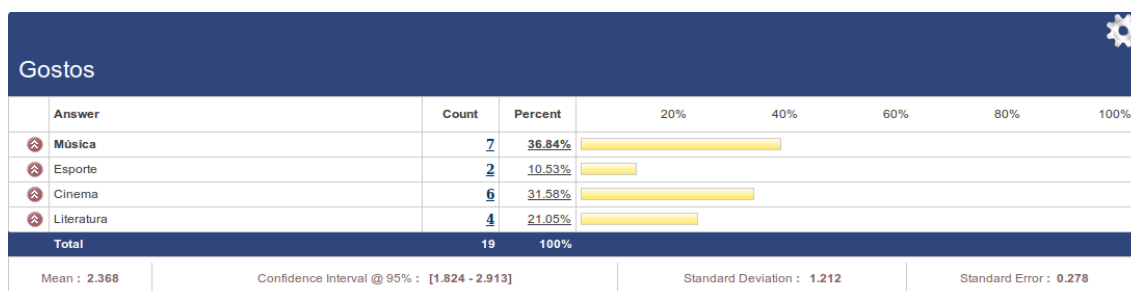


Figura 9 Análise Qualitativa QuestionPro

Além do gráfico básico é possível gerar uma grande quantidade de relatórios e análises. Pode-se gerar relatórios um pouco mais completos escolhendo quais perguntas serão mostradas.

Esse relatório gera alguns gráficos diferentes para as perguntas mas sem permitir que o usuário escolha qual o tipo de gráfico e seus parâmetros. Outro ponto negativo é que os cálculos mostrados na figura Análise Qualitativa QuestionPro continuam sendo mostrados. Também é possível gerar gráficos de correlação de perguntas, conforme mostra a figura Relatório Correlação QuestionPro.

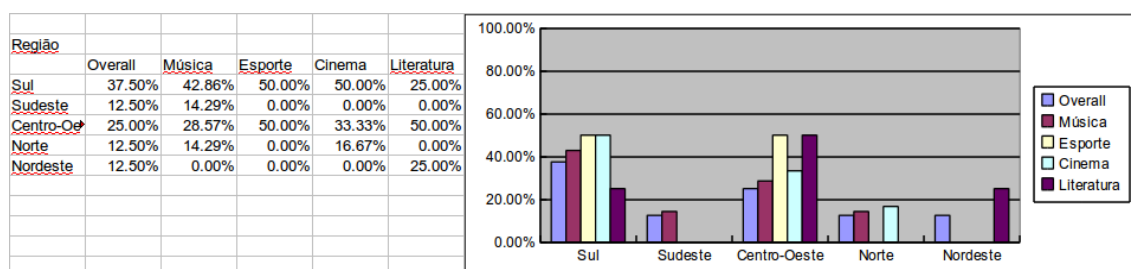


Figura 10 Relatório Correlação QuestionPro

Além disso, é possível criar análises de tendência, com relatórios que mostram as variações das respostas ao longo de divisões temporais como semanas ou meses, e análises TURF (abreviação de Total Unduplicated Reach and Frequency Analysis ou Análise Total Não Duplicada de Alcance e Frequência em português).

Análises TURF são análises feitas sobre questões de múltipla escolha, i.e., onde mais de uma resposta pode ser escolhida, e basicamente tenta responder qual a abrangência, dentre a população da pesquisa, de um conjunto de respostas (COHEN, 1993).

Um exemplo de uso de tal análise é a decisão de compra de espaço de propaganda em revistas. Supondo que um *survey* tenha sido feito entre o público alvo de um produto perguntando quais revistas eles liam. Muitos leitores podem ler mais de uma revista ao mesmo tempo, portanto não é possível simplesmente somar a quantidade de pessoas que lê cada revista para chegar no número total de pessoas que leem as duas revistas. Com análises TURF é possível obter a quantidade de respostas únicas para cada par, ou tripla, etc. de respostas.

Além disso, cada revista tem um custo de anúncio diferente. Portanto, o custo de anúncio nas revistas A e B pode ser diferente do custo nas revistas A e C ou B e C. Assim, dado um orçamento limitado, e o alcance de cada conjunto é possível calcular qual conjunto de revistas tem o maior alcance abaixo do orçamento previsto.

Então, apesar da inconsistência de fazer cálculos de média e desvio padrão em perguntas qualitativas o QuestionPro é, dentre as ferramentas analisadas, a mais completa.

4 Desenvolvimento

O projeto iniciou-se em meados de 2010 e, conforme descrito no capítulo Metodologia, foi dividido em algumas etapas, sendo elas: Levantamento de requisitos; Estudo comparativo de trabalhos relacionados; Construção do referencial teórico do projeto; Implementação em R de funções que possam sanar os principais requisitos funcionais levantados; Desenvolvimento e validação do módulo estatístico.

As etapas de construção do referencial teórico e estudo comparativo de trabalhos relacionados são, respectivamente, alvo dos capítulos 2 e 3. As demais são alvo dos subcapítulos a seguir.

4.1 Levantamento de Requisitos

Podemos considerar como os primeiros requisitos do sistema as premissas sobre as quais o mesmo foi concebido. Sendo elas:

- Capacidade de gerar análises estatísticas (usando R) diversas a partir de dados de surveys online;
- Deve ser de fácil uso, i.e., ter uma boa usabilidade;
- Deve ser um módulo da plataforma Polvo;
- Capacidade de geração de relatórios a partir das análises feitas;

- Permitir a adição de novas funções de análise sem que seja necessário fazer alterações no código do sistema.

