INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DO MARANHÃO
CAMPUS SÃO LUÍS - MONTE CASTELO
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JHONATHAN DA ROCHA DA CRUZ
KATERINY BISPO DE DEUS
SAULO FERRO MACIEL

Contribuições para o PostgreSQL

SÃO LUÍS – MA
2025

SUMÁRIO

1. Contribuições Técnicas

1.1. Teste

Esta contribuição consiste na elaboração de um **teste automatizado de regressão** para o PostgreSQL, utilizando a suíte oficial de testes do projeto, baseada na ferramenta **pg_regress**.

O objetivo do teste é **validar o comportamento de funções de agregação SQL na presença de valores NULL**, garantindo que operações como COUNT e AVG sigam o padrão esperado e não apresentem regressões em versões futuras do sistema.

1.1.1. Contexto da Contribuição

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto amplamente utilizado, cujo desenvolvimento é sustentado por uma comunidade ativa. Para assegurar a estabilidade e a confiabilidade do sistema, o projeto mantém uma ampla suíte de testes automatizados.

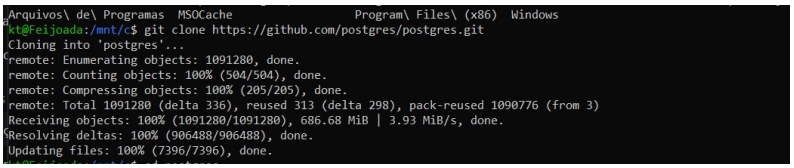
Dentro desse contexto, a contribuição realizada insere-se na categoria de **Testes Automatizados**, conforme previsto no menu de contribuições da disciplina.

1.1.2. Ambiente de Desenvolvimento

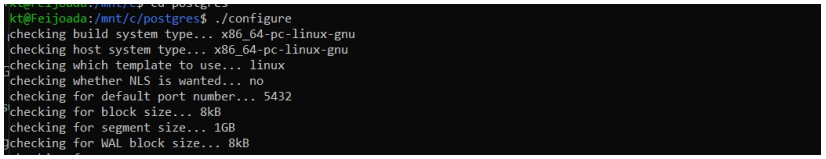
Para a realização da contribuição, foi utilizado o **Windows Subsystem for Linux (WSL)**, com distribuição Ubuntu, por oferecer um ambiente Linux compatível com o fluxo oficial de desenvolvimento do PostgreSQL.

As principais etapas de configuração do ambiente envolveram:

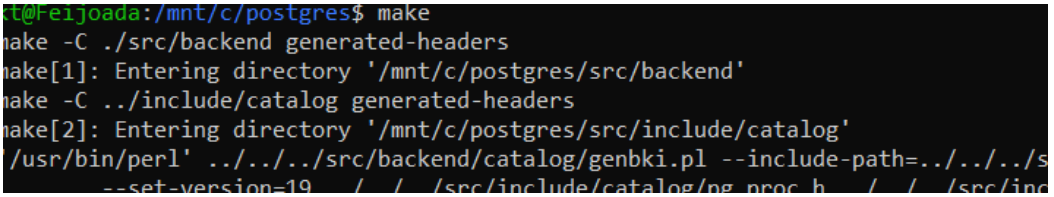
- Instalação das dependências de compilação (GCC, Make, bibliotecas de desenvolvimento);
- Clonagem do repositório oficial do PostgreSQL;



- Execução do script configure para verificação do ambiente;



- Compilação do sistema por meio do comando make.



1.1.3. Implementação do Teste

O teste foi implementado seguindo o padrão adotado pela comunidade do PostgreSQL, que utiliza arquivos SQL para definição dos testes e arquivos de saída esperada para validação dos resultados.

