

---

# Aprenda Google Analytics

*Gerando Impacto  
Comercial e Insights*

*Mark Edmondson*

AMOSTRA



ALTA BOOKS  
GRUPO EDITORIAL  
Rio de Janeiro, 2024

---

# Sumário

Prefácio.....	vii
1. O Novo Google Analytics 4 .....	1
2. Arquitetura e Estratégia de Dados .....	23
3. Ingestão de Dados.....	42
4. Armazenamento de Dados.....	93
5. Modelagem de Dados.....	149
6. Ativação de Dados.....	177
7. Caso de Uso: Compras Preditivas.....	214
8. Caso de Uso: Segmentação de Público .....	223
9. Caso de Uso: Previsão em Tempo Real .....	244
10. Próximos Passos .....	268
Índice.....	275

# O Novo Google Analytics 4

Este capítulo faz uma introdução ao novo Google Analytics 4 (GA4) e aborda por que ele foi criado. Veremos em que pontos o Google achou que seu predecessor, o Universal Analytics, estava deixando a desejar, e como o GA4 deve compensar nessas áreas com a base de um novo modelo de dados.

Também veremos como a integração da Google Cloud Platform (GCP) com o GA4 aumenta sua funcionalidade e os casos de uso que ajudarão a ilustrar os novos recursos do GA4, ajudando-o a dar início aos seus próprios projetos de dados.

## Apresentando o GA4

O Google Analytics 4 foi lançado a partir do seu beta e apresentado como o novo Google Analytics no início de 2021. Seu nome beta “App+Web” foi trocado por Google Analytics 4.

As principais diferenças entre o GA4 e o Universal Analytics destacadas no post de anúncio do GA4 foram suas habilidades de aprendizado de máquina, seu esquema de dados unificado na web e nos dispositivos móveis, e seu design voltado para a privacidade.

O Google vinha planejando o lançamento do GA4 há muitos anos antes do seu anúncio público. Depois do lançamento, o Google Analytics se tornou o sistema mais popular de análise da web. Contudo, em 2021, seu design ainda refletia os objetivos de design dos últimos 15 anos. Embora a plataforma tivesse sido aprimorada nos últimos anos pela dedicada equipe do Google Analytics, restavam alguns desafios modernos que eram mais difíceis de resolver: os usuários estavam pedindo visualizações de clientes individuais para apps da web e de dispositivos móveis em vez de precisarem enviar dados para duas propriedades separadas; o Google Cloud era o líder em tecnologias de aprendizado de máquina, mas o aprendizado de máquina ainda não era fácil de integrar com o modelo de dados do GA; e a privacidade do usuário era uma preocupação crescente que exigia controlar mais de perto para onde os dados fluíam.

Quando foi lançado em 2005, o Google Analytics abalou o setor de análises oferecendo uma versão completa e gratuita do que antes era disponibilizado apenas por produtos corporativos pagos. Reconhecendo que quanto mais desenvolvedores web estivessem cientes de seu tráfego, maior seria a probabilidade de investirem no AdWords (o atual Google Ads), o Google Analytics se tornou um investimento que beneficia a todos, dando acesso às opiniões dos usuários conforme navegavam seus sites.

Em 2020, o cenário das análises era bem diferente. Produtos de análise concorrentes eram lançados com modelos de dados mais simples que podiam trabalhar com diversas fontes de dados e eram mais adequados ao aprendizado de máquina e à privacidade (um recurso de usuário essencial). Podíamos usar a nuvem para tornar o sistema de análise mais

aberto, dando mais controle aos profissionais analíticos. Soluções de análise concorrentes podiam até ser executadas na própria infraestrutura de nuvem do Google, o que alterou a economia de criar ou comprar. As soluções de análise ideais teriam padrões sensíveis para aqueles que estavam procurando uma introdução rápida, mas poderiam ser personalizadas e ampliadas para atender as necessidades de clientes mais aventureiros.

## A Unificação da Análise Mobile e Web

Embora seu nome anterior (“App+Web”) tenha sido substituído por GA4 no lançamento, o nome descartado indicava melhor em que sentido o GA4 era diferente.