O capítulo Justificativa descreve os motivos da criação deste projeto e portanto a maiorias das premissas listadas, por isso elas não serão descritas novamente neste capítulo.

Já a possibilidade de adição de novas funções de análise ao sistema foi comentada no capítulo Motivação para a escolha de R mas não explicada. A motivação deste requisito funcional vem da percepção de que as funções de análise são implementadas em uma linguagem (R) externa ao sistema em que estão inseridas além de poderem ser independentes. Assim, qualquer pessoa com conhecimentos em R pode prover uma função ao sistema, desde que mantidas as precauções de segurança, sem que tenha conhecimento na plataforma Polvo.

Em seguida, os próximos requisitos funcionais e não funcionais foram levantados através de entrevistas com professores da UDESC, que no momento da entrevistas estavam usando o Polvo Survey para aplicar pesquisas de avaliação institucional docente e discente. Posteriormente, o estudo dos trabalhos relacionados também permitiu a obtenção de alguns requisitos.

Os requisitos levantados nestas entrevistas tiveram grande influência nas necessidades de análise das avaliações institucionais que, portanto, tiveram grande influência nas decisões das funcionalidades do sistema.

Os requisitos definidos desta forma foram:

- Foco na geração de gráficos;

- Facilidade de customização dos gráficos gerados;
- Possibilidade de criação de gráficos de correlação entre perguntas diferentes;
- Possibilidade de customização das categorias durante as análises qualitativas.

Os gráficos são muito importantes na estatística. Anscombe (1973) mostra a importância do uso de gráficos na análise estatística que podem ajudar o analista a notar e avaliar algumas características básicas dos dados.

Cálculos estatísticos geralmente se baseiam em suposições acerca da natureza dos dados analisados mas tais suposições podem estar erradas. É nesse ponto que os gráficos são de extrema valia, já que facilitam a visualização da natureza dos dados.

Além disso, gráficos são de fácil entendimento mesmo para pessoas sem conhecimento profundo em matemática e estatística. Por isso a fácil criação de gráficos e simples customização dos mesmo são requisitos importantes para um sistema que se propõe a simplificar ao máximo a criação de análises, por parte de cientistas sociais, de dados obtidos em pesquisas.

Os gráficos de correlação também são importantes para mostrar as diferenças, ou semelhanças, entre as respostas de uma determinada pergunta em relação a outra pergunta, e.g., tendo duas perguntas, uma perguntando o sexo do participante e outra perguntando se gosta de futebol. Um gráfico de correlação entre essas duas perguntas pode mostrar a diferença do gosto de futebol entre os homens e as mulheres.

Além disso, algumas vezes se faz necessário a junção de categorias qualitativas para melhor observação dos dados. Um exemplo de uso desta funcionalidade se dá quando uma pergunta sobre a qualidade do ensino é feita usando notas de 0 a 10, mas posteriormente é decidido mostrar os resultados em uma escala Likert. Deve ser possível portanto juntar, ou simplesmente renomear, categorias existentes em novas categorias, mantendo a consistência das análises, para uma apresentação diferente dos resultados.

Já na etapa de estudo comparativo dos trabalhos relacionados, os requisitos levantados visavam não incorrer nos mesmos erros cometidos pelas ferramentas estudadas e tentar reproduzir funcionalidades que por ventura fossem indispensáveis no desenvolvimento de uma ferramenta semelhante.

O estudo permitiu verificar que as ferramentas existentes geralmente geram relatórios simples, sem a possibilidade de escolha dos tipos de análises e gráficos ou mesmo das perguntas que fazem parte do relatório. Se o usuário desejar algo um pouco mais avançado deve exportar os dados para uma ferramenta de análise estatística.

Os problemas dessa abordagem já foram explicados na Introdução deste trabalho, e em face deles foi desenvolvida uma ferramenta que fosse simples o suficiente para ser usada por qualquer pessoa e ao mesmo tempo poderosa o suficiente para gerar análises das mais diversas, sendo altamente configurável.

Além disso, foi possível perceber que a total integração da ferramenta de análise com uma ferramenta de *survey* é de extrema importância para a

usabilidade da ferramenta de análise, pois automatiza a importação dos dados da pesquisa à ferramenta.

Assim, mais alguns requisitos foram obtidos:

- Deve haver uma integração com o Polvo Survey que facilite a análise das pesquisas criadas pela ferramenta;
- Possibilidade de escolha das análises inseridas nos relatórios gerados;
- Capacidade de posicionamento dos elementos inseridos ao relatório;
- Possibilidade de adição ao relatório de textos e imagens externos ao sistema;

4.2 Implementação de funções em R

Após o levantamento de requisitos foram implementadas funções de análise em R que pudessem sanar os principais requisitos funcionais obtidos relacionados à análises, sendo eles a criação de diversos tipos de gráficos customizáveis e a possibilidade de criação de categorias.

Depois da obtenção dos requisitos do sistema, foram estudadas as tecnologias envolvidas no desenvolvimento de trabalho, conforme o capítulo Referencial Teórico. Com o conhecimento das tecnologias supracitadas foi possível iniciar o desenvolvimento do sistema.

