

Registro de um questionário na base de dados de Apoio para Pesquisas Clínicas

Renan Mendanha¹, Xiao Yong Kong², Vinicius de Andrade Flores³

¹⁻⁵ Departamento de Computação – Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

{xiaokong, renanma}@dcc.ufrj.br

{viniciusflores93}@gmail.com

Resumo. *Este trabalho conta passo a passo como foi implementado um sistema de questionário para pesquisas clínicas, um site que permite a criação, edição e documentação do questionário.*

Sumário

1.	Introdução	2
2.	Ferramentas Utilizadas	2
3.	Modelagem	2
3.1.	SiteMap	2
4.	Implementação do Banco de Dados	3
4.1.	Script	3
5.	Queries	3
5.1.	Questionários	3
5.2.	Formulários	4
5.3.	questões	4
5.4.	Editar uma questão	5
5.5.	Adicionar uma nova questão	6
5.6.	Grupos de questões	6
6.	Criação do Sistema Web	7
6.1.	Levantamentos de Ferramentas/Tecnologias	7
6.2.	Implementação	7
7.	Problemas encontrados	8
7.1.	A base de dados	8
7.2.	Relatório	9
8.	Divisão do projeto por tarefa	10
9.	Oportunidades para melhorias futuras	10
9.1.	Criação de um banco de questões	10
9.2.	Editar o comentário sobre o grupo de questões	10
9.3.	Autenticação nas funcionalidades de remoção	10
9.4.	Exibição das questões aninhadas a uma questão pai	10
10.	Conclusão	10
11.	Bibliografia	10

1. Introdução

Na disciplina de banco de Dados I, ministrada pela professora Giseli Lopes, no PLE do ano de 2020, recebemos como trabalho final da matéria a tarefa de fazermos uma aplicação web em que fosse possível acessar o banco de dados para realizar consultas pré-elaboradas por nós. Os dados deveriam ser apresentados de maneira amigável, para futuras análises dos mesmos.

Da proposta, escolhemos a segunda - questionários para pesquisa clínica - como fonte de dados, por encontrarmos dados completos e amigáveis. Além disso, os dados eram complexos o suficiente para realizarmos as consultas necessárias e podermos modelar da forma pedida.

Usando estes dados o grupo modelou um sistema que atendesse a proposta, porém encontramos alguns problemas que serão relatados nas próximas seções.

2. Ferramentas Utilizadas

- Miro: Para criar o Sitemap e elucidar as funções a serem implementadas;
- MySQL 5.7.22: implementação das consultas e queries, integração com interface
- Vue.js: como *framework* front-end;
- Nuxt: framework SSR (Server Side Rendering) utilizada junto com o Vue.js para facilitar a utilização de componentes de interface de forma que os dados ficassem dispostos de maneira mais organizada;
- Express: api para lidar com as requisições e conexão com o banco de dados;
- Telegram: comunicação rápida, código, e agendamento de reuniões;
- Github ^[1]: compartilhamento de scripts e códigos remotamente;
- Discord: Ferramenta para conferências de áudio e vídeo que foi utilizada para os encontros do grupo.

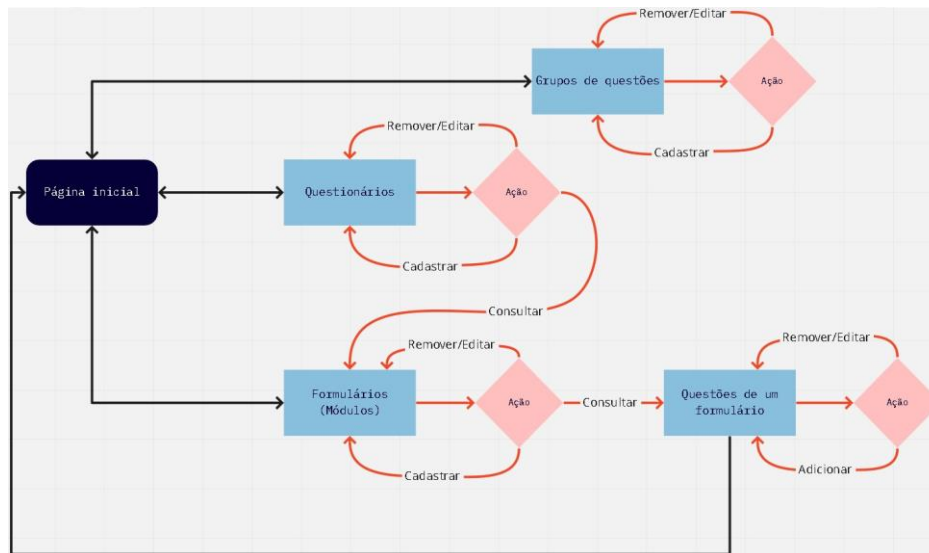
É essencial a instalação da última versão do Node.js e npm(6.14), que podem ser obtidos no site dos próprios desenvolvedores ^[2], para executar os frameworks, tanto front-end quanto back-end.