Até ser aposentado no final de 2019, o Google Analytics para apps de dispositivos móveis (Android/iOS) possuía seu próprio sistema de análise, à parte do sistema de análise web. Esses kits de desenvolvimento de software (SDKs) usavam um modelo de dados diferente, mais adequado para a análise de aplicativos, no qual conceitos como visualizações de página, sessões e usuários possuíam significados diferentes, o que quer dizer que não poderiam ser facilmente comparados com os dados da internet. Os usuários que visitavam o app e uma página na internet não costumavam ser vinculados.

O modelo de dados do GA4 segue uma estrutura personalizável e de eventos que foi adotada pelos apps de dispositivos móveis. O Universal Analytics impôs limitações para quando os dados podiam ser combinados, o que se tornou conhecido como escopo de dados, o que significa que os publicitários precisavam pensar em como seus dados se encaixavam nos escopos, como usuário, sessão ou eventos. Esses eram predeterminados pelo Google, de modo que éramos obrigados a adotar seu modelo de dados. Com a abordagem de apenas eventos do GA4, tínhamos mais flexibilidade de determinar como queríamos organizar nossos dados.

Quando os velhos SDKs para dispositivos móveis do Google Analytics foram aposentados em 2019, o Google incentivou seus usuários a usar os SDKs do Firebase, que foi desenvolvido como uma experiência de desenvolvedor mobile completa para iOS e Android com um SDK mobile para criar apps para dispositivos móveis do zero, agora incluindo a análise da web. Ademais, o novo GA4 representava um fluxo de dados adicional: o novo fluxo da internet. Fazer com que os fluxos do iOS, do Android e da internet usassem o mesmo sistema significava que, a partir de então, teríamos uma maneira realmente vinculada de medir a análise digital em todas essas fontes.

## Firebase e BigQuery — Os Primeiros Passos em Direção à Nuvem

Para muitos publicitários, o GA4 foi sua primeira introdução aos novos produtos da nuvem que são uma parte intrínseca da operação do GA4: o Firebase e o BigQuery.

O Firebase e o BigQuery são produtos da GCP, um extensivo serviço que o Google oferece para todos os serviços de nuvem. Este livro se concentra nos produtos que fazem parte dos serviços de análise de dados em nuvem, mas lembre-se de que são apenas um subconjunto da inteira plataforma em nuvem.

O Firebase é uma ampla estrutura de desenvolvimento mobile que agora inclui o Google Analytics. Os desenvolvedores mobile também o usam para criar apps de dispositivos móveis que não precisam de servidores e possuem recursos úteis, como configuração remota para alterar o código de aplicativos implantados sem precisar relançá-los na loja, APIs de apren-

dizado de máquina, como a modelagem preditiva, a autenticação, o alerta de dispositivos móveis e as integrações de publicidade do Google. O Firebase é um subconjunto dos serviços da GCP que, em alguns casos, são uma reformulação de produtos subjacentes da GCP — por exemplo, as Cloud Functions do Firebase são iguais às Cloud Functions da GCP.

O BigQuery pode ser considerado uma das joias da GCP; foi reconhecido como um dos produtos mais atraentes em comparação com os equivalentes usados em outros provedores de nuvem. O BigQuery é um banco de dados em SQL feito sob medida para serviços analíticos e um dos primeiros bancos de dados sem servidor a ser disponibilizado. Ele inclui inovações como um modelo de precificação que armazena dados a um custo baixo, cobrando as pesquisas sob demanda e um motor de busca veloz que roda com Dremel e, em alguns casos, oferece uma velocidade cem vezes maior em comparação com o MySQL. Os usuários do GA360 talvez já estejam familiarizados com ele, visto que uma de suas funcionalidades era exportar dados brutos e sem amostragem para o BigQuery — mas apenas para aqueles que compraram uma licença do GA360 (foi aí que comecei a trabalhar com a nuvem!). Os dados exportados pelo BigQuery do GA4 estarão disponíveis para todos, o que é emocionante, pois o BigQuery em si é um gateway do restante da GCP. Os recursos do BigQuery aparecem bastante neste livro.

## Implantação do GA4

Este livro não é um guia completo sobre a implementação do GA4; para isso, seria melhor consultar os recursos listados no Capítulo 10. Entretanto, este livro aborda as configurações comuns que lhe darão a visão geral, desde a coleta de dados ao valor de negócio.

