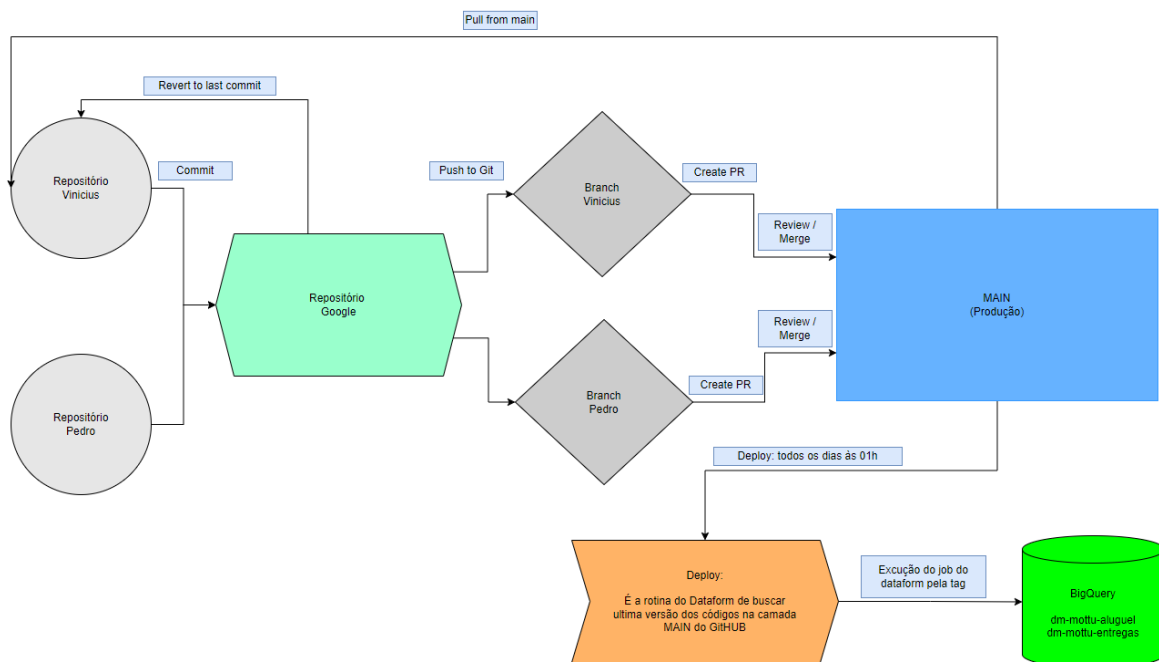


3.5. Levando seu código para produção - Ciclo de desenvolvimento e CI/CD

O Dataform possui um repositório local dentro da Google Cloud. E nos vinculamos o repositório do Dataform ao GitHub da Mottu para centralizar todos os códigos e customizar a esteira de CI/CD com camadas de aprovação, checklists e automatização de deploy com o GitHubActions.

Depois de vincular os repositórios, as alterações feitas em um espaço de trabalho de desenvolvimento do Dataform podem ser enviadas e extraídas do repositório GitHub.



Lógica de versionamento

1. Repositório

- **Local (workspace dentro do Dataform):** Um repositório local é uma cópia do seu projeto armazenada no seu próprio sistema de arquivos. Ele contém o histórico completo do projeto e todos os arquivos relevantes. O repositório local é onde você realiza suas alterações, faz commits e testes.
- **Remoto (Github):** Um repositório remoto é uma cópia do projeto armazenada em um servidor ou em outra máquina. Alterações feitas localmente são refletidas no repositório remoto a partir do push.

2. Commit

Um commit é uma operação no sistema de controle de versão Git que registra as alterações feitas nos arquivos do seu projeto. Cada commit representa um ponto específico no histórico do seu código e contém informações sobre as modificações realizadas. Os commits são fundamentais para o rastreamento do desenvolvimento, permitindo que você mantenha um histórico claro e reverta para versões anteriores, se necessário.

3. Push

Envia suas alterações locais para o repositório remoto. No caso do Dataform, alterações vão ser feitas apenas em branches de desenvolvimento locais.