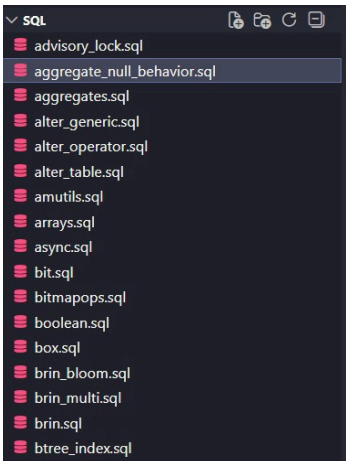
Foram criados os seguintes arquivos:

- Um arquivo .sql, contendo os comandos SQL a serem testados;

```sql
-- Teste de comportamento de agregações com valores NULL

CREATE TEMP TABLE t (x INT);

INSERT INTO t VALUES (1), (NULL), (2);

-- COUNT ignora NULL
SELECT COUNT(x) FROM t;

-- COUNT(*) conta todas as linhas
SELECT COUNT(*) FROM t;

-- AVG ignora NULL
SELECT AVG(x) FROM t;

-- Teste com tabela vazia
DELETE FROM t;

SELECT AVG(x) FROM t;
```

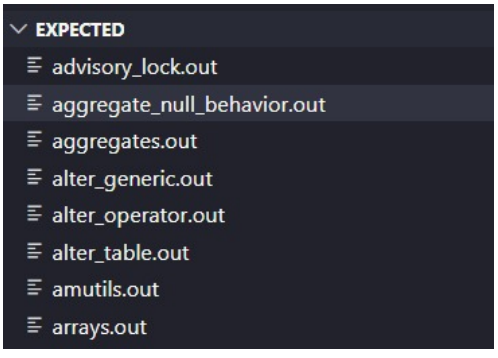
A imagem a seguir exibe o arquivo de contribuição de teste inserido no diretório clonado da comunidade:

Um arquivo .out, contendo o resultado esperado da execução desses comandos;



```
SQL
 count
-------
     2
(1 row)

 count
-------
     3
(1 row)

         avg
---------------------
 1.5000000000000000
(1 row)

         avg
---------------------

(1 row)
```

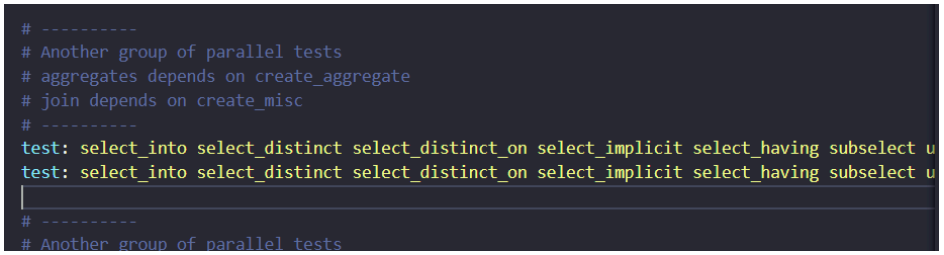
A imagem a seguir exibe o arquivo de contribuição de teste inserido no diretório clonado da comunidade:



Registro do teste no arquivo parallel_schedule, responsável por organizar a execução paralela dos testes.

SQL

```
# ----------
# Another group of parallel tests
# aggregates depends on create_aggregate
# join depends on create_misc
# ----------
test: select_into select_distinct select_distinct_on select_implicit select_having subselect union case join aggregates transactions random portals arrays btree_index hash_index update delete namespace prepared_xacts
test: select_into select_distinct select_distinct_on select_implicit select_having subselect union case join aggregates transactions random portals arrays btree_index hash_index update delete namespace prepared_xacts aggregate_null_behavior
```

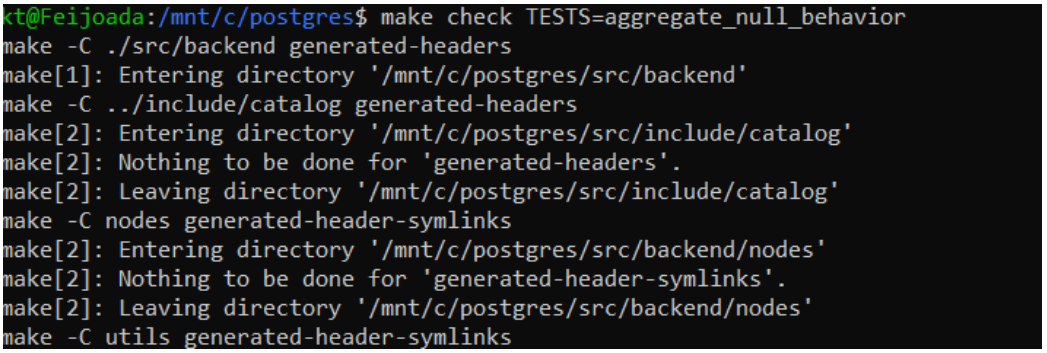


O teste verifica especificamente:

