

ETS

ROBERT BOSCH

Ariane Oliveira Silva

Gabriela Alejandra Bergamine dos Santos

Gustavo Bruno de Paiva Florentino

Israel Santana de Moraes

Luana Grandi Mota

Talita Cristina Alves Lobato

eLOGiar

Explicação do projeto em uma frase:

Boas atitudes não devem ficar invisíveis.

Campinas SP

SUMÁRIO

1	INTRODUÇÃO E JUSTIFICATIVA.....	3
2	OBJETIVOS.....	3
2.1.	Objetivos Gerais.....	3
2.2.	Objetivos Específicos	3
3	PRODUCT BACKLOG	4
4	REQUISITOS NÃO FUNCIONAIS	4
5	PREMISSAS	4
6	RESTRIÇÕES	4
7	Sprints.....	5
7.1	Diagrama de caso de uso.....	8
9.	Modelo de Dados.....	8
9.1	Diagrama de Entidade e Relacionamento	9
9.2	Modelo lógico do banco de dados.....	10
10.	Pesquisas realizadas.....	10
11.	Metodologia.....	11
12.	ANEXOS	12

1 INTRODUÇÃO E JUSTIFICATIVA

O **eLOGiar** surgiu com o propósito de ser um auxílio para tornar realidade uma visão que nasceu nas áreas de logística da Bosch. A logística dentro da Bosch é muito grande, tem aproximadamente 300 funcionários que são divididos em 7 áreas diferentes. Um gestor da área tem que cuidar de diversos colaboradores, por conta disso é muito complicado estar sempre atento as atitudes de cada um deles de forma individual até mesmo por conta da correria que o cargo exige, isso gerou incomodo a eles.

Pois como consequência ocorre de os colaboradores terem boas atitudes e elas não serem reconhecidas devidamente e nem se quer vistas, por conta dessa problemática surgiu o bordão: "Boas atitudes não devem ficar invisíveis".

2 OBJETIVOS

2.1. Objetivos Gerais

Objetivo principal do **eLOGiar** é servir como um meio de conectar os colaboradores da área Logística, reconhecendo atos de proatividade, eficiência em resultados, inovação e trabalho em equipe, por meio de indicações feitas entre seus colegas de trabalho, o qual são premiados com prêmios em um evento especial para o setor de logística, o ConectaLOG.

2.2. Objetivos Específicos

- Desenvolver uma ferramenta de fácil acesso para elogiar alguém;
- Conectar as pessoas das diversas áreas de logística da planta;
- Compartilhar ideias e elogiar as atitudes de cada um;
- Entrar no **eLOGiar**, e elogiar seus feitos, características;
- Servir como parâmetro em premiações na área;

3 PRODUCT BACKLOG

- **RF01** – Administradores
 - **RF01.1** – Cadastrar os usuários no sistema.
 - **RF01.2** – Modificar as categorias.
 - **RF01.3** – Exportar os dados de votação em Excel.
- **RF02** – Usuário padrão
 - **RF02.01** – Deve apenas selecionar a categoria.
 - **RF02.02** – Deve selecionar uma pessoa para votar.
 - **RF02.03** – Deve inserir uma mensagem opcional.
 - **RF02.04** – Deve enviar seu voto.

4 REQUISITOS NÃO FUNCIONAIS

- **RNF01** – O sistema permitir que o administrador realize operações de Criar, Atualizar e Remover usuários ou categorias;
- **RNF02** – O sistema deve permitir apenas acessar administração com login.
- **RNF03** – O usuário só é permitido votar em uma pessoa uma única vez por categoria;
- **RNF04** – O sistema deve ser capaz de aguentar várias requisições de votos;
- **RNF05** – O banco de dados deve armazenar todas as informações colocadas no momento da votação;
- **RNF06** – O servidor de hospedagem deve ser em Django.

5 PREMISSAS

- **PRE01** – Acessar o eLOGiar
- **PRE02** – Acessar a escolha de categoria
- **PRE03** – Escolher o usuário a ser votado e o votante
- **PRE04** – Realizar Voto
- **PRE05** – Confirmar voto através de vídeo.

6 RESTRIÇÕES

- **RES01** – Necessário hospedagem do servidor Django.
- **RES02** – Necessário um dispositivo para acesso do site.