3. Modelagem

Inicialmente, realizamos a modelagem de como deveria ser o sistema, isso nos permitiu, já desde o início do projeto, ter em mente de maneira bem clara quais seriam as funcionalidades e conteúdo de cada página.

3.1. SiteMap

O Sitemap apresenta o fluxo que será percorrido entre as páginas e suas funções. A seguir, está presente uma prévia do documento completo no Miro ^[3].



Site Map

4. Implementação do Banco de Dados

O Banco de Dados foi implementado, como já dito antes, usando MySQL. Usamos como base direta o MR e de acordo com o surgimento das necessidades do sistema, adicionamos restrições de integridade extras para algumas relações.

4.1. Script

O script em MySQL para a criação do banco de dados está disponível abertamente no GitHub ^[4]. Foi usado o próprio aplicativo para inserir um novo questionário (“Ficha de Investigação de Caso Suspeito de Novo Coronavírus”) na base de dados.

5. Queries

Para realizar as funções do sistema, foi necessário implementar queries que retornassem conjuntos de resultados específicos e fazer CRUDs mais elaboradas. Listamos a seguir os códigos em SQL que permitem a realização de cada função:

5.1. Questionários

5.1.1. Listar todos os questionários salvos no BD:

```
SELECT * FROM tb_questionnaire
```

5.1.2. Inserir Um novo questionário no BD

```
INSERT INTO `tb_questionnaire` (description) VALUES (?)
```

5.1.3. Remover um questionário do BD

```
DELETE FROM `tb_questionnaire` WHERE questionnaireID = ?
```

5.1.4. Atualizar a descrição de um questionário no BD

```
UPDATE `tb_questionnaire` SET description = ? WHERE questionnaireID = ?
```

5.2. Formulários

5.2.1. Listar os módulos (formulários) que estão presentes em um determinado questionário:

```
SELECT * FROM `tb_crfforms` WHERE questionnaireID = ?
```

5.2.2. Inserir um novo formulário em um questionário específico no BD

```
INSERT INTO `tb_crfforms` (description, questionnaireID) VALUES (?, ?)
```

5.2.3. Remover um formulário de um questionário específico no BD

```
DELETE FROM `tb_crfforms` WHERE crfFormsID = ?
```

5.2.4. Atualizar a descrição do formulário de um questionário específico no BD

```
UPDATE `tb_crfforms` SET description = ? WHERE crfFormsID = ?
```

5.3. questões

5.3.1. Resgatar as questões que estão dentro de um módulo CRF específico de forma ordenada:

```
SELECT *, f.description as form, q.description as question, qt.description  
questionType, qg.description questionGroup  
FROM `tb_questiongroupform` qgf JOIN `tb_questions` q  
on qgf.questionID=q.questionID  
LEFT OUTER JOIN `tb_questionGroup` qg  
on q.questionGroupID=qg.questionGroupID  
JOIN `tb_crfforms` f  
on qgf.crfFormsId=f.crfFormsId  
JOIN tb_questiontype qt  
on q.questionTypeID=qt.questionTypeID  
where f.crfFormsId = ?  
ORDER BY qgf.questionOrder ASC  
LIMIT 10 OFFSET ?
```

5.3.2. Seleção de uma questão específica em um formulário

```
SELECT * FROM `tb_questions` q  
JOIN `tb_questiongroupform` qgf on qgf.questionID=q.questionID  
JOIN `tb_crfforms` crf on crf.crfFormsID=qgf.crfFormsID  
where q.questionID=?
```

5.3.3. Seleção da ordem das questões em um grupamento de questões específico em um formulário

```
SELECT qgf.questionOrder, q.questionGroupID  
FROM `tb_questions` q  
JOIN `tb_questiongroupform` qgf on qgf.questionID=q.questionID  
where q.questionGroupID=?  
and qgf.crfFormsID=?  
LIMIT 1
```

