**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ**

*Banco de Dados – PJBL Fase 2*

KindAct

**Membros:**

João Victor - 40102065

Guilherme Reis Carvalho – 40093122

Nicolas Lobo - 40090033

Vinicius Trevisan - 40100664

Curitiba

2024

Sumário

[Descrição do Projeto 2](#_Toc181168820)

[Desafios 2](#_Toc181168821)

[Diagrama de Entidade e Relacionamento 3](#_Toc181168822)

[Lista de Requisitos e Comandos SQL 5](#_Toc181168823)

[VIEWS, PROCEDURES E FUNCTIONS 11](#_Toc181168824)

# \*\*\* NOTA: alteramos alguns requisitos do projeto. O código também foi alterado, com a criação de views, e mais tabelas necessárias para garantir o funcionamento dos requisitos.

\*\* COPIE E COLE NO GOOGLE PARA VER O CODIGO INTEIRO:

https://github.com/ViniMTrevisan/PUC-PR---Segundo-semestre/blob/main/banco%20de%20dados/programacao/kindact.sql

# Descrição do Projeto

O nosso projeto visa abrir oportunidades para ONGs e voluntários ao redor do Brasil, para que pessoas necessitadas sejam cada vez mais acessíveis e ajudadas para quem quer fazer a diferença. Nosso projeto conectará ONGs com Voluntários em um só site, onde ambos poderão se conectar com o outro, o que reduzirá a dificuldade das ONGs acharem ajuda e dos Voluntários acharem ONGs que se encaixem com o perfil deles.

# Desafios

Um dos desafios mais difíceis foi garantir a criação de tabelas suficientes para garantir todos os requisitos. Isso inclui a criação das tabelas que relacionam 2 tabelas separadas, que só pôde ser decidida de verdade após criarmos todas as tabelas principais. Outro desafio foi a criação de views úteis e corretas para o nosso sistema. Isso foi um desafio, pois acreditávamos que poderíamos realizar a maior parte das consultas a partir de SELECTs, porem foi necessário a criação de views. Percebemos que as regras de negócio e as relações entre as tabelas exigiam consultas mais complexas, especialmente envolvendo múltiplos JOINs. O ultimo problema foi relacionado a população inicial do banco. Acreditávamos que com poucos INSERTs e VALUES teríamos o suficiente para fazer um teste real, porém foi necessário a criação de mais INSERTs e maior população do banco para conseguirmos resultados de verdade.

# Diagrama de Entidade e Relacionamento

Insira aqui a versão finalizada e ajustada do Diagrama de Entidade Relacionamento.

Diagrama

O conteúdo gerado por IA pode estar incorreto.Uma imagem contendo Linha do tempo

O conteúdo gerado por IA pode estar incorreto.

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Tela de computador com jogo

O conteúdo gerado por IA pode estar incorreto.

# Lista de Requisitos e Comandos SQL

Aqui você irá inserir a lista de requisitos e seus respectivos comandos SQL conforme tabela abaixo:

|  |  |
| --- | --- |
| Req 1 | Cadastro de ONGs novas no sistema |
| SQL | INSERT INTO tb\_ong (ong\_nome, ong\_email, ong\_area\_atuacao, ong\_endereco, ong\_cnpj, ong\_telefone, ong\_site, ong\_nome\_responsavel, ong\_cep, ong\_logradouro, ong\_numero, ong\_bairro, ong\_complemento, ong\_cidade, ong\_uf) VALUES ( 'ONG Vida Verde', 'contato@vidaverde.org', 'Meio Ambiente', 'Rua das Árvores, 123, São Paulo', 99.888.777/0001-66', '(11) 97777-8888', 'www.vidaverde.org', 'João Verde', '01002-000', 'Rua das Árvores', '123', 'Jardim', '', 'São Paulo', 'SP'); |

|  |  |
| --- | --- |
| Req 2 | Lista de ONGs cadastradas no sistema |
| SQL | SELECT \* FROM tb\_ong; |

|  |  |
| --- | --- |
| Req 3 | O sistema deve permitir o cadastro de voluntários novos no sistema |
| SQL | INSERT INTO tb\_voluntario ( voluntario\_nome, voluntario\_email, voluntario\_idade, voluntario\_endereco, voluntario\_cpf, voluntario\_data\_nascimento, voluntario\_sexo, voluntario\_profissao, voluntario\_telefone, voluntario\_cep, voluntario\_logradouro, voluntario\_numero, voluntario\_bairro, voluntario\_complemento, voluntario\_cidade, voluntario\_uf) VALUES ( 'Mariana Oliveira', 'mariana.oliveira@email.com', 28, 'Rua das Flores, 456, São Paulo', '777.888.999-00', 1997-04-15', 'Feminino', 'Designer', '(11) 98888-9999', '01003-000', 'Rua das Flores', '456', 'Centro', '', 'São Paulo', 'SP’); |

