Logotipo, Ícone

Descrição gerada automaticamente

**Camadas ambiente de Dados**

[Fernando Sussmann](mailto:fernnandosussmann@arcoeducacao.com.br) [Marcello Otsuka](mailto:marcellootsuka@arcoeducacao.com.br) | Atualizado em: 2024-04-25

Status: Defined

**Contexto**

Para construir um ambiente de dados abrangente contendo todas as marcas do grupo Arco, é essencial definir uma estrutura organizada. A arquitetura em camadas é fundamental para separar responsabilidades e gerenciar o ciclo de vida dos dados, garantindo regras claras, limites definidos, fluxo de dados controlado e acesso restrito. Isso assegura a governança e escalabilidade do ambiente.

Esta documentação tem o objetivo de estabelecer uma compreensão clara da arquitetura de dados da empresa, além de definir padrões e boas práticas para sua gestão.

**Escopo e não-escopo**

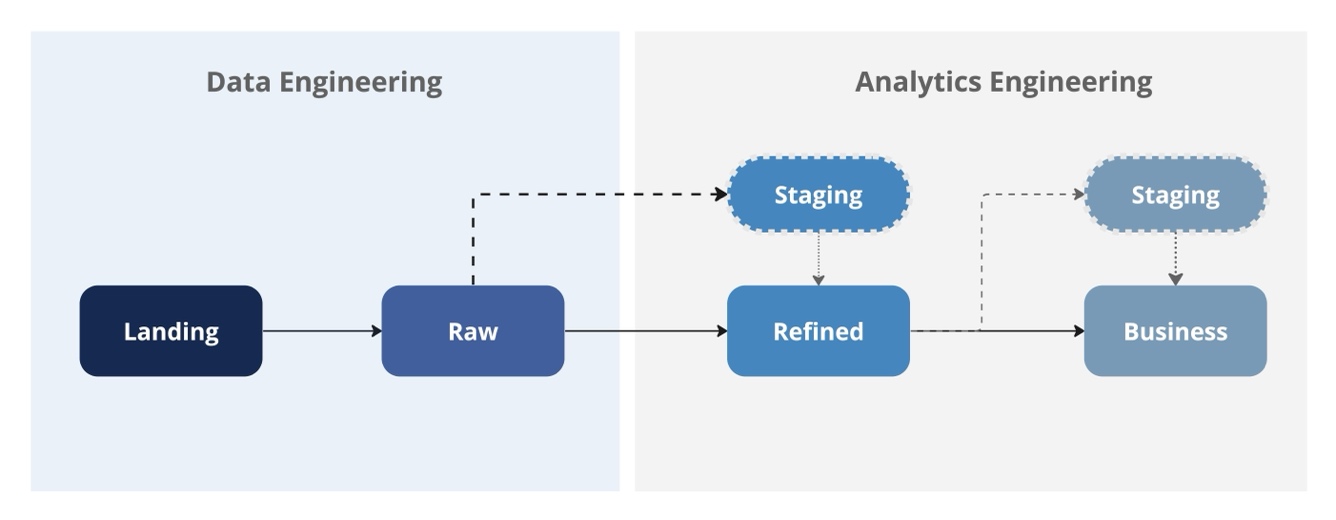
Escopo

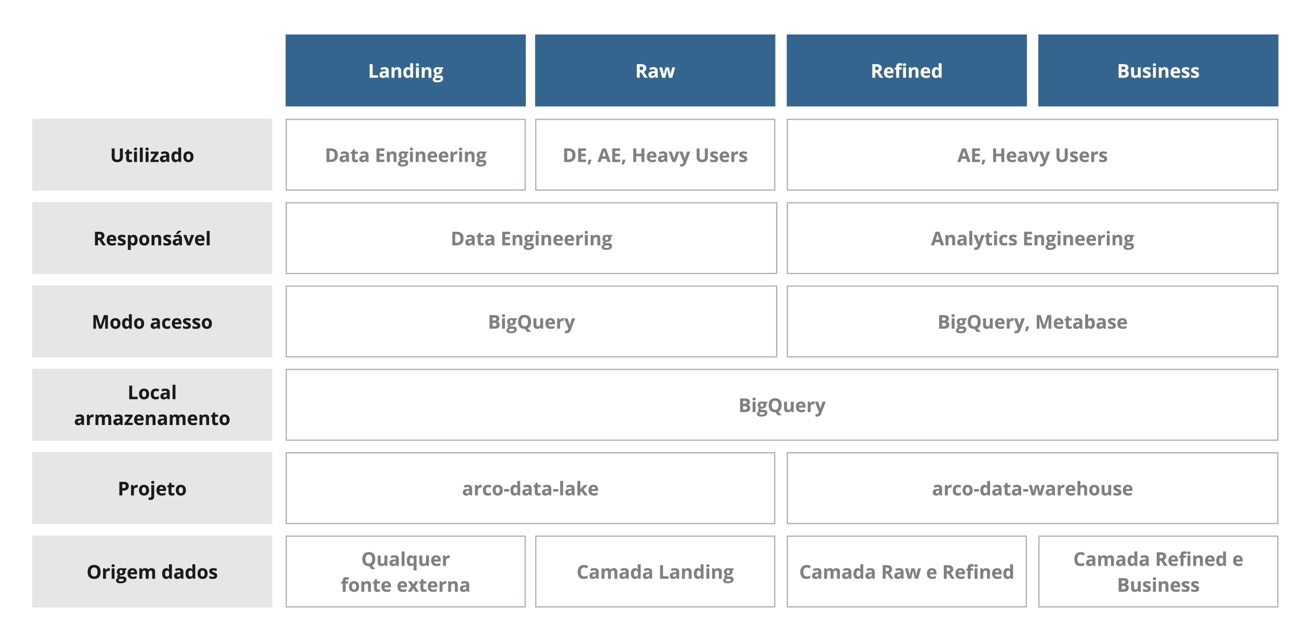
* Definir as camadas de dados da arquitetura
* Especificar as regras de uso de cada camada
* Definir os protocolos de desenvolvimento na nova arquitetura (incluindo documentação esperada)
* Especificar projetos que serão utilizados para cada camada na GCP

Não escopo

* Definir ferramentas a serem utilizadas
* Discutir implementação de casos específicos
* Definir como será a governança de dados do ambiente

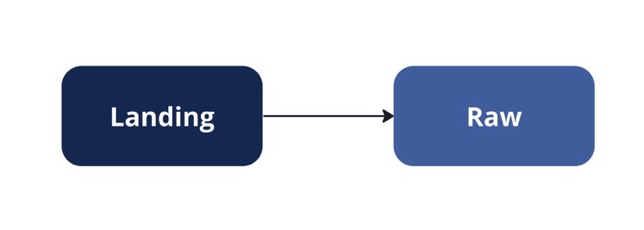
**Camadas**





Para garantir a integridade e eficiência dos dados em nossa plataforma, estabelecemos uma abordagem em camadas. Iniciamos com a camada “**Landing**", onde os dados são recebidos e avaliados para uso posterior. Em seguida, na camada "**Raw**", os dados são mantidos basicamente em seu estado original, com tratativas simples para facilitar o seu consumo. A camada "**Refined**" entra em cena como SSoT (Single Source of Truth), para unificar e padronizar dados de diferentes fontes para uma melhor consistência. Por fim, a camada "**Business**" é dedicada à criação de métricas e análises para suportar decisões estratégicas. A seguir nessa documentação detalharemos cada uma dessas camadas.