- O comportamento da função COUNT ao ignorar valores NULL;
- A diferença entre COUNT(coluna) e COUNT(*);
- O comportamento da função AVG em conjuntos de dados contendo valores NULL e em tabelas vazias.

1.1.4. Execução e Validação

Após a compilação do PostgreSQL, o teste foi executado de forma isolada utilizando o comando:



A execução do teste compara automaticamente a saída gerada pelo sistema com o arquivo de resultado esperado, validando a consistência do comportamento observado.

1.1.5. Relevância da Contribuição

A inclusão de novos testes automatizados contribui diretamente para:

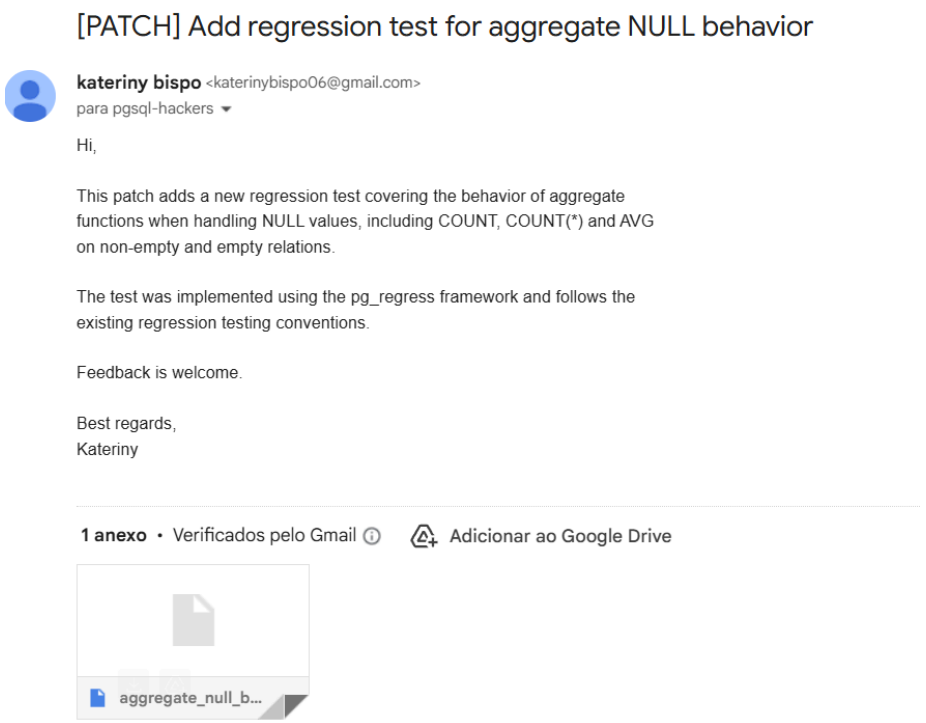
- Aumento da cobertura de testes do PostgreSQL;
- Prevenção de regressões em funcionalidades existentes;
- Fortalecimento da confiabilidade do sistema ao longo de sua evolução.

Esse tipo de contribuição é valorizado pela comunidade, pois auxilia na manutenção da qualidade do software livre sem a necessidade de alterações diretas no código-fonte principal.

1.1.6. Forma de Submissão à Comunidade

A contribuição foi estruturada de acordo com o fluxo oficial do PostgreSQL, por meio da geração de um arquivo .patch, que pode ser submetido à lista de discussão da comunidade (pgsql-hackers) para revisão e possível integração ao projeto.