5.3.4. Remoção de uma questão

```
DELETE FROM `tb_questions` where questionID = ?
```

5.4. Editar uma questão

5.4.1. Atualizar a descrição de uma questão em um formulário

Para verificar se a ação está somente mudando a descrição de uma questão, o grupo da questão e o ID não deve ser alterado, por isso há a verificação do novo grupo da questão, se ele for igual, a query muda somente a descrição.

```
if (questionGroupID_before == req.body.questionGroupID)
```

Se este for o caso, executamos a seguinte query:

```
UPDATE `tb_questions` SET description=?, questionTypeID=? where questionID=?
```

5.4.2. Selecionar o valor da ordem da última questão de um grupamento em um formulário

Se há mudança no grupamento da questão, executamos a query que será apresentada a seguir para adicionar corretamente uma ordem à nova questão(a consulta pode não retornar resultados se não existir nenhuma questão em um determinado grupamento, caso isso aconteça, a ordem dentro do grupamento será 0, caso contrário, será o maior valor de ordem somando-se 1).

Mas antes, devemos verificar se estamos tratando de uma questão genérica, caso seja uma questão genérica não possuirá grupo e, por consequência, terá o 'questionGroupID' nulo. Se tivermos uma verificação de nulo, devemos utilizar a declaração lógica 'is' ao invés de '='.

```
let dynamic_statement = '='  
if (req.body.questionGroupID == null)  
  dynamic_statement = 'is'
```

Executamos a seguinte query, inserindo a variável que define o tipo de comparação:

```
'SELECT MAX(qgf.questionOrder) as questionOrder  
from `tb_questions` q  
JOIN `tb_questiongroupform` qgf on q.questionID=qgf.questionID  
where q.questionGroupID ' + dynamic_statement + ' ? and qgf.crfFormsID = ?'
```

5.4.3. Editar completamente uma questão em um formulário

Primeiramente, alteramos as informações da questão propriamente dita:

```
UPDATE `tb_questions` SET description=?, questionTypeID=?, questionGroupID=?  
where questionID=?
```

Precisamos agora atualizar a ordem da questão no novo grupamento de questões, levando em conta sua última ordem, assim como supracitado:

```
UPDATE `tb_questiongroupform` SET questionOrder=? where questionID=?
```

5.5. Adicionar uma nova questão

5.5.1. Selecionar o valor da ordem da última questão de um grupamento em um formulário

Se há uma adição no grupamento da questão, executamos a query que será apresentada a seguir para adicionar corretamente uma ordem à nova questão(a consulta pode não retornar resultados se não existir nenhuma questão em um determinado grupamento, caso isso aconteça, a ordem dentro do grupamento será 0, caso contrário, será o maior valor de ordem somando-se 1)

Mas antes, devemos verificar se estamos tratando de uma questão genérica, caso seja uma questão genérica não possuirá grupo e, por consequência, terá o 'questionGroupID' nulo. Se tivermos uma verificação de nulo, devemos utilizar a declaração lógica 'is' ao invés de '='.

```
let dynamic_statement = '='
if (!req.body.questionGroupID)
  dynamic_statement = 'is'
```

Executamos a seguinte query, inserindo a variável que define o tipo de comparação:

```
'SELECT MAX(qgf.questionOrder) questionOrder
from `tb_questions` q
join `tb_questiongroupform` qgf on q.questionID=qgf.questionID
where qgf.crfFormsID=?
and q.questionGroupID ' + dynamic_statement + ' ?'
```

5.5.2. Adicionar uma questão em um formulário

Primeiramente, inserimos as informações da questão propriamente dita:

```
INSERT INTO `tb_questions` (description, questionTypeID, questionGroupID)
values (?, ?, ?)
```

Precisamos agora inserir a ordem da questão e fazer a ligação com o seu formulário no novo grupamento de questões, levando em conta sua última ordem, assim como supracitado:

```
INSERT INTO `tb_questiongroupform` (questionID, crfFormsID, questionOrder)
values (?, ?, ?)
```

5.6. Grupos de questões

5.6.1. Selecionar todos os grupamentos de questões

```
SELECT * FROM `tb_questiongroup`
```

5.6.2. Inserir um grupamento

```
INSERT INTO `tb_questiongroup` (description) VALUES (?)
```

5.6.3. Remover um grupamento

```
DELETE FROM `tb_questiongroup` WHERE questionGroupID = ?
```

5.6.4. Editar um grupamento

```
UPDATE `tb_questiongroup` SET description = ? WHERE questionGroupID = ?
```

6. Criação do Sistema Web

O desenvolvimento do Sistema teve início pela definição das consultas que seriam realizadas, assim partimos para prototipação, definindo como esses dados seriam expostos e organizados na interface. E assim, optamos pelas tecnologias que foram utilizadas no front e no back-end, desenvolvendo-as e depois as integrando, de maneira que as informações fossem visíveis no front.