**Landing**



A camada Landing é onde o dado extraído de quaisquer fontes (Databases, APIs, entre outros), chegará no ambiente de dados. Nessa camada não haverá transformações de dados, seu acesso deve ser limitado e desencorajado, pois os mesmos dados estarão disponíveis de uma maneira mais palatável na Raw.

Escopo e Governança

Essa camada será gerenciada pelo time de Data Engineering e seus dados devem ser utilizados apenas para o processo de formatação que os habilitará na camada raw e em cenários de debugging que exigem o entendimento de uma série de modificações que os dados sofreram ao longo do tempo. Não haverá nenhum outro tipo de uso nessa camada além do especificado.

Essa camada não terá acesso aos dados de outras camadas, a origem dos dados dela serão provenientes de diversas fontes produtivas de dados como tabelas sistêmicas, APIs, entre outros. Nenhum processo analítico deverá ser desenvolvido a partir dessa camada, o único processo oficial deve ser o tratamento e otimização dos dados para camada Raw.

Estrutura da camada

Essa camada terá todas as versões de todas as linhas da base de dados desde o início da replicação, ou seja, uma vez que a linha for replicada se ela for alterada na origem teremos uma linha adicional no nosso ambiente registrando a nova versão desse dado sem alterar ou remover a versão antiga (Exemplos: mudanças de endereço, status do pedido, alteração no preço de material, etc). Ela não receberá nenhum tratamento, ou seja, deverão ter a informação original com formato mais próximo do da origem de sua extração. Apenas metadados do processo de obtenção do dado e envio para o ambiente analítico podem ser adicionados às informações originais da tabela, como exemplo o datetime em que o dado foi enviado a ferramenta de ingestão.