- Email de submissão do Patch de testes



“Olá,

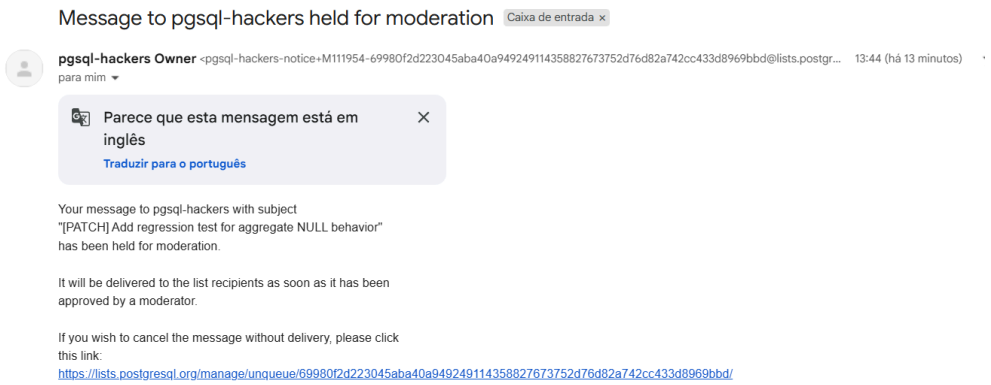
Este patch adiciona um novo teste de regressão que abrange o comportamento de funções de agregação
ao lidar com valores NULL, incluindo COUNT, COUNT(*) e AVG
em relações vazias e não vazias.

O teste foi implementado usando o framework pg_regress e segue as convenções de teste de regressão existentes.

Sugestões são bem-vindas.

Atenciosamente.”

- Resposta equipe moderadora do repositório:



Message to pgsql-hackers held for moderation Caixa de entrada ×

pgsql-hackers Owner <pgsql-hackers-notice+M111954-69980f2d223045aba40a9492491143588276737 52d76d82a742cc433d8969bbd@lists.postgr... 13:44 (há 13 minutos)
para mim

Parece que esta mensagem está em inglês
Traduzir para o português

Your message to pgsql-hackers with subject
"[PATCH] Add regression test for aggregate NULL behavior"
has been held for moderation.

It will be delivered to the list recipients as soon as it has been approved by a moderator.

If you wish to cancel the message without delivery, please click this link:
https://lists.postgresql.org/manage/unqueue/69980f2d223045aba40a9492491143588276737 52d76d82a742cc433d8969bbd/

> “Sua mensagem para pgsql-hackers com o assunto "[PATCH] Adicionar teste de regressão para comportamento de agregação NULL"
> foi liberada da moderação e será entregue a todos os membros da lista em breve.”

2. Contribuições de Documentação/Suporte

2.1. Replicação Lógica — logical-replication.sgml

A documentação:

- Explicava conceitos (publisher, subscriber, slot, publication)
- Tinha poucos exemplos completos
- Não deixava claro o fluxo real de configuração
- Não abordava erros comuns

Com a melhoria:

Nova seção: Exemplo prático completo de Replicação Lógica

Replicação lógica permite replicar dados de forma seletiva usando um modelo de publicação e assinatura.
Este exemplo demonstra um fluxo completo de configuração entre dois servidores PostgreSQL.

2.2. Autovacuum — maintenance.sgml

A documentação:

- Autovacuum descrito de forma teórica
- Pouca clareza sobre quando ele não roda
- Pouca explicação de parâmetros críticos

Com a melhoria

Comportamento do Autovacuum

O autovacuum é um processo automático responsável por:

- Remover tuplas mortas
- Atualizar estatísticas
- Prevenir o *wraparound* de IDs de transação

No entanto, o autovacuum pode não executar imediatamente mesmo quando o limite de tuplas mortas é atingido, dependendo de:

