

UNIVERSIDADE DE BRASÍLIA  
DEPARTAMENTO DE ESTATÍSTICA  
CURSO DE ESTATÍSTICA

Thiago Pfaffmann Diniz Moreti

*Shiny-R:*

Uma aplicação na análise da avaliação docente da Universidade de  
Brasília

Brasília  
2015

Thiago Pfaffmann Diniz Moreti

*Shiny-R:*

Uma aplicação na análise da avaliação docente da Universidade de  
Brasília

Relatorio apresentado ao Curso de Estatística da  
UNB, como requisito para a obtenção do grau de  
BACHAREL em Estatística.

**Orientador: Eduardo Monteiro de Castro Gomes**

**Doutor em Estatística Experimental - USP**

Brasília

2015

Thiago Pfaffmann Diniz Moreti

*Shiny-R:*

Uma aplicação na análise da avaliação docente da Universidade de  
Brasília

Relatorio apresentado ao Curso de Estatística da  
UNB, como requisito para a obtenção do grau de  
BACHAREL em Estatística.

Aprovado em

**BANCA EXAMINADORA**

---

Eduardo Monteiro de Castro Gomes

Doutor em Estatística Experimental - USP

---

Donald Matthew Pianto

Doutor em Matemática Computacional - UFPE

---

Luís Gustavo do Amaral Vinha

Mestre em Estatística - USP

## **Agradecimentos**

A Deus, porque se cheguei até esse momento é por honra e glória d'Ele. Aos meus pais e irmã que sempre me apoiaram nos momentos mais difíceis. E a minha namorada, Laryssa, que sempre me incentivou quando quis desistir.

*“Bem-aventurado o homem que suporta com paciência a provação! Porque, uma vez provado, receberá a coroa da vida, que o Senhor prometeu aos que o amam”.*

*Tiago 1, 12*

## Resumo

A avaliação docente, segundo o relatório de Autoavaliação Institucional 2013, tem subsidiado os departamentos e as faculdades no planejamento e na distribuição das disciplinas, além de ser utilizada como um instrumento de avaliação para fins de validação do Estágio Probatório, Progressão na Carreira Docente, e como forma de avaliação dos cursos pelos avaliadores externos do MEC. Então, tendo em vista a importância da avaliação docente para a Universidade de Brasília e que é papel do estatístico dar um bom tratamento aos dados e apresentá-los de forma clara e útil, elaborou-se um programa que permite a Universidade realizar a análise do questionário de avaliação docente do MatrículaWeb, dando foco às questões dissertativas do questionário, pois tal análise ainda não é feita pela UnB, sendo, dessa forma, uma novidade no âmbito da avaliação institucional. O presente estudo utilizou o pacote *Shiny* do *software* estatístico *R-Project*, pois apresenta inúmeras vantagens, como o fato de ser gratuito, apresentar uma interface simples e atraente, permitir a criação de várias análises estatísticas e além do grande diferencial de que não é necessário o conhecimento de código de programação para realizar a análise por meio do programa criado.

**Palavras-chave:** Shiny, Expressões Regulares, Text Mining, Avaliação Docente

# Sumário

<b>1</b>	<b>Introdução</b>	<b>6</b>
<b>2</b>	<b>Revisão Literária</b>	<b>9</b>
2.1	Expressões Regulares . . . . .	9
2.1.1	Sintaxe de Regex no R-Project . . . . .	9
2.1.2	Funções de Regex no R-Project . . . . .	12
2.2	<i>Text Mining</i> . . . . .	13
2.2.1	Tipos de Abordagens de Dados . . . . .	16
2.2.2	Preparação dos Dados . . . . .	16
2.2.3	Indexação e Normalização . . . . .	16
2.2.4	Cálculo da Relevância . . . . .	18
2.2.5	Seleção de Termos . . . . .	19
2.2.6	Análise de Resultados . . . . .	19
2.3	<i>Shiny</i> . . . . .	19
2.3.1	Interface do Usuário (UI) . . . . .	20
2.3.2	Server . . . . .	23
<b>3</b>	<b>Metodologia</b>	<b>25</b>
3.1	Materiais . . . . .	25
3.2	Métodos . . . . .	27
<b>4</b>	<b>Resultados e Discussões</b>	<b>29</b>
4.1	Tutorial . . . . .	30
4.2	Análise de Frequência . . . . .	31

4.2.1	Respostas . . . . .	31
4.2.2	Palavras . . . . .	36
4.3	Análise das Respostas . . . . .	40
4.3.1	Unidade Acadêmica . . . . .	40
4.3.2	Professor . . . . .	45
4.3.3	Disciplina . . . . .	48
4.3.4	Aluno . . . . .	51
<b>5</b>	<b>Considerações Finais</b>	<b>55</b>
	<b>Referências Bibliográficas</b>	<b>57</b>
<b>I</b>	<b>Anexo - Questionário de percepção discente sobre a disciplina, o desempenho docente e as condições de oferta (Modelo da Universidade de Brasília)</b>	<b>60</b>
<b>II</b>	<b>Anexo - Programação</b>	<b>64</b>



# 1 Introdução

A história da avaliação dos docentes pelos discentes, na Universidade de Brasília, vem desde a década de 1980. Em 1986, a elaboração da Proposta de Avaliação de Ensino Superior constituiu como o primeiro marco, quando todos os cursos de graduação da UnB foram avaliados. A partir de 2003, a Universidade começou realizar a avaliação dos docentes por meio de formulários padronizados que eram aplicados em sala de aula.

Em abril de 2004, foi criado pelo MEC o Sistema Nacional de Avaliação da Educação Superior (SINAES) que, segundo a Lei nº 10.861 de 2004, tem o seguinte objetivo:

Art.1º Fica instituído o Sistema Nacional de Avaliação da Educação Superior - SINAES, com o objetivo de assegurar processo nacional de avaliação das instituições de educação superior, dos cursos de graduação e do desempenho acadêmico de seus estudantes, nos termos do art 9º, VI, VIII e IX, da Lei no 9.394, de 20 de dezembro de 1996.

Com base no Art. 11 da mesma lei, a UnB constituiu a Comissão Própria de Avaliação (CPA) que tem como objetivo conduzir os processos de avaliação internos, de sistematização e de coleta de informações da instituição, nos quais se encontra a avaliação dos docentes pelos discentes. Assim, desde 2006, a Universidade utiliza os dados obtidos por meio dos questionários para elaborar o relatório de avaliação institucional, segundo as normas do SINAES.

A partir do segundo semestre de 2011, o novo formulário de avaliação dos docentes começou a ser disponibilizado na internet através do MatriculaWeb. Ele é de preenchimento opcional e apresenta um número reduzido de questões, que são divididas em quatro categorias: disciplina, professor, autoavaliação e apoio institucional. Em cada uma das categorias, encontra-se certo número de questões categorizadas, além de um espaço aberto para comentários dos pontos fortes e pontos a melhorar.

Essa avaliação se baseia na perspectiva do corpo discente, pois é ele quem convive diariamente com a realidade do ensino na universidade, sendo assim, o questionário é um ótimo “*feedback*” para a instituição e para o professor, ressaltando que, para isso, é necessário um comprometimento do aluno no preenchimento do formulário. Logo, a

importância disso tudo não é meramente administrativa, ou algo para ficar somente em um relatório, mas sim um mecanismo para se encontrar os defeitos das Instituições de Ensino Superior - IES e chegar às soluções cabíveis, sempre visando a melhora do ensino de graduação e, conseqüentemente, gerar a satisfação de seus alunos.

A Universidade é ampla e assim também é a diversidade entre seus docentes. Por muitas vezes, os professores apresentam um grande domínio do conteúdo, mas uma didática não tão boa ou vice-versa. Então, com o auxílio dos resultados obtidos no questionário de avaliação, o professor pode encontrar os seus pontos fracos que devem ser melhorados. Assim, é fundamental que o resultado da avaliação seja utilizado pelos colegiados, pois ela pode ser “um poderoso processo ao serviço da melhoria da qualidade pedagógica e da qualidade de ensino dos professores, gerando ambientes propícios à inovação, ao desenvolvimento profissional e, conseqüentemente, à melhoria das aprendizagens dos alunos” (FERNANDES, 2008).

Desse modo, quando a universidade utiliza dessa ferramenta de avaliação, ela é capaz de encontrar medidas para ofertar um maior conforto e um ensino de qualidade, garantindo uma maior satisfação de seus alunos, pois diminui o desencontro entre a expectativa do estudante e o que a instituição oferece. A definição de satisfação representa “o sentimento de prazer ou de desapontamento resultante da comparação do desempenho esperado pelo produto (ou resultado) em relação às expectativas da pessoa” (KOTLER, 1998), e, quando essa satisfação não é alcançada, pode interferir no nível de interação do estudante com a instituição, podendo implicar nas decisões de permanecer ou não na universidade (SCHLEICH; POLYDORO; SANTOS, 2006).

Para o presente trabalho, tem-se como objeto de estudo as respostas abertas do questionário de avaliação. Essas questões são de grande importância, pois é onde o aluno tem a liberdade para manifestar os pontos fortes e os a melhorar das quatro categorias do formulário que não puderam ser contempladas por algum motivo nas perguntas categorizadas. Porém, apesar de atualmente essas questões estarem no questionário, elas ainda não são analisadas pela Universidade de Brasília, como se pode constatar através do relatório de autoavaliação institucional 2013.

Então, diante da relevância levantada a cerca da avaliação institucional para a Universidade de Brasília, o presente estudo buscou criar um produto para gerar e facilitar essa análise do questionário de avaliação docente no MatriculaWeb ou, como foi

denominado pela Câmara de Ensino de Graduação (CEG), “Questionário de percepção discente sobre a disciplina, o desempenho docente e as condições de oferta” (ANEXO I). O produto é um programa de análise estatística das perguntas dissertativas do questionário de avaliação docente, que é o foco do trabalho, mas também apresenta campos de análise das perguntas categorizadas.

Para a elaboração do aplicativo foi utilizado um pacote do *R-project* chamado *Shiny*, que permite a criação de aplicativos interativos na *web*. Mas, devido ao fato de se estar trabalhando com uma base de dados textual, foi necessário a busca de algumas técnicas que permitissem a manipulação e organização desses dados. As técnicas foram as seguintes: expressões regulares (*regular expression*) e mineração de texto (*Text Mining*).

A vantagem do uso desse programa é que a Universidade de Brasília poderá realizar suas análises de forma fácil, simples, interativa e gratuita, pois o aplicativo *Shiny* permite que toda a análise estatística possa ser feita por um usuário que não possui o conhecimento de programação em um *software* e, além disso, o *R-Project* é livre e gratuito, disponível para *download* em seu site (<http://www.r-project.org/>). Portanto, por meio desse produto a UnB poderá fornecer um retorno para seus professores, alunos e departamentos, buscando atingir um ensino de excelência, sendo sempre uma referência de educação superior no Brasil.

## 2 Revisão Literária

### 2.1 Expressões Regulares

Atualmente, no ambiente dos computadores, depara-se muito com dados textuais, seja nas páginas da *web*, nos e-mails, documentos, etc. Os textos são compostos por sequências de caracteres que podem ser combinados com as expressões regulares (WATT, 2005). Muitas vezes, esses textos possuem uma grande dimensão e, por isso, necessita-se de um mecanismo para identificar alguns padrões específicos.

Então, a expressão regular (*Regular Expression* - regex) é um método formal para identificar ou não sequências de caracteres nos textos, ou seja, padrões textuais. Ela é interpretada como uma regra que, ao casar com uma entrada de texto, retornará um sucesso (JARGAS, 2012). Para utilizá-la é necessário o uso de um aplicativo ou uma linguagem de *script*.

Os padrões de expressões regulares consistem basicamente em uma combinação de caracteres especiais e alfanuméricos. A regex pode ser formada por combinações simples de apenas um ou uma série de caracteres. Logo, para construir expressões regulares, devem-se usar vários operadores para combinar pequenas expressões (SANCHEZ, 2013).

#### 2.1.1 Sintaxe de Regex no R-Project

São várias as linguagens de *script* para trabalhar com expressões regulares, por exemplo, Java, Perl, Python, etc. O R-Project também possui diversas funções de regex, porém, é necessário antes tratar de alguns elementos de expressões regulares (SANCHEZ, 2013).

#### Metacaracteres

Existem caracteres alfanuméricos que são expressões regulares e representações de si próprio, porém, existem caracteres especiais que possuem um *status* próprio, esses são chamados de metacaracteres. Para representá-los de forma literal no R, é necessário

utilizar duas barras `\\`. Os metacaracteres podem ser vistos pela tabela 2.1.

Tabela 2.1: Relação de metacaracteres e o modo de contornar o R para o usá-los de forma literal

Metacharacter	Literal meaning	Escape in R
.	the period or dot	<code>\\. </code>
\$	the dollar sign	<code>\\\$ </code>
*	the asterisk or star	<code>\\* </code>
+	the plus sign	<code>\\+ </code>
?	the question mark	<code>\\? </code>
	the vertical bar or pipe symbol	<code>\\  </code>
\\	the backslash	<code>\\\\ </code>
^	the caret	<code>\\^ </code>
[	the opening square bracket	<code>\\[ </code>
]	the closing square bracket	<code>\\] </code>
{	the opening curly bracket	<code>\\{ </code>
}	the closing curly bracket	<code>\\} </code>
(	the opening round bracket	<code>\\( </code>
)	the closing round bracket	<code>\\) </code>

Fonte: Sanchez, 2013

## Sequências

*Sequências* definem uma sequência de caracteres correspondente a algo, como por exemplo, a *sequência* `\\d` corresponde aos dígitos (números). A relação de algumas *sequências* podem ser vistas na tabela 2.2.

Tabela 2.2: Relação de *sequências* usuais no R

Anchor	Description
<code>\\d</code>	match a digit character
<code>\\D</code>	match a non-digit character
<code>\\s</code>	match a space character
<code>\\S</code>	match a non-space character
<code>\\w</code>	match a word character
<code>\\W</code>	match a non-word character
<code>\\b</code>	match a word boundary
<code>\\B</code>	match a non-(word boundary)
<code>\\h</code>	match a horizontal space
<code>\\H</code>	match a non-horizontal space
<code>\\v</code>	match a vertical space
<code>\\V</code>	match a non-vertical space

Fonte: Sanchez, 2013

## Classe de caracteres

Uma *classe de caracteres* é uma lista de caracteres dentro de colchetes [ ], que são usadas para identificar somente um elemento da lista dentre inúmeros caracteres. Exemplos de *classe de caracteres* estão na tabela 2.3.

Tabela 2.3: Relação de *classe de caracteres*

Anchor	Description
[aeiou]	match any one lower case vowel
[AEIOU]	match any one upper case vowel
[0123456789]	match any digit
[0-9]	match any digit (same as previous class)
[a-z]	match any lower case ASCII letter
[A-Z]	match any upper case ASCII letter
[a-zA-Z0-9]	match any of the above classes
[^aeiou]	match anything other than a lowercase vowel
[^0-9]	match anything other than a digit

Fonte: Sanchez, 2013

## POSIX classe de caracteres

A POSIX define algumas classes de caracteres utilizadas para fornecer um acesso padrão a alguns tipos de caracteres específicos, como pode se ver na tabela 2.4.

Tabela 2.4: Relação de POSIX utilizadas no R

Class	Description
[:lower:]	Lower-case letters
[:upper:]	Upper-case letters
[:alpha:]	Alphabetic characters ([:lower:] and [:upper:])
[:digit:]	Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
[:alnum:]	Alphanumeric characters ([:alpha:] and [:digit:])
[:blank:]	Blank characters: space and tab
[:cntrl:]	Control characters
[:punct:]	Punctuation characters: ! " # % & ' ( ) * + , - . / : ;
[:space:]	Space characters: tab, newline, vertical tab, form feed, carriage return, and space
[:xdigit:]	Hexadecimal digits: 0-9 A B C D E F a b c d e f
[:print:]	Printable characters ([:alpha:], [:punct:] and space)
[:graph:]	Graphical characters ([:alpha:] and [:punct:])

Fonte: Sanchez, 2013

## Quantificadores

Os *quantificadores* são elementos importantes no uso das expressões regulares, pois eles determinam a quantidade de elementos de um critério que deve ser correspondidos. A tabela 2.5 mostra os *quantificadores*.

Tabela 2.5: Relação de *quantificadores* utilizadas no R

Quantifier	Description
?	The preceding item is optional and will be matched at most once
*	The preceding item will be matched zero or more times
+	The preceding item will be matched one or more times
{n}	The preceding item is matched exactly n times
{n,}	The preceding item is matched n or more times
{n,m}	The preceding item is matched at least n times, but not more than m times

Fonte: Sanchez, 2013

### 2.1.2 Funções de Regex no R-Project

Depois de apresentados os elementos das expressões regulares, serão tratadas algumas funções importantes quando se fala em regex no R (SANCHEZ, 2013).

#### Principais funções

As principais funções do R de expressões regulares são descritas na tabela 2.6.

Tabela 2.6: Relação das principais funções regex do R

Function	Purpose	Characteristic
<code>grep()</code>	finding regex matches	which elements are matched (index or value)
<code>grepl()</code>	finding regex matches	which elements are matched (TRUE & FALSE)
<code>regexpr()</code>	finding regex matches	positions of the first match
<code>gregexpr()</code>	finding regex matches	positions of all matches
<code>regexec()</code>	finding regex matches	hybrid of <code>regexpr()</code> and <code>gregexpr()</code>
<code>sub()</code>	replacing regex matches	only first match is replaced
<code>gsub()</code>	replacing regex matches	all matches are replaced
<code>strsplit()</code>	splitting regex matches	split vector according to matches

Fonte: Sanchez, 2013

As cinco primeiras funções citadas na tabela 2.6 tem como objetivo principal encontrar padrões, já as duas próximas servem para substituir um caractere por outro

de interesse e a última função é usada para dividir um caractere em uma sequência de caracteres.