Organização dos dados

Todos os dados ingeridos na Landing deverão repousar em um Dataset que siga o seguinte padrão " landing\_\_<origin>[\_\_instance]" sendo "origin" o nome do database ou da api ou de qualquer outro tipo de origem e "instance" o nome da marca ou alguma outra subdivisão da origem que faça sentido ser exposta. No caso dos nomes das tabelas que vieram de banco de dados deverão ser "db\_<origin>\_<schema>\_<table>" para os dados originários de outras fontes utilizaremos "<origin>\_<domain>\_<table>". O prefixo "db\_" será usado para identificarmos tabelas que vieram de banco de dados. Já um exemplo de domain em outras origens, pegando o caso de API por exemplo, pode ser o nome do endpoint em que o dado é extraído. Exemplos:

* **Dataset:** landing\_\_protheus\_prod

**Tabela:** sc5030

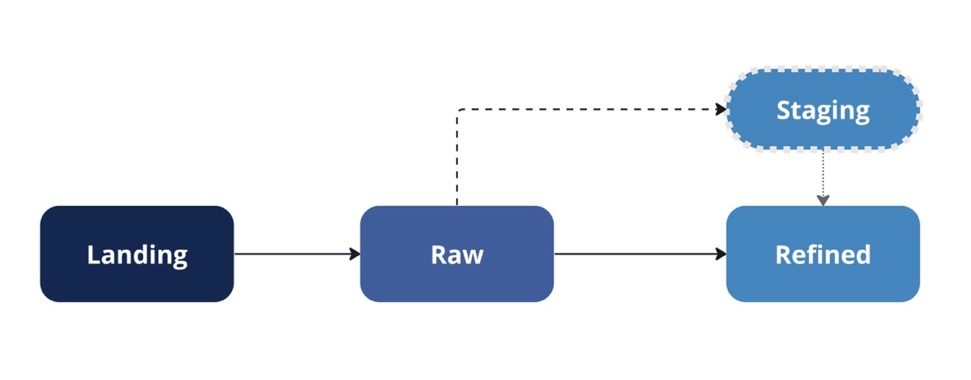
**Nome completo:** arco-data-lake.landing\_\_protheus\_prod.db\_PROTHEUS\_PROD\_dbo\_sc5030

* **Dataset:** landing\_\_ebs

**Tabela:** ap\_supplier\_sites\_all

**Nome completo:** arco-data-lake.landing\_\_ebs.db\_ebs\_apps\_ap\_supplier\_sites\_all

**Raw**



A camada Raw é onde teremos os dados brutos da camada landing, já padronizados e otimizados para a modelagem de dados. O intuito dela é evitarmos um processamento muito pesado direto da landing porque o dado de lá contém metadados irrelevantes para modelagem, pode conter registros duplicados, datas despadronizadas, entre outros. Essas ações facilitarão o consumo para modelagem desses dados.

Escopo e Governança

Essa camada será responsabilidade do time de Data Engineering que cuidará do processo de criação das tabelas, execução dos pipelines de transformação e definições de nome e utilização. Ela utilizará os dados da camada landing otimizando seu consumo e resolvendo alguns problemas que um dado igual a origem possa apresentar. Por essa razão o time de DE deverá levar em consideração inputs do time de Analytics Engineering, principal consumidor dessa camada, para adequar o formato desses dados e realizar ajustes no dado que veio da origem.

Dentre esses ajustes os que temos mapeados e deverão ser resolvidos nessa camada são:

* Remoção de registros duplicados
* Adequação de campos timestamp para formato de datetime
* Retirada de espaços em branco em colunas de texto para algumas fontes
* Remoção de estruturas desnecessárias
* Adição de data de processamento/atualização da tabela

Nessa camada a realização de consulta dos dados é livre, já a criação de outras tabelas, que não estão ligadas ao processo de ingestão, não devem ser persistidas nessa camada. Exemplos do que não deve estar nessa camada são agregações, cruzamento e tratamento de informações das tabelas dessa camada. O ideal é que a escrita desse processo seja feita na camada seguinte a Refined, de acordo com as regras descritas na definição dela nesse mesmo documento.

Estrutura da camada

Nessa camada habilitaremos duas estruturas de tabelas, as que chamaremos de "current" que serão um espelho da tabela do banco, apresentando apenas o estado atual dos dados e as "event" que tem estrutura de logs, ou seja, possuem todas as alterações que um dado sofreu desde que começamos sua replicação.