- **RES03** – Um usuário não pode votar nele mesmo ou em outro colega duas vezes na mesma categoria.
- **RES04** – Necessário o usuário estar cadastrado no banco de dados para votar ou ser votado.

7 Sprints

Primeiro Dia

Sorteio dos grupos de projeto do Hackthon;

Reunião de apresentação do projeto com a Roseli;

Houve agendamento de reunião para mais detalhes do projeto;

Início da prototipação;

Segundo Dia

Realizamos uma reunião com a Monique e Marcelo para entendermos mais sobre o projeto.

Definimos o design do Figma.

Fizemos 70% de todo design do Figma.

Programamos a tela inicial e principal do WebApp.

Definimos o escopo do backend, além de desenvolver o documento de levantamento de requisitos do software.

Fizemos escopo do backend para entender a lógica por trás.

Sorteio dos grupos de projeto do Hackton;

Terceiro Dia

Finalizamos as páginas principais no HTML, e fizemos toda estilização;

Desenvolvemos o design de páginas extras o Figma;

Tivemos uma breve reunião entre os integrantes sobre o vídeo pitch;

Desenvolvemos 60% de todo o backend.

Fizemos 100% de todo design do Figma.

Fizemos o escopo do banco de dados;

Começo do desenvolvimento do vídeo pitch e apresentação;

Começamos integração entre front e back.

Quarto Dia

Finalizamos o backend;

Finalizamos a integração do back com o front;

Gravamos uma parte do vídeo pitch;

Começamos elaborar o roteiro da apresentação;

Tivemos uma breve reunião entre os integrantes sobre o vídeo pitch;

Quinto Dia

Finalização de GIFs;

Realização do Vídeo Pitch;

Finalização do Frontend;

Reunião com os Focal Points;

Roteirização de Atividades para a próxima semana;

Sexto Dia

Finalização de GIFs;

Realização do Vídeo Pitch;

Finalização do Frontend;

Sétimo Dia

- Finalizamos o site;
- Realizamos ajustes no front;
- Demos prosseguimento para edição do vídeo pitch;
- Elaboramos a dinâmica da apresentação;
- Finalizamos a documentação;
- Demos início o desenvolvimento da documentação em Docsaurus.