|  |  |
| --- | --- |
| Req 4 | Lista de voluntários cadastrados no sistema |
| SQL | CREATE VIEW vw\_voluntarios\_cadastrados AS  SELECT      v.voluntario\_id,      v.voluntario\_nome,      o.ong\_id,      o.ong\_nome,      o.ong\_area\_atuacao  FROM      tb\_voluntario v      JOIN tb\_ong o ON v.voluntario\_id = o.ong\_id  WHERE      v.voluntario\_id IS NOT NULL; |

|  |  |
| --- | --- |
| Req 5 | Registro de novos eventos |
| SQL | INSERT INTO tb\_evento (evento\_titulo, evento\_data\_inicio, evento\_data\_termino, evento\_endereco, evento\_descricao,  evento\_cep, evento\_logradouro, evento\_numero, evento\_bairro, evento\_complemento, evento\_cidade, evento\_uf, fk\_ong\_id) VALUES ('Feira Verde', '2025-06-01', '2025-06-01', 'Rua das Árvores, 123, São Paulo', 'Feira para promover sustentabilidade', '01002-000', 'Rua das Árvores', '123', 'Jardim', ‘’ , 'São Paulo', 'SP', 1); |

|  |  |
| --- | --- |
| Req 6 | Lista de eventos cadastrados |
| SQL | CREATE VIEW vw\_eventos\_cadastrados AS  SELECT  e.evento\_id,  e.evento\_titulo,  e.evento\_data\_inicio,  e.evento\_data\_termino,  o.ong\_id,  o.ong\_nome  FROM  tb\_evento e  JOIN tb\_ong o ON e.fk\_ong\_id = o.ong\_id  WHERE  e.evento\_id IS NOT NULL; |

|  |  |
| --- | --- |
| Req 7 | Inscrição de voluntários em eventos |
| SQL | INSERT INTO tb\_voluntario\_evento (fk\_voluntario\_id, fk\_evento\_id) VALUES (3, 3); |

|  |  |
| --- | --- |
| Req 8 | Lista de voluntários inscritos por eventos |
| SQL | CREATE VIEW  vw\_voluntarios\_eventos AS  SELECT  v.voluntario\_nome,  e.evento\_titulo,  e.evento\_data\_inicio,  e.evento\_endereco  FROM  tb\_voluntario v  JOIN tb\_voluntario\_evento ve ON v.voluntario\_id = ve.fk\_voluntario\_id  JOIN tb\_evento e ON ve.fk\_evento\_id = e.evento\_id; |

|  |  |
| --- | --- |
| Req 9 | Registro de doações dos voluntários para ONGs especificas |
| SQL | INSERT INTO tb\_doacao ( doacao\_tipo, doacao\_valor, doacao\_data, doacao\_mensagem, fk\_voluntario\_id, fk\_ong\_id, fk\_destinatario\_id)  VALUES ('Dinheiro', 100.00, '2025-05-28', 'Doação para projetos ambientais', 1, 1, 1); |

|  |  |
| --- | --- |
| Req 10 | Lista de doações recebidas para cada ONG, por voluntario |
| SQL | CREATE VIEW  vw\_doacoes\_voluntarios AS  SELECT  v.voluntario\_nome,  d.doacao\_tipo,  d.doacao\_valor,  d.doacao\_data,  o.ong\_nome,  dest.destinatario\_nome  FROM  tb\_doacao d  JOIN tb\_voluntario v ON d.fk\_voluntario\_id = v.voluntario\_id  JOIN tb\_ong o ON d.fk\_ong\_id = o.ong\_id  JOIN tb\_destinatario dest ON d.fk\_destinatario\_id = dest.destinatario\_id; |

|  |  |
| --- | --- |
| Req 11 | Avaliação de voluntários por ONG |
| SQL | INSERT INTO tb\_avaliacao (avaliacao\_nota, avaliacao\_comentario, fk\_voluntario\_id)  VALUES (4, 'Ótimo desempenho no evento.', 1); |

|  |  |
| --- | --- |
| Req 12 | Registro de novos curadores |
| SQL | INSERT INTO tb\_curadoria (curadoria\_nome, curadoria\_requisitos\_minimos, fk\_ong\_id)  VALUES ('Curadoria de Sustentabilidade', 'Conhecimento em meio ambiente', 1); |

|  |  |
| --- | --- |
| Req 13 | Lista de curadores |
| SQL | SELECT \* FROM tb\_curadoria; |