Organização dos dados

Aqui ainda procuramos fazer referência aos nomes da origem, porém de maneira mais enxuta. Por conta disso o nome do dataset passa a ter o schema como opcional para as origens que trazemos apenas um schema " raw\_\_<origin>[\_\_schema] ". Já o nome das tabelas passa a apresentar também seu tipo de estrutura " <table-type> \_\_<table-name>". Exemplos:

* **Dataset:** raw\_\_protheus\_prod

**Tabela:** current \_\_sc5030

**Nome completo:** arco-data-lake.raw\_\_protheus\_prod. current \_\_sc5030

**Tabela:** event \_\_sc5030

**Nome completo:** arco-data-lake.raw\_\_protheus\_prod. event \_\_sc5030

* **Dataset (com schema):** raw\_\_ebs\_\_apps

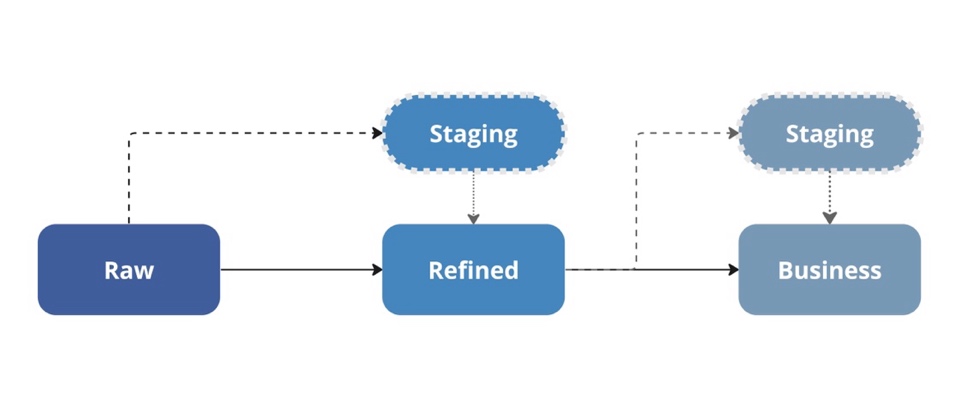
**Tabela:** current \_\_ap\_supplier\_sites\_all

**Nome completo:** arco-data-lake.raw\_\_ebs\_\_apps. current \_\_ap\_supplier\_sites\_all

**Tabela:** event\_\_ap\_supplier\_sites\_all

**Nome completo:** arco-data-lake.raw\_\_ebs\_\_apps.event\_\_ap\_supplier\_sites\_all

**Refined**



A camada Refined tem como objetivo a transformação dos dados provenientes da camada Raw. Nesta fase, os dados são refinados e consolidados para criar uma visão unificada e consistente de processos específicos. Por exemplo, um pedido, que pode ter várias fontes de origem na camada Raw (como Protheus, Oracle, SAP, etc.), será processado nesta camada para criar uma única visão validada e compreensível para ser utilizada nas modelagens da camada “Business”, que será tratada em detalhe mais adiante.

É importante ressaltar que a transformação dos dados nessa camada é realizada de forma ainda mais criteriosa em comparação com a camada Raw. Aqui, é dada uma atenção especial aos tipos de campos, definição de chaves primárias para melhora de performance e garantia de que os nomes dos campos sejam mais intuitivos e significativos, tornando mais acessíveis e compreensíveis para os usuários finais.

Governança

A camada Refined será gerenciada pelo time de Analytics Engineering que será responsável por definir as regras de uso e desenvolvimento. Todas as criações ou edições de tabelas nesta camada serão revisadas e aprovadas por esse time. Embora outras equipes tenham acesso à camada, devem seguir as diretrizes estabelecidas pelo time de AE para garantir a consistência e qualidade dos dados. As regras abordarão boas práticas para padronização de nomenclaturas de tabelas e colunas, tipo de dados, documentação e armazenamento de scripts.

Estrutura da camada

A camada Refined será configurada para ter acesso à camada Raw e nela mesma. A camada tem objetivo de trazer os dados na sua forma mais granular, portanto, agregações devem ser evitadas. Além disso, para criação de tabelas nessa camada, é recomendado a utilização apenas de tabelas RAW, evitando o consumo de tabelas da própria camada Refined em casos mais simples, visando minimizar a criação de dependências entre as tabelas até que tenhamos ferramentas apropriadas para fazer o monitoramento dessas dependências. É importante também avaliar a possibilidade de enriquecimento das tabelas já existentes ao invés de optar pela criação de novas. Porém, entendemos que haverá exceções que a utilização de tabelas da mesma camada será necessária.

Dentro da camada Refined, será criada uma subcamada chamada "**staging**" para auxiliar no processo de construção dos dados. Esta área conterá tabelas intermediárias destinadas exclusivamente a esse propósito, facilitando a compreensão e manutenção do código. A camada “Business” não terá acesso a esses dados, que são utilizados apenas para melhorar a organização do código, e não para construção de relatórios ou painéis. A visualização dessa subcamada também só ficará disponível para os times de AE e DE, evitando a consultas incorretas de outros usuários, podendo pegar informações incompletas acidentalmente.