### Funções de Regex em *stringr*

O R possui um pacote chamado *stringr* que necessita ser instalado, esse pacote apresenta uma série de operadores de regex (ver tabela 2.7).

Tabela 2.7: Relação das funções regex do pacote *stringr* do R

Function	Description
<code>str_detect()</code>	Detect the presence or absence of a pattern in a string
<code>str_extract()</code>	Extract <b>first</b> piece of a string that matches a pattern
<code>str_extract_all()</code>	Extract <b>all</b> pieces of a string that match a pattern
<code>str_match()</code>	Extract <b>first</b> matched group from a string
<code>str_match_all()</code>	Extract <b>all</b> matched groups from a string
<code>str_locate()</code>	Locate the position of the <b>first</b> occurrence of a pattern in a string
<code>str_locate_all()</code>	Locate the position of <b>all</b> occurrences of a pattern in a string
<code>str_replace()</code>	Replace <b>first</b> occurrence of a matched pattern in a string
<code>str_replace_all()</code>	Replace <b>all</b> occurrences of a matched pattern in a string
<code>str_split()</code>	Split up a string into a variable number of pieces
<code>str_split_fixed()</code>	Split up a string into a fixed number of pieces

Fonte: Sanchez, 2013

Todas elas apresentam uma característica em comum, que são os dois primeiros argumentos. O primeiro argumento é um vetor de *strings*, que é o que deve ser processado, e o segundo é um único padrão a ser combinado.

## 2.2 *Text Mining*

A mineração de texto (MT) é uma técnica de análise de informação textual que tem como objetivo primário encontrar padrões interessantes nos bancos de dados textuais (AGGARWAL; ZHAI, 2012). Então, segundo Feldman e Sanger (2007), MT pode ser definido como um processo intensivo onde o usuário interage com um conjunto de documentos ao longo do tempo usando um conjunto de ferramentas de análise, onde procura extrair informações úteis a partir de fontes de dados através da identificação e exploração de padrões interessantes.

Já segundo Hotho (2005), pode-se definir *text mining* de forma diferente, dependendo da perspectiva da área em que será utilizada, como:



- *Text Mining* = Extração de Informação: Essa primeira abordagem assume que a mineração corresponde a extração de informações em textos. Essa informação pode ser obtida de uma palavra, frase ou uma passagem.
- *Text Mining* = *Text Data Mining*: Nessa abordagem a mineração pode ser definida similar à Mineração de Dados, onde serão aplicados algoritmos e métodos estatísticos para a descoberta de algum padrão relevante nos dados.
- *Text Mining* = KDD (*Knowledge Discovery in Databases*): Essa abordagem segue a ideia da Descoberta de Conhecimento em Dados Estruturados, onde pode ser resumido como a extração de informações não descobertas em coleções de textos.

Sem dúvida, a mineração de texto se inspira na mineração de dados. Ambos os processos apresentam rotinas de pré-processamentos, algoritmos e elementos de apresentação. Porém, devido a sua maior complexidade, a mineração também se baseia em avanços de outras disciplinas da ciência da computação preocupadas com a manipulação de linguagem natural. A MT explora técnicas e metodologias das áreas de recuperação de informação, extração de informações, e linguística computacional de corpus (FELDMAN; SANGER, 2007).

As principais contribuições do *text mining* têm sido para a análise qualitativa de grandes bases textuais e para uma melhor compreensão dos textos disponíveis em documentos. Uma coisa que deve ficar clara, pois ainda há muita confusão quanto a isso, é que quando um usuário utiliza a ferramenta de MT, ele não a utiliza como uma ferramenta de busca em certo texto, mas sim para a análise de um documento.

De forma geral, segundo Feldman e Sanger (2007), as etapas de mineração de texto podem ser descritas como operações de pré-processamento, documentos processados, núcleo de mineração e apresentação, como mostra a Figura 2.1.

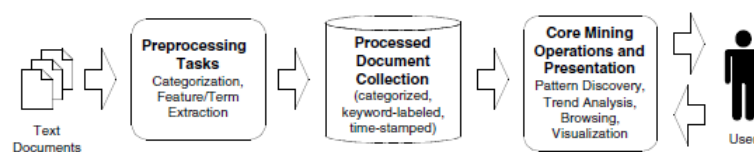


Figura 2.1: Etapas da Mineração de Texto. (Fonte: FELDMAN; SANGER, 2007)

Segundo Azevedo (2011), o pré-processamento mostrado por Feldman e Sanger (2007) corresponde à identificação e extração de características representativas dos documentos e, assim, essa operação transforma dados não estruturados em uma estrutura expressa em um modelo intermediário, que é baseado na escolha da unidade mínima do texto.

Já as operações que compõem o núcleo de mineração são: descoberta de padrões, análise de tendências e algoritmos incrementais para descoberta de conhecimento. Os mecanismos mais utilizados são as distribuições, proporções, associações e conjunto de conceitos frequentes. E, por fim, a apresentação dos resultados corresponde a interface do sistema (AZEVEDO, 2011).

Mas essas etapas da mineração podem sofrer alterações em alguns casos, como apresentado por Moraes e Ambrósio (2007), conforme a Figura 2.2. Elas serão descritas sucintamente a seguir.

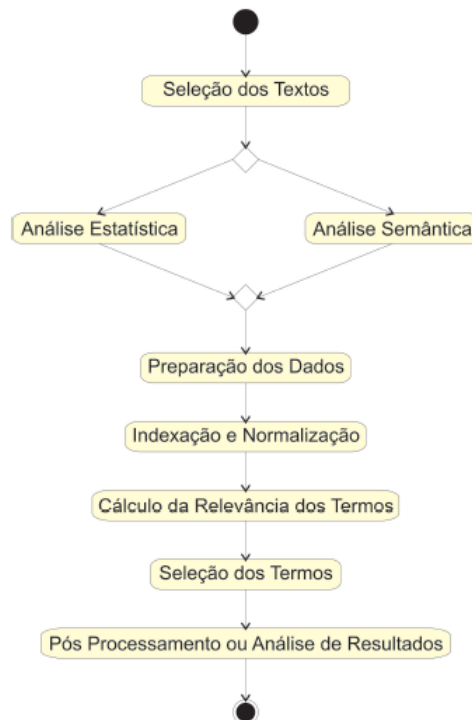


Figura 2.2: Etapas da Mineração de Texto. (Fonte: MORAIS; AMBRÓSIO, 2007)

### 2.2.1 Tipos de Abordagens de Dados

Existem dois tipos de abordagens de dados, segundo Ebecken (2003), citado por Morais e Ambrósio (2007), o primeiro é a *análise semântica* e o segundo é a *análise estatística*. A análise semântica é baseada na funcionalidade dos termos encontrados nos documentos textuais e é necessário o conhecimento morfológico, sintático, semântico, etc. E a análise estatística é baseada na frequência dos termos encontrados nos documentos textuais, ou seja, sua importância é dada de acordo com o número de vezes que certa palavra aparece no texto.

### 2.2.2 Preparação dos Dados

Essa etapa envolve a seleção dos dados que constituem a base de textos e o grupo que melhor expressa o conteúdo dos mesmos, com isso, obtêm-se uma redução na dimensão dos dados textuais. Mas o objetivo principal dessa etapa é identificar similaridades que é feita pela *função de similaridade*. Essa função busca encontrar uma relação entre os termos da consulta e os termos dos documentos.

### 2.2.3 Indexação e Normalização

Essa fase tem como objetivo principal facilitar a identificação de palavras no texto que possuem significados similares, considerando a variação morfológica e os problemas de sinonímias, ou seja, pretende facilitar a identificação de similaridade. O resultado dessa etapa é a criação de um índice, onde são colocadas as características identificadas nos documentos.

O texto pode ser indexado de duas formas: uma utilizando os termos técnicos (vocabulários) da área de interesse e a outra feita manualmente, onde a pessoa encarregada da indexação vai analisar todos os documentos e encontrar palavras-chave. Essas palavras, ao serem adicionadas ao índice, recebem o nome de *termos de índices*.

As principais fases da indexação e normalização são mostradas na Figura 2.3, elas têm como objetivo reduzir a dimensão do banco de dados, dando ênfase na carga semântica que a palavra possui.

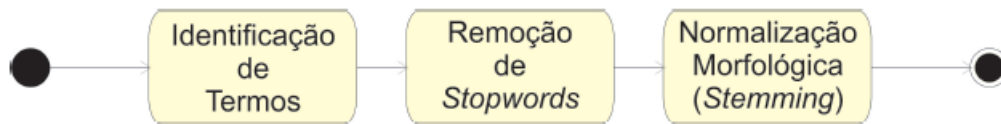


Figura 2.3: Etapas do processo de indexação. (Fonte: MORAIS; AMBRÓSIO, 2007)

### Identificação de Termos

Essa fase consiste principalmente na identificação de termos presentes nos documentos, sejam eles simples ou compostos. (MORAIS; AMBRÓSIO, 2007)

- **Identificação de termos simples:**

É feita a aplicação de um analisador léxico que serve para encontrar nos documentos as palavras de termos simples e retirar os símbolos e os caracteres de controle de arquivo ou de formatação.

Nessa fase pode ser feita a correção de ortografia e de sinônimos, converter as letras todas em maiúscula ou minúscula, padronizar as datas, retirar os hífen, entre outras coisas.

- **Identificação de termos compostos:**

Essa fase também é conhecida por *Word-phrase formation*, ela busca identificar palavras que apresentam diferenças nos seus significados quando são combinadas com uma outra palavra.

O processo de identificação consiste em utilizar um dicionário de expressões. Quando for adicionar essas expressões no índice, deve-se também adicionar as palavras individualmente, para evitar erro na hora de localizar certas palavras.

### Remoção de *Stopwords*

*Stopwords* são as palavras que não possuem relevância textual, como por exemplo as palavras que pertencem à classe gramatical dos pronomes, artigos, preposições, advérbios e outras classes de palavras auxiliares. Essa fase, portanto, consiste na retirada das *stopwords* do banco de dados textual, além de outras palavras que possuem uma

frequência muito grande, pois elas não são capazes de discriminar documentos (MORAIS; AMBRÓSIO, 2007).

As palavras classificadas como *stopwords* formam uma lista chamada *stoplist*, essas palavras não devem ser adicionadas ao índice, pois resultariam em um índice muito extenso e com informações desnecessárias.

### Normalização Morfológica (*stemming*)

Essa fase conhecida como *stemming*, consiste em reduzir uma palavra ao seu radical, pois muitas vezes é de interesse eliminar certas variações morfológicas de uma palavra, assim, é identificado o radical de uma palavra e é feita a eliminação de seu sufixo e prefixo. Dessa forma, elimina-se o gênero, número e grau das palavras. O objetivo, portanto, é diminuir o tamanho do índice, porém poderá gerar alguns problemas no processo de busca.

### 2.2.4 Cálculo da Relevância

As palavras presentes nos textos não apresentam o mesmo grau de importância (relevância), em geral, os termos que são observados com maior constância devem apresentar uma importância maior. Então, o cálculo de relevância de uma palavra pode ser obtido através de sua frequência no texto. Existem outras análises do grau de relacionamento da palavra com o texto, que também recebem o nome de peso, porém a frequência é a mais simples e, logo, a mais usual.

Algumas formas do cálculo do peso são: *frequência absoluta e frequência relativa*:

#### Frequência absoluta

Essa é a medida mais simples de frequência, ela mede a quantidade de vezes em que um termo aparece no documento ( $n_i$ ).

$$Freq_{abs} = n_i \quad (2.1)$$

## Frequência relativa

Essa medida leva em consideração a quantidade de palavras que um documento possui ( $n$ ). Para o cálculo da frequência relativa ( $f_i$ ) é necessário dividir a frequência absoluta ( $Freq_{abs}$ ) pela quantidade total de palavras do documento ( $n$ ).

$$f_i = \frac{Freq_{abs}}{n} \quad (2.2)$$

### 2.2.5 Seleção de Termos

Essa é justamente a etapa onde se seleciona as palavras retiradas do texto, após ter sido feito o pré-processamento e cálculo de relevância. Essa técnica é baseada principalmente no peso que cada termo do texto recebeu ou na posição sintática em relação ao texto.

### 2.2.6 Análise de Resultados

Por fim, nessa fase, são aplicadas técnicas de análise dos resultados de um *Sistema de Recuperação de Informações* (SRI). Tal análise pode ser realizada com base em técnicas de uma área conhecida como bibliometria, uma sub-área da biblioteconomia que é encarregada de estudar e aplicar métodos estatísticos e matemáticos em documentos textuais. Essas métricas podem ser utilizadas no SRI para testar se o mecanismo funcionou como deveria.

## 2.3 *Shiny*

O *Shiny* é um pacote do *R-project* que torna fácil a construção de aplicações interativas na *web*, permitindo o compartilhamento de análises e gráficos feitos pelo *software*. Assim, o usuário poderá interagir de forma rápida e simples com as análises sem mesmo saber programar.

Todo programa *Shiny* apresenta sempre a mesma estrutura, ou seja, apresenta sempre dois *scripts* (componentes) que são salvos juntos em um diretório do R. Esses componentes são:

- a interface do usuário (*user-interface script*) que é denominada `ui.R`;
- e o servidor (*server script*) recebe o nome de `server.R`.

Então, para criar um aplicativo no *Shiny*, basta criar um novo diretório no R (*new directory*) e salvar os dois *scripts*, `ui.R` e `server.R`, dentro desse diretório. Para rodar o aplicativo, basta apertar no botão *Run app*, como mostra na Figura 2.4.

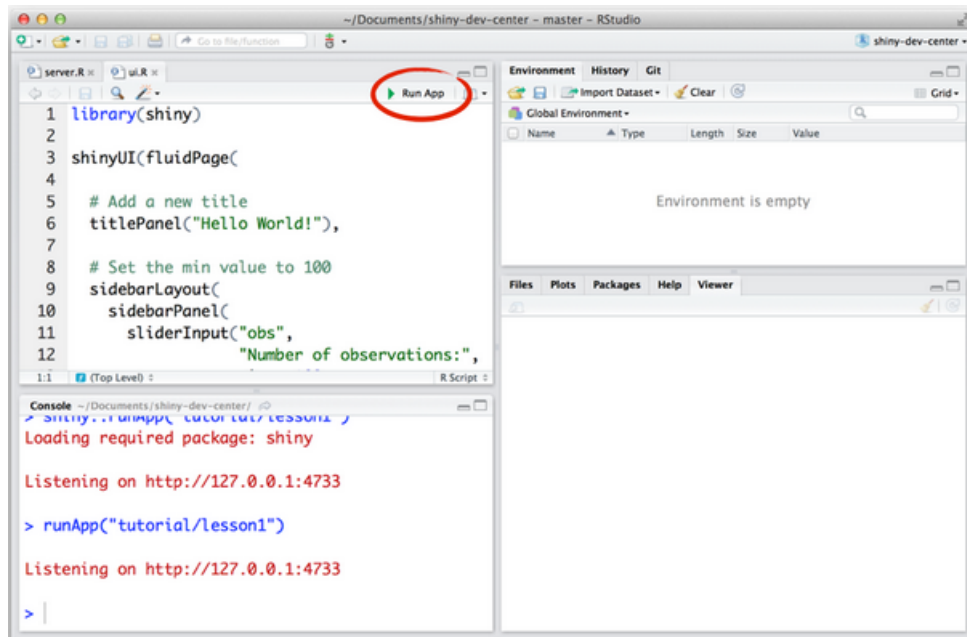


Figura 2.4: R com os dois *scripts*, `ui.R` e `server.R`. (Fonte: <http://shiny.rstudio.com/tutorial/lesson1/>)

### 2.3.1 Interface do Usuário (UI)

A interface do usuário (UI), como já foi dito anteriormente, é denominado `ui.R` e tem a função de controlar o *layout* e a aparência do aplicativo (programa). A programação UI do *Shiny* deve começar com a função `shinyUI()` e as principais funções que a compõem e dão o formato da interface do programa, são as de *Layout*, *Input* e *Output*.

#### UI Layout

A primeira função que deve vir dentro da `shinyUI()` é justamente o UI *Layout*, pois é a função que vai dizer justamente qual o *layout*, o formato, o modelo, que o programa

apresentará. Essas funções permitem a criação de vários tipos de páginas, painéis, *layout*, e etc. A relação das funções de UI *Layout* com suas descrições estão na Tabela 2.8.

Tabela 2.8: Funções UI Layout

Função	Descrição
<code>absolutePanel</code> ( <code>fixedPanel</code> )	Cria um painel cujo conteúdo tem um posicionamento absoluto
<code>bootstrapPage</code> ( <code>basicPage</code> )	Cria uma página Bootstrap
<code>column</code>	Cria uma coluna com uma definição UI
<code>conditionalPanel</code>	Cria um painel que é visível ou não, depende de uma expressão JavaScript
<code>fixedPage</code> ( <code>fixedRow</code> )	Cria uma página com layout fixo
<code>fluidPage</code> ( <code>fluidRow</code> )	Cria uma página composta de linhas que incluem colunas
<code>headerPanel</code>	Cria o cabeçalho do Painel
<code>helpText</code>	Cria um texto de ajuda
<code>icon</code>	Cria um ícone
<code>mainPanel</code>	Cria um painel principal
<code>navbarPage</code> ( <code>navbarMenu</code> )	Cria uma página com barras superiores de navegação
<code>navlistPanel</code>	Cria um painel com listas
<code>pageWithSidebar</code>	Cria uma página com uma barra lateral
<code>sidebarLayout</code>	Cria um layout com uma barra lateral e um menu principal
<code>sidebarPanel</code>	Cria um painel com uma barra lateral
<code>tabPanel</code>	Cria um painel com abas
<code>tabsetPanel</code>	Cria um menu principal com abas
<code>titlePanel</code>	Cria o título do Painel
<code>inputPanel</code>	Cria um painel com borda cinza e fundo cinza claro
<code>flowLayout</code>	Cria um layout com elementos da esquerda para direita e de cima para baixo
<code>splitLayout</code>	Cria um Layout horizontalmente, dividindo o horizonte em partes iguais
<code>verticalLayout</code>	Cria um layout com elementos verticais UI
<code>wellPanel</code>	Cria um painel com borda
<code>withMathJax</code>	Carrega MathJax e conjunto de expressões matemáticas

Fonte: <http://shiny.rstudio.com/reference/shiny/latest/> (Tradução Nossa)

## Conteúdo HTML

Existem funções que permitem elaborar textos com alguns avanços, é o caso das funções HTML. As funções e suas descrições podem ser encontradas na Tabela 2.9.



