

# Desafio time de dados tratados

## Contratos inteligentes ETH

### Introdução

Imagine que você precise vender uma casa. Este é um processo complicado e assustador, que envolve muita documentação, comunicação com diferentes empresas e pessoas, além de altos níveis de riscos. É por isso que a maioria absoluta dos vendedores de residências decide procurar um corretor imobiliário, que cuida de toda a papelada, negocia a propriedade e age como um intermediário quando as negociações começam, supervisionando o negócio até que ele seja fechado.

Além disso, a imobiliária fornece um serviço de caução, que é especialmente útil em tais transações, pois as somas envolvidas normalmente são muito grandes e você não pode realmente confiar totalmente na pessoa com quem estará lidando. Entretanto, após o sucesso do acordo, os agentes do vendedor e do comprador compartilharão cerca de sete por cento do preço de venda como comissão. Isso equivale a uma perda financeira substancial para o vendedor.

É numa situação como essa em que os contratos inteligentes podem realmente ser viáveis e efetivamente revolucionar uma indústria inteira, tornando todo o processo muito menos oneroso. Talvez o mais importante, eles resolveriam um problema de confiança. Os contratos inteligentes funcionam com um princípio "se-então", o que significa que a propriedade da casa só será repassada ao comprador quando a quantia de dinheiro acordada for enviada ao sistema.

Eles também funcionam como serviços de custódia, o que significa que tanto o dinheiro quanto o direito de propriedade serão armazenados no sistema e distribuídos para as partes participantes exatamente ao mesmo tempo. Além disso, a transação é testemunhada e verificada por centenas de pessoas, portanto a entrega sem falhas é garantida. Como a confiança entre as partes não é mais um problema, não há necessidade de um intermediário. Todas as funções que um agente imobiliário faz podem ser pré-programadas em um contrato inteligente, ao mesmo tempo em que economizam ao vendedor e ao comprador quantias consideráveis de dinheiro.

**Fonte:**

<https://cointelegraph.com.br/ethereum-for-beginners/what-are-smart-contracts-a-beginners-guide-to-automated-agreements>

# O desafio

Nosso time fictício está interessado em investigar a relevância dos contratos inteligentes Ethereum no mercado, principalmente para conseguir se mover rápido caso ocorra um pico na adoção dos mesmos.

Os dados de contratos inteligentes estão disponíveis em um dataset **público e gratuito** da Google. Este dataset fica dentro do **BigQuery**, banco de dados voltado para analytics, no seguinte endereço:

- **Projeto:** "bigquery-public-data"
- **Dataset:** "crypto\_ethereum"
- **Tabela:** "tokens"

Para acessar estes dados você precisará criar uma conta gratuita na **Google Cloud Platform** (GCP) onde é disponibilizado 1TB de consultas gratuitamente todo mês. Como os dados de contratos inteligentes tem em torno de 30MB isso deveria ser o suficiente.

Gostaríamos de montar algumas tabelas tratadas para acompanharmos o crescimento e relevância dos contratos em relação ao tempo. Precisamos que esse processo seja 100% server side, não podemos ter processos rodando na máquina dos nossos desenvolvedores.

## Perguntas que gostaríamos de responder com nossa(s) tabela(s):

- Quantos contratos (tokens) estão sendo criados por bloco?
- Quantos blocos contendo tokens tivemos em um dia?
- Em relação aos últimos 15 dias, qual foi a variação na quantidade de blocos contendo tokens em um dia?
- Em relação aos últimos 15 dias, qual foi a variação na quantidade de tokens em um dia?

Para a execução diária do código que criará as tabelas, você pode utilizar qualquer solução, desde que seja server side. Caso não consiga encontrar uma solução para executar as consultas diariamente, você pode versionar o SQL que responde a(s) pergunta(s) e explicar na hora de apresentar o desafio.

# Critérios de conclusão do desafio

- Para que possamos considerar o desafio entregue, precisamos que você disponibilize o código em algum serviço de git (ex.: Github, Gitlab, Bitbucket, etc) público ou privado (dando acesso aos revisores). Não serão considerados commits após a data de entrega do desafio;