

Nikolas Malinowski de Pedro
Lucas Aioria Serpa
Flavia Cristina Fagundes
João Augusto Schoembakla da Cruz

RELAÇÃO DE ARTEFATOS
ESPECIFICAÇÃO DO PROJETO
- FÓRUM SUSTENTÁVEL -

Curitiba

2025

SUMÁRIO

ARTEFATO 1: Quadro “3 Objetivos”	3
ARTEFATO 2: Quadro “é – não é – faz – não faz”	3
ARTEFATO 3: Quadro “Visão de Produto”.	4
ARTEFATO 4: Canvas PBB	5
ARTEFATO 5: Relação de User Stories.....	6
ARTEFATO 6: Modelo Relacional.....	13
ARTEFATO 7: Diagrama de Atividades	20
REFERÊNCIA BIBLIOGRÁFICAS	22

ÍNDICE DE ILUSTRAÇÕES

Figura 1 – Quadro “3 Objetivos”.	3
Figura 2 – Quadro “é – não é – faz – não faz”	3
Figura 3 – Quadro "Visão de Produto". Fonte: AGUIAR, F. 2018.	4
Figura 4 – Canvas PBB: "Product Backlog Building". Fonte: AGUIAR, F. 2018.	5
Figura 5 - User Stories e Critérios de Aceite. Fonte: AGUIAR, F. 2018.	12
Figura 6 – Exemplo: Modelo Relacional gerado por engenharia reversa (MySQL Workbench).	18

ARTEFATO 1: Quadro “3 Objetivos”

ARTEFATO 1: Quadro “3 Objetivos”	
NOME DO PRODUTO: DailyGreen	
OBJETIVOS	DESCRIÇÃO
1	Incentivar uma vida sustentável
2	Promover educação ambiental
3	Facilitar a organização e a participação em ações coletivas

Figura 1 – Quadro “3 Objetivos”.

ARTEFATO 2: Quadro “é – não é – faz – não faz”

ARTEFATO 2: Quadro “é – não é – faz – não faz”	
NOME DO PRODUTO: DailyGreen	
É Um fórum; uma rede social; Uma área educacional; é um site.	Não é Site de notícias; Site de delivery; E-commerce; Assistente virtual
Faz Incentiva a sustentabilidade; Conecta pessoas com interesses a natureza; postagens públicas; Curtir e comentar postagens;	Não faz Mandar mensagens por chat privado; dieta diária para o usuário.

Figura 2 – Quadro “é – não é – faz – não faz”.

ARTEFATO 3: Quadro “Visão de Produto”.

ARTEFATO 3: Quadro “Visão de Produto”	
NOME DO PRODUTO: DailyGreen	
CLIENTE-ALVO	Pessoas que gostam de preservar a natureza; tem interesse em sustentabilidade.
CATEGORIA-SEGMENTO	É um fórum e rede social
BENEFÍCIO-CHAVE	Aprender a ter uma vida mais sustentável
DIFERENCIADO-CHAVE	Visado na natureza e como a preservá-la
META-VALOR.	Promover ações de sustentabilidade; divulgar organizações sustentáveis

Figura 3 – Quadro “Visão de Produto”. Fonte: AGUIAR, F. 2018.