Também é recomendável aqui evitar a inclusão de regras de negócio específicas de uma única área e contexto, aplicando apenas aquelas mais genéricas que auxiliam na padronização de dados de diferentes fontes. Casos excepcionais devem ser cuidadosamente avaliados por pessoas com conhecimento na área, com documentação adequada e comunicação das regras.

Teremos duas segmentações possíveis de tabela nessa camada, são essas:

* **current**: Contém o estado atual dos dados em um determinado ponto no tempo. Por exemplo, em uma tabela de pedido poderíamos ter apenas a visão do status atual do mesmo, sem considerar seu histórico de status antigos.
* **timeline**: Mantém um histórico das alterações nos dados ao longo do tempo. Ela registra todas as versões anteriores dos dados, permitindo análises retrospectivas e rastreamento de alterações. No mesmo exemplo de uma tabela de pedidos poderíamos ter todas as alterações que ocorreram no status do pedido, tendo todo entendimento do seu ciclo.

Organização dos dados

* A princípio os datasets serão separados por entidades, como exemplo temos a entidade “orders”, contendo tabelas como orders\_type, orders\_lifecycle, orders\_item, etc. Usaremos então o seguinte padrão:
  + **current**: refined\_\_<entidade>.current\_\_<tabela>. Exemplo:
    - **Dataset**: refined\_\_contract

**Tabela**: current\_\_contract\_status

**Projeto:** arco-data-warehouse

**Nome completo: ‘**arco-data-warehouse.refined\_\_contract.contract\_status’

* + **timeline**: refined\_\_<entidade>.timeline\_\_<tabela>. Exemplo:
    - **Dataset**: refined\_\_contract

**Tabela**: timeline\_\_contract\_status

**Projeto:** arco-data-warehouse

**Nome completo: ‘**arco-data-warehouse.refined\_\_contract.timeline\_\_contract\_status’

* Para tabelas da subcamada **staging**, seguiremos com a mesma lógica de entidades, sendo como padrão staging\_\_refined\_\_<entidade>.<tabela> . Exemplo:
  + **Dataset**: staging\_\_refined\_\_orders

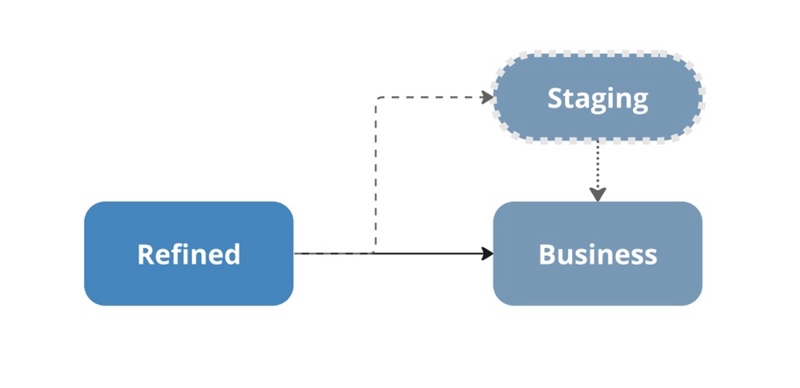
**Tabela**: orders\_status\_step1

**Projeto:** arco-data-warehouse

**Nome completo: ‘**arco-data-warehouse.staging\_\_refined\_\_orders. orders\_status\_step1

*Obs.: o nome do projeto não é necessário caso já esteja nele*

**Business**



A camada Business tem como objetivo principal a análise e utilização dos dados refinados provenientes da camada Refined. Nesta fase, os dados são transformados em informações estratégicas para criação de KPIs e OKRs com intuito de apoiar a tomada de decisão e atender demandas operacionais da empresa atuando na melhora de performance nos processos. Embora a principal fonte de dados seja a camada Refined, a camada Business pode ocasionalmente utilizar tabelas próprias para criar indicadores ou relatórios. No entanto, essa prática não é recomendada para evitar dependências. Aqui também é importante garantir que os nomes dos campos sejam mais intuitivos e significativos possíveis, com intuito de deixarmos o mais simples possível para os usuários finais.

Governança

A governança na camada Business, assim como na camada Refined, será supervisionada pelo time de AE, responsável por definir as diretrizes de uso e desenvolvimento. Destaque-se que essa governança será mais orientada para as necessidades e objetivos específicos do negócio, em vez de apenas para a consistência e qualidade dos dados, como por exemplo verificar se realmente a tabela deveria ser criada ou se a informação poderia ser adicionada em alguma outra já existente. A governança abrange a revisão e aprovação de todas as criações ou edições de tabelas nesta camada. Embora outras equipes tenham acesso de leitura e escrita, é crucial que sigam as orientações estabelecidas pelo time de AE.