Basicamente, existem três formas de configurar os dados coletados de sites: `gtag.js`, `analytics.js` ou *Google Tag Manager* (GTM). Em quase todos os casos, recomendo implementá-los por meio do GTM, abordado em mais detalhes no Capítulo 3. Os motivos para isso são a flexibilidade e a capacidade de desacoplar o trabalho da `dataLayer` da configuração da análise, o que minimizará o esforço de desenvolvimento necessário no site HTML. Os recursos do desenvolvedor serão mais eficientes ao implementar uma `dataLayer` mais limpa para o GTM, visto que isso cuidará de todas as nossas necessidades de rastreamento, não apenas das tags do GA4 ou do Google. Quaisquer mudanças adicionais na configuração de rastreamento poderão ser realizadas na interface web do GTM, sem a necessidade de usar nosso precioso tempo novamente para cada edição menor.

Com a introdução do Server Side (SS) do GTM, as possíveis configurações também podem incluir integrações diretas com o Google Cloud e sistemas de back-end com modificações de solicitações e respostas da chamada HTTP, dando-nos uma extrema flexibilidade.

## Universal Analytics Versus GA4

Dizem que o GA4 é a evolução de seu predecessor, o Universal Analytics (apelidado de GA3 desde o lançamento do GA4). Mas exatamente em que sentido ele é diferente?

Uma das primeiras perguntas que as pessoas fazem ao ouvir falar sobre o GA4 é: “Quão diferente ele é para a mudança ser atrativa? Por que eu deveria trocar de ferramenta, treinar e reaprender a usar um sistema que vem funcionando bem nos últimos 15 anos?” Esta é a pergunta-chave que será respondida nesta seção.

Um tópico de ajuda do Google também aborda essa questão.

## Um novo modelo de dados

A primeira grande mudança é no modelo de dados em si, visto posteriormente neste capítulo em “Modelo de Dados do GA4”.

O Universal Analytics se concentrava bastante nas métricas dos sites, nas quais conceitos como usuários, sessões e visualizações de página eram mais fáceis de definir; entretanto, esses conceitos eram mais difíceis de definir no caso de outras fontes de dados, como nos apps de dispositivos móveis e nos acessos a servidores. Em geral, isso quer dizer que soluções alternativas precisavam ser incorporadas ou que algumas métricas precisavam ser ignoradas nos relatórios quando os dados vinham de certas fontes. Também queria dizer que algumas métricas não funcionavam muito bem com outras ou eram impossíveis de consultar.

O GA4 se afastou de um esquema imposto de dados para algo muito mais livre: agora tudo é um evento. Essa flexibilidade nos permite definir nossas próprias métricas com mais facilidade, mas para os usuários que não querem lidar com esse nível de detalhes, ele também fornece tipos de eventos automáticos e padrão que oferecem algumas métricas mais familiares.

Isso também quer dizer que agora é possível coletar de forma automática alguns dados que, antes, precisavam ser configurados separadamente, como os cliques em links, de modo que as implementações do GA4 devem exigir menos experiência para serem implementadas corretamente, ajudando a diminuir a barreira de entrada para novos usuários de análise digital. O conhecimento especializado, como a diferença entre uma métrica de sessão e uma métrica de hit, seria menos crítico.

## Uma abordagem mais flexível das métricas

Os eventos do GA4 podem ser modificados depois de enviados. Isso nos permite corrigir erros de rastreamento ou padronizar eventos (“venda” versus “transação”) sem a necessidade de modificar os scripts de rastreamento — muito mais fáceis de executar.

Ao criarmos definições personalizadas para nossos eventos, não precisamos nos lembrar de nenhum esquema predefinido. Criamos nosso evento com parâmetros otimizados e os registramos na interface do GA4 para começar a ver o evento surgindo nos nossos relatórios.

## Exportações do BigQuery

Agora, as exportações do BigQuery, um antigo recurso do GA360, estão disponíveis inclusive para aqueles que não pagam pela versão corporativa do GA4. O Firebase Analytics para dispositivos móveis possuía esse recurso no lançamento, e como o GA4 não passa de um acréscimo, a análise web também o possui.

Isso muda tudo, porque a parte mais difícil de um projeto de dados costuma ser obter acesso aos dados brutos por trás de nossos aplicativos de uma forma que ainda possamos trabalhar com eles com facilidade. Com as exportações do BigQuery, precisamos apenas preencher alguns formulários na internet para fazer esses dados fluírem praticamente em tempo real, prontos para a análise, usando o SQL do BigQuery.