|  |  |
| --- | --- |
| Req 14 | Envio e recebimento de mensagens, funcionando para ONGs e voluntários |
| SQL | INSERT INTO tb\_mensagem (fk\_voluntario\_id, fk\_ong\_id, mensagem\_conteudo, mensagem\_data\_envio)  VALUES (1, 1, 'Gostaria de participar do próximo evento.', '2025-05-28 16:54:00'); |

|  |  |
| --- | --- |
| Req 15 | Consulta de mensagens trocadas entre ONGs e voluntários |
| SQL | CREATE VIEW  vw\_mensagens\_ongs AS  SELECT  v.voluntario\_nome,  o.ong\_nome,  m.mensagem\_conteudo,  m.mensagem\_data\_envio  FROM  tb\_mensagem m  JOIN tb\_voluntario v ON m.fk\_voluntario\_id = v.voluntario\_id  JOIN tb\_ong o ON m.fk\_ong\_id = o.ong\_id; |

|  |  |
| --- | --- |
| Req 16 | Edição de perfil da ONG |
| SQL | UPDATE tb\_ong  SET ong\_nome = ‘ONG nova, Vida nova’  WHERE ong\_id = 1; |

|  |  |
| --- | --- |
| Req 16 | Edição de perfil do voluntário |
| SQL | UPDATE tb\_voluntario  SET voluntario\_email = ‘emailnovo@email.com’  WHERE ong\_id = 4; |

|  |  |
| --- | --- |
| Req 17 | Edição de perfil do evento |
| SQL | UPDATE tb\_evento  SET evento\_endereco = ‘Rua Augusto Zibarth, 1245, Curitiba’  WHERE evento\_id = 2; |

|  |  |
| --- | --- |
| Req 18 | Edição de voluntários por evento |
| SQL | UPDATE tb\_voluntario\_evento  SET fk\_voluntario\_id = 4  WHERE fk\_evento\_id = 2; |

|  |  |
| --- | --- |
| Req 19 | Edição de avaliações |
| SQL | UPDATE tb\_avaliacao  SET avaliacao\_nota = ‘2’  WHERE fk\_voluntario\_id = 5; |

|  |  |
| --- | --- |
| Req 20 | Edição de curadores |
| SQL | UPDATE tb\_curadoria  SET curadoria\_nome = ‘Curadoria de jogos eletrônicos ’  WHERE fk\_ong\_id = 4; |

|  |  |
| --- | --- |
| Req 21 | Exclusão de perfil da ONG |
| SQL | DELETE FROM tb\_ong  WHERE ong\_id = 1; |

|  |  |
| --- | --- |
| Req 22 | Edição de perfil do voluntário |
| SQL | DELETE FROM tb\_voluntario  WHERE voluntario\_id = 5; |

|  |  |
| --- | --- |
| Req 23 | Exclusão de perfil do evento |
| SQL | DELETE FROM tb\_evento  WHERE evento\_id = 3; |

|  |  |
| --- | --- |
| Req 24 | Exclusão de voluntários por evento |
| SQL | DELETE FROM tb\_voluntario\_evento  WHERE fk\_evento\_id = 2; |

|  |  |
| --- | --- |
| Req 25 | Exclusão de avaliações |
| SQL | DELETE FROM tb\_avaliacao  WHERE fk\_voluntario\_id = 5; |

|  |  |
| --- | --- |
| Req 26 | Edição de curadores |
| SQL | DELETE FROM tb\_curadoria  WHERE fk\_ong\_id = 4 |

# 

# VIEWS, PROCEDURES E FUNCTIONS

Aqui você deve listar todas as Views, Procedures e Functions no Sistema,

CREATE VIEW

    vw\_voluntarios\_eventos AS

SELECT

    v.voluntario\_nome,

    e.evento\_titulo,

    e.evento\_data\_inicio,

    e.evento\_endereco

FROM

    tb\_voluntario v

    JOIN tb\_voluntario\_evento ve ON v.voluntario\_id = ve.fk\_voluntario\_id

    JOIN tb\_evento e ON ve.fk\_evento\_id = e.evento\_id;

Exibe os voluntários cadastrados em cada evento, com nome, data e endereço do evento.

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_doacoes\_voluntarios AS

SELECT

    v.voluntario\_nome,

    d.doacao\_tipo,

    d.doacao\_valor,

    d.doacao\_data,

    o.ong\_nome,

    dest.destinatario\_nome

FROM

    tb\_doacao d

    JOIN tb\_voluntario v ON d.fk\_voluntario\_id = v.voluntario\_id

    JOIN tb\_ong o ON d.fk\_ong\_id = o.ong\_id

    JOIN tb\_destinatario dest ON d.fk\_destinatario\_id = dest.destinatario\_id;

Mostra doações feitas por voluntários, incluindo tipo, valor, data, ONG e destinatário.