Tabela 2.9: Funções HTML

Função	Descrição
p	Parágrafo no texto
h1	Primeiro nível de cabeçalho
h2	Segundo nível de cabeçalho
h3	Terceiro nível de cabeçalho
h4	Quarto nível de cabeçalho
h5	Quinto nível de cabeçalho
h6	Sexto nível de cabeçalho
a	Hiper Link
br	Inserir uma linha em branco
div	Divisão do texto com estilo uniforme
span	Divisão em uma linha do texto com estilo uniforme
pre	Texto com uma fonte de largura fixa
code	Bloco de código formatado
img	Uma imagem
strong	Texto em negrito
em	Texto em itálico
HTML	Códigos em HTML

Fonte: <http://shiny.rstudio.com/tutorial/lesson2/> (Tradução Nossa)

## Widget (UI Input)

O *widget* é uma maneira do usuário enviar mensagens para o aplicativo *Shiny*. Então, é um modo de interação, onde o usuário irá selecionar um valor de entrada para o programa, por isso o *widget* é chamado também de *UI Input*. Na Figura 2.5 pode-se ver os tipos de *widgets* que o pacote *Shiny* possui, e na Tabela 2.10 estão as funções juntamente com suas descrições.

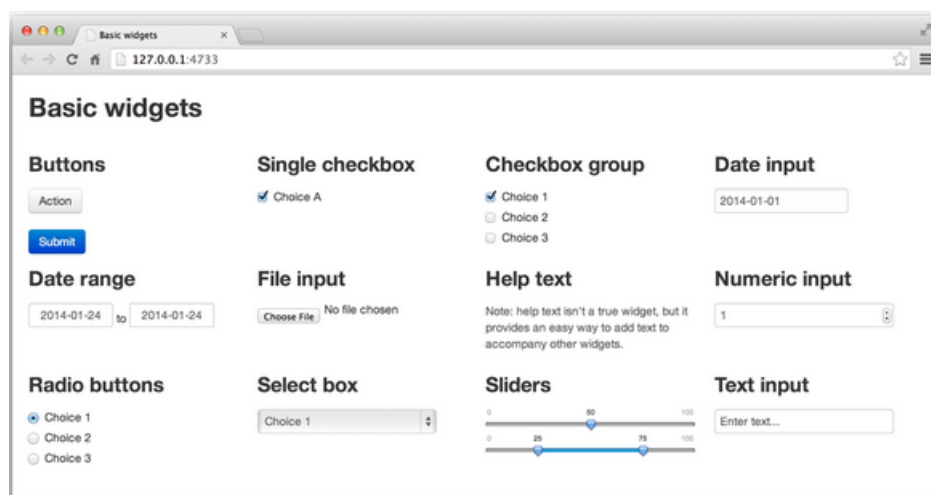


Figura 2.5: App Shiny com alguns exemplos de widgets. (Fonte: <http://shiny.rstudio.com/tutorial/lesson3/>)

Tabela 2.10: Funções Widget (UI Input)

Função	Descrição
<code>actionButton(actionLink)</code>	Cria botão de ação ou link cujo valor inicial é zero
<code>checkboxGroupInput</code>	Cria grupos de caixinhas que permitem várias escolhas independentes
<code>checkboxInput</code>	Cria uma caixinha para especificar um valor lógico
<code>dateInput</code>	Cria um calendário para selecionar uma data
<code>dateRangeInput</code>	Cria dois calendários para selecionar um período de tempo
<code>fileInput</code>	Cria um controle para selecionar um arquivo
<code>numericInput</code>	Cria um campo para digitar um número
<code>radioButtons</code>	Cria uma série de botões, onde apenas um pode ser selecionado
<code>selectInput(selectizeInput)</code>	Cria um campo que possui uma lista de elementos
<code>sliderInput(animationOptions)</code>	Cria uma barra de arrastar
<code>submitButton</code>	Cria um botão de submeter
<code>textInput</code>	Cria um campo para digitar um texto
<code>passwordInput</code>	Cria um campo para entrar com uma senha

Fonte: <http://shiny.rstudio.com/reference/shiny/latest/> (Tradução Nossa)

## UI Output

Existe uma série de funções no *Shiny* que, em conjunto com funções *renders*, que serão apresentadas na seção 2.3.2 *Server*, transformam objetos do R em uma saída para o usuário. Cada função apresenta um tipo específico de saída. Essas funções, juntamente com suas descrições, se encontram na Tabela 2.11.

Tabela 2.11: Funções UI Output

Função	Descrição
<code>htmlOutput(uiOutput)</code>	Cria uma saída com elementos em HTML
<code>imageOutput</code>	Cria uma saída com imagem
<code>plotOutput</code>	Cria uma saída com gráfico
<code>tableOutput(dataTableOutput)</code>	Cria uma saída com tabela
<code>textOutput</code>	Cria uma saída com texto
<code>verbatimTextOutput</code>	Cria uma saída com texto
<code>downloadButton(downloadLink)</code>	Cria um botão ou link para download

Fonte: <http://shiny.rstudio.com/reference/shiny/latest/> (Tradução Nossa)

### 2.3.2 Server

O *Shiny* necessita de informações para saber como construir seu objeto no R, esse é o papel do *script server*, ou seja, ele possui as instruções que o computador necessita para construir o seu aplicativo.

## Função Render

O código necessário para a construção do objeto deve ir em uma *function* (função) que deve aparecer dentro do *shinyServer()*. Essa função cria uma lista do tipo objeto, chamada de saída. Cada objeto do R deve conter sua própria entrada na lista, essa entrada pode corresponder a um elemento criado do *script* ui.R. As funções *render* podem ser vistas na Tabela 2.12.

Tabela 2.12: Funções Render

Função	Descrição
<code>renderImage</code>	Corresponde a uma saída com imagem
<code>renderPlot</code>	Corresponde a uma saída com gráfico
<code>renderPrint</code>	Corresponde a uma saída de qualquer tipo
<code>renderTable</code>	Corresponde a uma saída com tabela (data frame, matriz, etc)
<code>renderText</code>	Corresponde a uma saída com texto
<code>renderUI</code>	Corresponde a uma saída em Html

Fonte: <http://shiny.rstudio.com/reference/shiny/latest/> (Tradução Nossa)

## 3 Metodologia

### 3.1 Materiais

O presente trabalho busca criar um programa, utilizando a ferramenta *Shiny* do *R-project*, que torne simples e fácil a análise das respostas abertas (dissertativas) do questionário de avaliação docente, disponibilizado na internet por meio do MatriculaWeb. Então, para a realização desse estudo, foi utilizado o banco de dados fornecido pela Universidade de Brasília.

O questionário de avaliação docente, que também é chamado de “questionário de percepção discente sobre a disciplina, o desempenho docente e as condições de oferta” (ANEXO I), é de preenchimento opcional, sendo disponibilizado no MatriculaWeb todo final de semestre. Ele é composto por um número reduzido de questões, que são divididas em quatro categorias: disciplina, percepção sobre o desempenho do professor, autoavaliação e apoio institucional à disciplina. O formulário possui questões categorizadas, em uma escala de 1 a 5, sendo 1 insatisfatório e 5 excelente, é possível também assinalar a opção “não se aplica” em qualquer um dos itens. Além disso, o questionário possui um espaço aberto para comentários dos pontos fortes e pontos a melhorar, que é o objeto de estudo desse trabalho.

O banco de dados recebido possui um total de 45 variáveis, mas apenas 29 foram mantidas, devido ao fato de 16 variáveis não apresentarem repostas. O banco de dados foi dividido em duas partes, a fim de criar um banco de dados com as variáveis qualitativas ordinais, como mostra a Tabela 3.1, e outro com as variáveis dissertativas, como mostra a Tabela 3.2, com o intuito de facilitar a criação do aplicativo *Shiny*.

Sendo o objetivo do presente estudo apresentar o programa desenvolvido e não os dados, tomou-se um cuidado para preservar a confidencialidade da identidade dos professores, alunos e disciplinas, ou seja, os resultados apresentados são adaptados e não condizem com as matrículas apresentadas. Entretanto, o programa para uso na real avaliação dos dados verdadeiros está disponível.

Tabela 3.1: Variáveis do banco de dados qualitativo ordinal

Variável	Descrição
idDepartamento	Número de id do Departamento
Sigla	Sigla do Departamento
Departamento	Nome do Departamento
MatriculaAluno	Matricula do Aluno
idProfessor	Número de id do Professor
IdDisciplina	Número de id da Disciplina
Disciplina	Nome da Disciplina
Turma	Turma da Disciplina
q11	1.1 Clareza na descrição dos objetivos do programa
q12	1.2 Coerência entre objetivos, ementa e conteúdo ministrado
q13	1.3 Clareza quanto aos critérios de avaliação
q14	1.4 Adequação da bibliografia utilizada à proposta da disciplina
q15	1.5 Relevância da disciplina para a formação acadêmica e profissional do aluno
q21	2.1 Domínio do conteúdo ministrado
q22	2.2 Clareza na transmissão do conteúdo
q23	2.3 Adequação das atividades desenvolvidas para alcance dos objetivos propostos
q24	2.4 Capacidade de despertar o interesse dos estudantes em relação ao conteúdo
q25	2.5 Utilização de estratégias de ensino que facilitam a aprendizagem
q26	2.6 Capacidade de lidar com divergências de opinião
q27	2.7 Integração entre teoria, pesquisa, prática e aspectos da realidade
q28	2.8 Coerência entre nível de de complexidade das avaliações
q29	2.9 Discussão dos resultados de avaliação de aprendizagem
q210	2.10 Disponibilidade para esclarecer dúvidas e solucionar dificuldades dos alunos relacionados ao conteúdo da disciplina
q211	2.11 Pontualidade no cumprimento dos horários de início e término das aulas, pelo professor
q212	2.12 Assiduidade

Tabela 3.2: Variáveis do banco de dados dissertativo

Variável	Descrição
idDepartamento	Número de id do Departamento
Sigla	Sigla do Departamento
Departamento	Nome do Departamento
MatriculaAluno	Matricula do Aluno
idProfessor	Número de id do Professor
IdDisciplina	Número de id da Disciplina
Disciplina	Nome da Disciplina
Turma	Turma da Disciplina
q16	Ponto forte da Disciplina
q17	Ponto a melhorar da Disciplina
q213	Ponto forte do Professor
q214	Ponto a melhorar do Professor

## 3.2 Métodos

Tendo em vista que é papel do estatístico dar um bom tratamento aos dados e apresentá-los de forma clara e útil, buscou-se a elaboração desse programa para permitir que a Universidade de Brasília analise seus dados da avaliação docente. Para tanto, foi necessária a realização de quatro etapas para a obtenção do programa final, que será apresentado no próximo capítulo (4. Resultados e Discussões). As etapas foram as seguintes:

1. Organização do banco de dados;
2. Normalização das palavras das variáveis dissertativas do banco de dados;
3. Análise de frequência das respostas e das palavras;
4. E, por fim, a implementação dessa análise de frequência no pacote *Shiny*.

As programações realizadas no *R-project* em cada uma dessas etapas encontram-se no Anexo II desse trabalho. A seguir, serão descritas as técnicas utilizadas para a realização do presente estudo.

### Mineração de Texto

Para as etapas de organização e normalização do banco de dados textual, foi empregada a técnica de indexação e normalização da mineração de texto, que consiste nos seguintes passos:

- Identificou-se os termos simples, onde foi retirado qualquer tipo de pontuação e números contidos no texto, além de transformar todas as letras em minúsculas.
- Depois, foi retirado um total de 260 *stopwords*, ou seja, palavras que não apresentavam importância na análise textual.
- Por último, na fase de *stemming* (normalização morfológica), foram retiradas de algumas palavras principais as variações de gênero e número, o que permitiu uma melhor análise de frequência das palavras.

Para a etapa de frequência das palavras dos dados desse trabalho, foi utilizada a frequência absoluta e relativa de cada uma das palavras presentes no texto, pois esse é o primeiro passo para a fase do cálculo de relevância, que representa uma das fases mais importantes da mineração de texto.

### Expressões Regulares

Basicamente, o que permitiu a manipulação e organização desse banco de dados textual foi a técnica de expressões regulares. Então, fez-se uso de algumas funções importantes, como *grepl* e *gregexpr*, que permitiram localizar certos padrões de interesse nos dados, como encontrar a matrícula de algum aluno ou o id de um determinado professor. Outra função utilizada foi a *gsub*, que ajudou na organização dos dados, em particular na parte de normalização morfológica citada logo acima, pois ela identifica um padrão e o substitui por outro desejado.

E, por fim, para auxiliar na análise de frequência das palavras, usou-se a função *str\_split* que gera um vetor com todas as palavras existentes em cada uma das categorias, pois essa função dividi uma frase em um vetor de palavras seguindo um padrão.

### Shiny

Para a criação da interface do programa foram utilizadas algumas funções dentre as inúmeras disponíveis no pacote *Shiny* do *R-Project*. A principal foi a *navbarPage*, usada para criar barras superiores que dividem o aplicativo em seções com as diferentes análises existentes. Além disso, foi gerado um painel principal em cada uma das seções, por meio da função *mainPanel*, e um painel lateral, por meio da função *sidebarPanel*, tudo isso com o intuito de tornar a interface do programa mais organizada, agradável e de fácil entendimento.

Nos painéis laterais foram criados diversos *widgets* utilizando as funções *selectInput*, *sliderInput* e *textInput*, que permitem ao usuário selecionar a variável que deseja analisar em cada uma das seções do programa. E para a visualização das análises, em especial as de frequências que aparecem no painel principal, fez-se uso de gráficos de barra, *wordcloud* (Nuvem de Palavras) e tabelas, além de listas que também foram usadas para mostrar a média e o desvio-padrão das variáveis categorizadas. Utilizou-se as médias e os desvios-padrões porque hoje a UnB utiliza essa análise para as variáveis categorizadas.

## 4 Resultados e Discussões

Nesse capítulo será apresentado o produto do presente estudo, que é o programa de análise do questionário de avaliação docente do MatriculaWeb, usando o pacote Shiny do *R-project*. Para utilizar o programa criado, basta abrir os dois arquivos necessários (ui.R e server.R), e, após aberto no R os dois *scripts*, clique no botão *Run App*, como já foi ensinado anteriormente na seção 2.3 Shiny.

Após iniciar o aplicativo, deve-se entender como funciona o programa. Então, será explicada a seguir toda a programação passo-a-passo juntamente com a saída, ou seja, a interface do programa criado, e também sua funcionalidade na hora da análise. E para visualizar toda a programação dirija-se ao Anexo II do presente estudo.

A primeira operação a ser realizada pelo aplicativo é carregar os pacotes exigidos, depois entrar com os dados necessários para a análise, escolher o tipo de página que a interface do programa possuirá e o seu título.

```
ui.R
##### Pacote exigidos #####
require(shiny)

#####

shinyUI(navbarPage(h3("Análise do Questionário de Avaliação Docente"),

Server.R

##### Pacote exigidos #####
require(shiny)
require(stringr)
require(wordcloud)
require(markdown)

##### Importando arquivos #####

# Banco de dados
base.dis <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus documentos/
                    facudade-UNB/Monografia/Programação/Dados/base.dis.csv")
base.cat <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus documentos/
                    facudade-UNB/Monografia/Programação/Dados/base.cat.csv")

# Banco de dados normalizado
base.norm <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus documentos/
                    facudade-UNB/Monografia/Programação/Dados/base.norm.csv")

# Stoplist
stoplist <- scan(file="C:/Users/ThiagoMoreti/Documents/Meus documentos/
                    facudade-UNB/Monografia/Programação/Dados/Stoplist.txt",
                  what=character(0), sep="\n")
```



```
#####
# Criando nível para base.cat
for (i in 10:26){
  base.cat[,i] <- as.factor(base.cat[,i])
  levels(base.cat[,i])<- c(1,2,3,4,5)
}

#####
#####
shinyserver(function(input, output, session) {
  selectedData <- reactive({
    base.dis
  })
  selectedData1 <- reactive({
    base.cat
  })
  selectedNorm <- reactive({
    base.norm
  })
  selectedStop <- reactive({
    stoplist
  })
}
```

A função `navbarPage(h3("Análise do Questionário de Avaliação Docente"))` permite que se crie uma página com barras superiores e o título do programa, que nesse caso é *Análise do Questionário de Avaliação Docente*, conforme mostra a Figura 4.1.



Figura 4.1: Tipo de página com barras superiores

## 4.1 Tutorial

A função `tabPanel(h5("Tutorial"))` cria uma barra superior fixa chamada *Tutorial*, assim como mostra a Figura 4.2. A parte `includeMarkdown("Tutorial.md")` insere um arquivo do tipo *Markdown*, que contém todo o texto do Tutorial, como pode ser visto

na Figura 4.2. E a função HTML `br()` gera uma quebra no texto.

```
ui.R
##### Tutorial #####
tabPanel(h5("Tutorial"),
         includeMarkdown("Tutorial.md"),
         br()
        ),
```



Figura 4.2: Barra tutorial e o texto do tutorial

Quando o programa é rodado, abre diretamente na página do tutorial, mas caso o usuário saia e deseje retornar, clique na barra superior Tutorial. E qualquer dúvida a cerca do programa, basta consultar essa página.