Visto que o BigQuery está intimamente integrado ao restante da GCP, isso também quer dizer que ele possui integrações estreitas com o restante da pilha de dados da GCP, como Pub/Sub, Dataflow e Data Studio. Esses serviços permitem coletar dados diretamente do

BigQuery, e como suas APIs são abertas, elas também são uma fonte ou um depósito para muitos serviços de terceiros.

Tudo isso significa que o antigo problema de silos de dados, nos quais os dados necessários estavam trancados em bancos de dados com diferentes políticas corporativas, agora possui uma rota de solução enviando tudo para um único destino: o BigQuery. É assim que podemos começar a relacionar as vendas e o marketing ou consultar os dados de outra fonte, como a previsão do tempo, com mais facilidade. Segundo minha experiência, transferir todos os dados úteis para um único lugar teve um efeito muito transformador na maturidade digital do cliente, visto que um dos obstáculos mais comuns — “Como obter os dados?” — foi removido.

## **Nada de amostragem — tudo em tempo real**

Um dos motivos para usar as exportações do BigQuery do GA360 foi que essa era uma das formas de obter dados sem amostragem, o que também se aplica ao GA4. Embora os limites da amostragem sejam reduzidos na WebUI, os dados subjacentes sempre são sem amostragem e disponibilizados em tempo real. Se precisássemos de uma exportação sem amostragem, ela era disponibilizada via BigQuery ou utilizando a API Data gratuita. Isso evitava ter de pagar pelo GA360 para obter dados para alguns casos de uso que exigiam fontes de dados de análise de alta precisão e em tempo real.

## **Privacidade e dados de análise digital**

Os usuários estão positivamente mais cientes do valor de seus dados hoje, e a privacidade se tornou um assunto importante no setor. Todos reconhecem que os usuários devem ser plenamente informados da opção de aprovar a forma como seus dados são usados e o site tem a responsabilidade de ganhar a confiança e valorizar essa informação. Para nos ajudar com isso, disponibilizou-se o Google Consent Mode para remover os cookies e os identificadores pessoais armazenados para que eles não sejam disponibilizados ao Google Analytics até que o usuário dê seu consentimento. Contudo, os dados impessoais ainda podem ser úteis e o GA4 tem uma forma de modelar como seriam nossas sessões e conversões de dados se 100% dos nossos usuários consentem em nos fornecer seus dados. Como, em geral, nossos novos clientes tenderão a não confiar no nosso site e dar seu consentimento de início, essas informações podem ser úteis para aprimorar nosso desempenho.

## **Quando o GA4 é a resposta?**

Dadas as mudanças no GA4, segue um resumo das oportunidades que ele oferece em comparação com o Universal Analytics para ajudá-lo com as dúvidas mais frequentes:

- Como integrar nossos dados de análise digital com a GCP para fazer nossos dados operarem além dos serviços do GA4 (justamente sobre o que este livro mais fala!)?
- Como unificar o rastreamento de usuários em todas as nossas propriedades digitais, incluindo nossos apps de dispositivos móveis e sites?
- Como fazer mais implementações analíticas personalizadas em vez de padronizadas?
- Como acessar nossos dados de análise digital para alimentar nosso modelo de aprendizado de máquina?
- Como respeitar as opções de privacidade e ainda assim obter alguns dados sobre o desempenho do nosso site?

Esta seção explicou por que usar o GA4 e suas principais diferenças em relação ao Universal Analytics. A principal fonte dessas mudanças é a forma como o GA4 registra seus dados no seu novo modelo de dados, que aprofundaremos mais na próxima seção.

## Modelo de Dados do GA4

O modelo de dados do GA4 é o que o diferencia do Universal Analytics. Esse novo modelo permite que o GA4 ofereça recursos mais avançados. Esta seção analisa mais a fundo esse modelo de dados e explica como ele funciona.

Os principais elementos desse modelo de dados incluem:

### *Simplicidade*

Tudo é um evento do mesmo tipo. Nenhuma relação arbitrária é imposta aos dados.

### *Velocidade*

Visto que o modelo de dados é mais simples, o processamento reduzido dos eventos permite que tudo seja feito em tempo real.

### *Flexibilidade*

Os eventos podem ser nomeados até o limite da nossa cota (quinhentos por padrão). Parâmetros podem ser anexados a cada evento para ajustar seus metadados.