O primeiro passo foi criar as funções de análise em R que cobrissem os requisitos mínimos de criação de análises e gráficos do sistema. A partir dos

requisitos obtidos foi percebido que as principais funções necessárias eram de criação de gráficos diversos com certo nível de customização, inclusive permitindo a criação de categorias no caso de análises qualitativas.

Por isso, a maioria das funções criadas são de geração de gráficos, como gráficos de barras, pizza, caixas e linhas. A figura Exemplo Gráfico Gerado mostra o exemplo de um gráfico de barras gerado a partir de uma função criada em R.

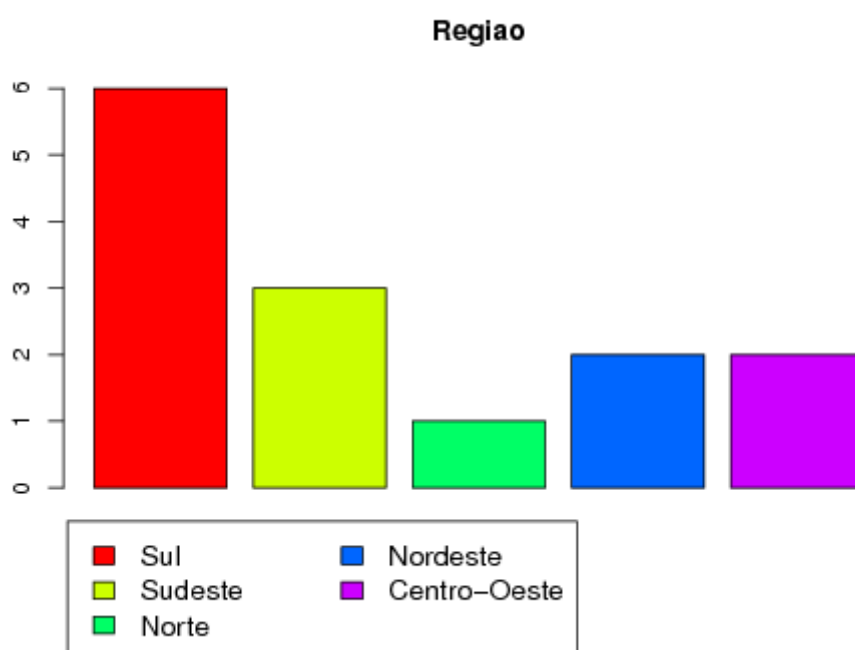


Figura 11 Exemplo Gráfico Gerado

Cada função possui certos parâmetros de configuração como o título de um gráfico ou o grau de confiança desejado em uma análise estatística. No exemplo a cima os parâmetros são a são a quantidade de colunas da legenda e o título do gráfico.

Além disso, as funções devem receber como parâmetro o conjunto de dados, o nome do arquivo de saída do resultado, seja um gráfico ou um texto e

em caso de uma função de análise qualitativa os possíveis valores de resposta, chamados de fatores e seus respectivos nomes para impressão.

O conjunto de dados é recebido como uma lista de respostas não contabilizadas. Em casos onde uma opção de resposta não tiver sido respondida nenhuma vez o conjunto de dados não terá nenhuma menção à opção não marcada. Isso impede que os fatores sejam extraídos diretamente dele e obriga a adição do parâmetro de fatores.

Como R não é uma linguagem orientada a objetos não é possível fazer com que cada função de análise contasse com um método de mesmo nome e uma assinatura comum a todas. Porém, quando a função for chamada pelo sistema para ser executada deve ser possível saber quais são os nomes dos parâmetros correspondentes aos parâmetros de dados, nomes, fatores e etc.

Por isso alguns nomes de parâmetros foram restringidos para serem usados obrigatória e exclusivamente pelo sistema. Assim, o criador de funções a serem adicionadas ao sistema deve estar ciente de tal fato.

Durante o desenvolvimento das funções de análise foi percebido que várias funções possuíam código em comum, todas liam os dados recebidos da mesma forma, as funções geradoras de gráficos definiam os padrões de gráfico do mesmo jeito e assim por diante. Existiam inclusive em alguns casos mais de uma função para a criação do mesmo tipo de gráfico mudando o tipo do rótulo, seja uma legenda como na figura Exemplo Gráfico Gerado ou rótulos inclinados abaixo de cada barra.

Por isso, foi introduzido o conceito de dependência de funções. Funções poderiam chamar outras para diminuir a repetição de código e melhorar sua

compreensão. Além disso, as funções que não fossem de análise não precisariam seguir as diretrizes de nomenclatura de parâmetros do sistema pois não serão chamadas diretamente por ele, mas somente por outras funções R.

Outro requisito funcional do sistema era a capacidade de criação de categorias. Uma categoria é um conjunto não vazio de fatores de uma análise qualitativa. O objetivo é fazer com que os dados relativos aos fatores pertencentes à categoria sejam considerados pela função de análise como dados de um só fator.

Isso foi feito de forma independente das funções de análise já que do ponto do vista da função a forma em que as respostas estão categorizadas é indiferente.

Sendo assim, uma função de agrupamento de fatores foi criada, essa função recebe como parâmetros o conjunto de dados, os fatores e a relação dos fatores pertencentes a cada categoria e retorna os dados categorizados. Em seguida os dados categorizados são passados como parâmetro para a função de análise da mesma forma que seriam sem a categorização. Já os fatores recebidos pela função de análise passam a ser as novas categorias e não mais os fatores originais.