ARTEFATO 4: Canvas PBB

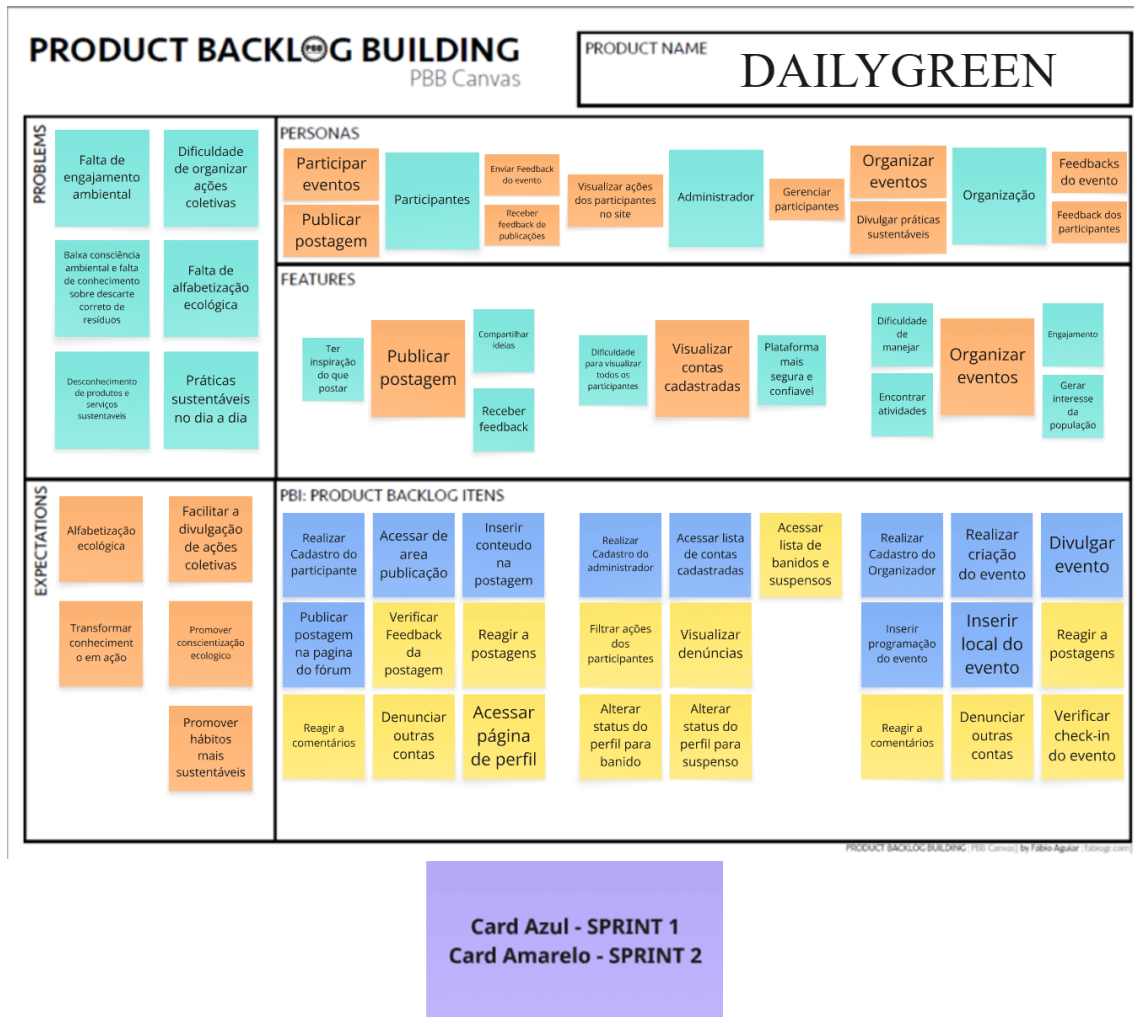


Figura 4 – Canvas PBB: "Product Backlog Building". Fonte: AGUIAR, F. 2018.

ARTEFATO 5: Relação de User Stories

SPRINT 1

HISTÓRIA DO USUÁRIO 1 - PBI: Publicar Postagem	
COMO: Participante. POSSO: Publicar postagem. PARA: Compartilhar ideias.	
Critério de Aceite 1	DADO QUE: Estou logado na plataforma. QUANDO: Preencho os campos obrigatórios de postagem, título e descrição. ENTÃO: Publico a postagem na plataforma.
Critério de Aceite 2	DADO QUE: Estou logado na plataforma. QUANDO: Não preencho o campo de descrição. ENTÃO: O sistema exibe um alerta, solicitando que o campo seja preenchido.

HISTÓRIA DO USUÁRIO 2 - PBI: Acessar Página De Postagem	
COMO: Participante. POSSO: Acessar página de postagem. PARA: Visualizar eventos e postagens.	
Critério de Aceite 1	DADO QUE: Possuo uma conta na plataforma. QUANDO: Realizo o login, preenchendo os campos de e-mail e senha. ENTÃO: Acesso a página de postagens.
Critério de Aceite 2	DADO QUE: Possuo uma conta na plataforma. QUANDO: Preencho o campo senha com a senha incorreta. ENTÃO: O sistema exibe um alerta informando que ou o e-mail ou a senha estão incorretos.

HISTÓRIA DO USUÁRIO 3 - PBI: Evento	
COMO: Organização. POSSO: Organizar evento. PARA: Gerar engajamento.	
Critério de Aceite 1	DADO QUE: Exista um evento programado. QUANDO: Público o evento preenchendo os campos obrigatórios, de título, descrição, data e hora de início e fim, endereço (local do evento) e link. ENTÃO: Posto um evento no DailyGreen.
Critério de Aceite 2	DADO QUE: Exista um evento programado. QUANDO: Não preencho o campo, data e hora. ENTÃO: O sistema exibe um alerta, solicitando que o campo seja preenchido.

HISTÓRIA DO USUÁRIO 4 - PBI: Acessar Área De Postagem.	
COMO: Organização. POSSO: Acessar área de postagem. PARA: Visualizar eventos e postagens.	
Critério de Aceite 1	DADO QUE: Possuo uma conta na plataforma. QUANDO: Realizo o login, preenchendo os campos de e-mail e senha. ENTÃO: Visualizo a área de postagens.
Critério de Aceite 2	DADO QUE: Possuo uma conta na plataforma. QUANDO: Insiro o Email incorreto do login. ENTÃO: O sistema exibe um alerta informando que ou o e-mail ou a senha estão incorretos.