Agora vamos nos aprofundar e explorar a sintaxe de como os resultados dos eventos do GA4 são criados.

## Eventos

Os eventos são a unidade atômica da coleta de dados no GA4. Cada ação que um usuário toma no nosso site, segundo nossas configurações, envia um evento aos servidores do Google.

Isto é um evento:

```
{"events": [{"name": "book_start"}]}
```

Simplesmente contar o número de eventos “book\_start” nos fornece informações úteis, como quantas pessoas começaram o livro, a média de leituras do livro por dia etc.

Para garantir que uma coleção de eventos seja associada a um usuário, esses eventos precisam de uma ID em comum. No GA4, isso significa enviar uma `client_id`, que é uma ID pseudônima que costuma ser encontrada nos cookies do GA4. Em geral, ela é gerada na forma de um número aleatório, com uma indicação de data e hora de quando foi criada:

```
{"client_id": "1234567.1632724800", "events": [{"name": "book_start"}]}
```

Essa linha representa o mínimo de dados necessários para os eventos enviados para a nossa conta do GA4.





As indicações de data e hora costumam ser fornecidas no formato do Unix, ou seja, os segundos desde a meia-noite de 1° de janeiro de 1970. Por exemplo, cookies com 1632724800 significam segunda-feira, 27 de setembro de 2021, 08h39m56s — o momento em que estou escrevendo esta frase.

Esses exemplos vêm do Measurement Protocol v2, que é uma forma de enviar eventos. Uma forma muito mais comum é usar os scripts de rastreamento do GA4 no nosso site ou no nosso app de iOS ou Android para compilar e criar esses eventos. Mas acho que é útil saber o que esse script faz.

O mesmo evento enviado de um rastreador web que usa `gtag()` seria assim:

```
gtag('event', 'book_start')
```

A biblioteca JavaScript do GA4 se certifica de que o cookie fornecerá a `client_id`, de modo que só precisamos fornecer nosso nome de evento personalizado.

Ao usar os scripts de rastreamento do GA4, a biblioteca tentará nos ajudar a não configurar tipos de eventos comuns fornecendo eventos coletados automaticamente. Esses incluem eventos úteis, como visualizações de página, visualizações de vídeos, cliques, downloads de arquivos e scrolling. Já é uma vantagem em comparação com o Universal Analytics: o que antes precisaríamos configurar já é padrão no GA4. Menos configuração significa implementações mais rápidas e menos probabilidades de bugs. Para usar esses eventos automáticos, podemos escolher quais ativar com as configurações avançadas de medidas.

Também temos os *eventos recomendados*, que são os eventos que implementamos, mas que seguem uma estrutura de nomenclatura recomendada pelo Google. Eles são mais personalizados ao nosso site e incluem recomendações para atividades paralelas, como viagens, e-commerce ou sites de emprego. Também vale a pena usá-los, pois os relatórios futuros podem se basear nessas convenções de nomenclatura para apresentar novos recursos. Os eventos genéricos recomendados incluem logins de usuários, compras e compartilhamento de conteúdo.

Como esses eventos automáticos e recomendados são padronizados, se coletarmos nossos próprios eventos personalizados, devemos nos certificar de não duplicar seus nomes para evitar conflitos e confusões. Espero que esteja conseguindo ver como o sistema é flexível em sua tentativa de oferecer padronização com padrões sensíveis para não precisar reinventar a roda em cada implementação.

## Parâmetros Personalizados

Não obstante, a contagem de eventos por si só não é suficiente para um sistema de análise útil. Para cada evento, pode haver muitos — ou nenhum — parâmetros que nos dão informações adicionais sobre eles.

Por exemplo, um evento de login nos dará o número de logins no nosso site. Mas talvez queiramos saber como o usuário faz seu login — usando seu e-mail ou pelas redes sociais. Nesse caso, nosso evento `login` também sugerirá um parâmetro `method` para especificar isso:

```
gtag('event', 'login', {
  'method': 'Google'
})
```

Se isso fosse realizado com um protocolo de medida mais fundamental, seria mais ou menos assim:

```
{
  "client_id": "a-client-id",
  "events": [
    { "name": "login",
      "params": {
        "method": "Google"
      }
    }
  ]
}
```

Perceba que incluímos um array de `params` com as informações adicionais.