4. Pull

Obtém as alterações do repositório remoto e as mescla no seu branch local. No dataform, é possível dar pull da branch padrão (main) ou da versão remota da branch de desenvolvimento.

5. Merge

Mescla alterações de uma branch para outra.

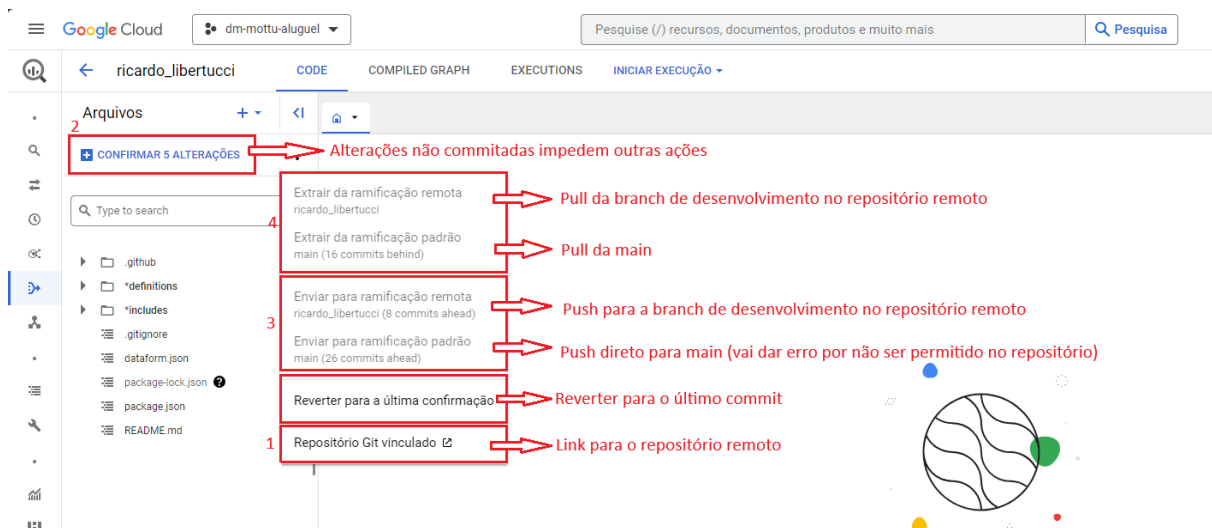
6. Pull request

Um pull request (PR), também conhecido como merge request em alguns sistemas de controle de versão, é uma solicitação para mesclar as alterações feitas em uma branch para outra branch. As alterações só serão refletidas na branch destino após aprovação dos reviewers.

7. Branch

Uma branch, em sistemas de controle de versão, é uma linha de desenvolvimento independente que contém seu próprio conjunto de alterações no código-fonte.

- **Branch de desenvolvimento:** No Dataform, cada espaço de trabalho (workspace) funciona como uma branch com o mesmo nome do workspace. As alterações feitas nele são salvas apenas na versão local da branch quando é feito um commit. Quando é feito um push, a versão remota é atualizada com as alterações commitadas localmente.
 - **Branch principal (main):** A branch principal no Dataform é a versão da main salva no repositório remoto. Para alterações nas branches de desenvolvimento entrarem para produção, elas precisam ser aprovadas por meio de PR's. Uma boa prática é sempre dar um pull da main antes de começar a alterar a sua branch de desenvolvimento para evitar conflitos entre versões.