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_avaliacoes\_voluntarios AS

SELECT

    v.voluntario\_nome,

    a.avaliacao\_nota,

    a.avaliacao\_comentario

FROM

    tb\_avaliacao a

    JOIN tb\_voluntario v ON a.fk\_voluntario\_id = v.voluntario\_id;

Lista as avaliações dos voluntários com notas e comentários.

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_voluntarios\_cargos AS

SELECT

    v.voluntario\_nome,

    c.cargo\_nome

FROM

    tb\_voluntario v

    JOIN tb\_voluntario\_cargo vc ON v.voluntario\_id = vc.fk\_voluntario\_id

    JOIN tb\_cargos c ON vc.fk\_cargo\_id = c.cargo\_id;

Relaciona voluntários com os cargos que ocupam.

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_ongs\_eventos AS

SELECT

    o.ong\_nome,

    COUNT(e.evento\_id) AS total\_eventos

FROM

    tb\_ong o

    LEFT JOIN tb\_evento e ON o.ong\_id = e.fk\_ong\_id

GROUP BY

    o.ong\_nome;

Apresenta cada ONG com a quantidade de eventos que organizou.

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_curadorias\_ongs AS

SELECT

    c.curadoria\_nome,

    o.ong\_nome,

    c.curadoria\_requisitos\_minimos

FROM

    tb\_curadoria c

    JOIN tb\_ong o ON c.fk\_ong\_id = o.ong\_id;

Mostra curadorias associadas às ONGs e seus requisitos mínimos.

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_mensagens\_ongs AS

SELECT

    v.voluntario\_nome,

    o.ong\_nome,

    m.mensagem\_conteudo,

    m.mensagem\_data\_envio

FROM

    tb\_mensagem m

    JOIN tb\_voluntario v ON m.fk\_voluntario\_id = v.voluntario\_id

    JOIN tb\_ong o ON m.fk\_ong\_id = o.ong\_id;

Lista mensagens enviadas por voluntários para ONGs, com data e conteúdo.

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_notificacoes\_voluntarios AS

SELECT

    v.voluntario\_nome,

    n.notificacao\_msg,

    n.notificacao\_data

FROM

    tb\_notificacao n

    JOIN tb\_voluntario v ON n.fk\_voluntario\_id = v.voluntario\_id;

Exibe notificações recebidas pelos voluntários, com data e conteúdo.

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_voluntarios\_ongs\_enderecos AS

SELECT

    v.voluntario\_nome,

    v.voluntario\_endereco,

    o.ong\_nome,

    o.ong\_endereco

FROM

    tb\_voluntario v

    JOIN tb\_ong o ON v.voluntario\_id = o.ong\_id;

Relaciona voluntários e ONGs pelos seus nomes e endereços (possivelmente incorreta).

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_aprovado\_pendentes\_voluntarios\_curadorias AS

SELECT

    v.voluntario\_nome,

    c.curadoria\_nome,

    vc.voluntario\_curadoria\_status\_aprovacao

FROM

    tb\_voluntario\_curadoria vc

    JOIN tb\_voluntario v ON vc.fk\_voluntario\_id = v.voluntario\_id

    JOIN tb\_curadoria c ON vc.fk\_curadoria\_id = c.curadoria\_id

WHERE

    vc.voluntario\_curadoria\_status\_aprovacao IN ('Aprovado', 'Pendente');

Lista o status de aprovação (Aprovado/Pendente) dos voluntários nas curadorias.

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_voluntarios\_cadastrados AS

SELECT

    v.voluntario\_id,

    v.voluntario\_nome,

    o.ong\_id,

    o.ong\_nome,

    o.ong\_area\_atuacao

FROM

    tb\_voluntario v

    JOIN tb\_ong o ON v.voluntario\_id = o.ong\_id

WHERE

    v.voluntario\_id IS NOT NULL;

Exibe voluntários cadastrados e a ONG à qual estão associados (possível erro de JOIN).

--------------------------------------------------------------------------------------------------------

CREATE VIEW

    vw\_eventos\_cadastrados AS

SELECT

    e.evento\_id,

    e.evento\_titulo,

    e.evento\_data\_inicio,

    e.evento\_data\_termino,

    o.ong\_id,

    o.ong\_nome

FROM

    tb\_evento e

    JOIN tb\_ong o ON e.fk\_ong\_id = o.ong\_id

WHERE

    e.evento\_id IS NOT NULL;

Mostra eventos cadastrados com seus dados e a ONG responsável.