HISTÓRIA DO USUÁRIO 5 - PBI: Visualizar Contas Cadastradas	
COMO: Administrador. POSSO: Visualizar contas cadastradas. PARA: Ter o controle de acessos na plataforma.	
Critério de Aceite 1	DADO QUE: Posso uma conta na plataforma. QUANDO: Preencho os campos obrigatórios de login, e-mail e senha. ENTÃO: Acesso a página de visualizar contas cadastradas na plataforma.
Critério de Aceite 2	DADO QUE: Não possuo uma conta na plataforma. QUANDO: Preencho os campos obrigatórios de login, e-mail e senha. ENTÃO: O sistema exibe um alerta informando que ou o e-mail ou a senha estão incorretos.

SPRINT 2

HISTÓRIA DO USUÁRIO 6 - PBI: Reagir a Comentários	
COMO: Participante. POSSO: Reagir a comentários. PARA: Interagir com os participantes.	
Critério de Aceite 1	DADO QUE: Visualizo os comentários de uma postagem. QUANDO: Clico no botão de reação. ENTÃO: A minha reação é armazenada e fica visível para outros usuários.
Critério de Aceite 2	DADO QUE: Visualizo os comentários de uma postagem. QUANDO: Não localizo nenhum comentário. ENTÃO: O sistema mostra na área de comentários, a mensagem: “Essa postagem ainda não possui comentários”.

HISTÓRIA DO USUÁRIO 7 - PBI: Denunciar Outras Contas.	
COMO: Participante. POSSO: Denunciar outras contas. PARA: Garantir a segurança e o bom convívio na plataforma.	
Critério de Aceite 1	DADO QUE: Visualizo uma postagem ou comentário com conteúdo inadequado para a plataforma. QUANDO: Clico na foto do perfil do usuário, clico na opção "Denunciar" e escolho uma opção do motivo da denúncia. ENTÃO: A denúncia é registrada com sucesso e o sistema exibe uma mensagem de confirmação.
Critério de Aceite 2	DADO QUE: Visualizo uma postagem ou comentário com conteúdo inadequado para a plataforma. QUANDO: Clico na foto do perfil do usuário, clico na opção "Denunciar", mas não seleciono o motivo da denúncia. ENTÃO: A denúncia não é registrada e o sistema exibe uma mensagem de erro solicitando que selecione o motivo da denúncia.

HISTÓRIA DO USUÁRIO 8 - PBI: Acessar Página de Perfil.	
COMO: Participante. POSSO: Acessar página de perfil. PARA: Visualizar e gerenciar minhas informações e publicações.	
Critério de Aceite 1	DADO QUE: Estou logado da plataforma. QUANDO: Clico no botão: Perfil. ENTÃO: O sistema mostra a minha página de perfil, com as minhas publicações.
Critério de Aceite 2	DADO QUE: Não estou logado na plataforma. QUANDO: Acesso a página de perfil. ENTÃO: O sistema me redireciona para a página de login.

HISTÓRIA DO USUÁRIO 9 - PBI: Reagir a Postagens.	
COMO: Organização.	
POSSO: Reagir a postagens.	
PARA: Interagir com os participantes e outras organizações.	
Critério de Aceite 1	DADO QUE: Visualizei uma postagem que eu gostei. QUANDO: Clico no botão de reação. ENTÃO: A minha reação é armazenada e fica visível para outros usuários.
Critério de Aceite 2	DADO QUE: Quero tirar a minha reação de uma postagem. QUANDO: Eu clicar novamente no ícone da reação que coloquei na postagem. ENTÃO: A minha reação deve ser removida da postagem e o contador de reações deve ser atualizado.

HISTÓRIA DO USUÁRIO 10 - PBI: Verificar Check-In	
COMO: Organização.	
POSSO: Verificar check-in.	
PARA: Visualizar o número de participantes que pretendem ir ao evento.	
Critério de Aceite 1	DADO QUE: Publiquei um evento na plataforma. QUANDO: Clico em: Lista Check-In. ENTÃO: Visualizo a quantidade de usuários, junto com a lista dos nomes dos usuários que fizeram o Check-In no meu evento.
Critério de Aceite 2	DADO QUE: Publiquei um evento na plataforma. QUANDO: Clico em: Lista Check-In, mas nenhum usuário fez Check-In no evento. ENTÃO: O sistema exibe a mensagem: “Não há registros de check-in para este evento.”

HISTÓRIA DO USUÁRIO 11 - PBI: Acessar Lista de Banidos.	
COMO: Administrador. POSSO: Acessar lista de banidos. PARA: Monitorar e gerenciar usuários que violaram as regras da plataforma.	
Critério de Aceite 1	DADO QUE: Estou logado na plataforma. QUANDO: Clico no botão: Lista Banidos. ENTÃO: O sistema apresenta a lista de usuários banidos, incluindo o motivo do banimento e a data em que a medida foi aplicada.
Critério de Aceite 2	DADO QUE: Não estou logado na plataforma. QUANDO: Acesso a lista de banidos. ENTÃO: O sistema exibe a mensagem: “Ainda não há usuários banidos!”.