## 4.2 Análise de Frequência

### 4.2.1 Respostas

Essa área do aplicativo serve para fazer a análise de quantas respostas foram dadas pelo aluno em cada uma das categorias (Ponto Forte da Disciplina, Ponto a Melhorar da Disciplina, Ponto Forte do Professor e Ponto a Melhorar do Professor) e quantas respostas os professores ou as disciplinas receberam em cada uma delas.

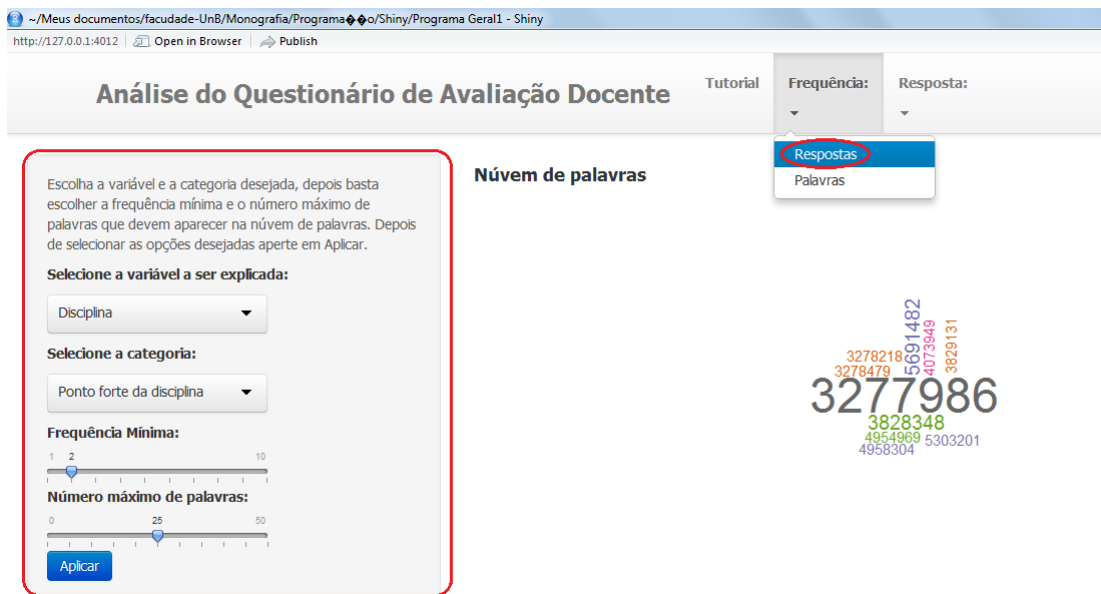


Figura 4.3: Opção de página da análise de Frequência das Respostas

Essa parte do *script* ui.R apresenta a função `navbarMenu(h5("Frequência:"))` que cria uma barra superior chamada Frequência, do tipo menu, ou seja, ao clicar nela abrirá mais opções de páginas. A função `tabPanel("Respostas")` cria dentro da barra Frequência a opção de página chamada Respostas. Ao clicar na opção Respostas, irá direto para a página que faz a análise de frequência das respostas do questionário. (Figura 4.3)

```
ui.R
##### Respostas #####
      navbarMenu(h5("Frequência:"),
                tabPanel("Respostas"),
```

Para criar o painel lateral da Figura 4.3, usa-se o comando `sidebarPanel`. Dentro desse painel encontra-se um texto de ajuda, que para ser incluído usa-se o `helpText`.

```

sidebarLayout(
  sidebarPanel(
    helpText("Escolha a variável e a categoria desejada,",
             "depois basta escolher a frequência mínima e",
             "o número máximo de palavras que devem aparecer",
             "na nuvem de palavras. Depois de selecionar",
             "as opções desejadas aperte em Aplicar."),
    selectInput("id", h5("Selecione a variável a ser explicada:"),
               choices = c("Disciplina" = "IdDisciplina",
                           "Professor" = "idProfessor",
                           "Aluno" = "MatriculaAluno")),
    selectInput("categoria", h5("Selecione a categoria:"),
               choices = c("Ponto forte da disciplina" = "Ponto.Forte.da.Disciplina",
                           "Ponto a melhorar da disciplina" = "Ponto.a.Melhorar.da.Disciplina",
                           "Ponto forte do professor" = "Ponto.Forte.do.Professor",
                           "Ponto a melhorar do professor" = "Ponto.a.Melhorar.do.Professor")),
    sliderInput("f",
               h5("Frequência Mínima:"),
               min = 1, max = 10, value = 2, step = 1),
    sliderInput("max",
               h5("Número máximo de palavras:"),
               min = 0, max = 50, value = 25, step = 5),
    submitButton("Aplicar")
  ),
),

```

O comando `selectInput("id", h5("Selecione a variável a ser explicada:"))` cria o *widget* que seleciona as variáveis a serem explicadas (Professor, Disciplina e Aluno), assim como mostra a Figura 4.4. E `selectInput("categoria", h5("Selecione a categoria:"))` cria o *widget* que seleciona as categorias (Ponto Forte da Disciplina, Ponto a Melhorar da Disciplina, Ponto Forte do Professor e Ponto a Melhorar do Professor), conforme a Figura 4.5.

Escolha a variável e a categoria desejada, depois basta escolher a frequência mínima e o número máximo de palavras que devem aparecer na nuvem de palavras. Depois de selecionar as opções desejadas aperte em Aplicar.

**Selecione a variável a ser explicada:**

Disciplina

Disciplina

Professor

Aluno

**Frequência Mínima:**

1 2 10

**Número máximo de palavras:**

0 25 50

Aplicar

Figura 4.4: Widget da variável a ser explicada

Escolha a variável e a categoria desejada, depois basta escolher a frequência mínima e o número máximo de palavras que devem aparecer na núvem de palavras. Depois de selecionar as opções desejadas aperte em Aplicar.

**Selecione a variável a ser explicada:**

Disciplina ▼

**Selecione a categoria:**

Ponto forte da disciplina ▲

Ponto forte da disciplina

Ponto a melhorar da disciplina

Ponto forte do professor

Ponto a melhorar do professor

Aplicar

Figura 4.5: Widget da categoria

Para gerar a barra de arrastar responsável por selecionar a frequência mínima, que varia de 1 até 10, e o número máximo de palavras, que varia de 0 até 50, que a *wordcloud* deve retornar, usa-se *sliderInput("f")* e *sliderInput("max")*, respectivamente (Figura 4.6). Após tudo selecionado, clique no botão Aplicar, que foi gerado com *submitButton("Aplicar")*.

Escolha a variável e a categoria desejada, depois basta escolher a frequência mínima e o número máximo de palavras que devem aparecer na núvem de palavras. Depois de selecionar as opções desejadas aperte em Aplicar.

**Selecione a variável a ser explicada:**

Disciplina ▼

**Selecione a categoria:**

Ponto forte da disciplina ▼

**Frequência Mínima:**

1 2 10

**Número máximo de palavras:**

0 25 50

Aplicar

Figura 4.6: Widget da frequência mínima e máximo de palavras da wordcloud

Assim, no painel principal, o programa retornará a *wordcloud* e também uma lista com as matrículas das variáveis selecionadas segundo alguma categoria, como aparece na Figura 4.7. Para gerar esse painel principal juntamente com a saída, é necessário fazer

como a programação a seguir.

```

ui.R

mainPanel(
  h4("Núvem de palavras"),
  plotOutput("wordcloud1"),

  h4("Frequência de respostas"),
  verbatimTextOutput("freq")
),
)

Server.R
##### Respostas #####
id <- reactive({
  selectedData()[,input$id]
})
var <- reactive({
  selectedData()[,input$categoria]
})
coment <- reactive({
  grepl("[a-z]", var())
})
freq <- reactive({
  sort(summary(as.factor(id[coment])),maxsum=3664,decreasing=T)
})
output$wordcloud1 <- renderPlot({
  if (max(freq()) < input$F) {
    wordcloud("None", 1, scale=c(4,1), max.words= input$max,
             colors=brewer.pal(8,"Dark2"))}
  else {wordcloud(names(freq()), freq(), scale=c(4,.3), min.freq = input$F,
                 max.words=input$max, random.order=FALSE, rot.per=.15,
                 random.color=T, colors=brewer.pal(8,"Dark2"))}
})
output$freq <- renderPrint({
  freq()
})

```

### Núvem de palavras



### Frequência de respostas

3277986	3335319	3278218	3828348	5366015	5632409	3989037	5249377	5568435	3279697	3324154	34220	
29	3308349	3324386	3899485	3277522	3285729	3422290						
	52	20	19	18	18	18	16	15	15	14	14	
14	13	13	13	12	12	12						
	4031957	3278479	5337131	5539783	3629524	3915319	4233159	3248319	3336305	4954969	3372729	35126
54	4871507	4950039	3422812	3519063	3534462	3596290						
	12	11	11	11	10	10	10	9	9	9	8	
8	8	8	7	7	7	7						

Figura 4.7: Painel principal com a wordcloud e a lista com as frequências

### 4.2.2 Palavras

Nessa área, pode-se fazer a análise da frequência absoluta das palavras que existem em cada uma das categorias (Ponto Forte da Disciplina, Ponto a Melhorar da Disciplina, Ponto Forte do Professor e Ponto a Melhorar do Professor). A visualização se dá de três maneiras: por meio da Nuvem de palavras, gráfico de barras ou de uma tabela.

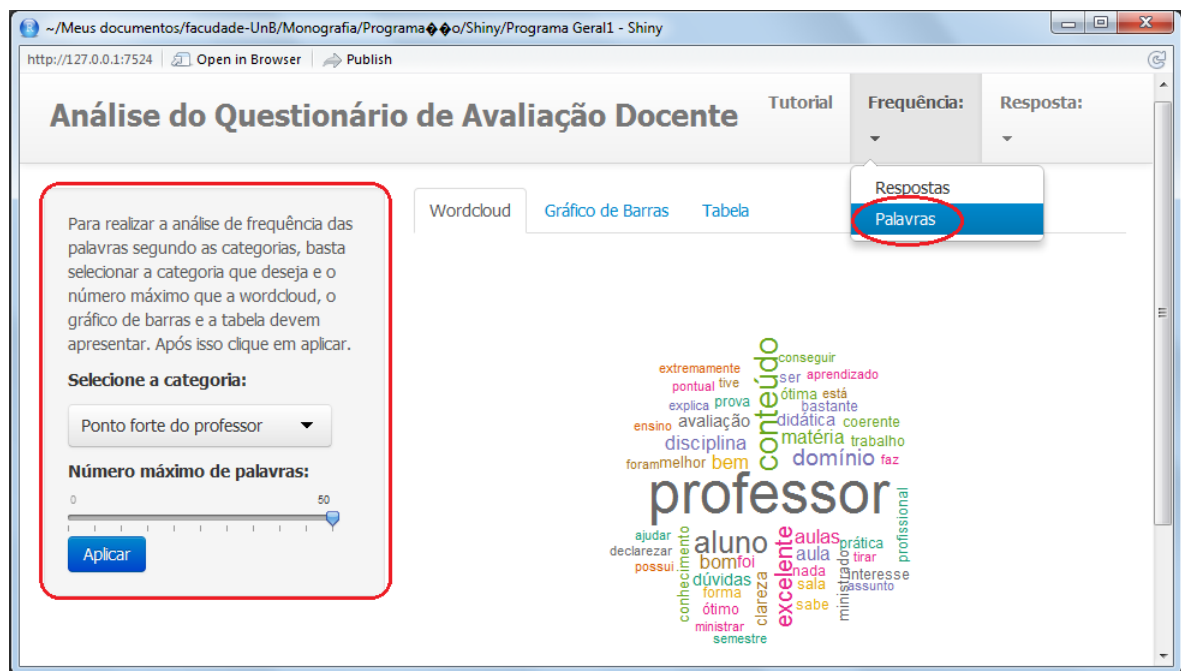


Figura 4.8: Página da análise de frequência das palavras

Essa parte do *script* ui.R apresenta a função `tabPanel("Palavras",` que cria dentro da barra Frequência a opção de página chamada Palavras. Ao clicar na opção Palavras, irá direto para a página da Figura 4.8.

```
ui.R
##### Palavras #####
tabPanel("Palavras",|
```

Para criar a barra lateral da Figura 4.8, usa-se o comando `sidebarPanel`. Dentro dessa barra, encontra-se um texto de ajuda que para ser incluído usa-se o `helpText`.

```

ui.R

sidebarLayout(
  sidebarPanel(
    helpText("Para realizar a análise de frequência das palavras",
             "segundo as categorias, basta selecionar a categoria",
             "que deseja e o número máximo que a wordcloud, o gráfico",
             "de barras e a tabela devem apresentar.",
             "Após isso clique em aplicar."),
    selectInput("categoria1", h5("Selecione a categoria:"),
               choices = c("Ponto forte da disciplina" =
                           "Ponto.Forte.da.Disciplina" =
                           "Ponto a melhorar da disciplina" =
                           "Ponto.a.Melhorar.da.Disciplina",|
                           "Ponto forte do professor" =
                           "Ponto.Forte.do.Professor" =
                           "Ponto a melhorar do professor" =
                           "Ponto.a.Melhorar.do.Professor")),
    sliderInput("max1",
               h5("Número máximo de palavras:"),
               min = 0, max = 50, value = 25, step= 5),
    submitButton("Aplicar")
  ),
),

```

O comando `selectInput("categoria1", h5("Selecione a categoria:"))` cria o widget que seleciona as categorias (Ponto Forte da Disciplina, Ponto a Melhorar da Disciplina, Ponto Forte do Professor e Ponto a Melhorar do Professor), como mostra a Figura 4.9.

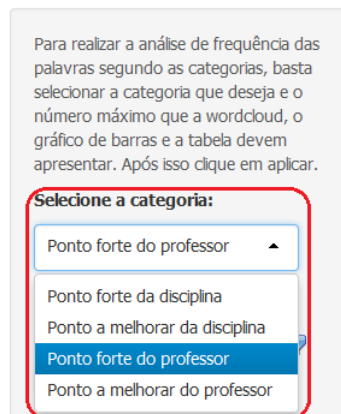


Figura 4.9: Widget da categoria

Para gerar a barra de arrastar com o máximo de palavras que devem aparecer na *wordcloud*, gráfico de barras e na tabela, usa-se `sliderInput("max1"`. (Figura 4.10)



Para realizar a análise de frequência das palavras segundo as categorias, basta selecionar a categoria que deseja e o número máximo que a wordcloud, o gráfico de barras e a tabela devem apresentar. Após isso clique em aplicar.

**Selecione a categoria:**

Ponto forte do professor ▼

**Número máximo de palavras:**

0 50

Aplicar

Figura 4.10: Widget número máximo de palavras

Após tudo selecionado clique no botão Aplicar, que foi gerado com `submitButton("Aplicar")`. Assim, no painel principal, o programa retornará a *wordcloud* (Figura 4.11), o gráfico de barras (Figura 4.12) e a tabela com a frequência absoluta e relativa de cada palavra (Figura 4.13). Para gerar as abas do menu principal, usa-se `tabsetPanel(type = "tabs"` dentro da função `mainPanel`.

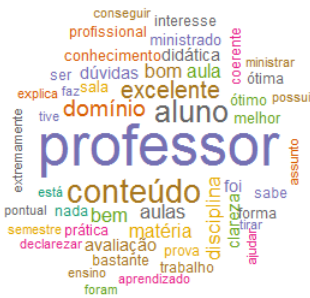
ui.R

```

mainPanel(
  tabsetPanel(type = "tabs",
    tabPanel("wordcloud",
      plotOutput("wordcloud2")),
    tabPanel("Gráfico de Barras",
      plotOutput("hist")),
    tabPanel("Tabela",
      tableOutput("tabela"))),
  h4("Total de palavras na categoria"),
  textOutput("total1")
)
),
),
),

```

Wordcloud Gráfico de Barras Tabela

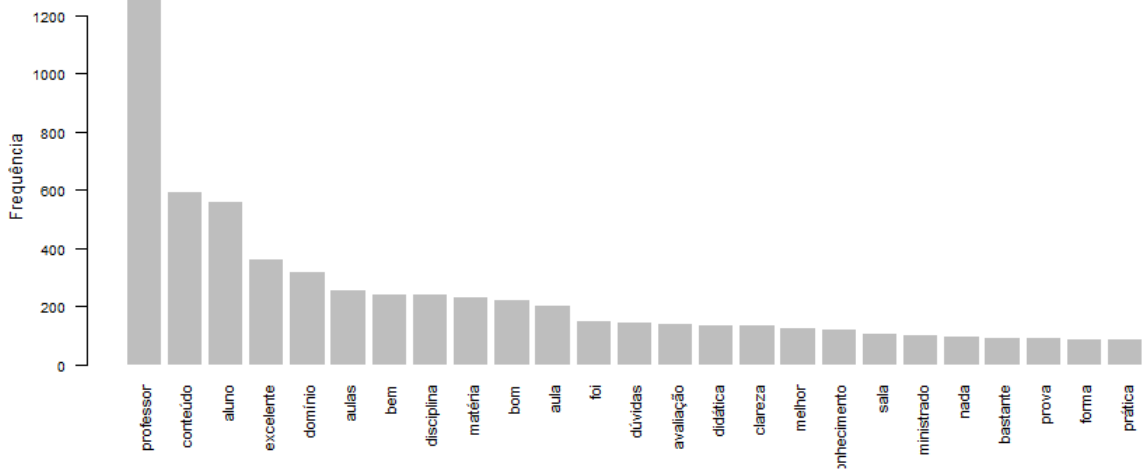


Total de palavras na categoria

[1] 3909

Figura 4.11: Saída Wordcloud

Wordcloud Gráfico de Barras Tabela



Total de palavras na categoria

[1] 3909

Figura 4.12: Saída gráfico de barras

Wordcloud Gráfico de Barras **Tabela**

	Palavras	Frequência	Freq Relativa
1	professor	1287	0.33
2	conteúdo	594	0.15
3	aluno	560	0.14
4	excelente	363	0.09
5	domínio	316	0.08
6	aulas	254	0.06
7	bem	243	0.06
8	disciplina	240	0.06
9	matéria	231	0.06
10	bom	220	0.06

**Total de palavras na categoria**  
[1] 3909

Figura 4.13: Saída tabela de frequência

Mas é necessário que se especifique no *script* `server.R` qual o procedimento que o R deve realizar para gerar as saídas acima. Para isso, foram feitos os seguintes comandos.

```
server.R
output$tabela <- renderTable({
  tabela <- matrix(c(words_n(),freq_n(),freqr_n()),ncol=3,nrow=input$max1,byrow=F)
  colnames(tabela) <- c("Palavras", "Frequência", "Freq Relativa")
  rownames(tabela) <- c(1:input$max1)
  tabela <- as.data.frame(tabela)
})

output$hist <- renderPlot({
  barplot(freq_n(), border = NA, names.arg = words_n(),
    las = 2, ylab = "Frequência",
    ylim = c(0,max(freq_n())),cex.names=0.9, cex.axis=0.8)
})

output$wordcloud2 <- renderPlot({
  wordcloud(words_n(), freq_n(), scale=c(4,.8), min.freq= min(freq()),
    max.words=input$max1, random.order= FALSE, rot.per=.15,
    random.color=T, colors=brewer.pal(8,"Dark2"))
})

output$total1 <- renderPrint({
  total1()
})
```

## 4.3 Análise das Respostas

### 4.3.1 Unidade Acadêmica

Nessa área pode-se ter acesso a todas as respostas (avaliação) que uma determinada Unidade Acadêmica recebeu, sendo possível visualizar tanto a frequência, a média e o desvio-padrão das respostas categorizadas, quanto as respostas dissertativas.