A figura Gráfico com Junção de Categorias mostra um gráfico gerado com os mesmos dados e pela mesma função do gráfico mostrado na figura Exemplo Gráfico Gerado mas com as regiões Sul e Sudeste juntas em uma só categoria, chamada de Sul-Sudeste, e as regiões Norte, Nordeste e Centro-Oeste também juntas em uma só categoria, chamada de Norte-Nordeste.

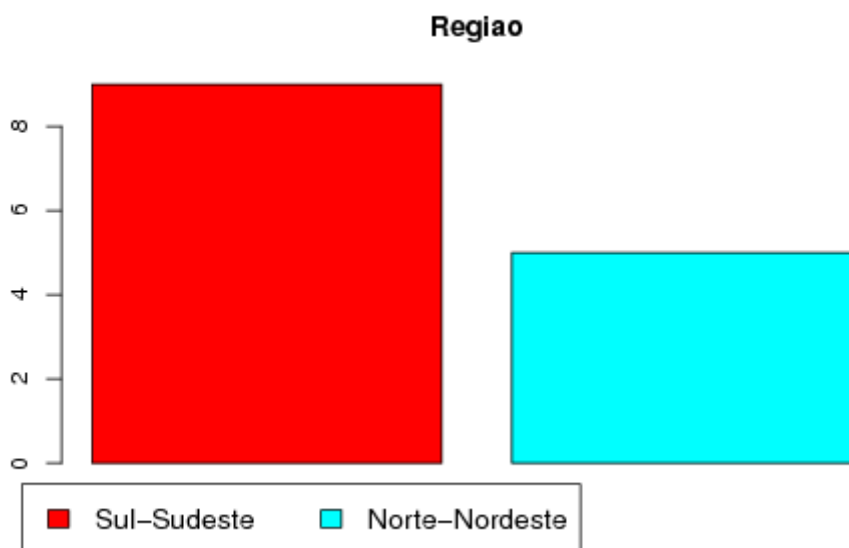


Figura 12 Gráfico com Junção de Categorias

Também foram criadas funções para a criação de gráficos de correlação, como mostra a figura Gráfico Correlação. A execução dessas funções é feita de forma semelhante a execução das outras funções. A única mudança é a necessidade de se juntar os dados e fatores das duas perguntas antes de passá-los ao R.

Os gráficos de correlação, conforme mencionado no subcapítulo Levantamento de Requisitos, são muito úteis para mostrar as diferenças, ou semelhanças, entre as respostas de duas perguntas. O exemplo, apesar de contar com um pequeno espaço amostral, mostra os gostos de respondedores de uma pesquisa separados pela suas respectivas regiões de origem.

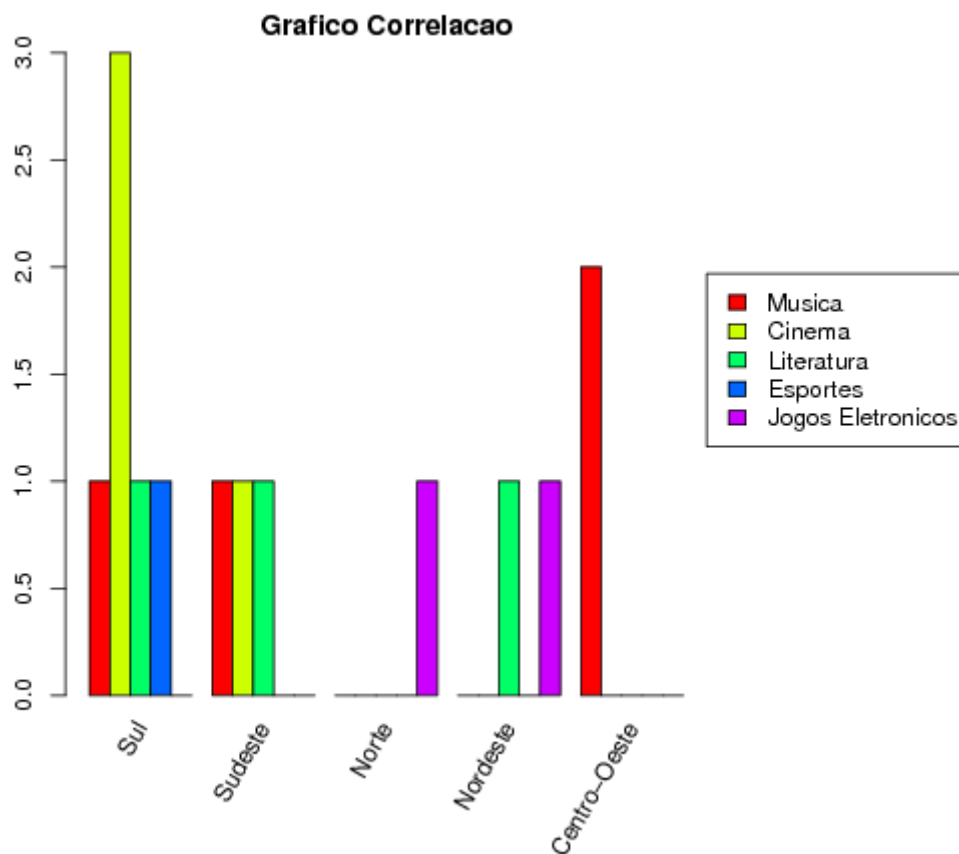


Figura 13 Gráfico Correlação

4.3 Desenvolvimento do Módulo Estatístico

A partir dos requisitos do sistema, do conhecimento das tecnologias envolvidas no desenvolvimento de trabalho – conforme o capítulo Referencial Teórico – e do entendimento e definição do funcionamento das funções R foi possível iniciar o desenvolvimento do sistema propriamente dito.