HISTÓRIA DO USUÁRIO 12 - PBI: Visualizar Denúncias	
COMO: Administrador. POSSO: Visualizar denúncias. PARA: Manter a plataforma mais segura e confiável.	
Critério de Aceite 1	DADO QUE: Estou logado na plataforma. QUANDO: Clico no botão: Lista Denúncias. ENTÃO: O sistema exibe a lista de denúncias, contendo o identificador da denúncia, o título da denúncia, o motivo da denúncia e a data que ela foi registrada. Junto com o botão para analisar a denúncia.
Critério de Aceite 2	DADO QUE: Não estou logado na plataforma. QUANDO: Acesso a lista de denúncias. ENTÃO: O sistema me redireciona para a página de login.

HISTÓRIA DO USUÁRIO 13 - PBI: Alterar Status do Perfil Para Suspenso	
COMO: Administrador. POSSO: Alterar status do perfil para suspenso. PARA: Restringir o acesso de usuários que violam as regras da plataforma.	
Critério de Aceite 1	DADO QUE: Existe uma denúncia registrada na lista de denúncias. QUANDO: A denúncia é analisada e considerada passível de suspensão, clico no botão "Suspend". ENTÃO: O sistema exibe uma mensagem de confirmação, o usuário é adicionado à lista de suspensos e à graylist e o status da denúncia muda para "Resolvida".
Critério de Aceite 2	DADO QUE: Existe uma denúncia registrada na lista de denúncias. QUANDO: A denúncia é analisada e não considerada passível de punição, clico no botão "Arquivar". ENTÃO: O sistema redireciona para a página principal, o status da denúncia muda para "Arquivada", e o usuário continua na white list.

Figura 5 - User Stories e Critérios de Aceite. Fonte: AGUIAR, F. 2018.

Links:

[Trello DailyGreen](#)

[Github DailyGreen](#)