Para organização interna do grupo utilizamos o GitHub para versionamento de código de maneira a facilitar a organização do grupo.

6.1. Levantamentos de Ferramentas/Tecnologias

Devido a familiaridade de alguns integrantes do grupo com node e a *framework* Vue.js optamos pelo uso da mesma para evitar surpresas indesejadas e facilitar o processo de criação da interface da aplicação web. Além disso, contamos também com a familiaridade de conhecimento sobre integração de MySQL com a stack mencionada, podendo passar para todo o grupo como deveria ser feito.

6.2. Implementação

O Web-app está organizado em duas páginas principais, a página de questionário (a qual possibilita o acesso às páginas de formulário e de questões de um formulário) e a de grupos de questões (que é responsável mostrar e permitir manipulação dos agrupamentos de questões). É possível transitar livremente entre essas duas abas devido ao menu lateral de navegação, tais páginas e subpáginas terão seu funcionamento explicado a seguir:

6.2.1. Página de questionário

A página de questionário exibe os questionários presentes no banco através da consulta do tópico 5.1.1. Além disso, nesta página, temos a opção de adicionar um questionário, que abre uma janela modal com um campo de inserção e executa a query para adicionar um questionário (função implementada pela query do tópico 5.1.2.), temos também a opção de editar sua descrição, novamente abrindo uma modal com campo de inserção e executando a query de edição (função implementada pela query do tópico 5.1.4.), por fim, temos a função que nos permite remover um questionário através de um botão que executa a query de remoção (função implementada pela query do tópico 5.1.3.).

6.2.2. Página de formulários

Ao selecionar um questionário, será exibida a página de formulários, contendo todos os módulos de um questionário (função executada pela query do tópico 5.2.1.).

A partir da lista que nos é apresentada, nos surge a opção de remover um formulário, através de um botão que implementa a query de remoção (função executada pela query do tópico 5.2.3.) e também a opção de editar a descrição de um formulário, através de uma janela modal que surge na tela, necessitando uma entrada de texto e após executando a query de atualização na tabela (função executada pela query do tópico 5.2.4.).

Por fim, podemos adicionar um novo formulário clicando no botão “Adicionar novo módulo”, que abrirá também uma janela modal com um campo de texto e executará a query de inserção de um formulário (função executada pela query do tópico 5.2.2.).

6.2.3. Página de questões

Após clicar em um formulário, é exibida na tela a página com as questões do mesmo, organizadas por agrupamento e ordem (função executada pela query do tópico 5.3.1.).

A partir da tabela de questões, podemos remover uma em específico através de um botão que implementa a query de remoção (função executada pela query do tópico 5.3.4.).

Se optarmos por editar uma questão, surgirá uma janela modal, requisitando a seleção do agrupamento da questão, texto de descrição e tipo da questão, executando a query de atualização na tabela (função executada pela sequência de queries de consultas e updates do tópico 5.4.).

Por fim, temos a possibilidade adicionar uma nova questão clicando no botão “Adicionar nova questão”, que abrirá também uma janela modal requisitando a seleção do agrupamento da questão, texto de descrição e tipo da questão, executando a query de

inserção na tabela (função executada pela sequência de queries de consulta e inserts do tópico 5.5.).

6.2.4. Página de grupos de questões

Na página de grupos de questões está presente a exibição de uma lista com diversos itens composta pela descrição de um grupo de questão e seu comentario (função implementada pela query do tópico 5.6.1.).

A partir dessa lista, podemos editar a descrição de um grupamento, ao clicar na ação de editar, surgirá uma janela modal com um campo de texto que servirá como a nova descrição para o grupamento (função implementada pela query do tópico 5.6.4.).

Outra ação possível é a de remoção de um grupamento, que acontece ao clicar no botão “lixeira” na coluna de ações, isso faz com que a query de remoção seja executada (função implementada pela query do tópico 5.6.3.).

Por fim, cobrimos a função de adicionar um grupamento novo, esta funcionalidade pode ser realizada ao clicar-se no botão “Adicionar novo grupo de questões”, que abre uma janela modal com o campo necessário para atribuir uma descrição para a nova questão. Ao clicar em “Criar novo grupo de questões”, dentro da janela, será executada a query de inserção de uma nova questão (função implementada pela query do tópico 5.6.2.).