O desenvolvimento do sistema foi dividido em três etapas. A primeira consistiu em desenvolver um método de executar as funções de análise pela ferramenta; A segunda em implementar a funcionalidade de criação,

armazenamento e execução de análises; E a terceira em implementar a funcionalidade de geração e armazenamento de relatórios. As três etapas serão explanas em detalhes nos subcapítulos seguintes.

Vale ressaltar que conforme mencionado no capítulo Metodologia o sistema consiste em um módulo da plataforma Polvo e conforme descrito no capítulo Polvo cada módulo seu implementa o padrão de arquitetura MVC.

A criação dos modelos, controladores e visões, além das tabelas de banco de dados não serão comentadas em seções específicas para parte do MVC e sim conforme foram criadas, ou seja, cada etapa dará os detalhes dos elementos relativos ao seu desenvolvimento.

4.3.1 Método de Execução de Funções R

O primeiro passo para desenvolver um método de execução das funções de análise foi definir uma forma de armazenar as funções no sistema. Para isso algumas características das funções tiveram que ser consideradas.

A primeira característica é dependência de funções. Deve existir um meio da ferramenta saber quais são as dependências da função a ser executada.

A segunda é o fato de existirem tanto funções de análise, que devem ser mostradas como opções de análise ao usuário, como funções auxiliares, que não devem ser mostradas ao usuário.

Outra característica são os parâmetros de customização das análises. O sistema deve saber quais são os parâmetros da função para que possam ser mostrados ao usuário, que poderá defini-los e, idealmente, quais os tipos dos parâmetros para que a ferramenta possa validá-los.

O requisito funcional que diz que deve ser possível adicionar funções ao sistema sem alterar seu código impede que os parâmetros, relações de dependência e o tipo da função (de análise ou auxiliar) sejam guardados, implícita ou explicitamente dentro do código.

Se faz então necessário o uso de banco de dados para armazenar esses dados e adquiri-los quando necessário. Sendo assim, para executar uma função o sistema precisa acessar o banco de dados para obter os dados das funções. Para aproveitar esse acesso ao banco de dados o código das funções também foi armazenado no BD.

Para armazenar as funções e seus dados no banco de dados foram criadas três tabelas. A primeira guarda a função em si, i.e., seu nome, código, tipo, nome para exibição e descrição; A segunda, a relação de dependência entre as funções, ou seja, quais funções devem ser lidas para compor o código em R que será executado; Já a terceira guarda os parâmetros das funções. Os campos dessa tabela são o nome do parâmetro, qual função ele pertence, um nome para ser exibido ao usuário, uma descrição e o tipo do parâmetro.

Depois de definida a forma de armazenamento das funções foi definido o método de execução das funções em si.

Não existe uma API R para PHP, por isso não havia como fazer chamadas R como funções PHP. Entretanto, existem algumas interfaces web

para R. Banfield (1999) e Jockers (2008) implementaram duas interfaces web distintas para R executando os códigos da linguagem através de comandos *shell*. Isso é feito usando a função *exec* do PHP.

Assim, para executar uma função o PHP lê do banco de dados a função a ser executada, todas as funções da qual ela é dependente – aproveitando os dados da tabela das relações de dependência – e a função de agrupamento de categorias que são concatenadas em uma *string* pra uso posterior.

Em seguida, os parâmetros da função (restritos e de customização) são concatenados na forma *nome=valor* que também são guardados para uso posterior. Vale notar que a forma de obtenção dos parâmetros, sendo eles os dados que serão analisados, os fatores, as categorias e os parâmetros de customização será tema do subcapítulo Funcionalidade de Análise.

O passo seguinte é criar todo o comando R que será executado via *shell*. Ele é feito concatenando a chamada para a função a ser executada, os parâmetros restritos ao sistema, os parâmetros de customização da análise e o código de todas as funções obtidas. Se for o caso também é concatenada a chamada para a função de agrupamento de categorias e o encaminhamento do resultado desta para a função de análise. Por fim o comando criado é executado.

4.3.2 Funcionalidade de Análise

Um requisito não funcional do sistema é a usabilidade do mesmo, com isso em mente a criação de análises foi projetada, sob o ponto de vista do usuário, com foco na facilidade de uso e simplicidade da interface.

Por causa da integração do módulo estatístico com o módulo de *survey* do Polvo a criação de análises de dá depois da criação do pesquisa. O usuário seleciona o *survey* para o qual ele deseja criar análises e é direcionado para a página de análises, que lista as análises já criadas e permite a criação ou edição das mesmas.

A criação é simples, a cada página o usuário escolhe um atributo da análise. Primeiro escolhe a função desejada dentre as apresentadas na tela, essas funções são as funções de análise definidas na tabela de funções.

Em seguida escolhe qual pergunta deseja analisar. Em caso de uma análise que envolva mais de uma pergunta, como uma correlação, o usuário seleciona quantas perguntas forem necessárias à função selecionada. As perguntas mostradas são as perguntas do *survey* selecionado anteriormente pelo usuário.