Estrutura da camada

Diferente da camada Refined, na camada Business as agregações e regras de negócio devem ser aplicadas para deixar as informações o mais mastigado possível para usuário final e facilitar na criação de dashboards e relatórios. Também aqui é recomendado para a criação de tabelas evitar o uso de tabelas já existentes dentro da própria camada, visando minimizar a criação de dependências entre as tabelas e até mesmo criação de novas visões desnecessárias. É importante avaliar a possibilidade de enriquecimento das tabelas já existes ao invés de optar pela criação de novas.

Dentro dessa camada, também teremos uma subcamada chamada "**staging**" para auxiliar no processo de construção dos dados, seguindo os mesmos padrões da subcamada presente na Refined. Importante ressaltar aqui que as ferramentas de consulta e visualização de dados não terão acesso a essa subcamada. A visualização dessa camada também só ficará disponível para os times de AE e DE, evitando a consultas incorretas de outros usuários, podendo pegar informações incompletas acidentalmente.

Aqui teremos também duas segmentações opcionais possíveis de tabela nessa camada, são essas:

* **current**: Contém apenas as informações mais recentes e atualizadas. Ela reflete o estado atual dos dados em um determinado ponto no tempo. Em uma tabela de contas a receber poderíamos ter apenas a visão do mês atual, sem considerar históricos anteriores.
* **timeline**: mantém um histórico das alterações nos dados ao longo do tempo. Ela registra todas as versões anteriores dos dados, permitindo análises retrospectivas e rastreamento de alterações. No exemplo de uma tabela de contas a receber poderíamos ter a foto do fechamento de cada mês.

Organização dos dados

* A princípio os datasets serão separados domínio, ou seja, se a tabela será usada pelo time de Contas a receber, ela faz parte do domínio Finance, teriamos um dataset com nome ”finance”. Aqui o nome do domínio, não é o nome da área (consultar documentação de [Domínios e Entidades](https://arcoeducacao.sharepoint.com/:w:/r/sites/CV-DataEngineering/_layouts/15/Doc.aspx?sourcedoc=%7BF3D70AB9-FE3C-4B3C-AC35-0B91B17DFBD5%7D&file=Mapeamento%20Dom%C3%ADnios%20e%20Entidades%20Arco.docx&action=default&mobileredirect=true)).
* Tendo então como padrão usaremos:
  + **current**: business\_\_<dominio>.current\_\_<tabela>
    - **Dataset**: business\_\_accounts\_recivable

**Tabela**: current\_\_accounts\_recivable

**Projeto:** arco-data-warehouse

**Nome completo:**

**‘**arco-data-warehouse. business\_\_accounts\_recivable.current\_\_accounts\_ receivable’

* + **timeline:** business\_\_<dominio>.timeline\_\_<tabela>
    - **Dataset**: business\_\_accounts\_recivable

**Tabela**: timeline\_\_accounts\_recivable

**Projeto:** arco-data-warehouse

**Nome completo:**

**‘**arco-data-warehouse. business\_\_accounts\_recivable.timeline\_\_accounts\_ receivable’

* Para tabelas da subcamada **staging**, seguiremos com a mesma lógica do time solicitante, sendo como padrão staging\_\_business\_\_< timesolicitante>.<tabela>. Exemplo:
  + **Dataset**: staging\_\_business\_\_accounts\_recivable

**Tabela**: general\_ledger\_sas

**Projeto:** arco-data-warehouse

**Nome completo:**

**‘**arco-data-warehouse. staging\_\_business\_\_accounts\_recivable. general\_ledger\_step1’

*Obs.: o nome do projeto não é necessário caso já esteja nele*

**Padronizações Refined e Business**

Como padrão todas as tabelas e respectivas colunas nessa camada deverão estar em **inglês,** sempre em lowercase. O nome da tabela deverá corresponder ao assunto que está sendo tratado, como por exemplo orders, school, contract. Abreviações em nome de tabelas e colunas deverão ser evitadas, a menos que sejam de conhecimento geral. A separação deverá ser feita com underscore (“\_”).

Será feito um documento apenas para tratar desse assunto mais afundo, contendo a padronização de colunas.