- Carga do sistema
- Número de workers disponíveis
- Prioridade do processo

2.3. Freeze e Wraparound — mvcc.sgml

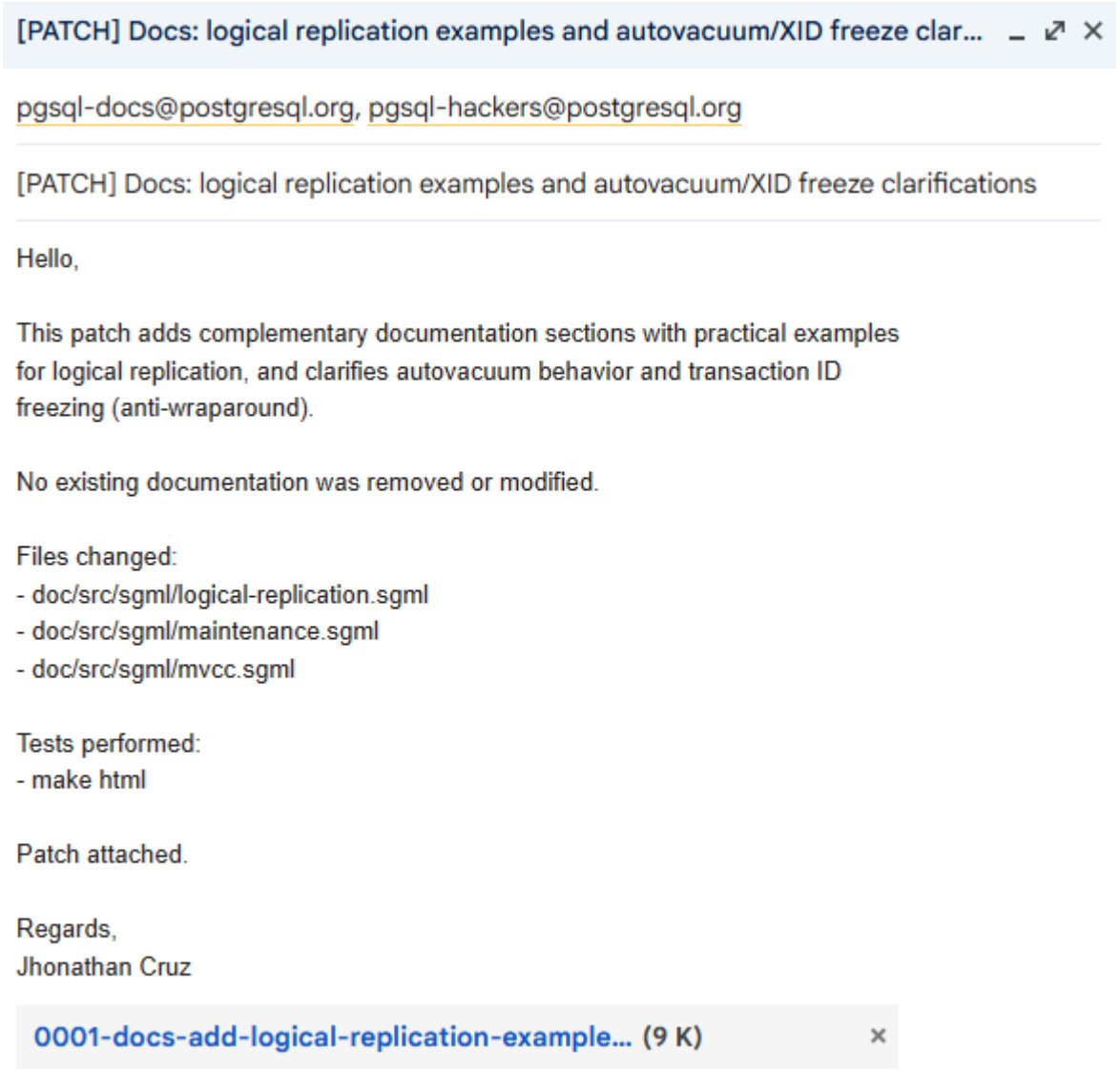
A documentação:

- Conceito de Freeze explicado
- Pouca ligação prática com Autovacuum
- Pouca orientação operacional

Com a melhoria

- Relação com Autovacuum
- Alerta crítico adicionado

- E-mail com o .path com as novas sessões adicionadas.