Depois, caso seja uma análise qualitativa, o usuário vê a lista de fatores da pergunta e pode criar categorias. A cada categoria criada a lista de fatores é atualizada para refletir a mudança feita. Nesta lista também é possível editar ou remover as categorias criadas. Vale notar que é possível criar categorias com um só fator o que na prática funciona como a renomeação de um fator.

Na próxima página o usuário define os valores dos parâmetros da função. Terminando assim a criação da análise por parte do usuário.

O último passo antes da execução da análise é ler os dados da pergunta selecionada do banco de dados do *survey*. Finalmente, a análise é executada e o resultado, que foi salvo em um arquivo, mostrado ao usuário que decide se deseja salvá-la ou descartá-la.

Caso o usuário escolha atributos incompatíveis para a análise, e.g., uma função de análise quantitativa e uma pergunta qualitativa ele será avisado, neste momento, que alguns dos atributos escolhidos são incompatíveis e que ele deve corrigir o erro. Não é possível salvar uma análise que não consiga ser executada.

Em caso de sucesso também é possível baixar o resultado caso o usuário deseje. A figura Visão de Resultado de Análise Gerada mostra um exemplo de uma análise criada e das opções dadas ao usuário.

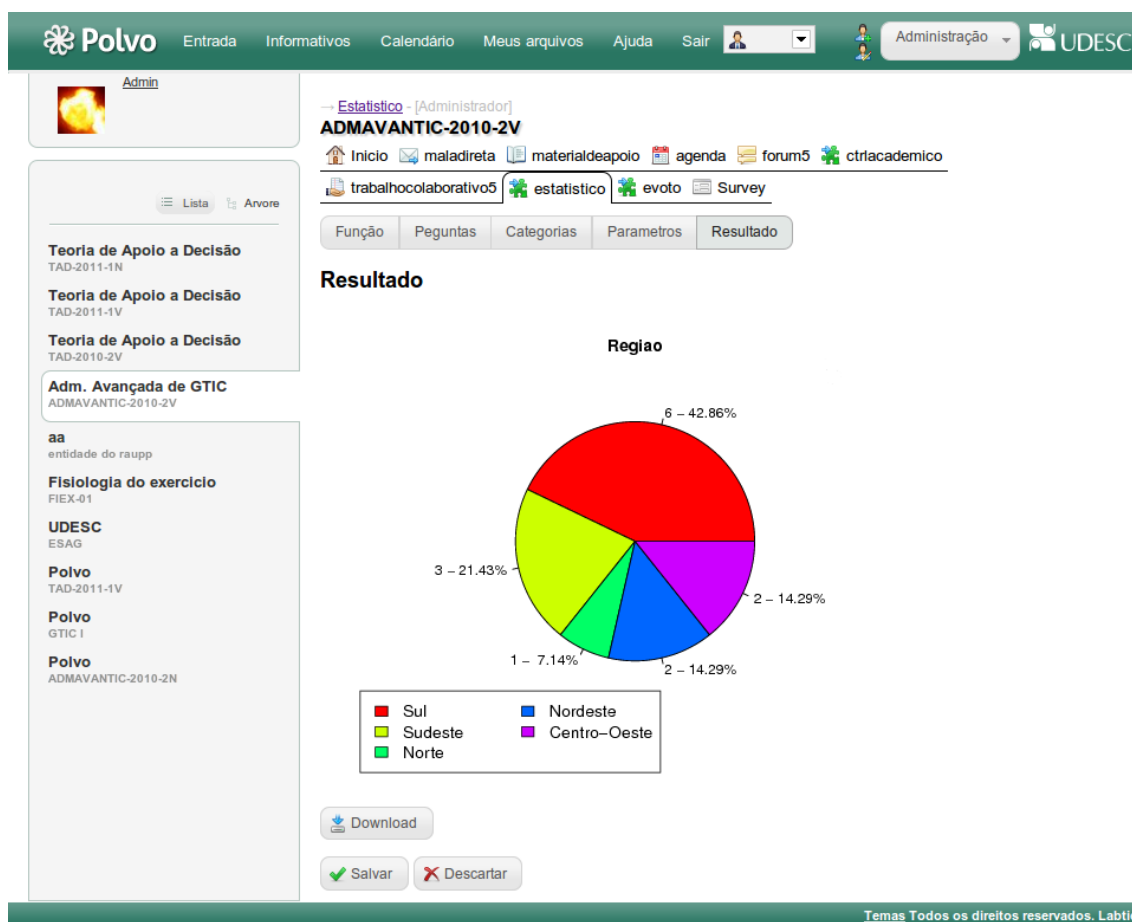


Figura 14 Visão de Resultado de Análise Gerada

Na figura a cima também é possível ver botões de navegação direta para as páginas de criação da análise. No começo da criação tais botões são desabilitados e a cada atributo da análise definido pelo usuário os botões relacionados vão sendo habilitados.

Os botões de navegação são muito úteis caso o usuário não esteja satisfeito com o resultado da análise. Com eles o usuário poderá, por exemplo, voltar à página de edição de categorias, criar ou remover uma categoria e voltar ao resultado para verificar se as alterações o satisfizeram.

O usuário então pode salvar a análise que será listada na página de análises. A qualquer momento depois da criação da análise o usuário poderá editar ou a análise criada.

A edição funciona exatamente como a criação, mas já inicia com todos os atributos definidos, o que permite o uso dos botões de navegação para acessar diretamente o atributo que deseja alterar e em seguida ir diretamente para o resultado. Vale ressaltar que em caso de alteração da função ou da pergunta um outro atributo é apagado, os parâmetros e as categorias respectivamente, o que obriga a definição desses atributos e, portanto, impede o acesso direto ao resultado da análise.