**Documentação**

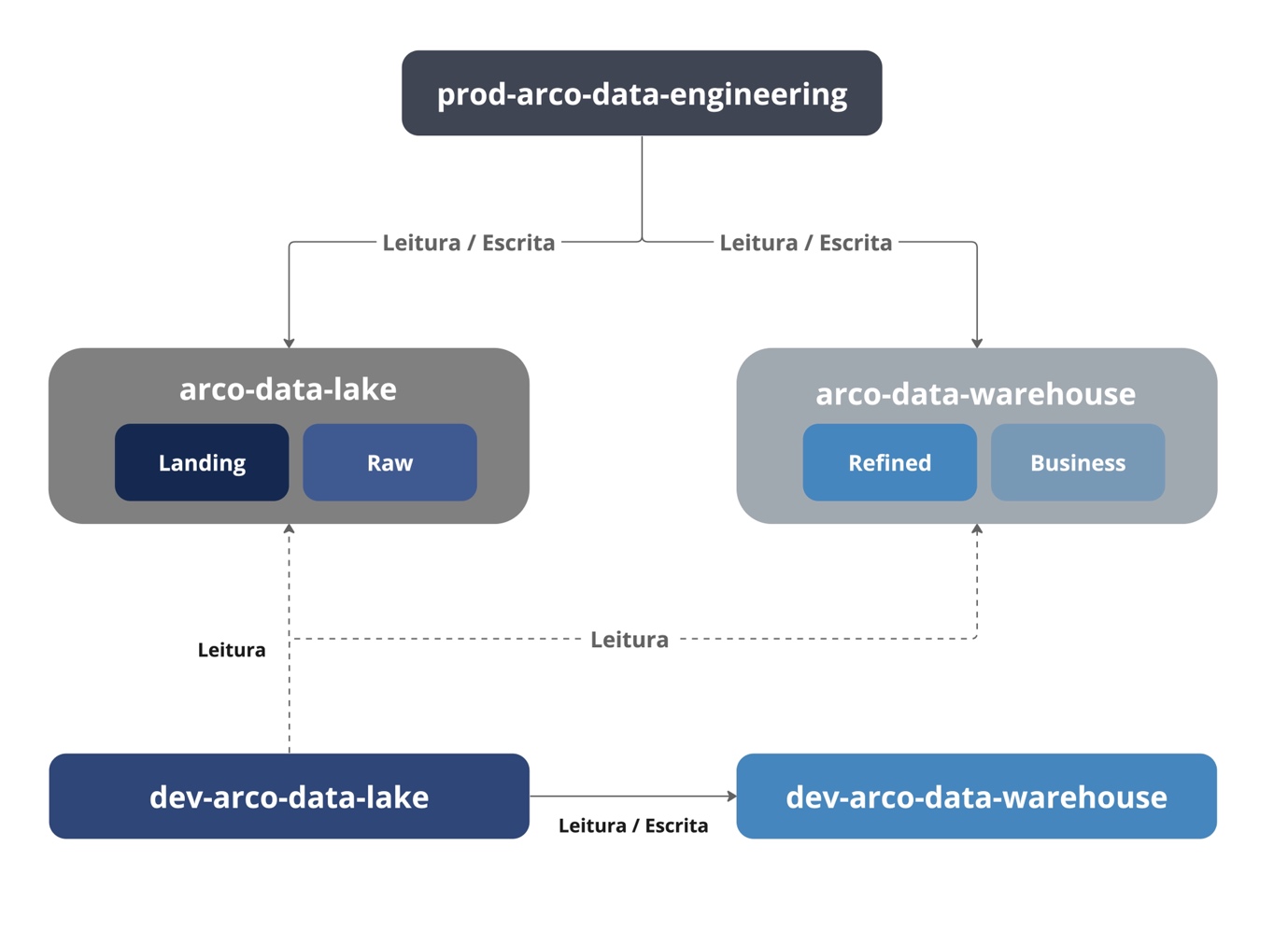
Neste primeiro momento, para as camadas Refined e Business, trabalharemos com um dicionário de dados centralizado, contendo as informações de todas as tabelas em um arquivo excel. Porém, estamos analisando ferramentas para implantarmos no futuro e deixar esse processo de documentação mais automatizado

