

Instituto Federal de Educação Ciência e Tecnologia de São Paulo - Campus Birigui

Rangel Vieira Perico

TALK TO ME

Birigui
2022

Sumário

1. O Dicionário de Dados.....	3
2. A biblioteca de Estrutura de Dados.....	6
3. Projeto de Interface.....	8
4. Referências.....	10

1. O Dicionário de Dados

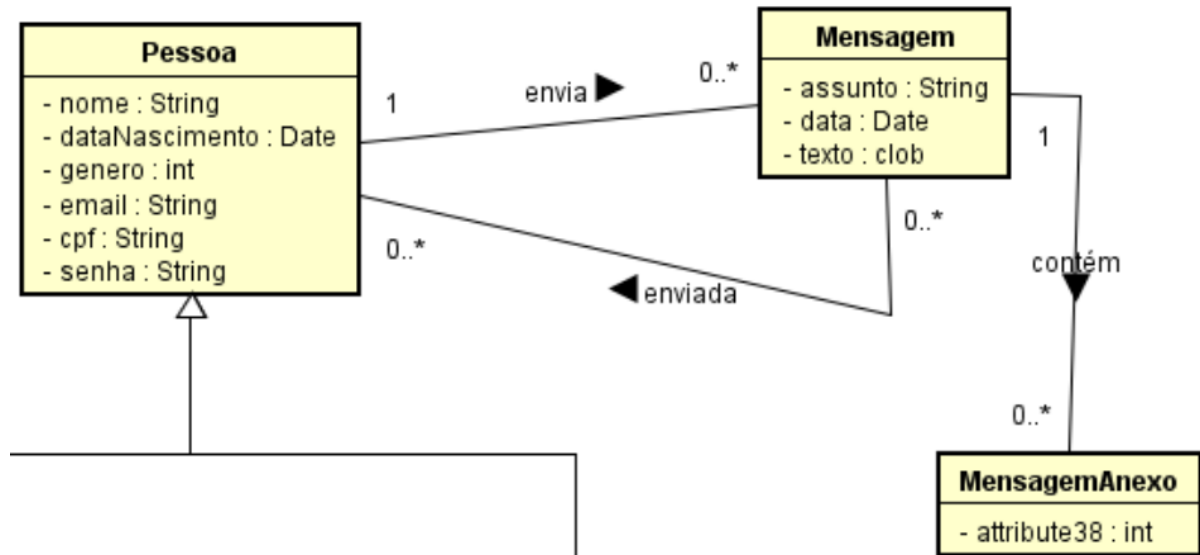
O dicionário de dados é um recurso muito útil quando se necessita catalogar cada função e variável contida em um sistema. Estas estão espalhadas nos mais diversos formatos e tipos, então se torna inviável manter um sistema onde pouquíssimos desenvolvedores possuem pleno domínio. Para resolver isto, uma espécie de dicionário pode ser criado, visando auxiliar e explicar para o que e, como, determinadas funções devem ser utilizadas.

De acordo com o site AGUIA (Agência USP de Gestão da Informação Acadêmica), existem algumas especificações que auxiliam na hora de identificar um dado e dar mais significado a ele. É proposto que se faça uma tabela que conste:

- Nome da variável: Exatamente como foi escrita no código;
- Nome de variável legível: Este serve para especificar para que propósito aquela variável foi criada, no caso, o que ela armazena, exemplo: peso, idade e outros;
- Unidade de medida: Se for tempo, horas, segundos, dias, meses, e outros. Se for temperatura, celsius, graus. E por aí vai;
- Valores permitidos: Como diz o nome, o que aquele campo recebe, são apenas números? apenas letras? quantas letras ou números aceita? e afins;
- Definição da variável: Uma forma descrita de como uma variável deve ser utilizada;
- (Opcional) São citados sinônimos (outros termos cujo tem o mesmo significado da variável, portanto, cria-los seria de certo modo um pleonismo no código);
- (Opcional) Descrição da variável (Descrever detalhadamente o motivo daquela variável e para que serve);

Portanto, seguindo estas dicas de modelagem de projeto e analisando um trecho do seguinte diagrama que foi sugerido em sala de aula:

Figura 1 - Trecho de diagrama proposto em sala de aula, para inclusão de um sistema de mensagens em um projeto.