Oitavo Dia

Finalização do site

Treinamento para apresentação

Nono Dia

Treinamento para apresentação e conclusão do Docsaurus

7.1 Diagrama de caso de uso

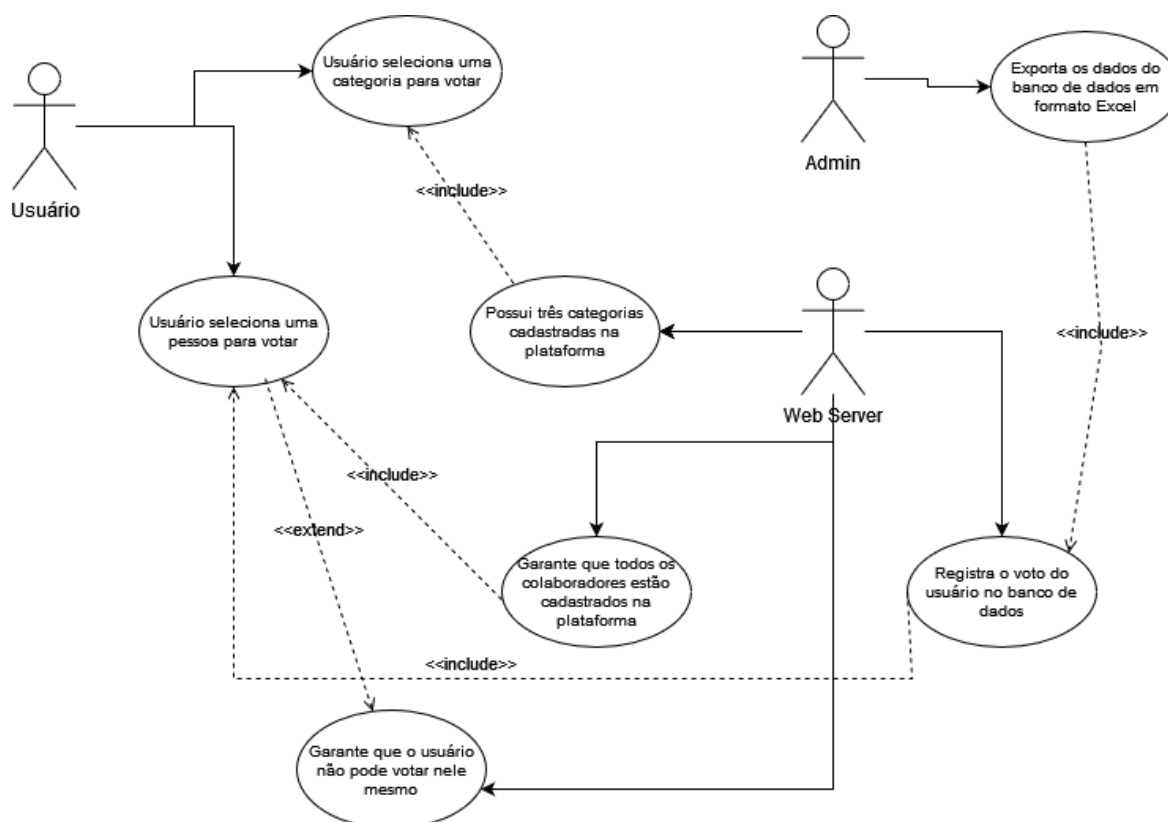


Imagem 2 – Diagrama de Caso de uso

9. Modelo de Dados

O banco de dados foi desenvolvido usando o SQLite como SGBD integrado a framework de Python, Django. Toda sua estrutura lógica e conceitual estará descrita abaixo. O modelo é composto por três entidades principais:

Categoria: Armazena informações sobre as categorias de votação, como o nome da categoria.

Usuário: Armazena informações sobre os usuários que podem votar, como o nome do usuário.

Votos: Armazena os votos em si, incluindo a mensagem votada, a categoria e o usuário associados ao voto.

Relacionamentos:

Categoria - Votos: Um relacionamento de um para muitos, onde uma categoria pode ter múltiplos votos associados a ela.

Usuário - Votos: Um relacionamento de um para muitos, onde um usuário pode realizar múltiplos votos.

Chaves:

Chaves Primárias: `id_categoria`, `id_usuario` e `id_votos` identificam exclusivamente cada registro nas respectivas tabelas.

Chaves Estrangeiras: `id_categoria` e `id_usuario` na tabela Votos estabelecem os relacionamentos com as tabelas Categoria e Usuário, respectivamente.

Este modelo fornece uma estrutura básica para um sistema de votação, permitindo o registro de votos de usuários em diferentes categorias. Ele pode ser expandido e adaptado para atender a requisitos mais complexos, como a inclusão de informações adicionais sobre usuários, categorias ou votos.