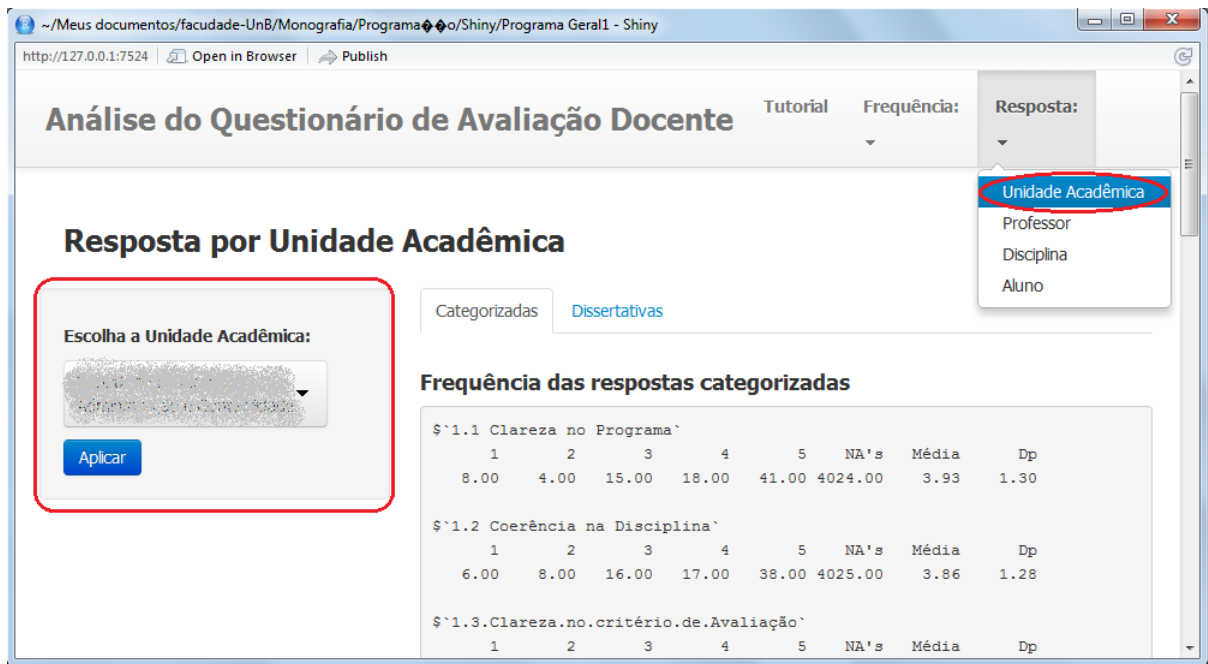


Figura 4.14: Página da análise das respostas por unidade acadêmica

Essa parte do *script* ui.R apresenta a função `navbarMenu(h5("Resposta:"))`, que cria uma barra superior chamada Resposta, do tipo menu, ou seja, ao clicar nela abrirá mais opções de páginas. A função `tabPanel("Unidade Acadêmica")` cria dentro da barra Resposta a opção de página chamada Unidade Acadêmica e, ao clicar nela, irá direto para a página da Figura 4.14.

```
ui.R
##### Departamento #####
      navbarMenu(h5("Resposta:"),
                tabPanel("unidade Acadêmica",
```

Para criar a barra lateral da Figura 4.15, usa-se o comando `sidebarPanel`. Dentro dessa barra, encontra-se um *widget* que seleciona uma das 24 unidades acadêmicas disponíveis no banco de dados. Para isso, utiliza-se o comando `selectInput("depart", h5("Escolha a Unidade Acadêmica:"))`.

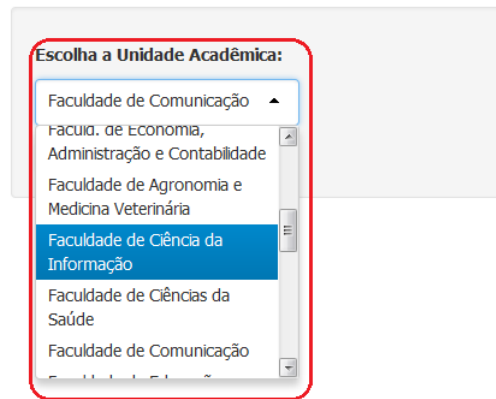


Figura 4.15: Widget que seleciona as unidades acadêmicas

```

ui.R
headerPanel(h3("Resposta por Unidade Acadêmica")),
sidebarLayout(
  sidebarPanel(
    selectInput("depart", h5("Escolha a Unidade Acadêmica:"),
      choices = c("Decanato de Pesquisa e Pós-Graduação",
        "Direção da Faculdade de Arquitetura e Urbanismo",
        "Direção da Faculdade de Ciências da Saúde",
        "Direção do Instituto Ciências Biológicas",
        "Direção do Instituto de Psicologia",
        "Faculd. de Economia, Administração e Contabilidade",
        "Faculdade de Agronomia e Medicina Veterinária",
        "Faculdade de Ciência da Informação",
        "Faculdade de Ciências da Saúde",
        "Faculdade de Comunicação",
        "Faculdade de Educação",
        "Faculdade de Educação Física" =
        "Faculdade de Ed. Física" =
        "Faculdade de Estudos Sociais Aplicados",
        "Faculdade de Medicina",
        "Faculdade de Tecnologia",
        "Gabinete do Reitor",
        "Instituto de Artes",
        "Instituto de Ciências Biológicas",
        "Instituto de Ciências Exatas",
        "Instituto de Ciências Humanas",
        "Instituto de Ciências Sociais",
        "Instituto de Física",
        "Instituto de Geociências",
        "Instituto de Letras")),
    submitButton("Aplicar")
  ),

```

Para gerar as abas do painel principal (Figura 4.16), usa-se `tabsetPanel(type = "tabs"` no `script` ui.R. Também é necessário dizer no `server.R` qual procedimento o R deverá retornar dentro de cada uma das abas. Como mostra a programação a seguir.

```

ui.R
  mainPanel(
    tabsetPanel(type = "tabs",
      tabPanel("Categorizadas",
        h4("Frequência das respostas categorizadas"),
        verbatimTextOutput("freqd")),
      tabPanel("Dissertativas",
        h4("Respostas dissertativas por unidade acadêmica"),
        tableOutput("respd"))
    )
  ),

server.R
output$freqd <- renderPrint({
  listal()
})
output$respd <- renderTable({
  selectedData()[proc11(),c(4,10:13,6:9)]
})

```

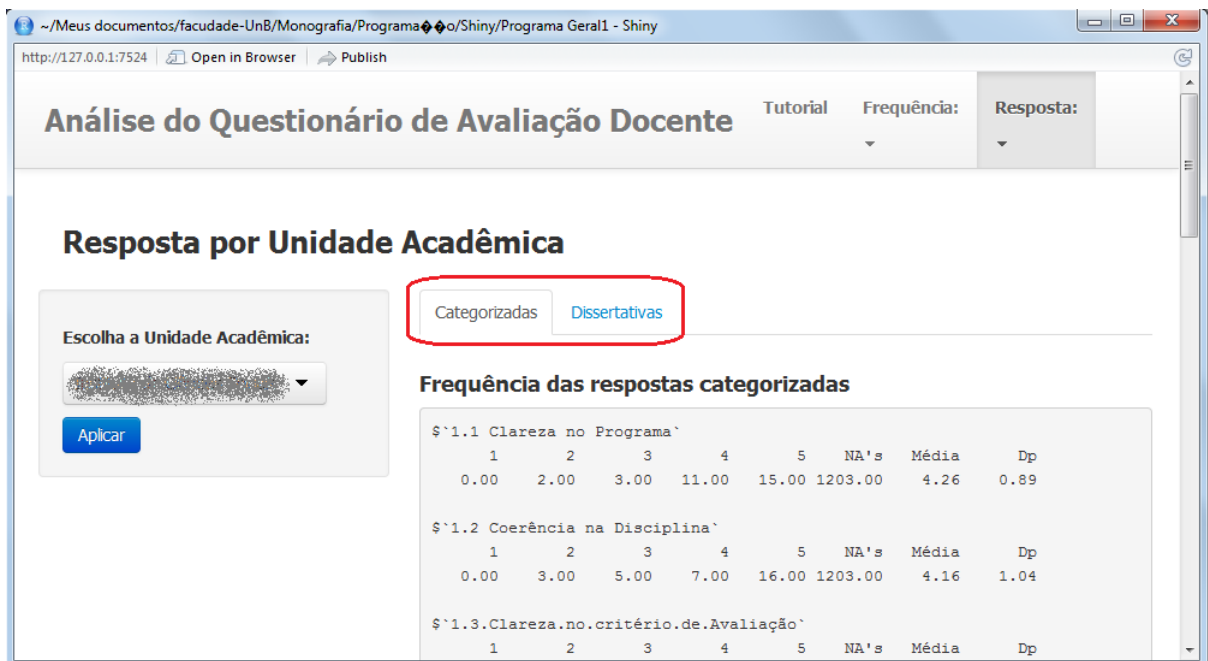


Figura 4.16: Abas do painel principal

Depois de selecionada a Unidade Acadêmica, basta clicar no botão Aplicar. Assim, no painel principal, o programa retornará duas saídas, a primeira é uma lista com a frequência, média e desvio-padrão das variáveis qualitativas ordinais (Figura 4.17), e a segunda é uma tabela com todas as respostas que essa Unidade recebeu dos alunos (Figura 4.18).

Categorizadas **Dissertativas**

### Frequência das respostas categorizadas

S`1.1 Clareza no Programa`								
1	2	3	4	5	NA's	Média	Dp	
1.00	2.00	4.00	9.00	17.00	1577.00	4.18	1.07	
S`1.2 Coerência na Disciplina`								
1	2	3	4	5	NA's	Média	Dp	
1.00	2.00	5.00	10.00	15.00	1577.00	4.09	1.07	
S`1.3.Clareza.no.critério.de.Avaliação`								
1	2	3	4	5	NA's	Média	Dp	
4.00	2.00	6.00	5.00	16.00	1577.00	3.82	1.42	
S`1.4 Adequação da Bibliografia`								
1	2	3	4	5	NA's	Média	Dp	
3.00	1.00	2.00	11.00	13.00	1580.00	4.00	1.26	
S`1.5 Relevância da Disciplina`								
1	2	3	4	5	NA's	Média	Dp	
3.00	1.00	4.00	5.00	19.00	1578.00	4.12	1.31	
S`2.1 Domínio do Conteúdo`								
1	2	3	4	5	NA's	Média	Dp	
41.00	39.00	127.00	330.00	1071.00	2.00	4.46	0.93	

Figura 4.17: Frequência das variáveis categorizadas

Categorizadas **Dissertativas**

### Respostas dissertativas por unidade acadêmica

Ponto.Forte.da.Disciplina	Ponto.a.Melhorar.da.Disciplina	Ponto.Forte.do.Professor	Ponto.a.Melhorar.do.Professor
		clareza na explicação da matéria	dificuldade em lidar com opiniões diferentes
		não tem	não tem
		professora se desdobrou e conseguiu atingir o objetivo mesmo depois da greve uma guerreira parabéns	nada a declarar
		professor pontual coerente com nível de provas e trabalhos compromisso e responsável	
		ótimo domínio do conteúdo boa vontade em atender aos alunos	

Figura 4.18: Respostas recebidas por uma determinada Unidade Acadêmica

### 4.3.2 Professor

Nessa seção pode-se ter acesso a todas as respostas (avaliação) que um determinado professor recebeu de seus alunos, sendo possível visualizar tanto a frequência, a média e o desvio-padrão das respostas categorizadas, quanto as respostas dissertativas.

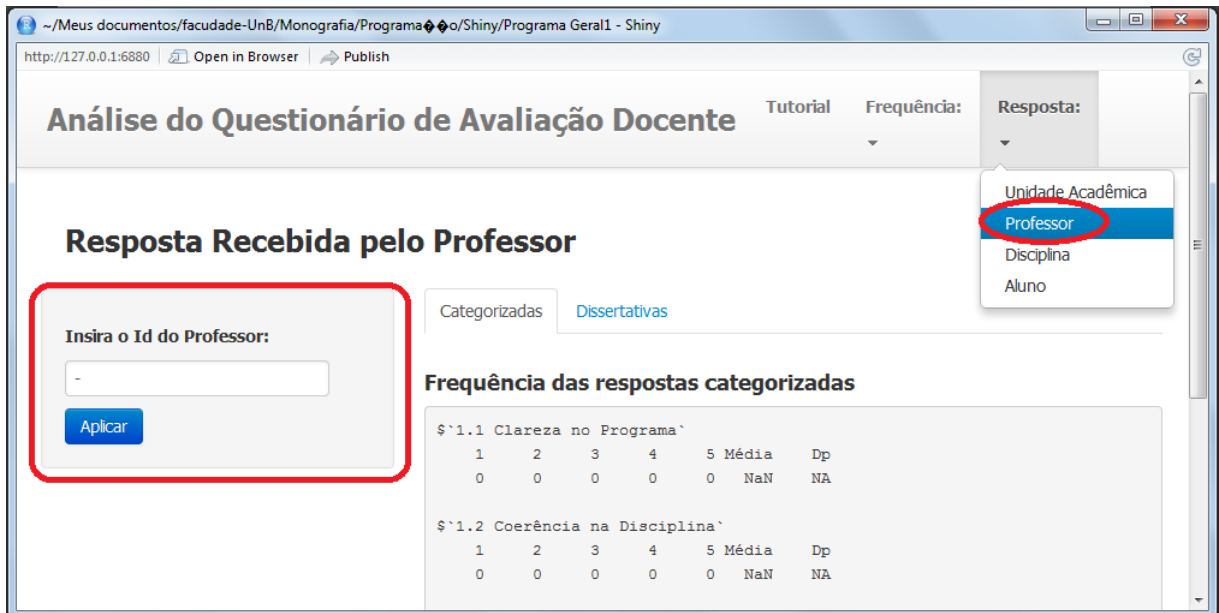


Figura 4.19: Página da análise das respostas recebidas pelo professor

Essa parte do *script* ui.R apresenta a função `tabPanel("Professor"`, que cria dentro da barra Resposta a opção de página chamada Professor e, ao clicar nela, irá direto para a página da Figura 4.19.

```
ui.R
##### Professor #####
  tabPanel("Professor",
    headerPanel(h3("Resposta Recebida pelo Professor")),
    sidebarLayout(
      sidebarPanel(
        textInput("idp", h5("Insira o Id do Professor:"),
                  value = "-"),
        submitButton("Aplicar")
      ),
    ),
```

Para criar a barra lateral da Figura 4.19, usa-se o comando `sidebarPanel`. Dentro dessa barra, encontra-se um *widget* que permite insirir o id do professor que se deseja consultar, como mostra a Figura 4.20. Para tanto, utiliza-se o comando `textInput("idp", h5("Insira o Id do Professor:"))`.