ARTEFATO 6: Modelo Relacional

ARTEFATO 6: Modelo físico

```
-- Schema db_dailygreen
-- -----
CREATE SCHEMA IF NOT EXISTS db_dailygreen;
USE db_dailygreen;

-- Tabela db_dailygreen.lista
-- -----
CREATE TABLE IF NOT EXISTS lista (
    id_lista INT NOT NULL AUTO_INCREMENT,
    tipo_lista VARCHAR(20),
    PRIMARY KEY (id_lista),
);

INSERT INTO lista (tipo_lista)
VALUES ("blackList");

INSERT INTO lista (tipo_lista)
VALUES ("grayList");

INSERT INTO lista (tipo_lista)
VALUES ("whiteList");

-- Tabela db_dailygreen.participante
-- -----
CREATE TABLE IF NOT EXISTS participante (
    id_participante INT NOT NULL AUTO_INCREMENT,
    id_lista INT NOT NULL DEFAULT 3,
    profile_pic VARCHAR(255),
    banner_pic VARCHAR(255),
    biografia VARCHAR(255),
    username VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL,
    genero CHAR(1) NOT NULL,
    create_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (id_participante),
    CONSTRAINT fk_listaParticipante FOREIGN KEY (id_lista) REFERENCES
lista(id_lista)
);
```

```
-- Tabela organizacao
--
CREATE TABLE IF NOT EXISTS organizacao (
    id_organizacao INT NOT NULL AUTO_INCREMENT,
    id_participante INT NOT NULL,
    nome VARCHAR(100) NOT NULL,
    cnpj VARCHAR(18) NOT NULL,
    PRIMARY KEY (id_organizacao),
    CONSTRAINT fk_participanteOrganizacao FOREIGN KEY (id_participante)
REFERENCES participante(id_participante)
);

-- Tabela administrador
--
CREATE TABLE IF NOT EXISTS administrador (
    id_administrador INT NOT NULL AUTO_INCREMENT,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    PRIMARY KEY (id_administrador)
);

-- Tabela post
--
CREATE TABLE IF NOT EXISTS post (
    id_post INT NOT NULL AUTO_INCREMENT,
    id_autor INT NOT NULL,
    titulo VARCHAR(50) NOT NULL,
    descricao VARCHAR(255),
    PRIMARY KEY (id_post),
    CONSTRAINT fk_participantePost FOREIGN KEY (id_autor) REFERENCES
participante (id_participante)
);

-- Tabela reacaoPost
--
CREATE TABLE IF NOT EXISTS reacaoPost (
    id_reacaoPost INT NOT NULL,
    id_autor_reacao INT NOT NULL,
    tipo_gostei INT NOT NULL DEFAULT 0,
    tipo_parabens INT NOT NULL DEFAULT 0,
    tipo_apoio INT NOT NULL DEFAULT 0,
    tipo_amei INT NOT NULL DEFAULT 0,
```

```
tipo_genial INT NOT NULL DEFAULT 0,
tipo_divertido INT NOT NULL DEFAULT 0,
PRIMARY KEY (id_reacaoPost),
CONSTRAINT fk_autorReacaoPost FOREIGN KEY (id_autor_reacao) REFERENCES
participante(id_participante)
);

-- -----
-- Tabela midia
-- -----

CREATE TABLE IF NOT EXISTS midia (
    id_midia INT NOT NULL AUTO_INCREMENT,
    id_post INT NOT NULL,
    midia_ref VARCHAR(255) NOT NULL,
    PRIMARY KEY (id_midia),
    CONSTRAINT fk_postMidia FOREIGN KEY (id_post) REFERENCES post(id_post)
);

-- -----
-- Tabela evento
-- -----

CREATE TABLE IF NOT EXISTS evento (
    id_evento INT NOT NULL AUTO_INCREMENT,
    id_organizacao INT NOT NULL,
    id_post INT NOT NULL,
    data_hora_inicio DATETIME NOT NULL,
    data_hora_fim DATETIME NOT NULL,
    local VARCHAR(255) NOT NULL,
    link VARCHAR(255) NOT NULL,
    PRIMARY KEY (id_evento),