[PATCH] Docs: logical replication examples and autovacuum/XID freeze clar...

pgsql-docs@postgresql.org, pgsql-hackers@postgresql.org

[PATCH] Docs: logical replication examples and autovacuum/XID freeze clarifications

Hello,

This patch adds complementary documentation sections with practical examples
for logical replication, and clarifies autovacuum behavior and transaction ID
freezing (anti-wraparound).

No existing documentation was removed or modified.

Files changed:
- doc/src/sgml/logical-replication.sgml
- doc/src/sgml/maintenance.sgml
- doc/src/sgml/mvcc.sgml

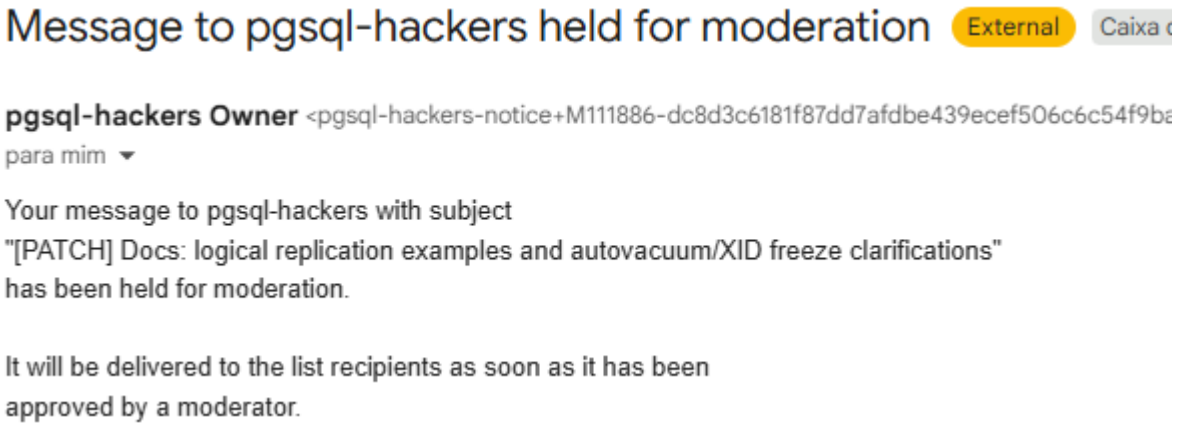
Tests performed:
- make html

Patch attached.

Regards,
Jhonathan Cruz

0001-docs-add-logical-replication-example... (9 K)

- Resposta do Repositório:



Message to pgsql-hackers held for moderation External Caixa

pgsql-hackers Owner <pgsql-hackers-notice+M111886-dc8d3c6181f87dd7afdbe439ecef506c6c54f9ba
para mim

Your message to pgsql-hackers with subject
"[PATCH] Docs: logical replication examples and autovacuum/XID freeze clarifications"
has been held for moderation.

It will be delivered to the list recipients as soon as it has been
approved by a moderator.

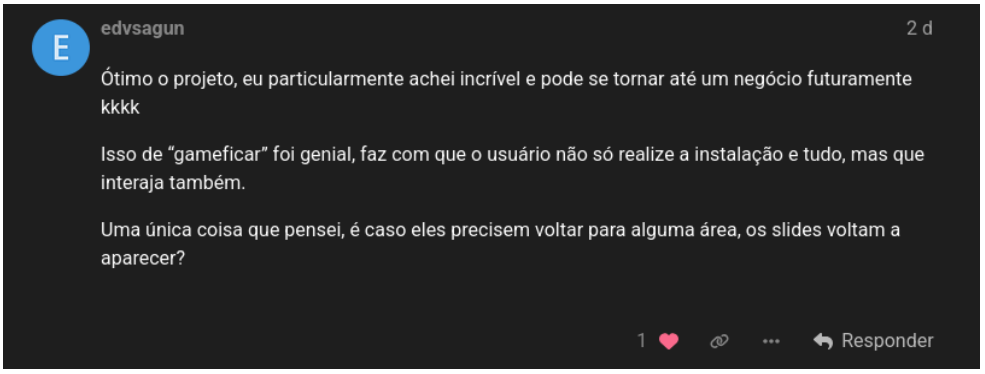
2.3. DISTINCT ON + ORDER BY e JIT with aggregation