## Itens de e-commerce

Uma classe especial de parâmetros personalizados são os itens, que são um array mais aninhado nos parâmetros personalizados que contém todas as informações sobre o item. Em geral, o e-commerce representa os fluxos de dados mais complexos porque diversos itens, atividades e dados estão associados às vendas.

Todavia, os princípios são basicamente os mesmos: nesse caso, o parâmetro personalizado é um array que contém alguns campos recomendados, como `item_id`, `price` e `item_brand`:

```
{
  "items": [
    {
      "item_id": "SKU_12345",
      "item_name": "jeggings",
      "coupon": "SUMMER_FUN",
      "discount": 2.22,
      "affiliation": "Google Store",
      "item_brand": "Gucci",
      "item_category": "pants",
      "item_variant": "Black",
      "price": 9.99,
      "currency": "USD"
    }
  ]
}
```

Se combinarmos isso com os eventos de e-commerce recomendados, como `purchase` e outros parâmetros, o evento completo será assim:

```

{
  "client_id": "a-client-id",
  "events": [{
    "name": "purchase",
    "params": {
      "affiliation": "Google Store",
      "coupon": "SUMMER_FUN",
      "currency": "USD",
      "items": [{
        "item_id": "SKU_12345",
        "item_name": "jeggings",
        "coupon": "SUMMER_FUN",
        "discount": 2.22,
        "affiliation": "Google Store",
        "item_brand": "Gucci",
        "item_category": "pants",
        "item_variant": "Black",
        "price": 9.99,
        "currency": "USD",
        "quantity": 1
      }],
      "transaction_id": "T_12345",
      "shipping": 3.33,
      "value": 12.21,
      "tax": 1.11
    }
  ]
}

```

Considerando que esse código representa alguns dos eventos mais complexos enviados ao GA4, espero que você consiga perceber a simplicidade do modelo subjacente. Usando apenas eventos e parâmetros, o GA4 pode ser configurado para coletar interações complexas no nosso site.

## Propriedades do Usuário

Além dos dados no nível dos eventos, também é possível configurar dados no nível do usuário. Esses dados estão associados a `client_id` ou `user_id` que registramos. Isso pode ser usado para configurar o segmento de clientes ou as preferências de idioma.



Nesse ponto, lembre-se de respeitar as escolhas de privacidade do usuário. Se estiver acrescentando informações de um usuário específico, então leis como GDPR (Regulamento Geral de Proteção de Dados) da UE exigem que obtenhamos primeiro o consentimento do usuário para coletar seus dados para os fins indicados.

Enviar as propriedades do usuário é muito parecido com enviar eventos, mas usamos o campo `user_properties` e quaisquer eventos que desejemos enviar:

```

{
  "client_id": "a-client-id",
  "user_properties": {
    "user_type": {
      "value": "bookworm"
    }
  },
  "events": [
    {
      "name": "book_start",
      "params": {
        "title": "Learning Google Analytics"
      }
    }
  ]
}

```

Se usássemos `gtag()`, o código ficaria assim:

```

gtag('set', 'user_properties', {
  'user_type': 'bookworm'
});
gtag('event', 'book_start', {
  'title': 'Learning Google Analytics'
});

```

Nesta seção, vimos várias formas de enviar eventos do GA4, como o protocolo de medida e `gtag`, e a sintaxe de enviar eventos com parâmetros e propriedades do usuário. Agora veremos como processar os eventos que vêm do GA4 por meio de suas integrações com a GCP.

## Google Cloud Platform

Agora, a GCP pode ser inserida no sistema do GA4 por meio dos seus sistemas de análise de dados preexistentes. Ela oferece serviços em tempo real, de aprendizado de máquina e que podem ser estendidos à escala de 1 bilhão, e pelos quais pagamos apenas quando usamos, evitando a parte chata relacionada com manutenção, segurança e atualizações. Dê à sua empresa a possibilidade de se concentrar no seu campo de especialização, permitindo que a nuvem cuide das demais tarefas. Graças à estrutura da nuvem de pagar conforme o uso, as equipes pequenas podem criar serviços que, antes, teriam exigido mais pessoas e recursos de TI.

Nesta seção, analisaremos os serviços da GCP que você provavelmente mais usará ao fazer sua integração com o GA4, as habilidades e os papéis que sua equipe precisará exercer para utilizar essas ferramentas, como dar os passos iniciais, administrar os custos e escolher o serviço de nuvem certo para o seu caso.