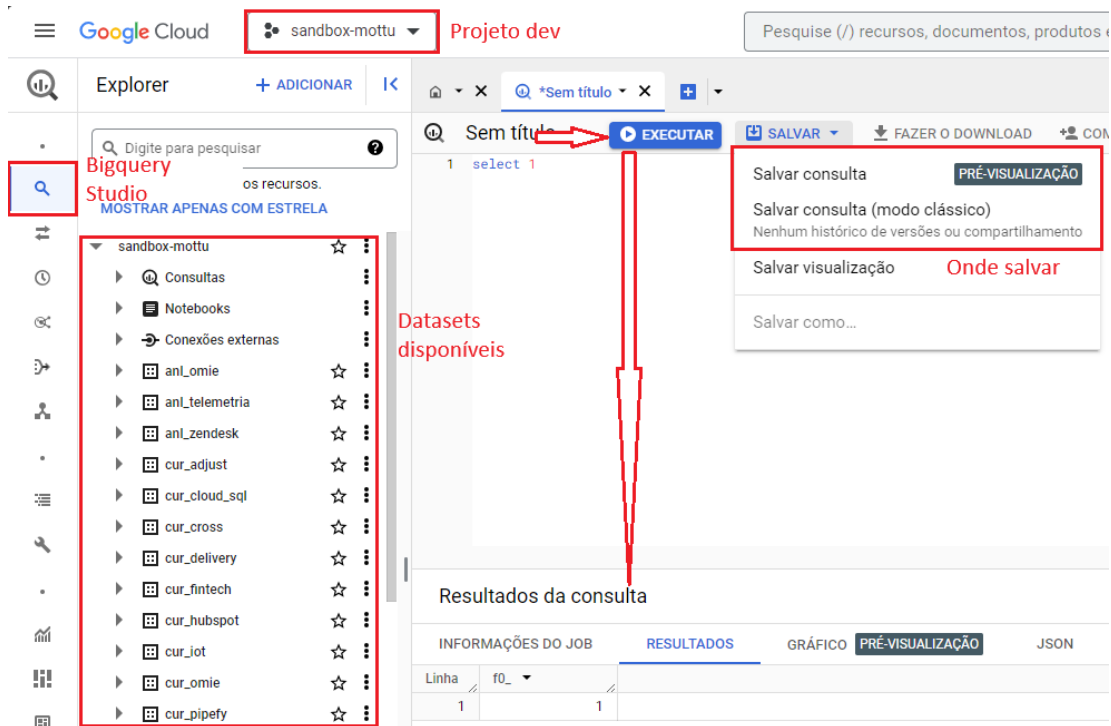
Ambientes

1. Desenvolvimento

O projeto destinado para desenvolvimento é o `sanbox-mottu`. As queries neste ambiente são rodadas dentro do BigQuery Studio, o script sql pode ser salvo apenas como consulta pessoal (SALVAR>SALVAR CONSULTA MODO CLASSICO) e o resultado da consulta não vai poder ser salvo (apenas visualizado nos resultados de consulta).

Os dados presentes nesse projeto são desatualizados e ele é destinado apenas para validar queries e a modelagem dos dados:

- Verificar a sintaxe da sua query
- Compreender o que a query está fazendo em termos de lógica
- Considerar a otimização da query para melhorar o desempenho/custo
- Conferir se a tabela resultante tem a modelagem ideal
- Conferir se os agrupamentos são os mais otimizados



2. Homologação

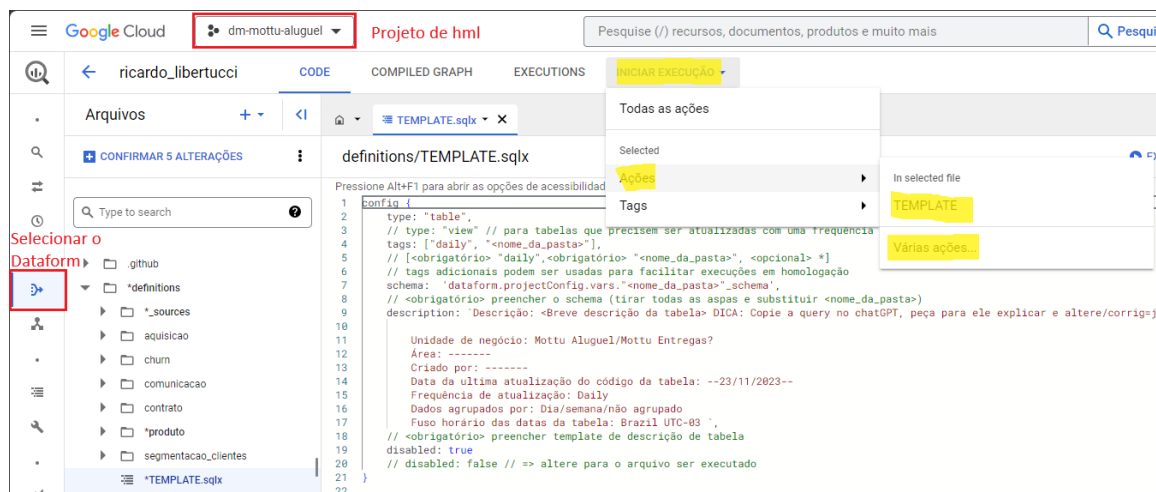
O projeto destinado para homologação é o mesmo que o de produção `dm-mottu-aluguel` ou `dm-mottu-entregas`, a diferença está no dataset de destino do resultado das queries rodadas pela esteira de homologação.