7. Problemas encontrados

7.1. A base de dados

Durante o desenvolvimento do sistema, surgiram alguns problemas relacionados à base de dados, o principal deles foi a falta de padronização para os identificadores das relações, para isso, foram feitas as seguintes modificações no modelo original:

7.1.1. Adicionada a chave primária 'questionnaireID' na relação 'tb_questionnaire'

```
ALTER TABLE `tb_questionnaire`  
ADD PRIMARY KEY (`questionnaireID`);
```

7.1.2. Adicionada a chave primária 'questionID' na relação 'tb_questions'

```
ALTER TABLE `tb_questions`  
ADD PRIMARY KEY (`questionID`);
```

7.1.3. Adicionada a chave primária 'questionGroupID' na relação 'tb_questiongroup'

```
ALTER TABLE `tb_questiongroup`  
ADD PRIMARY KEY (`questionGroupID`);
```

7.1.4. Mudado 'questionnaireID' da tabela 'tb_questionnaire' para Auto_Increment

Essa mudança permite designar um id automaticamente aos novos questionários adicionados à relação.

```
ALTER TABLE `tb_questionnaire`  
MODIFY `questionnaireID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
```


7.1.5. Mudando o campo 'crfFormsID' da tabela 'tb_crfforms' para Auto_Increment

Essa mudança permite designar um id automaticamente aos novos Formulários(módulos) adicionados à relação.

```
ALTER TABLE `tb_crfforms`  
MODIFY `crfFormsID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
```

7.1.6. Mudando o campo 'questionID' da tabela 'tb_questions' para Auto_Increment

Essa mudança permite designar um id automaticamente às novas questões adicionadas na tabela.

```
ALTER TABLE `tb_questions`  
MODIFY `questionID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=256;
```

7.1.7. Mudando 'questionGroupID' de 'tb_questiongroup' para Auto_Increment

Essa mudança permite designar um id automaticamente aos novos grupos de questões adicionados.

```
ALTER TABLE `tb_questiongroup`  
MODIFY `questionGroupID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;
```

Além disso, surgiram problemas mais complexos que serão abordados no tópico 9.1. deste relatório.

7.2. Relatório

Desconhecimento da formatação causou certo espanto e entrave inicial, especialmente pelo limite de páginas imposto. Fazendo com que o grupo revisasse o relatório algumas vezes, deixando somente as informações relevantes.

8. Divisão do projeto por tarefa

- Manutenção do Repositório, levantamento de tecnologias para front-end: Xiao Kong, Vinicius Flores;
- Brainstorm/Reuniões iniciais para definir o tema do projeto: Todos participaram.
- Pesquisa, implementação de queries: Renan Mendanha e Vinicius Flores;
- Modelagem BD Físico e Sistema: Renan Mendanha e Vinicius Flores;
- Armazenamento do Banco de Dados: Todos Participaram;
- Relatório (rascunho inicial, revisão, correções): Renan Mendanha e Xiao Kong.

9. Oportunidades para melhorias futuras

Devido ao curto espaço de tempo para desenvolvimento do app focamos em funcionalidades básicas, porém, existem algumas correções do esquema do banco de dados e funcionalidades adicionais que melhoram a experiência do usuário com o sistema. Dentre elas, podemos citar:

9.1. Criação de um banco de questões

Atualmente, devido à estrutura do banco de dados, estamos limitados a um determinado número de questões em um formulário de um questionário, pois suas ordens necessitam de um formato de identificação específico para funcionar.

Ao criar um banco de questões, será possível não só expandir as possibilidades em relação ao número de questões, mas também utilizar uma questão globalmente, de forma que todos os módulos consigam referencia-la, evitando repetições de uma questão igual em formulários diferentes.

9.2. Editar o comentário sobre o grupo de questões

9.3. Autenticação nas funcionalidades de remoção

9.4. Exibição das questões aninhadas a uma questão pai

10. Conclusão

O processo de modelagem do sistema, elaboração das consultas, a prototipação e como tudo isso se interliga a fim de formar uma aplicação web agregou muito a formação acadêmica do grupo. Uma vez que vimos na prática como a disciplina de Banco de Dados se relaciona com assuntos que estão em alta no âmbito profissional.

11. Bibliografia

[1]<https://github.com/xiaoyongkong/TrabFinalBD-PLE>

[2]<https://nodejs.org/en/download/>

[3]<https://miro.com/welcomeonboard/ITkjHOEOlpYASwgudhrgEeCRsuP7zkUnXnyKgHi2G4kajFRGXM9vP0TJGP7i8IBk>

[4]<https://github.com/xiaoyongkong/TrabFinalBD-PLE/blob/main/Banco%20de%20Dados/vodan.sql>