9.1 Diagrama de Entidade e Relacionamento

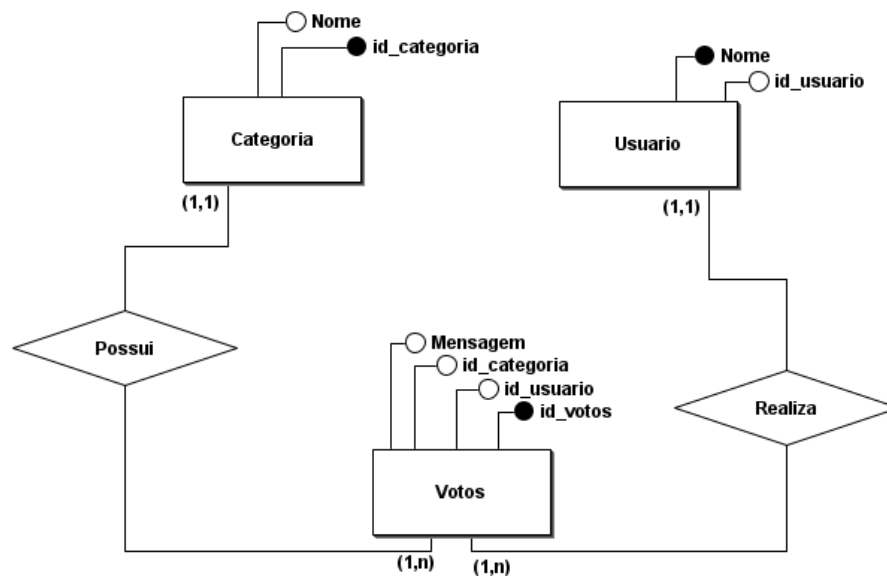


Imagem 6 – Diagrama de Entidade e relacionamento

9.2 Modelo lógico do banco de dados

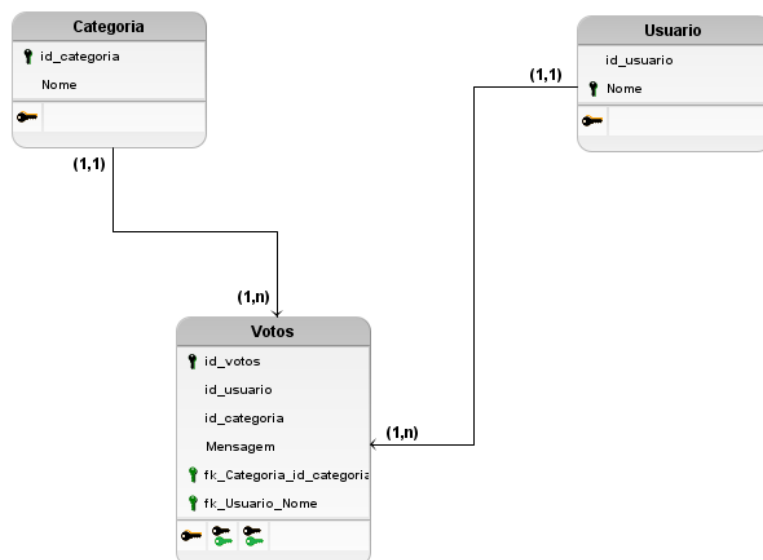


Imagem 7 – Modelo logico do banco de dados

10. Pesquisas realizadas

Conversamos com a área para tentar entender quais eram suas dores em relação ao projeto. Ondes após algumas reuniões entendemos e entre os integrantes conversamos e chegamos as conclusões para desenvolver o projeto.

11. Metodologia

A metodologia ágil foi adotada para garantir flexibilidade e entregas rápidas no desenvolvimento deste projeto. Usamos reuniões diárias (daily stand-ups), para alinhar o que foi feito no dia anterior e planejar o que seria feito no dia seguinte, garantindo que todos estivessem sempre atualizados e cientes do progresso.

O projeto foi dividido em sprints com prazos definidos para cada etapa. As fases principais foram:

ETAPA 1.

Sprints:

Design UI/UX: Criação da interface e experiência do usuário, visando usabilidade e estética.

Escopo BackEnd: Desenvolvimento do escopo do backend, junto com a linguagem que seria usada.

Escopo FrontEnd: Escolha das linguagens que seriam usadas no FrontEnd.

Mascote: Desenvolvido a mascote junto com suas variações para o projeto.

Definição de cores: Definição de cores de acordo com a mascote para o projeto.

ETAPA 2.

Sprints:

Back-end: Desenvolvimento da estrutura de servidor e banco de dados.

Front-end: Implementação da parte visual e interativa, conectando-se ao back-end.

Integração Back-end/Front-end: Garantir a integração do Front com Back.

3.1. Documentação: Elaboração da documentação técnica do BackEnd e banco de dados.

Roteiro vídeo pitch: Roteirização do vídeo pitch seguido de seu objetivo.

4.1. Roteiro: Definido o roteiro, realizamos o levantamento do que seria necessário para gravar e editar, e o que usaríamos.

4.2. Gravação do vídeo: Realizado a gravação do vídeo pitch.

ETAPA 3.

Sprints:

Edição vídeo pitch: Realizada a edição do vídeo.

Finalização dos slides: Realizado a finalização e ajustes nos slides de apresentação

Treinamento para Apresentação: Ensaios para alinhar todos os detalhes do projeto e apresentação.

Revisão do projeto: Revisar o projeto e realizar possíveis ajustes.

Para organização e desenvolvimento das Sprints, usamos Trello. para facilitar a do tempo. No desenvolvimento de Sprints junto com o Backlog, usamos o trello e criamos Abas para cada Sprint descrita acima, onde após concluída, era adicionada na Aba de concluídos.

12. ANEXOS

Wireframe 1 – Light Mode