Dois detalhes importantes devem ser comentados. O primeiro é que as análises só são salvas na tela de resultados, portanto não é possível nem criar nem editar parcialmente uma análise e salvá-la. E mesmo que uma análise esteja completamente criada ela só será salva em caso de confirmação do usuário.

Essa política foi implementada para impedir que o usuário tenha salvo uma análise incompleta ou com a qual ele não esteja satisfeito. Impedindo que o esquecimento desta análise incompleta faça com que seja gerado um relatório incorreto.

O segundo detalhe é que o resultado em si não é salvo, somente os atributos da análise. Essa abordagem permite que o criador do *survey* crie as análises antes mesmo da pesquisa ter sido completada. Assim, a qualquer momento é possível reexecutá-la a qualquer momento, aproveitando-se dos dados mais recentes. Caso o resultado fosse salvo ao invés da análise, depois de mais algumas respostas o resultado salvo estaria inconsistente com os dados mais novos.

Sob o ponto de vista do sistema a cada novo atributo definido pelo usuário o objeto de análise é salvo na sessão do usuário. Somente após o explícito salvamento da análise é que ela é salva no banco de dados.

Existem alguns pontos negativos do uso da sessão para salvamento de dados. Como o protocolo HTTP não possui o conceito de estados, para usar a sessão, a cada requisição o servidor deve procurar pelos dados de sessão localmente, o que limita a escalabilidade e aumenta o trabalho que o servidor deve fazer para recuperar a sessão.

O aconselhado é salvar os objetos no banco de dados e só carregar o mínimo necessário a cada nova requisição, porém o objeto de análise é pequeno e seria todo carregado a cada requisição, portanto as duas abordagens geram o mesmo trabalho ao servidor.

Além disso, como a análise só é realmente salva quando o usuário confirma seu salvamento, salvá-la no banco de dados ao invés da sessão exigiria um controle de confirmação de salvamento da análise, que é desnecessário com o salvamento em sessão.

4.3.3 Funcionalidade de Geração de Relatórios

A segunda grande funcionalidade da ferramenta desenvolvida é a criação de relatórios. Após a criação das análises o usuário pode criar relatórios escolhendo quais análises ele deseja usar, a disposição das mesmas e pode inclusive adicionar textos, imagens e *links* externos.

Do ponto de vista do usuário a criação de um relatório se inicia de forma semelhante à criação de uma análise. O usuário seleciona o *survey* para o qual ele deseja criar relatórios e é direcionado para a página de análises e relatórios, que lista as análises e os relatórios já criadas e permite a criação ou edição dos mesmos.

Clicando no botão de novo relatório o usuário é direcionado à página de edição de relatórios. Esta página possui quatro componentes: O campo de nome do relatório, o de privacidade, i.e., se ele será público ou privado, o editor de relatórios e a lista de análises do usuário. A figura Visão Edição de Relatórios mostra esta página com os dois campos de configuração na parte superior, o editor ao centro, ocupando a maior parte da tela e a lista de análises à direita.

Figura 15 Visão Edição de Relatórios

O editor usado é o [TinyMCE](#), que é um editor *richtext* What-You-See-Is-What-You-Get (WYSIWYG) *open-source*, sob a licença LGPL, desenvolvido em javascript pela Moxiecode Systems AB.

O editor foi escolhido devido a sua grande quantidade de recursos, capacidade de customização e integração com o jQuery – biblioteca javascript já usada na plataforma Polvo. Com ele é possível definir tamanho, cor e fonte de texto; criação de tabelas, listas numeradas ou com *bullets*; inserção de textos sobre e subscritos, imagens e *links*; formatação de textos com negrito, itálico e sublinhado; alinhamento do texto à esquerda, à direita, centralizado ou mesmo justificado; entre muitas outras.

Além disso, o TinyMCE aceita que imagens sejam arrastadas para dentro do editor, inserindo automaticamente a imagem ao texto. Como saída de dados o editor produz um HTML.

O segundo componente da visão é a lista de análises. A lista mostra os resultados de todas as análises criadas pelo usuário. Para adicionar uma análise ao relatório o usuário só precisa arrastar essa imagem até a posição desejada e redimensioná-la conforme desejado. Resultados textuais podem ser copiados e colados no editor ou mesmo arrastados.

Quando o usuário desejar ele pode salvar o relatório, que será listado na lista de relatórios já mencionada. A partir da lista o usuário pode editar o relatório, voltando a página de edição ou pode visualizar o relatório criado, como uma página HTML ou ainda baixar o relatório em PDF.

Para fornecer a visão de edição de relatórios o sistema deve executar todas as análises criadas pelo usuário para poder listá-las. A edição é toda controlada pelo TinyMCE portanto a ferramenta só precisa atuar no salvamento do relatório. Ele é feito recebendo o nome, a privacidade e o relatório em si, em HTML. Os dados são salvos em uma tabela de relatórios no banco de dados.

Conforme mencionado anteriormente existem duas forma de visualização do relatório. A primeira é via HTML. Para fornecer essa visualização o sistema deve, antes de mostrar o HTML, verificar se o *survey* já esta finalizado. Em caso negativo a ferramenta deve executar as funções pertencentes ao relatório para atualizá-las. Isso garante que as análise estão usando os dados mais atualizados possível. Já caso o *survey* estiver finalizado pode-se aproveitar os últimos resultados gerados, caso eles existam.