## Serviços Relevantes da GCP

Este livro se concentra mais nos serviços de aplicativos de dados da GCP, mas esse ainda é um grande conjunto de serviços que está sendo constantemente atualizado. Para uma aná-

lise mais detalhada que vai além do âmbito deste livro, recomendo o livro *Data Science on the Google Cloud Platform* [sem publicação no Brasil], de Valliappa Lakshmanan.

Os serviços de nuvem essenciais a seguir são empregados em casos de uso mais adiante neste livro e vêm se mostrando fundamentais no meu trabalho em geral. Existem muitos serviços de nuvem, e escolher o certo para você pode ser um tanto intimidador quando se está começando. Sugiro que faça uma pesquisa dos serviços destacados aqui, pois são úteis para quem está começando.

Vamos nos familiarizar com esses serviços ao longo do livro, em uma ordem aproximada de utilidade:

#### *BigQuery*

Como mencionado anteriormente, o BigQuery servirá bastante como destino e fonte para análise e grupos de dados. Ele possui até capacidades de modelagem com o ML BigQuery.

#### *Cloud Functions*

Servindo como a ligação entre os serviços, as Cloud Functions nos permitem executar pequenos fragmentos de código — Python, por exemplo — em um ambiente sem servidor.

#### *Pub/Sub*

Pub/Sub é um sistema de fila de mensagens que garante que cada uma seja entregue pelo menos uma vez em uma escala que pode lidar com toda a internet sendo enviada por sua fila.

#### *Cloud Build*

Cloud Build é uma ferramenta de integração contínua/desenvolvimento contínuo (CI/CD) que nos permite ativar contêineres de Docker em batch em resposta aos pushes do GitHub. É um recurso valioso que contribuiu para várias de minhas soluções.

#### *Cloud Composer/Airflow*

Cloud Composer/Airflow é um orquestrador que nos permite criar fluxos de dados interdependentes, complexos e confiáveis, além de agendamentos.

#### *Dataflow*

Dataflow é uma solução de batch e streaming de dados em tempo real bem integrada com muitos serviços da GCP.

#### *Cloud Run*

Cloud Run é parecido com as Cloud Functions, mas nos permite executar contêineres Docker que incluem qualquer código desejado.

Em geral, existem muitas maneiras de criar o que precisamos e as diferenças podem ser sutis, mas recomendo que seja pragmático e escolha algo que funcione primeiro, então otimize qual serviço é melhor executar depois. Por exemplo, talvez você precise importar dados todos os dias que passam por uma fila agendada do BigQuery, mas então descobre que o Cloud Composer é uma ferramenta melhor para coordenar a importação.

Entretanto, fica o aviso de que nenhuma dessas ferramentas é do tipo aponte e clique. É necessário saber programar para que elas entreguem o que precisamos. Assim, veremos quais habilidades são necessárias para tirar o máximo de proveito na próxima seção.

## Habilidades de Programação

Um dos aspectos mais intimidadores de aplicar essas integrações pode ser que elas exigem habilidades que talvez você imagine que apenas os programadores possam. Talvez se considere “não técnico”.

Eu costumava achar a mesma coisa. Lembro de dizer no início da minha carreira: “Eu não sei usar o JavaScript.” Então ficava esperando seis semanas para que um desenvolvedor ficasse livre para escrever cinco linhas de código em um site. Depois que encontrei tempo e disposição, comecei a fazer alguns experimentos por conta própria, cometendo vários erros ao longo do caminho. Também descobri que os profissionais também cometem vários erros, e a única diferença é que eles têm a motivação de seguir em frente. Outra descoberta foi que grande parte do que eu estava fazendo em Excel era mais complicado e difícil do que se estivesse usando uma ferramenta mais apropriada para o trabalho. Realizar uma tarefa em Excel exigia mais energia cerebral do que fazer a mesma coisa em R, por exemplo.