    CONSTRAINT fk_organizacaoEvento FOREIGN KEY (id_organizacao) REFERENCES
organizacao(id_organizacao),
    CONSTRAINT fk_postEvento FOREIGN KEY (id_post) REFERENCES post(id_post)
);

-- -----
-- Tabela comentario
-- -----

CREATE TABLE IF NOT EXISTS comentario (
    id_comentario INT NOT NULL AUTO_INCREMENT,
    id_post INT NOT NULL,
    id_autor_comentario INT NOT NULL,
    titulo_comentario VARCHAR(50),
    descricao_comentario VARCHAR(225),
    PRIMARY KEY (id_comentario),
    CONSTRAINT fk_postComentario FOREIGN KEY (id_post) REFERENCES
post(id_post),
```

```
        CONSTRAINT fk_autorComentario FOREIGN KEY (id_autor_comentario)
REFERENCES participante(id_participante)
);

-----
-- Tabela reacaoComentario
-----

CREATE TABLE IF NOT EXISTS reacaoComentario (
    id_reacaoComentario INT NOT NULL AUTO_INCREMENT,
    id_autor_reacao INT NOT NULL DEFAULT 0,
    tipo_gostei INT NOT NULL DEFAULT 0,
    tipo_parabens INT NOT NULL DEFAULT 0,
    tipo_apoio INT NOT NULL DEFAULT 0,
    tipo_amei INT NOT NULL DEFAULT 0,
    tipo_genial INT NOT NULL DEFAULT 0,
    tipo_divertido INT NOT NULL DEFAULT 0,
    PRIMARY KEY (id_reacaoComentario),
    CONSTRAINT fk_autorReacaoComentario FOREIGN KEY (id_autor_reacao)
REFERENCES participante(id_participante)
);

-----
-- Tabela denuncia
-----

CREATE TABLE denuncia (
    id_denuncia INT NOT NULL AUTO_INCREMENT,
    id_relator INT NOT NULL,
    id_relatado INT NOT NULL,
    id_administrador INT NULL,
    id_post INT NULL,
    titulo VARCHAR(50) NOT NULL,
    motivo VARCHAR(255) NOT NULL,
    status VARCHAR(45) NOT NULL DEFAULT 'Pendente',
    data_registro TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (id_denuncia),
    FOREIGN KEY (id_relator) REFERENCES participante(id_participante),
    FOREIGN KEY (id_relatado) REFERENCES participante(id_participante),
    FOREIGN KEY (id_administrador) REFERENCES
administrador(id_administrador),
    FOREIGN KEY (id_post) REFERENCES post(id_post)
);

-----
-- Tabela banido
-----

CREATE TABLE IF NOT EXISTS banido (
    id_banido INT NOT NULL AUTO_INCREMENT,
```



```
id_administrador INT NOT NULL,  
id_participante_banido INT NOT NULL,  
motivo VARCHAR(100) NOT NULL,  
create_time TIMESTAMP NOT NULL,  
PRIMARY KEY (id_banido),  
CONSTRAINT fk_participanteBanidio FOREIGN KEY (id_participante_banido)  
REFERENCES participante (id_participante),  
CONSTRAINT fk_administradorBanido FOREIGN KEY (id_administrador)  
REFERENCES administrador (id_administrador)  
);  
  
-----  
-- Tabela suspenso  
-----  
CREATE TABLE IF NOT EXISTS suspenso (  
    id_suspenso INT NOT NULL AUTO_INCREMENT,  
    id_administrador INT NOT NULL,  
    id_participante_suspenso INT NOT NULL,  
    motivo VARCHAR(100) NOT NULL,  
    data_hora_inicio DATETIME NOT NULL,  
    data_hora_fim DATETIME NOT NULL,  
    PRIMARY KEY (id_suspenso),  
    CONSTRAINT fk_administradorSuspenso FOREIGN KEY (id_administrador)  
REFERENCES administrador (id_administrador),  
    CONSTRAINT fk_participanteSuspenso FOREIGN KEY  
(id_participante_suspenso) REFERENCES participante (id_participante)  
);
```