Figura 4.20: Widget para inserir o id do Professor

Para gerar as abas do painel principal (Figura 4.21), usa-se `tabsetPanel(type = "tabs"` no `script ui.R`, mas também é necessário dizer no `server.R` qual procedimento o R deverá retornar dentro de cada uma das abas. Como mostra a programação a seguir.

```
ui.R
mainPanel(
  tabsetPanel(type = "tabs",
    tabPanel("Categorizadas",
      h4("Frequência das respostas categorizadas"),
      verbatimTextOutput("freqp")),
    tabPanel("Dissertativas",
      h4("Respostas dissertativas recebidas pelo Professor"),
      tableOutput("respp"))
  )
)

Server.R
output$freqp <- renderPrint({
  lista2()
})
output$respp <- renderTable({
  selectedData()[proc21(),c(6,10:13,7:9)]
})
```

Frequência das respostas categorizadas									
`\$`1.1 Clareza no Programa`									
1	2	3	4	5	NA's	Média	Dp		
0	0	0	1	0	10	4	NA		
`\$`1.2 Coerência na Disciplina`									
1	2	3	4	5	NA's	Média	Dp		
0	0	0	1	0	10	4	NA		

Figura 4.21: Abas do Painel Principal

Depois de inserir o id do professor, basta clicar no botão Aplicar. Assim, no painel principal, o programa retornará duas saídas, a primeira é uma lista com a frequência, a média e desvio-padrão das variáveis qualitativas ordinais (Figura 4.22), e a segunda é uma tabela com todas as respostas que esse professor recebeu de seus alunos (Figura 4.23).

Categorizadas
Dissertativas

---

**Frequência das respostas categorizadas**

\$`1.1 Clareza no Programa`							
1	2	3	4	5	NA's	Média	Dp
0	0	0	1	0	10	4	NA
\$`1.2 Coerência na Disciplina`							
1	2	3	4	5	NA's	Média	Dp
0	0	0	1	0	10	4	NA
\$`1.3.Clareza.no.critério.de.Avaliação`							
1	2	3	4	5	NA's	Média	Dp
0	0	0	0	1	10	5	NA
\$`1.4 Adequação da Bibliografia`							
1	2	3	4	5	NA's	Média	Dp
0	0	0	0	1	10	5	NA
\$`1.5 Relevância da Disciplina`							
1	2	3	4	5	NA's	Média	Dp
0	0	0	1	0	10	4	NA

Figura 4.22: Frequência das variáveis categorizadas

Categorizadas
Dissertativas

---

**Respostas dissertativas recebidas pelo Professor**

	Ponto.Forte.da.Disciplina	Ponto.a.Melhorar.da.Disciplina	Ponto.Forte.do.Professor	Ponto.a.Melhorar.do.Professor
1364				esse professor não ministrou aulas nesta disciplina no referido semestre o sistema da unb não deveria me dar a possibilidade de avaliá lo
1383			o conteúdo do professor e seu empenho em despertar o interesse dos alunos pela pesquisa e pela extensão é uma verdadeira referência em nossa universidade	a forma de transmitir o conteúdo de ministrar a aula propriamente mesmo sabendo muito a transmissão do conhecimento pelo professor se mostra pouco didática
1391		nada a observar	o professor é uma pessoa muito agradável e que sabe transmitir o conteúdo de maneira efetiva e prazerosa para o aluno muito qualificado de fato	um pouco mais de firmeza com certos horários e datas dos alunos

Figura 4.23: Respostas recebidas por um determinado Professor

### 4.3.3 Disciplina

Nessa área pode-se ter acesso a todas as respostas (avaliação) que uma determinada disciplina recebeu dos alunos que a cursaram, sendo possível visualizar tanto a frequência, a média e o desvio-padrão das respostas categorizadas, quanto as respostas dissertativas.

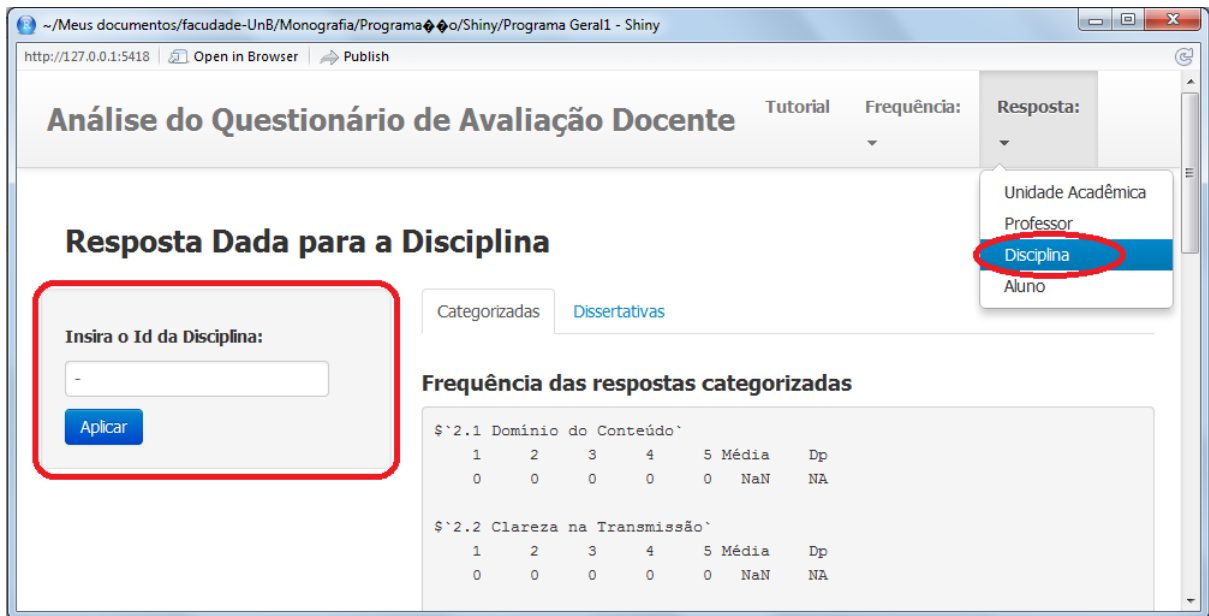


Figura 4.24: Página da análise das respostas recebidas pela disciplina

Essa parte do *script* ui.R apresenta a função `tabPanel("Disciplina",` que cria dentro da barra Resposta a opção de página chamada Disciplina e, ao clicar nela, irá direto para a página da Figura 4.24.

ui.R

```
##### Disciplina #####
```

```
tabPanel("Disciplina",
  headerPanel(h3("Resposta Dada para a Disciplina")),
  sidebarLayout(
    sidebarPanel(
      textInput("idd", h5("Insira o Id da Disciplina:"),
        value = "-"),
      submitButton("Aplicar")
    ),
  ),
```

Para criar a barra lateral da Figura 4.24, usa-se o comando `sidebarPanel`. Dentro dessa barra, encontra-se um *widget* que permite inserir o id da disciplina que se deseja consultar, como mostra a Figura 4.25. Para isso, utiliza-se o comando `textInput("idd", h5("Insira o Id da Disciplina:"))`.

Figura 4.25: Widget para inserir o id da disciplina

Para gerar as abas do painel principal (Figura 4.26), usa-se `tabsetPanel(type = "tabs"` no `script ui.R`, mas também é necessário dizer no `server.R` qual procedimento o R deverá retornar dentro de cada uma das abas. Como mostra a programação a seguir.

```

ui.R
##### Disciplina #####
mainPanel(
  tabsetPanel(type = "tabs",
    tabPanel("Categorizadas",
      h4("Frequência das respostas categorizadas"),
      verbatimTextOutput("freqdis")),
    tabPanel("Dissertativas",
      h4("Respostas dissertativas dadas para a Disciplina"),
      tableOutput("respdis"))
  )
)

Server.R
output$freqdis <- renderPrint({
  lista3()
})
output$respdis <- renderTable({
  selectedData()[proc31(),c(8,10:13,4,6,9)]
})

```



Figura 4.26: Abas do painel principal

Depois de inserir o id da disciplina, basta clicar no botão Aplicar. Assim, no painel principal, o programa retornará duas saídas, a primeira é uma lista com a frequência, a média e desvio-padrão das variáveis qualitativas ordinais (Figura 4.27), e a segunda é uma tabela com todas as respostas que a disciplina recebeu dos alunos que a cursaram (Figura 4.28).



Figura 4.27: Frequência das variáveis categorizadas

Categorizadas **Dissertativas**

**Respostas dissertativas dadas para a Disciplina**

Ponto.Forte.da.Disciplina	Ponto.a.Melhorar.da.Disciplina	Ponto.Forte.do.Professor	Ponto.a.Melhorar.do.Professor
excelentes métodos de ensino paciência por parte do professor sem disposto a tirar dúvidas e ajudar o aluno	pele fato de a aula ser no anfiteatro às vezes se tornava um pouco complicado para todos tirarem dúvidas		
			esta avaliação é lida pelo menos os professores tem acesso
	didática da professora exemplos em sala de aula e adequação à aplicabilidade		
			a professora não tem didática e suas provas nem sempre correspondem ao conteúdo passado em sala e nas revisões de listas
		domina a matéria	a didática
		dinamico e direto	tratar o conteudo com um pouco mais de fundamentalismo

Figura 4.28: Respostas recebidas por uma determinada disciplina

#### 4.3.4 Aluno

Nessa área é possível ter acesso a todas as respostas (avaliação) feitas por um determinado aluno, sendo possível visualizar tanto a frequência, a média e o desvio-padrão das respostas categorizadas, quanto as respostas dissertativas.

**Análise do Questionário de Avaliação Docente** Tutorial Frequência: Resposta:

Unidade Acadêmica  
Professor  
Disciplina  
**Aluno**

Categorizadas Dissertativas

**Resposta Dada pelo Aluno**

Insira a Matrícula do Aluno:  
-  
Aplicar

**Frequência das respostas categorizadas**

\$^1.1 Clareza no Programa`

1	2	3	4	5	Média	Dp
0	0	0	0	0	NaN	NA

\$^1.2 Coerência na Disciplina`

1	2	3	4	5	Média	Dp
0	0	0	0	0	NaN	NA

Figura 4.29: Página da análise de respostas dadas pelo aluno

Essa parte do *script* ui.R apresenta a função `tabPanel("Aluno",` que cria dentro

da barra Resposta a opção de página chamada Aluno e, ao clicar nela, irá direto para a página da Figura 4.29.

```
ui.R
##### Aluno #####
  tabPanel("Aluno",
    headerPanel(h3("Resposta Dada pelo Aluno")),
    sidebarLayout(
      sidebarPanel(
        textInput("ida", h5("Insira a Matrícula do Aluno:"),
          value = "-"),
        submitButton("Aplicar")
      ),
    ),
```

Para criar a barra lateral da Figura 4.29, usa-se o comando *sidebarPanel*. Dentro dessa barra, encontra-se um *widget* que permite inserir a matrícula do aluno que se deseja consultar, como mostra a Figura 4.30. Para isso, utiliza-se o comando *textInput("ida", h5("Insira a Matrícula do Aluno:"))*.

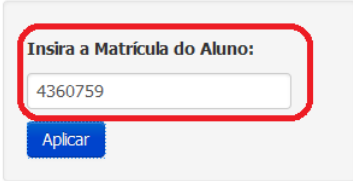


Figura 4.30: Widget para inserir a matrícula do aluno

Para gerar as abas do painel principal (Figura 4.31), usa-se *tabsetPanel(type = "tabs"* no *script* ui.R, mas também é necessário dizer no *server.R* qual procedimento o R deverá retornar dentro de cada uma das abas. Como mostra a programação a seguir.

```
ui.R
  mainPanel(
    tabsetPanel(type = "tabs",
      tabPanel("Categorizadas",
        h4("Frequência das respostas categorizadas"),
        verbatimTextOutput("freqa")),
      tabPanel("Dissertativas",
        h4("Respostas dissertativas dadas pelo aluno"),
        tableOutput("respa"))
    )
  )

Server.R
output$freqa <- renderPrint({
  lista4()
})
output$respa <- renderTable({
  selectedData()[proc41(),c(5,10:13,4,7:9)]
})
```

~\Meus documentos\faculdade-UnB\Monografia\Programa\Shiny\Programa Geral1 - Shiny

http://127.0.0.1:6880 | Open in Browser | Publish

## Análise do Questionário de Avaliação Docente

Tutorial    Frequência:    Resposta:

### Resposta Dada pelo Aluno

Insira a Matrícula do Aluno:

4360759

Aplicar

Categorizadas    Dissertativas

#### Frequência das respostas categorizadas

```

$`1.1 Clareza no Programa`
  1   2   3   4   5 NA's Média  Dp
0   0   0   0   0   89  NaN  NA

$`1.2 Coerência na Disciplina`
  1   2   3   4   5 NA's Média  Dp
0   0   0   0   0   89  NaN  NA

```

Figura 4.31: Abas do painel principal

Depois de inserir a matrícula do aluno, basta clicar no botão Aplicar. Assim, no painel principal, o programa retornará duas saídas, a primeira é uma lista com a frequência, a média e desvio-padrão das variáveis qualitativas ordinais (Figura 4.32), e a segunda é uma tabela com todas as respostas dadas pelos alunos (Figura 4.33).

Categorizadas    Dissertativas

#### Frequência das respostas categorizadas

```

$`2.1 Domínio do Conteúdo`
  1   2   3   4   5 NA's Média  Dp
0.00 0.00 6.00 19.00 63.00 1.00 4.65 0.61

$`2.2 Clareza na Transmissão`
  1   2   3   4   5 Média  Dp
1.00 1.00 9.00 28.00 50.00 4.40 0.81

$`2.3 Adequação das Atividades`
  1   2   3   4   5 NA's Média  Dp
1.00 2.00 21.00 27.00 37.00 1.00 4.10 0.92

$`2.4 Dispersar o interesse`
  1   2   3   4   5 Média  Dp
2.00 5.00 16.00 29.00 37.00 4.06 1.02

```

Figura 4.32: Frequência das variáveis categorizadas



Categorizadas

Dissertativas

**Respostas dissertativas dadas pelo aluno**

MatriculaAluno	Ponto.Forte.da.Disciplina	Ponto.a.Melhorar.da.Disciplina	Ponto.Forte.do.Professor	Ponto.a.Melhorar.do.Professor
4360759			a professora nos ensinou bastante durante todo o semestre sobre técnicas didáticas para lidar com diferentes grupos de alunos	a professora dificilmente respeita os horários tanto de início quanto de término das aulas houve desorganização quanto aos conteúdos ministrados durante o semestre
4360759			nada a declarar	nada a declarar
4360759			ela sabe de verdade o que está ministrando	pontualidade e assiduidade
4360759			boa didática	organização
4360759			domínio do conteúdo e boas estratégias para passar aos alunos ótimas avaliações onde aplica-se muito bem a prática	

Figura 4.33: Respostas dadas por um determinado aluno

## 5 Considerações Finais

A avaliação docente, segundo o relatório de Autoavaliação Institucional 2013, tem subsidiado os departamentos e as faculdades no planejamento e na distribuição das disciplinas, além de ser utilizada como um instrumento de avaliação para fins de validação do Estágio Probatório, Progressão na Carreira Docente, e como forma de avaliação dos cursos pelos avaliadores externos do MEC.

Identificada a relevância que o questionário de avaliação docente possui para a Universidade de Brasília, no presente estudo optou-se por criar uma ferramenta que permitisse a análise desses dados, principalmente das variáveis dissertativas, que ainda não são analisadas pela instituição. Para o desenvolvimento desse aplicativo, utilizou-se o pacote *Shiny* do *software* estatístico R.

Um ponto importante que levou ao uso do pacote *Shiny* do R para a criação do programa, é a escassez de profissionais no mercado que dominem alguma linguagem de programação. Pois, como foi possível ver ao longo desse trabalho, o programa criado por meio do *Shiny* permite ao usuário realizar toda a análise sem esse conhecimento prévio de programação, o que se mostra como uma grande vantagem em relação a muitos outros *softwares*.

Mas as vantagens do uso desse aplicativo não se baseiam apenas nisso, pois existem diversas outras que são importantes, como o fato do programa ter uma interface agradável, ser visualmente moderna e por ter um aspecto *clean*, que permite facilmente o aprendizado e a localização dos itens desejados. Isso faz com que pessoas com pouca habilidade e intimidade com funções computacionais possam utilizar o programa de análise de forma fácil e imediata.

Sabe-se que qualquer gasto financeiro que possa ser evitado por uma instituição pública ou privada é extremamente importante, então, a utilização do programa criado permite o corte de gastos em uso de *softwares* de análise estatística, pois utiliza o *software R-Project* que é livre e gratuito para qualquer pessoa física ou jurídica.

Esse aplicativo criado é uma primeira versão, logo, vale ressaltar que ele pode ser sempre aprimorado de forma que venha a atender cada vez mais as necessidades da

---

Universidade de Brasília. E, além disso, o programa pode ser adaptado e vir a ser utilizado por qualquer outra instituição de ensino que tenha interesse em uma ferramenta poderosa, simples e gratuita que realize análise estatística de banco de dados textuais.

## Referências Bibliográficas

- [1] AGGARWAL, C. ZHAI, C. **Mining Text Data**. Heidelberg: Springer, 2012.
- [2] ARANHA, C. PASSOS, E. A Tecnologia de Mineração de Textos. **RESI- Revista Eletrônica de Sistemas de Informação**, N°2, 2006.
- [3] AZEVEDO, B. F. T.. **MineraFórum**: um recurso de apoio para análise qualitativa em fórum de discussão. 2011.
- [4] BRASIL, Lei N° 10.861, de 14 de abril em 2004. Institui o Sistema Nacional de Avaliação da Educação Superior SINAES e dá outras providências. Brasília, 2004. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/ato2004-2006/2004/lei/110.861.htm](http://www.planalto.gov.br/ccivil_03/ato2004-2006/2004/lei/110.861.htm). Acesso em: 18 setembro 2014.
- [5] CORRÊA, G. N.; MARCACINI, R. M; REZENDE, S. O. **Mineração de textos na análise exploratória de artigos científicos**. 2012
- [6] EBECKEN, N; LOPES, M; COSTA, M. Mineração de Textos, REZENDE, S. O., editor, **Sistemas Inteligentes: Fundamentos e Aplicações**. Manole, 2003, Pág 337 a 370.
- [7] FELDMAN, R; SANGER, J. **The Text Mining Handbook**: Advanced Approaches in Analyzing Unstructured Data. New York: Cambridge University Press, 2007.
- [8] FERNANDES, D. **Avaliação do desempenho docente**: desafios, problemas e oportunidades. Lisboa: Texto Editores, 2008.
- [9] FRIEDL, J. E. F. **Mastering Regular expressions**. O'Reilly Media, Inc, 2006.
- [10] HOTH, A; NÜRNBERGER, A; PAAB, G. **A Brief Survey of Text Mining**. 2005.
- [11] INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA. **Objetivos**. Brasília 2011. Disponível em: <http://portal.inep.gov.br/superior-sinaes-objetivos>. Acesso em: 18 setembro 2014.