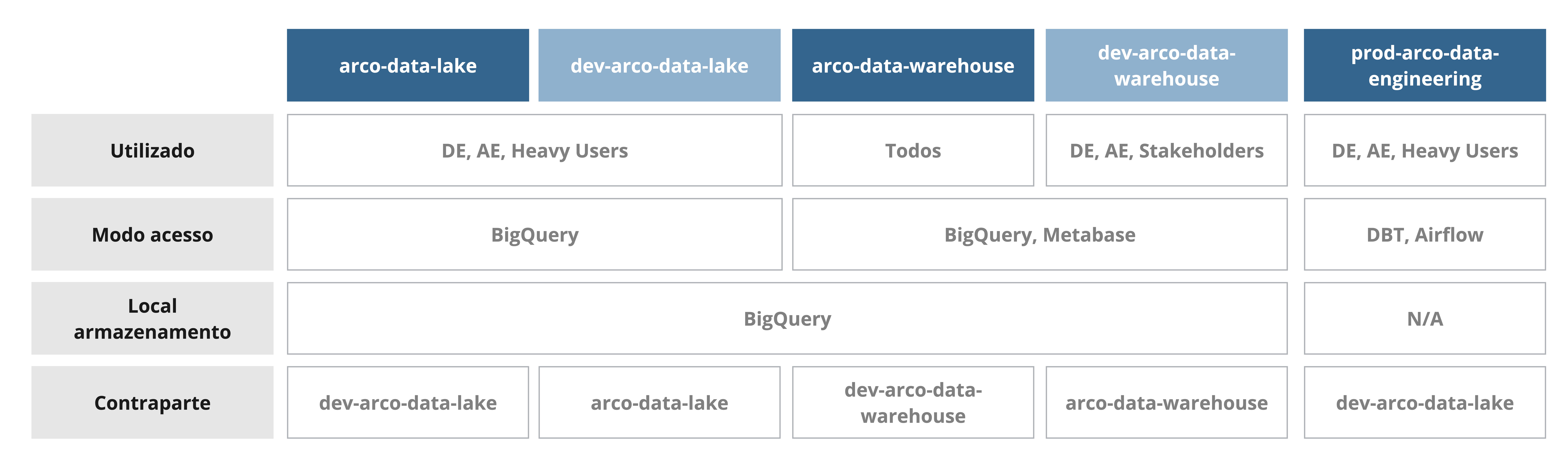
O dicionário pode ser encontrado [aqui](https://arcoeducacao.sharepoint.com/:x:/r/sites/CV-DataEngineering/_layouts/15/Doc.aspx?sourcedoc=%7B54A3E265-0BD1-422B-A184-2A12E79DCEEA%7D&file=Dicion%C3%A1rio%20de%20Dados.xlsx&action=default&mobileredirect=true).

**Monitoramento**

Entendemos que o monitoramento é de suma importância para garantirmos a confiabilidade dos dados que estamos trabalhando. Porém, nesse primeiro momento, não teremos nenhum processo/ferramenta de monitoramento, mas discutiremos essas definições em um segundo momento.

**Projetos GCP**





**Projetos**

Para ter essas camadas vamos precisar ter projetos na GCP e estamos elaborando essa proposta de estrutura de projetos

**arco-data-lake**

Projeto onde teremos 2 camadas, uma landing e a raw. Tanto os processos de otimização dos dados da landing para a raw quanto às modelagens geradas a partir da raw e camadas subsequentes serão processados no projeto prod-arco-data-engineering. Essa divisão de projeto ocorre por questões de organização e limites da GCP.

**dev-arco-data-lake**

Esse projeto terá dados de desenvolvimento para a construção de processos que utilizarão como fonte os dados as camadas presentes no projeto arco-data-lake. Por meio dele será possível elaborar e validar novas modelagens. Além disso esse projeto também será a versão de desenvolvimento do projeto prod-arco-data-engineering, contendo as mesmas ferramentas, possibilitando sua utilização para desenvolvimento de novos processos. Não criamos um dev-arco-data-engineering porque julgamos que a quantidade de execuções em desenvolvimento será inferior à de produção, não atingindo os limites de recursos da GCP. Esse projeto poderá tanto escrever nele mesmo quanto no arco-dev-data-warehouse que explicarei a seguir.

**arco-data-warehouse**

Projeto em que haverá 2 camadas, a refined e a business. Fora as consultas executadas pelo BigQuery ou DataViz, todos os processamentos dos dados desse projeto também serão realizados no projeto prod-arco-data-engineering.

**dev-arco-warehouse**

Nesse projeto será habilitado somente a leitura dos dados, para a validação de processos que utilizarão como fonte os dados das camadas presentes no projeto arco-data-warehouse. As novas modelagens geradas partirão do projeto dev-arco-data-lake que conseguirá fazer a leitura e escrita em ambos os projetos.

**prod-arco-data-engineering**

Projeto exclusivo para processamento de dados, logo ele terá acesso de leitura e escrita nos projetos arco-data-lake e arco-data-warehouse. Manteremos esse projeto por já possuir nossas conexões de rede com as origens dos dados. Sendo que esse setup possui um grau de complexidade e dependência com outros times. Esse projeto idealmente não será acessado diretamente pelas equipes da empresa, com exceção da equipe de Plataforma que gerenciará o projeto. Nele as definições dos processamentos de dados serão criadas por padrão no DBT, caso haja alguma exceção seguiremos com DAGs do Airflow. O Airflow também deverá ser utilizado para execução de processos administrativos como criação de recursos efêmeros ou chamadas de execuções de ferramentas da GCP que não tem relação direta aos dados.