Fonte: Helen de Freitas Santos¹

É possível criar uma tabela de exemplo, como a seguinte:

Tabela 1 - Dicionário de Dados

Nome	Nome legível	Unidade	Valores permitidos	Definição	Sinônimo	Descrição
nome	Nome do usuário	-	String de até 50 caracteres	Devem ser digitados nomes inteiros separados por espaços	sobrenome, último nome,	Este campo serve para armazenar o nome completo do usuário
dataNascimento	Data de Nascimento do usuário	ano, mês, dia, hora, minuto, segundo	Dados em formato de data, os valores de ano, mês e dia são obrigatórios	Para o formato da data será adotado ano-mês-dia-hora:minuto:segundo	-	Este campo armazena a data de nascimento do usuário
genero	Gênero do usuário	-	Apenas valores do tipo inteiro	Serão digitados números que referem a um gênero	-	-

¹ Disponível em: <https://docs.google.com/presentation/d/1_4DEr6vwFU6redoT8yYc2jrQDgFk-fWB/edit#slide=id.g1267775eaeb_0_16>. Acesso em: 30 abr. 2022.

email	E-mail	-	String de tamanho até 50, deve conter um '@'	Será digitado um endereço email que possua um único '@' em sua composição	-	Armazena o e-mail para contato do usuário
cpf	Cadastro de Pessoa Física	-	String de tamanho até 14	Deverá ser escrito no formato xxx.xxx.xxx-xx	-	Armazena o CPF do usuário
senha	Senha do usuário	-	String de tamanho até 12	A senha deve conter ao menos um número, uma letra maiúscula e minúscula e algum caractere especial	Password, Chave, palavra-passe	Armazena a senha do usuário
assunto	Assunto da mensagem	-	String de tamanho até 100	Uma curta descrição do que o usuário está tratando no corpo da mensagem	Título, Motivo	Recebe o título da mensagem
data	Data de envio	-	Dados em formato de data, os valores de ano, mês e dia são obrigatórios	Para o formato da data será adotado ano-mês-dia-hora:minuto:segundo	Criado em	Recebe a data em que o email foi criado
texto	Texto da mensagem	-	String de tamanho até 1000, anexos	Uma longa descrição da mensagem a ser tratada no e-mail	Mensagem, Conteúdo, Descrição	Armazena o texto a ser enviado
attribute 38	Anexo da mensagem	-	É um tipo especial de estrutura de dados que recebe a extensão, título, datas de criação e edição, autor e tamanho	Será responsável por armazenar os metadados referentes aos anexos incluídos no corpo da mensagem	-	-

Uma sugestão para este modelo seria a inclusão de um campo que indica qual o “parentesco” dessa variável, visto que quando trabalhamos com bancos de dados podem existir campos de mesmo nome em tabelas diferentes, e isso traria problemas de duplicidade, sendo possível apenas descrever a diferença de cada uma pelo campo descrição, que é opcional. Em casos de código fonte, de certa forma haveria o mesmo problema, pois geralmente as linguagens permitem escopos diferentes, nos quais as variáveis internas não interferem no resto do programa, e portanto, podem incluir nomes iguais ao de outras funções.

Outro dado que falta, no caso de bancos de dados, é se o componente é uma chave primária, estrangeira, única ou nenhum deles. Neste caso, a modelagem também não mostrou, então seria interessante ter noção desse campo, para não ter problemas quando os campos tiverem que ser implementados no BD.

2. A biblioteca de Estrutura de Dados

A biblioteca se refere a um elemento que não foi definido originalmente em uma ferramenta, por exemplo, em linguagens de programação temos componentes, criados por usuários, que manipulam o teclado, horário, *threads* e diversas outras coisas em um computador, que não foram implementadas pelos desenvolvedores da mesma. Em nosso caso, temos os anexos, que são estruturas de dados (possuem muitos campos de diferentes tipos, exemplo: inteiros, string, data e outros, todos incluídos no mesmo objeto), dessa forma, não são pré definidos, portanto, fica por nossa parte fazer sua implementação em um banco de dados, por exemplo.

Como a ideia do Talk to Me é armazenar os e-mails, para que possamos enviar para outro usuário, devemos também guardar os dados dos anexos, de forma que seja possível ler quando necessário. Claro, existe a possibilidade de trabalhar esses dados diretamente, colocando-os em pastas e utilizando sua localização relativa quando necessário, mas como foi proposto, é necessário colocá-los no BD.

Primeiramente, devemos especificar o que será salvo. Tomei a iniciativa de olhar os metadados, mais importantes, que os arquivos costumam possuir. Sendo assim, é possível citar:

- Nome;
- Extensão (png, jpg, pdf, etc);
- Data de criação e modificação;
- Tamanho;
- (Em imagens) Altura e largura;

São atributos genéricos, que estão presentes em todos os arquivos (pelo menos no windows), dessa forma fica mais fácil definir o que será necessário para os anexos.