- [12] JARGAS, A. M. **Expressões regulares : uma abordagem divertida**. São Paulo : Novatec Editora, 2012.
- [13] KOTLER, P. **Administração e Marketing**. 5 ed. São Paulo: Atlas, 1998.
- [14] MORAIS, E. A. M. AMBRÓSIO, A. P. L. **Mineração de Textos**. 2007.
- [15] REZENDE, S. O.; PUGLIESI, J. B.; MELANDA, E. A.; PAULA, M. F. Mineração de dados. REZENDE, S. O., editor, **Sistemas Inteligentes: Fundamentos e Aplicações**. Manole, 2003, Pág. 307 a 335.
- [16] R-PROJECT. **The R Project for Statistical Computing**. Disponível: <http://www.r-project.org/>. Acesso em: 29 de maio de 2015.
- [17] RSTUDIO. **Shiny by Rstudio**. Disponível: <http://shiny.rstudio.com/>. Acesso em: 29 de maio de 2015.
- [18] RSTUDIO. **Welcome to Rstudio**. Disponível: <http://www.rstudio.com/>. Acesso em: 29 de maio de 2015.
- [19] SANCHEZ, G. **Handling and Processing Strings in R**. Berkeley, Trowchez Editions, 2013.
- [20] SCHLEICH, A. L.; POLYDORO, S. A. J.; SANTOS, A. A. Escala de satisfação com a experiência acadêmica de estudantes do ensino superior. **Avaliação Psicológica, Itatiba**, v. 5, n. 1, p. 11-20, 2006.
- [21] SOUZA, S. A.; REINERT J. N. Avaliação de um Curso de Ensino Superior Através da Satisfação/Insatisfação Discente. **Avaliação (Campinas; Sorocaba)**, São Paulo, v. 15, n. 1, p. 159-176, 2010.
- [22] UNIVERSIDADE DE BRASÍLIA. **Relatório de autoavaliação institucional 2011**. Brasília, 2012. Disponível em: [http://www.dpo.unb.br/documentos/relatorio\\_autoavaliacao\\_2011.pdf](http://www.dpo.unb.br/documentos/relatorio_autoavaliacao_2011.pdf). Acesso em: 19 setembro 2014.
- [23] UNIVERSIDADE DE BRASÍLIA. **Relatório de autoavaliação institucional 2013**. Brasília, 2014. Disponível em: [http://www.dpo.unb.br/documentos/relatorio\\_autoavaliacao\\_2013.pdf](http://www.dpo.unb.br/documentos/relatorio_autoavaliacao_2013.pdf). Acesso em: 19 setembro 2014.

- [24] UNIVERSIDADE DE BRASÍLIA. **Avaliação de professores será feita online.** Brasília, 2011. Disponível em: <http://www.unb.br/noticias/unbagencia/unbagencia.php?id=5580>. Acesso em: 18 setembro 2014.
- [25] WATT, A. **Beginning regular expressions.** Wiley Publishing, Inc; 2005.

**I Anexo - Questionário de percepção  
discente sobre a disciplina, o desempenho  
docente e as condições de oferta (Modelo da  
Universidade de Brasília)**





		1	2	3	4	5	NA
2.7	Integração entre teoria, pesquisa, prática e aspectos da realidade.						
2.8	Coerência entre nível de complexidade das avaliações (trabalho, testes, provas, exercícios, etc.) e o conteúdo ministrado.						
2.9	Discussão dos resultados de avaliação de aprendizagem.						
2.10	Disponibilidade para esclarecer dúvidas e solucionar dificuldades dos alunos relacionados ao conteúdo da disciplina.						
2.11	Pontualidade no cumprimento dos horários de início e término das aulas, pelo professor						
2.12	Assiduidade (Cumprimento do calendário acadêmico estabelecido pelo CEPE)						

Relacione pontos fortes e pontos a melhorar neste quesito:

PONTOS FORTES	PONTOS A MELHORAR

### 3. Autoavaliação

		1	2	3	4	5	NA
3.1	Participação nas atividades desenvolvidas na disciplina.						
3.2	Estudo extraclasse do conteúdo da disciplina						
3.3	Aprofundamento do conteúdo da disciplina por meio de pesquisa bibliográfica e leitura						
3.4	Capacidade de aplicar os conhecimentos da disciplina em outras situações e contextos.						
3.5	Relacionamento com colegas da disciplina.						
3.6	Relacionamento com professor.						
3.7	Pontualidade no cumprimento dos horários de início e término das aulas.						
3.8	Assiduidade (Presença nas atividades desenvolvidas na disciplina)						

Relacione pontos fortes e pontos a melhorar neste quesito:

PONTOS FORTES	PONTOS A MELHORAR

#### 4. Apoio Institucional à Disciplina

		1	2	3	4	5	NA
4.1	Qualidade das instalações destinadas às aulas teóricas.						
4.2	Qualidade das instalações destinadas às aulas práticas.						
4.3	Acesso à bibliografia da disciplina em bibliotecas da UnB.						
4.4	Disponibilidade de equipamentos.						
4.5	Demais condições necessárias ao desenvolvimento das atividades da disciplina						

Relacione pontos fortes e pontos a melhorar neste quesito:

PONTOS FORTES	PONTOS A MELHORAR

## II Anexo - Programação

```

                                Organização Banco de Dados
##### Pacote exigidos #####
require(stringr)

##### Importando arquivos #####

# Banco de dados
dados <- read.csv("C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/avaliacaoDEG.csv"
,sep=";")

#####
##### Organizando a Base Dados #####
#####

# Retirando variáveis desnecessárias
names(dados)
dados <- dados[,-c(25:40,46)]

# Renomeando as variáveis
names(dados)
nomes <- c("idDepartamento", "Sigla", "Departamento", "MatriculaAluno",
"IdProfessor", "IdDisciplina", "Disciplina", "Turma",
"1.1 Clareza no Programa", "1.2 Coerência na Disciplina",
"1.3 Clareza no critério de Avaliação", "1.4 Adequação da
Bibliografia",
"1.5 Relevância da Disciplina", "Ponto Forte da Disciplina", "Ponto a
Melhorar da Disciplina",
"2.1 Domínio do Conteúdo", "2.2 Clareza na Transmissão", "2.3
Adequação das Atividades",
"2.4 Dispersar o interesse", "2.5 Estratégia de Ensino", "2.6 Lidar
com Divergência",
"2.7 Integração", "2.8 Coerência Prova e Conteúdo", "2.9 Discussão
dos Resultados",
"2.10 Disponibilidade", "2.11 Pontualidade", "2.12 Assiduidade",
"Ponto Forte do Professor",
"Ponto a Melhorar do Professor")
names(dados) <- nomes

# Separando os dados em questões categorizados e dissertativas
names(dados)
base.cat <- dados[,c(1:13,16:27)]
base.dis <- dados[,c(1:8,14,15,28,29)]

##### Organizando a Base.cat #####

# Retirando qualquer valor diferente 1 a 5 e NA
names(base.cat)
for (i in 9:25){
  base.cat[,i] <- ifelse(base.cat[,i]>5,NA,base.cat[,i])
}

# Substituindo Educação Física por Ed. Física
var_dep <- gsub("Faculdade de Educação Física", "Faculdade de Ed. Física",
base.cat[,3])
base.cat[,3] <- var_dep

# Retirando observações com menos de 50% respondido
q11 <- is.na(base.cat[,9])
q12 <- is.na(base.cat[,10])
q13 <- is.na(base.cat[,11])
q14 <- is.na(base.cat[,12])
q15 <- is.na(base.cat[,13])
q21 <- is.na(base.cat[,14])
q22 <- is.na(base.cat[,15])
q23 <- is.na(base.cat[,16])

```

## Organização Banco de Dados

```
q24 <- is.na(base.cat[,17])
q25 <- is.na(base.cat[,18])
q26 <- is.na(base.cat[,19])
q27 <- is.na(base.cat[,20])
q28 <- is.na(base.cat[,21])
q29 <- is.na(base.cat[,22])
q210 <- is.na(base.cat[,23])
q211 <- is.na(base.cat[,24])
q212 <- is.na(base.cat[,25])
filtro <- q11+q12+q13+q14+q15+q21+q22+q23+q24+q25+q26+q27+q28+q29+q210+q211+q212
base.cat <- base.cat[filtro<9,]
```

```
##### Organizando a Base.dis #####
```

```
# Criando função para organizar a base.dis
org <- function(q){
  # Retirando células que apresentaram Erro:509
  q <- gsub(pattern = "Erro:509", replacement = "", q)
  # Retirando pontuação
  q <- gsub(pattern = "[[:punct:]]", replacement = " ", q)
  # Colocando tudo em letra minúscula
  q <- tolower(q)
  # Retirando células com apenas números
  q <- gsub(pattern = "[[:digit:]]", replacement = "", q)
  return(q)
}
```

```
# Aplicando a função nas quatro variáveis de interesse
names(base.dis)
var_mod <- apply(base.dis[,c(9:12)],2,org)
```

```
# Gerando o banco de dados com as variáveis modificadas
base.dis[,c(9:12)] <- var_mod
```

```
# Substituindo Educação Física por Ed. Física
var_dep <- gsub("Faculdade de Educação Física", "Faculdade de Ed. Física",
base.dis[,3])
base.dis[,3] <- var_dep
```

```
# Retirando observações em branco
q16 <- base.dis[,9] == ""
q17 <- base.dis[,10] == ""
q213 <- base.dis[,11] == ""
q214 <- base.dis[,12] == ""
filtro <- q16+q17+q213+q214
base.dis <- base.dis[filtro!= 4,]
```

```
# Identificando observações NA
q16 <- is.na(base.dis[,9])
q17 <- is.na(base.dis[,10])
q213 <- is.na(base.dis[,11])
q214 <- is.na(base.dis[,12])
filtro <- q16+q17+q213+q214
base.dis <- base.dis[filtro!=4, ]
```

```
#####
##### salvar alguma planilha #####
#####
```

```
write.csv2(base.cat, "C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/base.cat.csv")
```

## Organização Banco de Dados

```
write.csv2(base.dis, "C:/Users/ThiagoMoreti/Documents/Meus  
documentos/facudade-UnB/Monografia/Programação/Dados/base.dis.csv")
```

## Normalização Banco de Dados

```
##### Pacote exigidos #####
require(stringr)

##### Importando arquivos #####

# Banco de dados
base.dis <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus
documentos/faculdade-UNB/Monografia/Programação/Dados/base.dis.csv")

##### Normalização Morfológica #####
norm <- function(q){

  q <- gsub("alunos", "aluno", q)
  q <- gsub("aulas", "aulas", q)
  q <- gsub("avaliações", "avaliação", q)
  q <- gsub("avalio", "avaliar", q)
  q <- gsub("bibliografias", "bibliografia", q)
  q <- gsub("bo[a,m,n]s*", "bom", q)
  q <- gsub("claramente", "clareza", q)
  q <- gsub("claros?", "clareza", q)
  q <- gsub("clara", "clareza", q)
  q <- gsub("coerência", "coerente", q)
  q <- gsub("coerentes", "coerente", q)
  q <- gsub("compreendam", "compreensão", q)
  q <- gsub("consegue", "conseguir", q)
  q <- gsub("conseguisse", "conseguir", q)
  q <- gsub("conseguiu", "conseguir", q)
  q <- gsub("conteúdos", "conteúdo", q)
  q <- gsub("conteudo", "conteúdo", q)
  q <- gsub("cursos", "curso", q)
  q <- gsub("did[a,á]tic[a,o]", "didática", q)
  q <- gsub("dinâmicas", "dinâmica", q)
  q <- gsub("disciplinas", "disciplina", q)
  q <- gsub("domina", "domínio", q)
  q <- gsub("entendeu", "entender", q)
  q <- gsub("explica[c,ç][o,ô]es", "explicação", q)
  q <- gsub("matérias", "matéria", q)
  q <- gsub("materia", "matéria", q)
  q <- gsub("melhores", "melhor", q)
  q <- gsub("ministrou", "ministrar", q)
  q <- gsub("paciencia", "paciência", q)
  q <- gsub("práticas", "prática", q)
  q <- gsub("professora", "professor", q)
  q <- gsub("professores", "professor", q)
  q <- gsub("prof ", "professor ", q)
  q <- gsub("profissionais", "profissional", q)
  q <- gsub("provas", "prova", q)
  q <- gsub("relevância", "relevante", q)
  q <- gsub("trabalhos", "trabalho", q)

  return(q)
}

#Aplicando a função nas quatro variáveis de interesse
names(base.dis)
var_norm <- apply(base.dis[,c(10:13)],2,norm)

#Gerando o banco de dados com as variáveis modificadas
base.dis[,c(10:13)] <- var_norm
base.dis <- base.dis[, -c(1)]

##### Salvar alguma planilha #####
```

## Normalização Banco de Dados

```
write.csv2(base.dis, "C:/Users/ThiagoMoreti/Documents/Meus  
documentos/facudade-UnB/Monografia/Programação/Dados/base.norm.csv")
```



## Análise de Frequência

```
##### Pacote exigidos #####
require(stringr)
require(wordcloud)

##### Importando arquivos #####

# Banco de dados
base.dis <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/base.dis.csv")
base.cat <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/base.cat.csv")

# Banco de dados normalizado
base.norm <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/base.norm.csv")

# Stoplist
stoplist <- scan(file="C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/Stoplist.txt",
                what=character(0), sep="\n")

#####
##### Análise de frequência das respostas #####
##### Disciplina e Professor #####
#####

# Criando a função func para encontrar as frequências e as respostas
func <- function(var_id,var_resp){
  id <- base.dis[,var_id]
  variavel <- base.dis[,var_resp]
  coment <-grep("[a-z]", variavel)
  soma <- sum(coment)
  freq <- sort(summary(as.factor(id[coment]),maxsum=3664),decreasing= TRUE)
  resp <- as.factor(base.dis[coment,var_resp])
  return(list(resp, freq ,soma))
}

# Criando a função cloud para retornar a wordcloud
cloud <- function(var_id,var_resp){
  id <- base.dis[,var_id]
  variavel <- base.dis[,var_resp]
  coment <-grep("[a-z]", variavel)
  freq <- sort(summary(as.factor(id[coment]),maxsum=3664))
  cor <- brewer.pal(8,"Dark2")
  if(var_resp == 12 | var_resp == 13){
    w = 4
  } else{
    w = 1
  }
  if(var_id == 7){
    k = 4
  } else{
    k = 2
  }
  wordcloud(names(freq),freq, scale=c(k,.2), min.freq= w,
           max.words=Inf, random.order=FALSE, rot.per=.15,
           random.color=T, colors=cor)
}

# Criando a função resp para retornar as respostas por matrícula da disciplina
# ou do professor
resp <- function(id){
  proc <- sapply(base.dis[,6],is.element,id)
  ret <- base.dis[proc,-c(1:5)]
  return(ret)
}
```

## Análise de Frequência

```
# Colocar o número da coluna da variável
names(base.dis)
func(6,10)
cloud(7,13)
resp()

#####
##### Criando lista de palavras #####
#####

# Selecionando a variável
q <- base.norm$Ponto.Forte.da.Disciplina

# Separando as observações em palavras
palavras <- str_split(q, pattern = " ")

# Criando um vetor(lista) com todas as palavras
vetor <- unlist(palavras)

# Eliminando da lista as palavras repetidas
sem_rep <- unique(vetor)

# Ordenando as palavras
ordenado <- order(sem_rep)
sem_rep_ord <- sem_rep[ordenado]

# Retirando células em branco
branco <- sem_rep_ord == ""
sem_rep_ord <- sem_rep_ord[!branco]

#Criando tabela com as palavras sem repetição
sem_rep_ord1 <- as.data.frame(sem_rep_ord)

# Total de palavras
total <- length(sem_rep_ord)
total

#####
##### Retirando stopwords #####
#####

# Comparando Banco de dados com a stoplist
prog <- sapply(sem_rep_ord, is.element, set=stoplist)

# Retirando as stopwords
free_stopword <- sem_rep_ord[!prog]

# Criando um data.frame das palavras sem stopwords
sem_rep_ord2 <- as.data.frame(free_stopword)

# Total de palavras no banco sem stopwords
total1 <- length(free_stopword)
total1

# Proporção da redução com a organização do banco de dados
prop <- 1 - (total1/total)
prop

#####
##### Análise de frequência de palavras #####
#####

# Vetor de contagem
qntd <- rep(0, total1)
```