Assim, se estiver disposto, incentive-o a seguir em frente. Se sentir que as coisas estão difíceis, isso não acontece necessariamente porque você não tem talento — essas coisas são difíceis para todo mundo no início. Programar pode parecer bem complexo em alguns casos, e as coisas podem dar errado se esquecemos de usar um único “;”. Contudo, assim que aprendemos mais uma área, a próxima fica mais fácil. Comecei sendo um bom usuário de Excel. Então, aprendi Python e JavaScript. Por fim, apaixonei-me por R. Então, precisei aprender a gostar de SQL e bash. Hoje, uso um pouco de Go. A natureza da programação é tal que, à medida que aprendemos e nos tornamos melhores, ao olharmos para trás, para um programa que escrevemos há seis meses, ele parecerá horrível. Isso é natural; o importante é conseguir olhar para trás e ver algum progresso. Quando conseguimos fazer alguma coisa que funciona, ganhamos experiência, e ela continuará crescendo nos 10 anos seguintes, quando estiver escrevendo um livro sobre isso.

No meu caso, o código aberto foi algo que aumentou minhas habilidades, visto que divulgar códigos e obter feedback multiplicou qualquer experiência que obteria executando esse código. É por isso que sou grato pelo feedback que recebo hoje, pelo GitHub ou qualquer outra fonte. Os códigos deste livro também estarão disponíveis em um repositório do GitHub [conteúdo em inglês] para acompanhar o livro, os quais tentarei manter atualizados e sem bugs.



Seguindo essa mesma lógica, se você ler algum dos meus códigos e tiver feedback sobre como escrevê-lo melhor, entre em contato comigo! Estou sempre aprendendo.

Os casos de uso neste livro incluem exemplos de código que empregam as seguintes linguagens:

### *JavaScript*

É essencial para todo o rastreamento de páginas da internet que envolvem HTML e costuma ser usado para a coleta de dados via tags. Também é bastante usado no GTM para criar modelos personalizados.

### *Python*

Por ser uma linguagem muito popular suportada por várias plataformas, é útil saber programar em Python, visto que ele pode ser considerado a segunda melhor

linguagem para tudo. Também possui uma forte representação de aprendizado de máquina. Contudo, você provavelmente não precisará disso a menos que esteja trabalhando com implementações avançadas.

R

Embora usar o Python talvez já seja suficiente, a comunidade data science do R faz com que seja a melhor linguagem para data science na minha opinião. Suas bibliotecas e comunidade de código aberto abrangem tudo, desde ingestão de dados até ativação de dados via painéis interativos e relatórios. Atribuo grande parte do meu raciocínio de como abordar os fluxos de dados à mentalidade que obtive com o R, de modo que ele influencia meus projetos mesmo quando não o uso diretamente.

*bash*

Ao interagir com servidores de nuvem, eles provavelmente usarão sistemas baseados em Linux, como Ubuntu ou Debian, que empregam bastante o bash para funcionar, em vez de uma interface gráfica, como o Windows. Também é útil saber algumas linhas de comando da programação em bash ao lidar com arquivos muito grandes que não podem ser facilmente importados em outras linguagens. O gcloud e outras CLIs também exigem algum conhecimento de shell scripts, sendo que o mais popular é o bash.

SQL

Na maioria dos casos, os dados brutos com os quais trabalhamos estarão em um banco de dados e o SQL será o melhor método para extraí-los. O SQL também apresenta uma forma útil de encarar os objetos de dados.

Embora seja possível ter sucesso copiando e colando tudo, realmente recomendo que você leia cada linha e entenda pelo menos o que cada seção de código faz.

Supondo que você tenha algum código à sua disposição agora, graças às suas próprias habilidades ou às da sua equipe, vejamos como começar a trabalhar na GCP e implantar seu primeiro código na nuvem.

## Entrando na GCP

GCP é um grande componente de negócio do Google, possuindo fluxos totalmente isolados do Google Analytics nos quais você aprenderá a navegar.

Você pode começar a usá-la gratuitamente, mas a primeira coisa que precisa saber é que, para qualquer coisa mais sofisticada, precisará fornecer os dados de seu cartão de crédito para o uso da nuvem. Entretanto, poderá usá-lo por vários meses de graça.

A página de inicialização do Google o ajudará a fazer seu primeiro login.



Se já tiver um Projeto no Google Cloud, ainda poderá valer a pena criar um novo a partir dos exemplos deste livro para garantir que sejam ativados com as versões mais atuais das APIs. Por exemplo, você provavelmente precisará ativar as APIs Reporting e Admin do Google Analytics, e do Cloud Build, além de verificar se a API do BigQuery está ativa por padrão.