ARTEFATO 6: Modelo Relacional

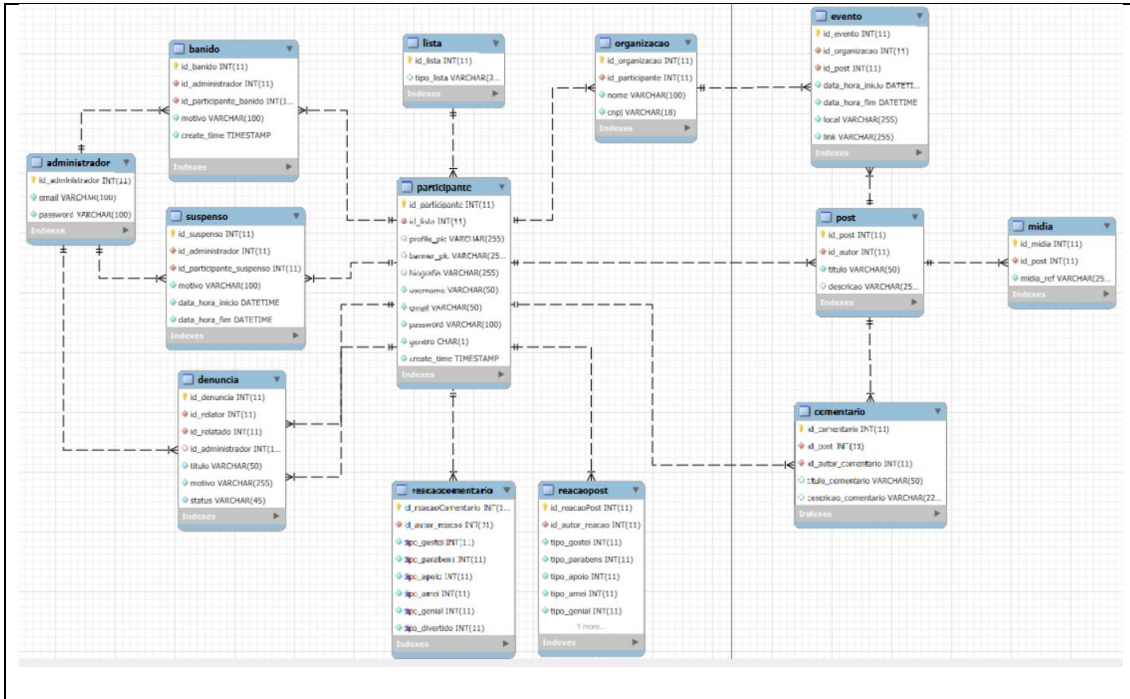
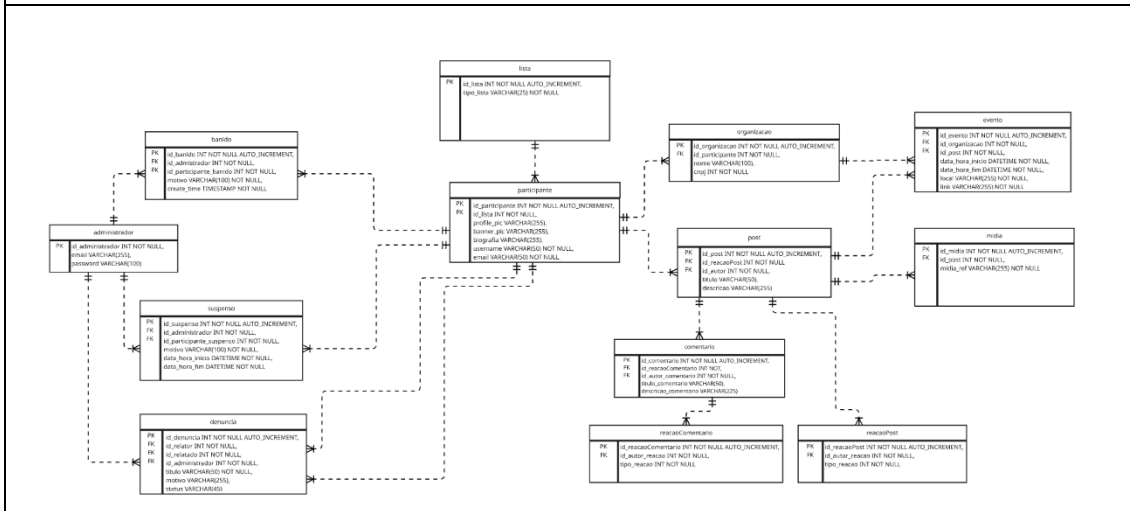
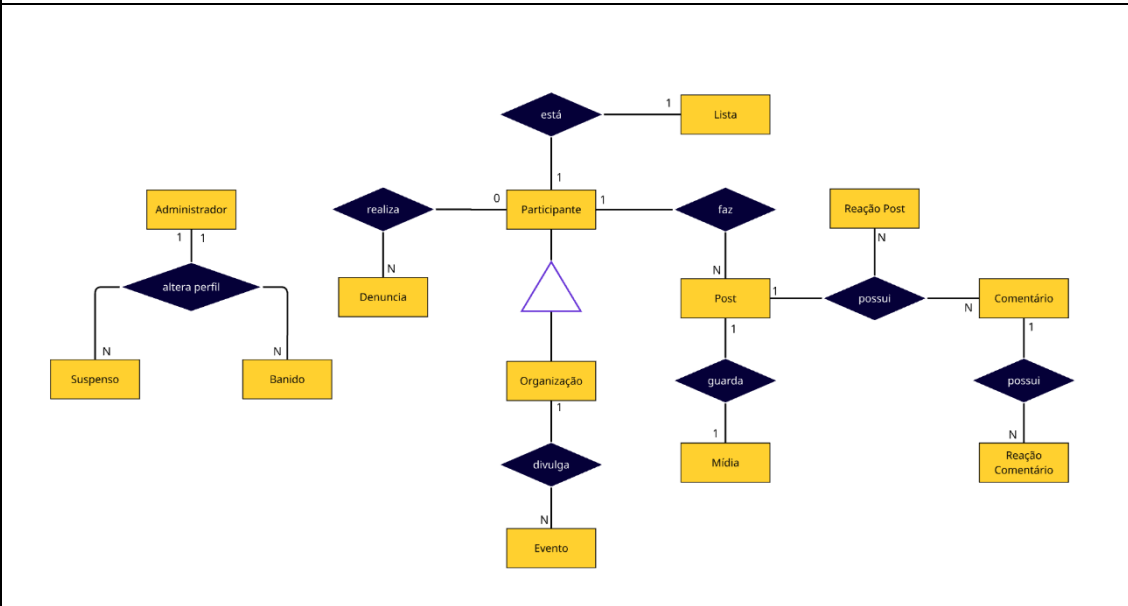