## Análise de Frequência

```
# número de ocorrências
for (i in 1:total) {
  qntd[i] = sum(vetor == free_stopword[i])
}
qntd

# Colocando o valor do índice em ordem decrescente
ord_30 <- order(qntd, decreasing = TRUE)[1:30]

# Selecionando as 30 maiores frequências
freq_30 <- sort(qntd, decreasing = TRUE)[1:30]

# Selecionando 30 palavras com maiores frequências
words_30 <- free_stopword[ord_30]
words_30

# tabela
tabela <- matrix(c(words_30, freq_30), ncol=2, nrow=30, byrow=F)
colnames(tabela) <- c("Palavras", "Frequência")
rownames(tabela) <- c(1:30)
tabela <- as.data.frame(tabela)

# barplot
barplot(freq_30, border = NA, names.arg = words_30,
        las = 2, ylim = c(0,20), cex.names=0.6, cex.axis=0.8,
        main="Frequência da variável ponto forte da disciplina",
        cex.main=0.9)

# wordcloud
cor <- brewer.pal(8, "Dark2")
wordcloud(free_stopword, qntd, scale=c(3,.2), min.freq=5,
          max.words=Inf, random.order=FALSE, rot.per=.15,
          random.color=T, colors=cor)

#####
##### Análise de freq das categorias #####
#####

# Criando os níveis para as variáveis
for (i in 10:26){
  base.cat[,i] <- as.factor(base.cat[,i])
  levels(base.cat[,i])<- c(1,2,3,4,5)
}

# Criando uma função que retorna uma lista das freq, médias e dp
freqc <- function(id){
  proc <- sapply(base.cat[,6], is.element, id)
  j<-1
  lista <-list()
  for (i in 10:26){
    lista[[j]] <-
    round(c((summary(base.cat[proc,i])), mean(as.numeric(base.cat[proc,i]), na.rm =
T),
          sd(as.numeric(base.cat[proc,i]), na.rm = T)), 2)
    if (length(lista[[j]]) < 8){
      names(lista[[j]])[6:7] <- c("Média", "Desvio-padrão")
    } else{
      names(lista[[j]])[7:8] <- c("Média", "Desvio-padrão")
    }
    j <- j+1
  }
  return(lista)
}

freqc()
```

```

                                Shiny Ui
##### Pacote exigidos #####
require(shiny)

#####

shinyUI(navbarPage(h3("Análise do Questionário de Avaliação Docente"),

##### Tutorial
#####

      tabPanel(h5("Tutorial"),
                includeMarkdown("Tutorial.md"),
                br()
      ),

##### Respostas
#####

      navbarMenu(h5("Frequência:"),
                 tabPanel("Respostas",
                           sidebarLayout(
                             sidebarPanel(
                               helpText("Escolha a variável e a
categoria desejada,",
                                     "depois basta escolher a
frequência mínima e",
                                     "o número máximo de palavras
que devem aparecer",
                                     "na nuvem de palavras.
Depois de selecionar",
                                     "as opções desejadas aperte
em Aplicar."),
                               selectInput("id", h5("Selecione a
variável a ser explicada:"),
                                     choices = c("Disciplina"
                                               "Professor" =
                                               "Aluno" =
                                               "MatriculaAluno")),
                               selectInput("categoria",
                                     choices = c("Ponto forte
                                               "Ponto a
                                               "Ponto forte
                                               "Ponto a
da disciplina" = "Ponto.Forte.da.Disciplina",
melhorar da disciplina" = "Ponto.a.Melhorar.da.Disciplina",
do professor" = "Ponto.Forte.do.Professor",
melhorar do professor" = "Ponto.a.Melhorar.do.Professor")),
                               sliderInput("f",
                                     h5("Frequência Mínima:"),
                                     min = 1, max = 10, value
= 2, step = 1),
                               sliderInput("max",
                                     h5("Número máximo de
palavras:"),
                                     min = 0, max = 50,
value = 25, step= 5),
                               submitButton("Aplicar")
                             )
                 ),
    )

```

```

shiny ui
  mainPanel(
    h4("Núvem de palavras"),
    plotOutput("wordcloud1"),

    h4("Frequência de respostas"),
    verbatimTextOutput("freq")
  )
),

##### Palavras
#####

  tabPanel("Palavras",
    sidebarLayout(
      sidebarPanel(
        helpText("Para realizar a análise de
frequência das palavras",
"segundo as categorias,
basta selecionar a categoria",
"que deseja e o número
máximo que a wordcloud, o gráfico",
"de barras e a tabela devem
apresentar. Após isso clique em aplicar."),
selectInput("categoria1",
h5("Selecione a categoria:"),
da disciplina" = "Ponto.Forte.da.Disciplina",
melhorar da disciplina" = "Ponto.a.Melhorar.da.Disciplina",
do professor" = "Ponto.Forte.do.Professor",
melhorar do professor" = "Ponto.a.Melhorar.do.Professor")),
sliderInput("max1",
h5("Número máximo de
palavras:"),
value = 25, step= 5),
submitButton("Aplicar")
),
mainPanel(
  tabsetPanel(type = "tabs",
    tabPanel("wordcloud",
plotOutput("wordcloud2")),
    tabPanel("Gráfico de
Barras",
plotOutput("hist")),
    tabPanel("Tabela",
tableOutput("tabela")),
    h4("Total de palavras na categoria"),
    textOutput("total1")
  )
)

```

```

shiny ui
)
)
),
##### Departamento
#####
navbarMenu(h5("Resposta:"),
            tabPanel("Unidade Acadêmica",
                    headerPanel(h3("Resposta por Unidade
Acadêmica")),
                    sidebarLayout(
                        sidebarPanel(
                            selectInput("depart", h5("Escolha a
Unidade Acadêmica:"),
                                choices = c("Decanato de
Pesquisa e Pós-Graduação", "Direção da
Faculdade de Arquitetura e Urbanismo", "Direção da
Faculdade de Ciências da Saúde", "Direção do
Instituto Ciências Biológicas", "Direção do
Instituto de Psicologia", "Faculd. de
Economia, Administração e Contabilidade", "Faculdade
de Agronomia e Medicina Veterinária", "Faculdade
de Ciência da Informação", "Faculdade
de Ciências da Saúde", "Faculdade
de Comunicação", "Faculdade
de Educação", "Faculdade
de Educação Física" = "Faculdade de Ed. Física",
"Faculdade
de Estudos Sociais Aplicados", "Faculdade
de Medicina", "Faculdade
de Tecnologia", "Gabinete do
Reitor", "Instituto
de Artes", "Instituto
de Ciências Biológicas", "Instituto
de Ciências Exatas", "Instituto
de Ciências Humanas", "Instituto
de Ciências Sociais", "Instituto
de Física", "Instituto
de Geociências", "Instituto
de Letras"))),
submitButton("Aplicar")

```

```

shiny ui
),
mainPanel(
  tabsetPanel(type = "tabs",
    tabPanel("Categorizadas",
      h4("Frequência
das respostas categorizadas"),
verbatimTextOutput("freqd")),
    tabPanel("Dissertativas",
      h4("Respostas
dissertativas por unidade acadêmica"),
tableOutput("respd"))
  )
),
)),
##### Professor
#####
tabPanel("Professor",
  headerPanel(h3("Resposta Recebida pelo
Professor")),
  sidebarLayout(
    sidebarPanel(
      textInput("idp", h5("Insira o Id do
Professor:")),
      value = "-"),
      submitButton("Aplicar")
    ),
    mainPanel(
      tabsetPanel(type = "tabs",
        tabPanel("Categorizadas",
          h4("Frequência
das respostas categorizadas"),
verbatimTextOutput("freqp")),
        tabPanel("Dissertativas",
          h4("Respostas
dissertativas recebidas pelo Professor"),
tableOutput("respp"))
      )
    ),
  ),
##### Disciplina
#####
tabPanel("Disciplina",
  headerPanel(h3("Resposta Dada para a
Disciplina")),
  sidebarLayout(
    sidebarPanel(
      textInput("idd", h5("Insira o Id da
Disciplia:")),
      value = "-"),

```





## Shiny Server

```
##### Pacote exigidos #####
require(shiny)
require(stringr)
require(wordcloud)
require(markdown)

##### Importando arquivos #####

# Banco de dados
base.dis <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/base.dis.csv")
base.cat <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/base.cat.csv")

# Banco de dados normalizado
base.norm <- read.csv2("C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/base.norm.csv")

# Stoplist
stoplist <- scan(file="C:/Users/ThiagoMoreti/Documents/Meus
documentos/facudade-UnB/Monografia/Programação/Dados/Stoplist.txt",
                what=character(0), sep="\n")

#####

# Criando nível para base.cat
for (i in 10:26){
  base.cat[,i] <- as.factor(base.cat[,i])
  levels(base.cat[,i])<- c(1,2,3,4,5)
}

#####
#####
shinyServer(function(input, output, session) {

  selectedData <- reactive({
    base.dis
  })

  selectedData1 <- reactive({
    base.cat
  })

  selectedNorm <- reactive({
    base.norm
  })

  selectedStop <- reactive({
    stoplist
  })

##### Respostas #####

  id <- reactive({
    selectedData()[,input$id]
  })

  var <- reactive({
    selectedData()[,input$categoria]
  })

  coment <- reactive({
    grepl("[a-z]", var())
  })
}
```

## Shiny Server

```
freq <- reactive({
  sort(summary(as.factor(id()[coment()])),maxsum=3664),decreasing=T)
})

output$wordcloud1 <- renderPlot({
  if (max(freq()) < input$f) {
    wordcloud("None", 1, scale=c(4,1), max.words= input$max,
              colors=brewer.pal(8,"Dark2"))}

  else {wordcloud(names(freq()), freq(), scale=c(4,.3), min.freq = input$f,
                 max.words=input$max, random.order=FALSE, rot.per=.15,
                 random.color=T, colors=brewer.pal(8,"Dark2"))}

})

output$freq <- renderPrint({
  freq()
})
```

##### Palavra #####

```
palavras <- reactive({
  q <- selectedNorm()[,input$categorial]
  palavras <- str_split(q, pattern = " ")
})

vetor <- reactive({
  unlist(palavras())
})

sem_rep_ord1 <- reactive({
  sem_rep <- unique(vetor())
  ordenado <- order(sem_rep)
  sem_rep_ord <- sem_rep[ordenado]
  branco <- sem_rep_ord == ""
  sem_rep_ord1 <- sem_rep_ord[!branco]
})

sem_rep_ord2 <- reactive({
  prog <- sapply(sem_rep_ord1(), is.element, set=selectedStop())
  free_stopword <- sem_rep_ord1()[!prog]
})

total <- reactive({
  length(sem_rep_ord1())
})

total1 <- reactive({
  length(sem_rep_ord2())
})

qntd <- reactive({
  qntd <- rep(0, total1())
  for (i in 1:total1()) {
    qntd[i] = sum(vetor() == sem_rep_ord2()[i])
  }
  qntd
})

ord_n <- reactive({
  order(qntd(), decreasing = TRUE)[1:input$max1]
})

freq_n <- reactive({
  sort(qntd(), decreasing = TRUE)[1:input$max1]
})
```

## Shiny Server

```

freqr_n <- reactive({
  relat <- freq_n()/total1()
  round(relat,2)
})

words_n <- reactive({
  sem_rep_ord2()[ord_n()]
})

output$tabela <- renderTable({
  tabela <-
matrix(c(words_n(),freq_n(),freqr_n()),ncol=3,nrow=input$max1,byrow=F)
  colnames(tabela) <- c("Palavras", "Frequência", "Freq Relativa")
  rownames(tabela) <- c(1:input$max1)
  tabela <- as.data.frame(tabela)
})

output$hist <- renderPlot({
  barplot(freq_n(), border = NA, names.arg = words_n(),
    las = 2, ylab = "Frequência",
    ylim = c(0,max(freq_n())),cex.names=0.9, cex.axis=0.8)
})

output$wordcloud2 <- renderPlot({
  wordcloud(words_n(), freq_n(), scale=c(4,.8), min.freq= min(freq()),
    max.words=input$max1, random.order= FALSE, rot.per=.15,
    random.color=T, colors=brewer.pal(8,"Dark2"))
})

output$total1 <- renderPrint({
  total1()
})

##### Departamento #####
proc11 <- reactive({
  grepl(input$depart,selectedData()[,4])
})

proc12 <- reactive({
  grepl(input$depart,selectedData1()[,4])
})

lista1 <- reactive({
  j<-1
  lista <-list("1.1 Clareza no Programa"=NULL,"1.2 Coerência na
Disciplina"=NULL,"1.3.Clareza.no.critério.de.Avaliação"=NULL,
"1.4 Adequação da Bibliografia"=NULL,"1.5 Relevância da
Disciplina"=NULL,"2.1 Domínio do Conteúdo"=NULL,
"2.2 Clareza na Transmissão"=NULL,"2.3 Adequação das
Atividades"=NULL,"2.4 Dispertar o interesse"=NULL,
"2.5 Estratégia de Ensino"=NULL,"2.6 Lidar com
Divergência"=NULL,"2.7 Integração"=NULL,
"2.8 Coerência Prova e Conteúdo"=NULL,"2.9 Discussão dos
Resultados"=NULL,"2.10 Disponibilidade"=NULL,
"2.11 Pontualidade"=NULL,"2.12 Assiduidade"=NULL)
  for (i in 10:26){
    lista[[j]] <-
round(c((summary(selectedData1()[proc12(),i])),mean(as.numeric(selectedData1()[p
roc12(),i]),na.rm = T),
sd(as.numeric(selectedData1()[proc12(),i]),na.rm =
T)),2)
    if (length(lista[[j]]) < 8){
      names(lista[[j]][6:7] <- c("Média","Dp")
    } else{
      names(lista[[j]][7:8] <- c("Média","Dp")
    }
  }
}

```

```

    j <- j+1
  }
  lista
})
output$freqd <- renderPrint({
  lista1()
})
output$respd <- renderTable({
  selectedData()[proc11(),c(4,10:13,6:9)]
})

##### Professor #####
proc21 <- reactive({
  sapply(selectedData()[,6],is.element,input$idp)
})

proc22 <- reactive({
  sapply(selectedData1()[,6],is.element,input$idp)
})

lista2 <- reactive({
  j<-1
  lista <-list("1.1 Clareza no Programa"=NULL,"1.2 Coerência na
Disciplina"=NULL,"1.3.Clareza.no.critério.de.Avaliação"=NULL,
"1.4 Adequação da Bibliografia"=NULL,"1.5 Relevância da
Disciplina"=NULL)
  for (i in 10:14){
    lista[[j]] <-
round(c((summary(selectedData1()[proc22(),i]),mean(as.numeric(selectedData1()[p
roc22(),i]),na.rm = T),
          sd(as.numeric(selectedData1()[proc22(),i]),na.rm =
T)),2)
    if (length(lista[[j]]) < 8){
      names(lista[[j]])[6:7] <- c("Média","Dp")
    } else{
      names(lista[[j]])[7:8] <- c("Média","Dp")
    }
  }
  j <- j+1
}
lista
})
output$freqp <- renderPrint({
  lista2()
})
output$respp <- renderTable({
  selectedData()[proc21(),c(6,10:13,7:9)]
})

##### Disciplina #####
proc31 <- reactive({
  sapply(selectedData()[,7],is.element,input$idd)
})

proc32 <- reactive({
  sapply(selectedData1()[,7],is.element,input$idd)
})

lista3 <- reactive({
  j<-1

```

```

                                Shiny Server
lista <-list("2.1 Domínio do Conteúdo"=NULL,"2.2 Clareza na
Transmissão"=NULL,"2.3 Adequação das Atividades"=NULL,
"2.4 Dispertar o interesse"=NULL,"2.5 Estratégia de
Ensino"=NULL,"2.6 Lidar com Divergência"=NULL,
"2.7 Integração"=NULL,"2.8 Coerência Prova e Conteúdo"=NULL,"2.9
Discussão dos Resultados"=NULL,
"2.10 Disponibilidade"=NULL,"2.11 Pontualidade"=NULL,"2.12
Assiduidade"=NULL)
for (i in 15:26){
  lista[[j]] <-
round(c((summary(selectedData1()[proc32(),i])),mean(as.numeric(selectedData1()[p
roc32(),i]),na.rm = T),
                                sd(as.numeric(selectedData1()[proc32(),i]),na.rm =
T)),2)
  if (length(lista[[j]]) < 8){
    names(lista[[j]][6:7] <- c("Média","Dp")
  } else{
    names(lista[[j]][7:8] <- c("Média","Dp")
  }
  j <- j+1
}
}
lista
})

output$freqdis <- renderPrint({
  lista3()
})

output$respdis <- renderTable({
  selectedData1()[proc31(),c(8,10:13,4,6,9)]
})

##### Aluno #####
proc41 <- reactive({
  sapply(selectedData1()[,5],is.element,input$ida)
})

proc42 <- reactive({
  sapply(selectedData1()[,5],is.element,input$ida)
})

lista4 <- reactive({
  j<-1
  lista <-list("1.1 Clareza no Programa"=NULL,"1.2 Coerência na
Disciplina"=NULL,"1.3.Clareza.no.critério.de.Avaliação"=NULL,
"1.4 Adequação da Bibliografia"=NULL,"1.5 Relevância da
Disciplina"=NULL,"2.1 Domínio do Conteúdo"=NULL,
"2.2 Clareza na Transmissão"=NULL,"2.3 Adequação das
Atividades"=NULL,"2.4 Dispertar o interesse"=NULL,
"2.5 Estratégia de Ensino"=NULL,"2.6 Lidar com
Divergência"=NULL,"2.7 Integração"=NULL,
"2.8 Coerência Prova e Conteúdo"=NULL,"2.9 Discussão dos
Resultados"=NULL,"2.10 Disponibilidade"=NULL,
"2.11 Pontualidade"=NULL,"2.12 Assiduidade"=NULL)
  for (i in 10:26){
    lista[[j]] <-
round(c((summary(selectedData1()[proc42(),i])),mean(as.numeric(selectedData1()[p
roc42(),i]),na.rm = T),
                                sd(as.numeric(selectedData1()[proc42(),i]),na.rm =
T)),2)
    if (length(lista[[j]]) < 8){
      names(lista[[j]][6:7] <- c("Média","Dp")
    } else{
      names(lista[[j]][7:8] <- c("Média","Dp")
    }
  }
}

```

## shiny Server

```
j <- j+1
}
lista
})
output$freqa <- renderPrint({
  lista4()
})
output$respa <- renderTable({
  selectedData()[proc41(),c(5,10:13,4,7:9)]
})
})
```