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” 2](#_Toc256625337)

[ARTEFATO 2: Quadro “é – não é – faz – não faz” 3](#_Toc1054680460)

[ARTEFATO 3: Quadro “Visão de Produto”. 4](#_Toc1653315020)

[ARTEFATO 4: Canvas PBB 5](#_Toc1029958449)

[ARTEFATO 5: Relação de User Stories 6](#_Toc1933579701)

[ARTEFATO 6: Modelo Relacional 13](#_Toc274977675)

[ARTEFATO 7: Diagrama de Atividades 20](#_Toc1968429483)

[REFERÊNCIA BIBLIOGRÁFICAS 22](#_Toc144049300)

ÍNDICE DE ILUSTRAÇÕES

[Figura 1 – Quadro “3 Objetivos”. 3](#_Toc198185799)

[Figura 2 – Quadro “é – não é – faz – não faz”. 4](#_Toc198185800)

[Figura 3 – Quadro "Visão de Produto". Fonte: AGUIAR, F. 2018. 5](#_Toc198185801)

[Figura 4 – Canvas PBB: "Product Backlog Building". Fonte: AGUIAR, F. 2018. 6](#_Toc198185802)

[Figura 5 - User Stories e Critérios de Aceite. Fonte: AGUIAR, F. 2018. 8](#_Toc198185803)

[Figura 6 – Exemplo: Modelo Relacional gerado por engenharia reversa (MySQL Workbench). 13](#_Toc198185804)

# 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

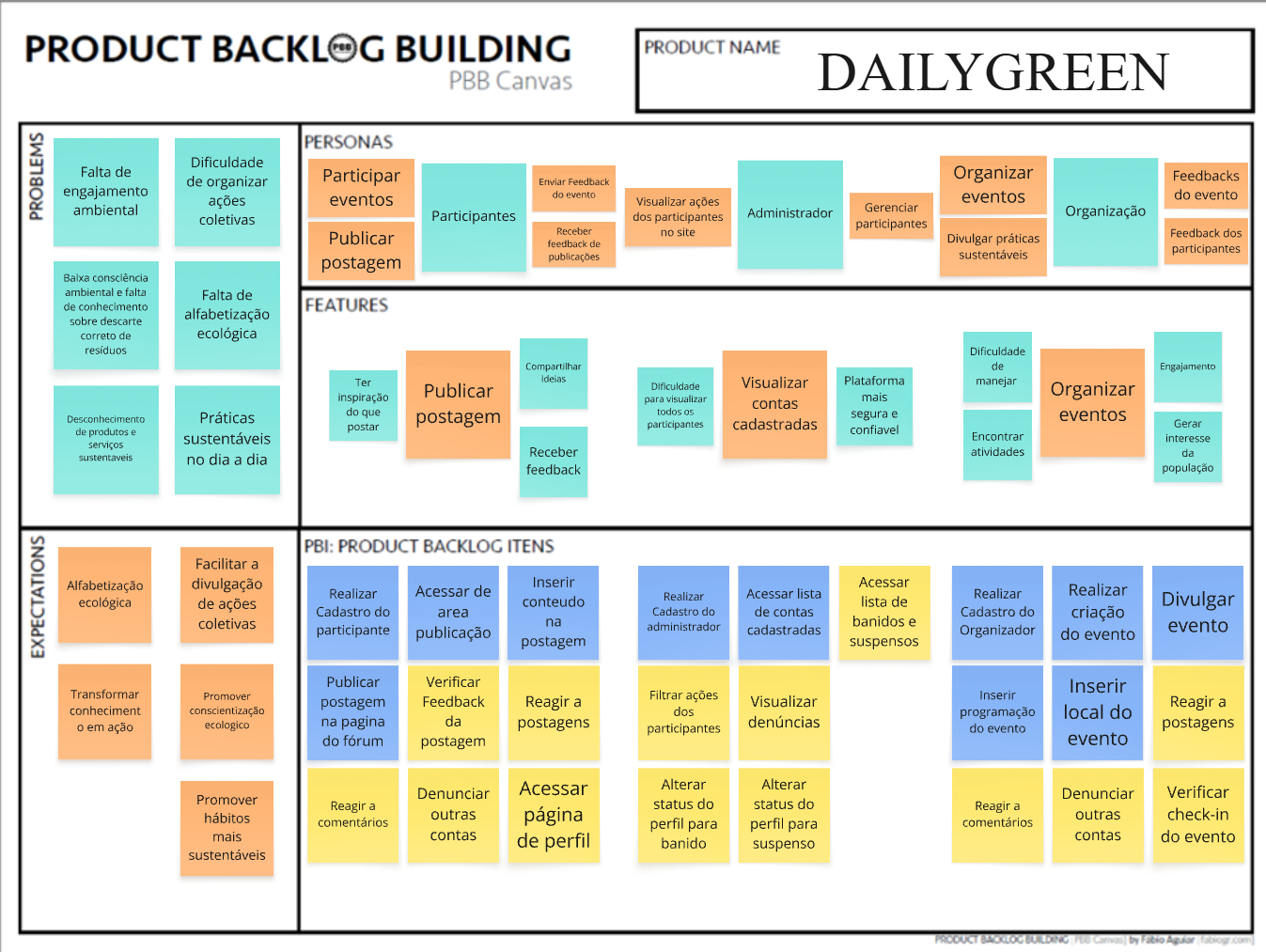


Figura 4 – Canvas PBB: "Product Backlog Building". Fonte: AGUIAR, F. 2018.

# 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 e-mail 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 acesos na plataforma. | |
| **Critério de**  **Aceite 1** | **DADO** **QUE**: Possuo 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 inapropriado 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 inapropriado 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 "Suspender".  **ENTÃO**: O sistema exibe uma mensagem de confirmação, o usuário é adicionado à lista de suspensos e à graylist e o staus 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](https://trello.com/b/U448eKZC/dailygreen)

[Github DailyGreen](https://github.com/Garudaakuma/DailyGreen-Project)

# 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

|  |
| --- |
| **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.