



escola  
britânica de  
artes criativas  
& tecnologia

# Profissão: Analista de dados



## 4º PROJETO: PIPELINE DE DADOS DO TELEGRAM



## GUIA DA AULA 1



# Entenda o projeto



Acompanhe aqui  
os temas que  
serão tratados  
na videoaula

Chatbot

Telegram

Arquitetura



# 1. Chatbot

Um **chatbot** é um tipo de software que interage com usuários através de conversas automatizadas em plataformas de mensagens. Uma aplicação comum de **chatbots** é o seu uso no atendimento ao cliente, onde, de maneira geral, ajudam clientes a resolver problemas ou esclarecer dúvidas recorrentes antes mesmo que um atendente humano seja acionado.



## 2. Telegram

**Telegram** é uma plataforma de mensagens instantâneas *freeware* (distribuído gratuitamente) e, em sua maioria, open source. É muito popular entre desenvolvedores por ser pioneiro na implantação da funcionalidade de criação de **chatbots**, que, por sua vez, permitem a criação de diversas automações.



### 3. Arquitetura

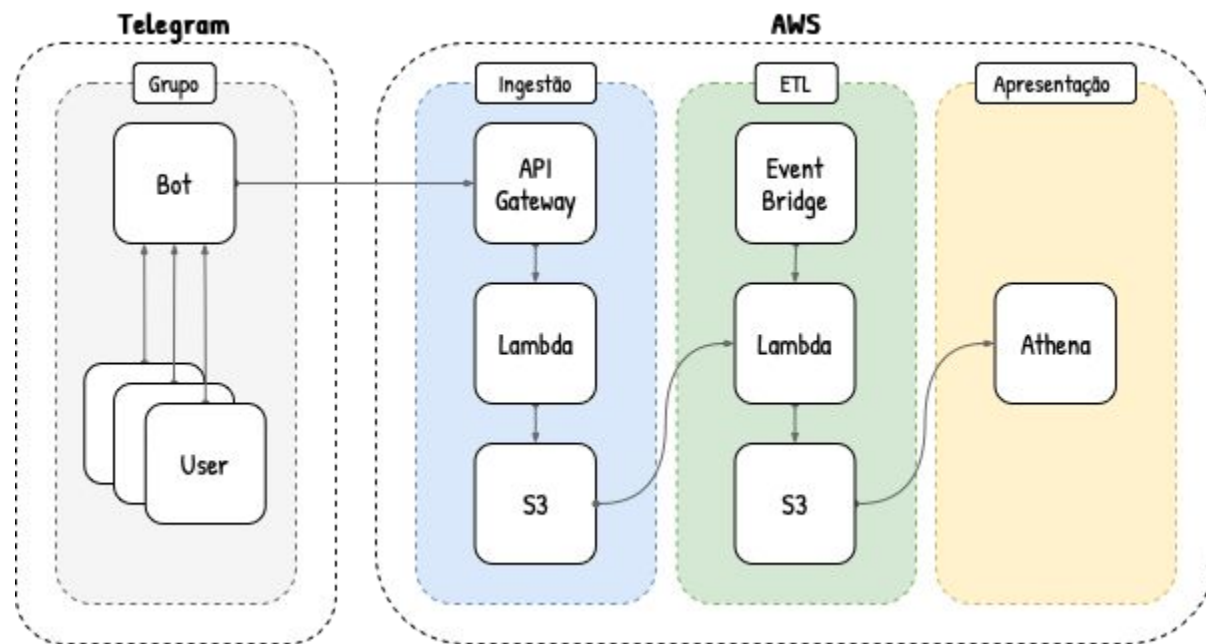
Uma atividade analítica de interesse é a de realizar a análise exploratória de dados enviadas a um **chatbot** para responder perguntas como:

1. Qual o horário que os usuários mais acionam o *bot*?
2. Qual o problema ou dúvida mais frequente?
3. O *bot* está conseguindo resolver os problemas ou esclarecer as dúvidas?
4. Etc.



Portanto, vamos construir um pipeline de dados que ingira, processe, armazene e exponha mensagens de um grupo do **Telegram** para que profissionais de dados possam realizar análises. A arquitetura proposta é dividida em duas: transacional, no **Telegram**, onde os dados são produzidos, e analítica, na Amazon Web Services (AWS), onde os dados são analisados.







- **Telegram**

O Telegram representa a fonte de dados transacionais. Mensagens enviadas por usuários em um grupo são capturadas por um *bot* e redirecionadas via *webhook* do *backend* do aplicativo para um endpoint (endereço web que aceita requisições HTTP) exposto pelo *AWS API Gateway*. As mensagens trafegam no corpo ou *payload* da requisição.



- **AWS | Ingestão**

Uma requisição HTTP com o conteúdo da mensagem em seu *payload* é recebida pelo `AWS API Gateway` que, por sua vez, as redireciona para o `AWS Lambda`, servindo assim como seu gatilho. Já o `AWS Lambda` recebe o *payload* da requisição em seu parâmetro *event*, salva o conteúdo em um arquivo no formato JSON (original, mesmo que o *payload*) e o armazena no `AWS S3` particionado por dia.



- **AWS | ETL**

Uma vez ao dia, o AWS Event Bridge aciona o AWS Lambda que processa todas as mensagens do dia anterior (atraso de um dia ou D-1), denormaliza o dado semi-estruturado típico de arquivos no formato JSON, salva o conteúdo processado em um arquivo no formato Apache Parquet e o armazena no AWS S3 particionado por dia.



- **AWS | Apresentação**

Por fim, uma tabela do AWS Athena é apontada para o *bucket* do AWS S3 que armazena o dado processado: denormalizado, particionado e orientado a coluna. Profissionais de dados podem então executar consultas analíticas (agregações, ordenações, etc.) na tabela utilizando o SQL para a extração de *insights*.