Os datasets em que o resultado das queries são salvos tem os nomes no formato `z_<prefixo de área>_homologacao`.

As consultas neste ambiente deveriam ser feitas com a intenção de validar o resultado da query com as informações do banco em produção e a configuração do arquivo `.sqlx` do dataform.

Para serem feitas as consultas neste ambiente, é necessário: acessar o Dataform → selecionar a sua branch (workspace) → criar/alterar o arquivo `.sqlx`

que gera a tabela/view → iniciar uma execução → selecionar a ação correta → gerar uma execução.



- Ação: é o processo que compila o código SQL e gera a sua tabela/view tem o nome do arquivo .sqlx processado. **É importante selecionar apenas a ação que gera a tabela/view (e dependências) sendo testada para não gerar custos desnecessários que podem ser altos**
- Botão “Executar”: tentar rodar o script por esse botão vai resultar em um erro porque ele roda o script com o seu usuário que não tem as permissões necessárias. Já o botão “Iniciar Execução” roda o script com uma conta de serviço que tem as permissões necessárias e portanto não gera erro
- Dependências: quando a tabela/view sendo gerada depende de uma outra tabela/view criada por dentro do mesmo repositório, ela provavelmente não vai existir dentro do conjunto de dados de homologação e por isso pode ser necessário incluir as dependências

Execute

Execute todas as ações ou selecione um subconjunto de ações. A conta de serviço `service-454570439594@gcp-sa-dataform.iam.gserviceaccount.com` será usada.
[Saiba mais](#)

[ALL ACTIONS](#)
[SELECTION OF ACTIONS](#)
[SELECTION OF TAGS](#)

Selecionar ações a serem executadas
 dm-mottu-aluguel.z_grw_homologacao.contrato

Opções de execução

☒ Incluir dependências ?
 ☐ Incluir dependentes ?
 ☐ Executar com atualização completa ?

1 ação selecionada para execução

Destino	Tipo	Arquivo
dm-mottu-aluguel.z_grw_homologacao.contrato	Table	definitions/contrato/contrato.sqlx

- Acompanhar a execução: para ver se a execução agendada rodou sem erros, basta ir na aba de log de execuções do fluxo de trabalho (workflow execution logs). Podendo clicar no pop-up que surge quando a ação é executada ou voltando uma página e clicando na aba.

Google Cloud | dm-mottu-aluguel

Pesquise (/) recursos, documentos, produtos e muito mais

growth

DEVELOPMENT WORKSPACES | RELEASE CONFIGURATIONS | WORKFLOW CONFIGURATIONS | **WORKFLOW EXECUTION LOGS** | SETTINGS

Nesta página estão todas as execuções anteriores do Dataform no seu repositório [ATUALIZAR](#)

3. Produção

Este ambiente é destinado para tabelas/views já validadas e com a modelagem revisada. Só deveriam estar nele dados confiáveis e modelados de forma a facilitar a consulta e geração de relatórios.

A execução dos scripts que estão em produção, a versão do repositório contida na branch main, só pode ser feita manualmente por meio de uma execução

manual do fluxo de trabalho (workflow) selecionando a configuração de versão (release configuration) 'prod'.

Mas a execução manual não é recomendada e o usuário padrão não vai poder executá-la. A versão de produção tem a execução agendada por meio de um cron job. A execução pode ser por hora, dia, semana, ou um dia específico do mês, sendo definido por meio das tags.

Cron jobs não podem ser criados ou alterados por usuários padrão.