Vale notar que as funções que geram resultados em modo texto não são atualizadas dentro do relatório, pois não é possível diferenciar um texto que foi copiado de um resultado de uma análise de outro que foi copiado de outra fonte.

A segunda forma de visualização é por PDF. O arquivo é criado pelo PHP a partir do HTML do relatório usando a ferramenta [dompdf](#) que converte código HTML em PDF. O resultado então pode ser baixado pelo usuário. O PDF gerado não é salvo pelo sistema, portanto, caso o usuário queira disponibilizá-lo a outras pessoas ele deve fazê-lo por si próprio.

5 Considerações Finais

O projeto foi iniciado em meados de 2010 e desenvolvido intermitentemente até meados de 2011. No seu decorrer foi desenvolvida uma ferramenta capaz de facilitar a análise estatística de respostas de *surveys* online além de permitir a composição de relatórios de tais análises. Assim todos os objetivos propostos foram alcançados.

A principal dificuldade durante o desenvolvimento do projeto foi a implementação de funções R devido à total falta de experiência anterior com a linguagem.

Existem alguns *bugs* conhecidos no sistema que não foram corrigidos a tempo da entrega deste trabalho. Esses *bugs* não impedem o uso completo do sistema mas afetam de forma negativa a experiência do usuário. A correção dos mesmos será feita posteriormente.

Além disso, algumas características do sistema podem ser melhoradas e novas funcionalidades implementadas.

As funções de análise implementadas no desenvolvimento deste projeto suprem as necessidades básicas de análises expostas no requisitos do sistema. Entretanto, outras análises, sem funções implementadas, podem ser necessárias.

Porém, adicionar novas funções à ferramenta exige o conhecimento do funcionamento do armazenamento das funções no banco de dados. Assim,

deve existir um meio – seja um formulário ou uma ferramenta de importação de funções – de facilitar a adição de novas funções ao sistema.

Durante o desenvolvimento do projeto com implementada uma integração à ferramenta de *survey* do Polvo. Essa integração facilita o uso do módulo estatístico pelos usuário do módulo Survey, mas impede o uso da ferramenta com dados externos, por isso pode-se implementar a funcionalidade de importação de dados externos ao sistema.

O foco do sistema foi a criação de gráficos que estão perfeitamente integrados em todas as etapas de análise e criação de relatórios. Porém, os resultados não gráficos das análises não contam com esta perfeita integração. Não existe, por exemplo, um tratamento específico para tabelas, que venham a ser geradas por uma análise. Assim, tal tratamento pode ser melhorado.

Por fim, o design do sistema pode ser melhorado. Mais especificamente o uso de Ajax pode ser feito para melhorar a usabilidade do sistema e a velocidade do sistema eliminando a troca de páginas durante a criação de análises.

Assim, sucintamente, os trabalhos futuros propostos são:

- Facilitação da inserção de novas funções R;
- Implementação de novas funções R;
- Implementação de funcionalidade de importação de dados externos ao Polvo;

- Melhorar o tratamento de resultados não gráficos da ferramenta, principalmente de tabelas;
- Melhorar o design do sistema através do uso de Ajax.

6 Referências

SCHEUREN, F. *What is Survey*. American Statistical Association, 2004. 68p.

ZEND TECHNOLOGIES. *Programmer's Reference Guide*, Disponível em: <<http://framework.zend.com/manual/en/learning.quickstart.intro.html>>. Acesso em 18 de nov. 2010.

ZEND TECHNOLOGIES. *Zend_Db_Adapter*, Disponível em: <<http://framework.zend.com/manual/en/zend.db.adapter.html>>. Acesso em 19 de nov. 2010.

PHP.net. *PHP Data Objects*, Disponível em: <<http://www.php.net/manual/en/book.pdo.php>>. Acesso em 19 de nov. 2010.

REENSKAUG, Trygve. *THING-MODEL-VIEW-EDITOR: an Example from a planning system*, 1979.

REENSKAUG, Trygve. *MODELS – VIEWS – CONTROLLERS*, 1979.

SHAN, T. C. HUA, W. W. *Taxonomy of Java Web Application Frameworks*, Proceedings of the IEEE International Conference on e-Business Engineering, 2006, ISBN: 0-7695-2645-4, p. 378-385, publisher:IEEE Computer Society.

Paul Teetor. *R Cookbook*. O'Reilly, first edition, 2011. ISBN: 978-0-596-80915-7.

VANCE, A. Data Analysts Captivated by R's Power. *New York Times*, Nova Iorque, p. B6, 07 jan. 2009.

HORNIK, K. *The R FAQ*. Disponível em: <<http://CRAN.R-project.org/doc/FAQ/R-FAQ.html>>. Acesso em 28 de ago. 2010.

ANSCOMBE, F. J. Graphs in Statistical Analysis. *The American Statistician*, v. 27, n. 1, p. 17-21, 1973.

BANFIELD, J. Rweb:Web-based Statistical Analysis, *Journal of Statistical Software*, v. 4, n. 1, p. 1-15, 1999.

JOCKERS, M. L. Executing R in Php, Disponível em: <<https://www.stanford.edu/~mjockers/cgi-bin/drupal/node/25>>. Acesso em 22 de set. 2010.

COHEN, E. TURF analysis. *QUIRK'S Marketing Research Review*. 1993.