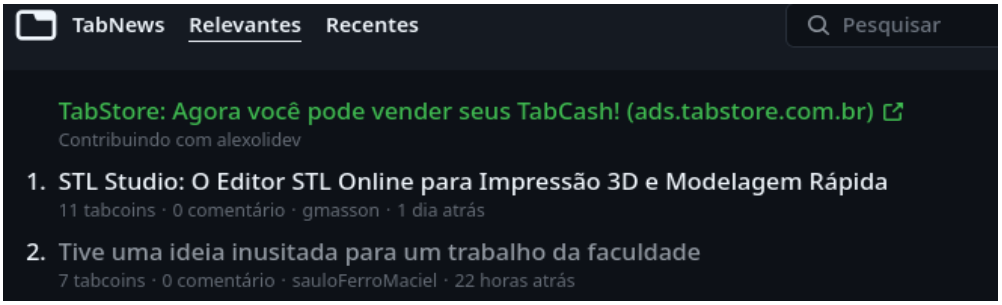
3. Contribuições Criativas/Divulgação

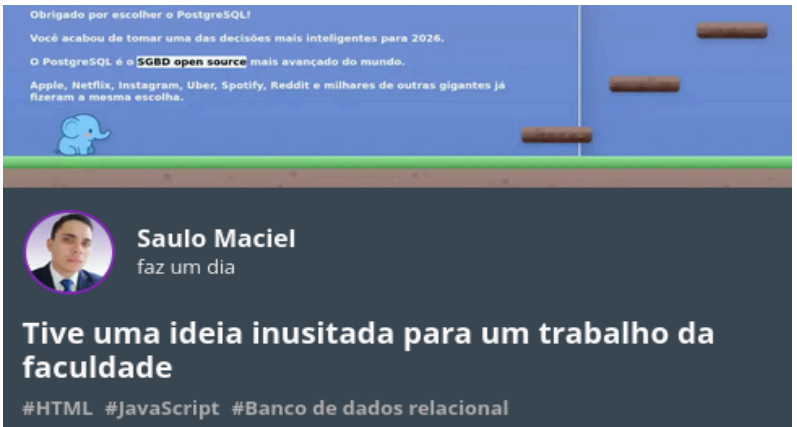
A equipe propôs criar alternativas para melhorar a divulgação aos novos usuários. Sendo assim, foi analisado o funil de marketing e clipping em redes sociais sobre o usufruto do PostgreSQL para poder ter uma noção dos problemas do projeto com sua base de usuários estruturais.

Foi constatado via comentários em fóruns que a maior problemática é a falta de uma unificação acessível para a introdução ao PostgreSQL.

Com isso, o integrante Saulo aproveitou sua game engine própria, na qual batizaram de Game Slide, para fazer um videojogo que unificasse os principais passos para aprendizado do SGBD.







Depois de 3 dias de desenho de elementos (cenário e personagens), o jogo de plataforma foi divulgado em fóruns e no youtube como alternativa de aprendizado que usa o viés de ***"gamificação infantil"***; principio de simplificar processos para que todas as faixas etárias consigam absorver partes do conteúdo com pouco esforço.

Como trata-se de um protótipo, o jogo foi disponibilizado via Github Pages com licença MIT. Pode usufruí-lo pelo link: rebrand.ly/qsduw7w

O projeto foi bem abraçado pelas comunidades. Alguns sugeriram que

transformassem o Game Slide em um serviço educacional, como pode-se observar nas imagens retiradas do Diolinux (link: rebrand.ly/eeq1qr4).

Podemos observar que o projeto obteve aceitação por vários usuários que forneceram feedbacks coesos, sensatos e humorados, o que evidencia êxito da equipe em inovar na comunicação mais assertiva e divertida.

Em outros fóruns, o projeto teve bastante visualizações. Chegando a ficar nos tópicos mais relevantes como foi no caso do TabNews (link: rebrand.ly/3kqennm) onde ficou em 2º lugar e no Dio (link: rebrand.ly/xjdaa7j) que passou de 100 visualizações e impulsionamentos.