Como não há um modelo (pelo menos não encontrei), vou criar uma estrutura padronizada que tenta abranger os dados contidos na biblioteca de estrutura de dados, com base no dicionário de dados. Primeiramente, existiria uma tabela que diz quais as estruturas de dados contidas no sistema, por exemplo:

Tabela 2 - Exemplo de biblioteca de estrutura de dados

Nome	Ações	Descrição
attribute38	Ler Gravar Apagar Modificar	Manipula os anexos incluídos pelo usuário no envio de mensagens

Fonte: O autor

A tabela acima descreve de forma genérica as estruturas do sistema, em nosso caso, só há a de anexos no trecho relacionado às mensagens. Uma outra tabela explicando cada uma e seus campos poderia ser criada. Por exemplo:

Tabela 3 - Tabela explicando o que é a estrutura de dados do anexo

attribute38	Campos	Nome legível	Valores permitidos	Descrição
	nome	Nome	string de até 1000 caracteres	Recebe o título do arquivo
	extensao	Extensão	string de até 10 caracteres	Armazena qual o tipo do arquivo, seja pdf, png ou outros
	dataCriacao	Data de Criação	data no formato ano-mes-dia-hora:minuto:segundo	Guarda a data em que o arquivo foi criado
	dataModificacao	Data de Modificação	data no formato ano-mes-dia-hora:minuto:segundo	Recebe a data da última modificação do arquivo

	tamanho	Tamanho	string de até 10 caracteres para o dígito seguido da unidade (GB, MB, KB, ...)	Armazena o tamanho do arquivo
--	---------	---------	--	-------------------------------

Fonte: O autor

3. Projeto de Interface

Uma interface intuitiva, bonita e rápida é o sonho de todo desenvolvedor front-end, visto que com a alta popularização dos computadores, diversas “bolhas” de pessoas foram atingidas e isto criou a necessidade de interfaces cada vez mais auto explicativas. Não apenas isso, mas visto que o volume de dados disparados pelo monitor aumenta significativamente a cada ano, formas mais simples e rápidas são desenvolvidas para auxiliar os usuários quanto a utilização de sistemas.

Alguns pontos devem ser levados em consideração quando uma nova interface é criada, e dentre eles os mais importantes:

- O tipo de interação com o usuário;
- A forma de apresentar e receber as informações;
- A paleta de cores;
- Tratamento de eventos (Erros, Cliques do usuário, em ambientes gráficos, e outros);

Por ser um sistema que envia e recebe mensagens deveremos ter alguns elementos, como assunto, caixa de texto, remetente, data, destinatário, anexos, recuperação de e-mails (enviados, recebidos, talvez rascunho) e botão enviar. Um problema notável é a falta de uma interface modelo, visto que apenas possui o modelo entidade relacional, portanto, não há como definir cores ou layout herdados do sistema, logo, a aparência do site será dada apenas por sugestões e que podem diferir completamente do original.

A começar pela página principal de mensagens, como não há uma *master page*, vou estar sugerindo os layouts para páginas inteiras, ao invés de containers dentro das páginas do sistema.

Figura 2 - Sugestão de design para a página principal do sistema de mensagens



Fonte: O autor

O layout, como deve ter percebido, remete muito ao *gmail*. Isto é totalmente proposital, uma vez que essa é uma ferramenta muito difundida pela organização Google e possui milhares de usuários todos os dias, portanto, sua aparência já é bem conhecida. Para as cores, decidi manter uma paleta baseada no portal <https://bri.ifsp.edu.br/>, uma vez que não tenha à disposição algum material que diga quais as cores principais do Talk to Me. Outro motivo é o estudo da psicologia das cores, que busca explicar e atribuir significados e sensações que cada uma delas passam para as pessoas, por exemplo, dentre os diversos significados de cada cor, poderia atribuir que a cor verde nos dá sensação de calma, a cor vermelha passa sensação de energia, a cor azul de confiança, a cor amarela otimismo e a cinza/prata de tecnológica. (CLEMENTE, 2020). O *design* da página de envios teria mesmo layout, porém o texto “Remetente - Assunto” seria substituído por “Destinatário - Assunto”.

E como não podemos esquecer, um sistema mensageiro precisa de uma interface para enviar as mensagens. Levando em consideração alguns tópicos abordados anteriormente, quanto a alguns requisitos de envio da mensagem, criei este protótipo de interface:

Figura 3 - Protótipo para a interface de mensagens

Destinatário
Assunto
Texto
Anexos: arquivo_d...pdf exemplo.png +

Enviar

Fonte: O autor

Não é glamuroso, muito menos bonito, mas dá para ter uma ideia de algum layout interessante (grandemente inspirado no da Google), que englobe os aspectos mais importantes a serem mostrados a quem irá escrever a mensagem.

4. Referências

Dados Científicos: como construir metadados, descrição, readme, dicionário-de-dados e mais. **AGUIA**, 2021. Disponível em: <<https://www.aguia.usp.br/noticias/dados-cientificos-como-construir-metadados-descricao-readme-dicionario-de-dados/>>. Acesso em: 28 abr. 2022.

CLEMENTE, Matheus. Psicologia das cores: o que é e como usar no Marketing. **rockcontent**, 22 jul. 2020. Disponível em: <<https://rockcontent.com/br/blog/psicologia-das-cores/#:~:text=A%20Psicologia%20das%20Cores%20%C3%A9%20um%20estudo%20que%20busca%20compreender,de%20desejos%20e%20muito%20mais.>>>. Acesso em: 02 abr. 2022.