Figura 6 – Exemplo: Modelo Relacional gerado por engenharia reversa (MySQL Workbench).

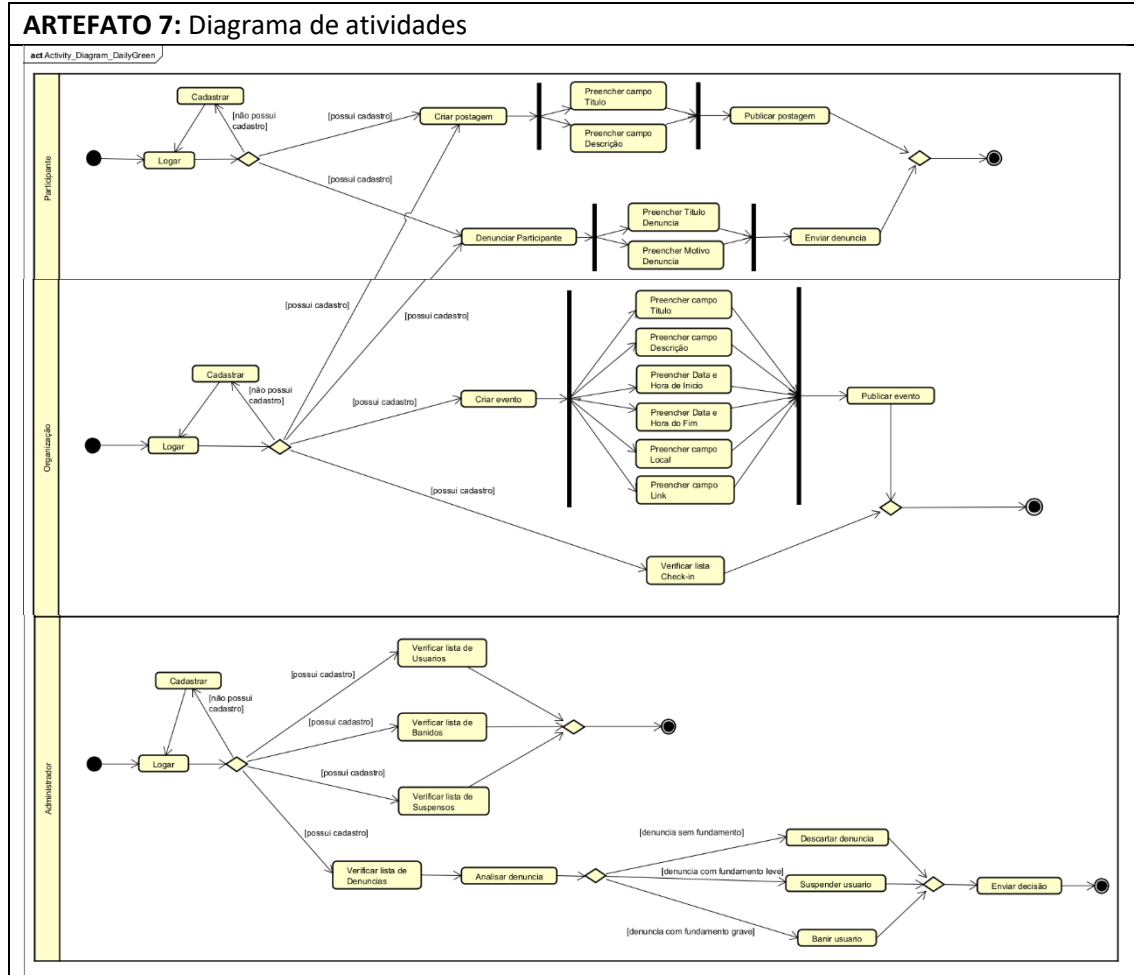
ARTEFATO 6: Modelo logico

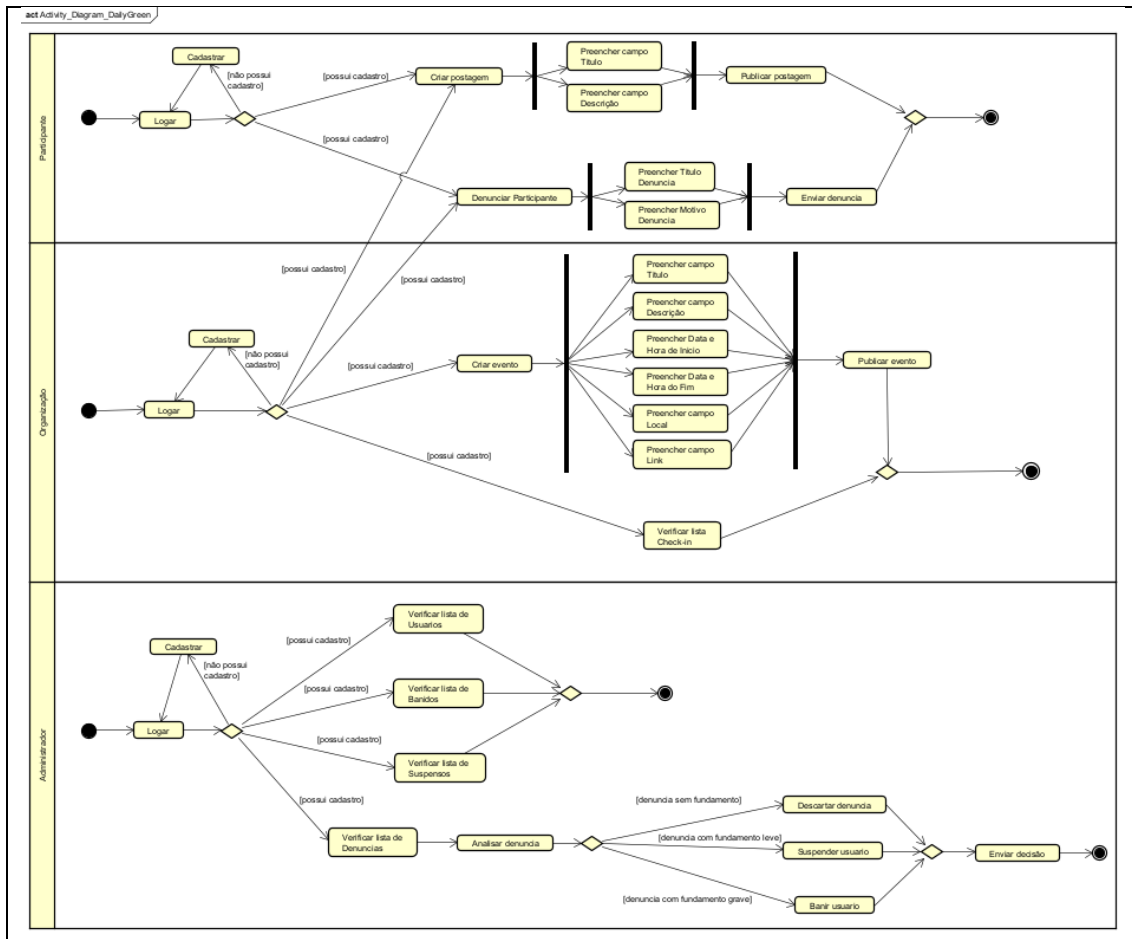


ARTEFATO 6: Modelo conceitual



ARTEFATO 7: Diagrama de Atividades





REFERÊNCIA BIBLIOGRÁFICAS

AGUIAR, F. **Product backlog building: concepção de um product backlog efetivo**. 2018. Disponível em: <https://speakerdeck.com/fabiogr/product-backlog-building>. Acesso em: 10 fevereiro 2022.

AGUIAR, F. **PBB_Canvas Template**. 2018. Disponível em: http://www.productbacklogbuilding.com/canvas/PBB_Canvas.pdf. Acesso em 10 de fevereiro de 2022.

SCHWABER, K.; SUTHERLAND, J. **Guia do SCRUM - o guia definitivo para o Scrum: as regras do jogo**. 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-European.pdf>. Acesso em: 10 fevereiro 2022.