

Arquitetura Orientada a Eventos

Antes de entender o que é a arquitetura orientada a eventos, é interessante entender o que é um evento:

Um evento é qualquer ocorrência ou mudança de estado significativa no software ou hardware de um sistema. Uma notificação de evento é uma mensagem enviada pelo sistema para avisar outra parte do mesmo sistema que algo ocorreu.

A Origem de um evento pode ser interna ou externa, podem ser cliques em um botão do mouse, o clique em uma tecla do teclado, ou até o carregamento de um programa.

Alguns exemplos de eventos:

Origem: (Produtor)

Segurança doméstica: movimentos muito suspeitos são detectados do lado de fora de uma casa;

Resposta: (Consumidor)

A câmera começa a gravar e as luzes da casa acendem, além de disparar o alarme a acionar a polícia.

Origem: (Produtor)

E-commerce: um usuário faz um pedido;

Resposta: (Consumidor)

Uma verificação de cartão é emitido, a compra é finalizada, um e-mail é enviado, o estoque é atualizado e depois de ser aprovado pelo banco, uma nota fiscal é emitida. (Assíncrona).

Tendo em vista o que é um evento, podemos entender o que é a arquitetura orientada a eventos:

Essa arquitetura é basicamente composta por produtores e consumidores de eventos. Um produtor detecta ou percebe um evento, e o representa como uma mensagem, ele não conhece o consumidor nem o resultado do evento.

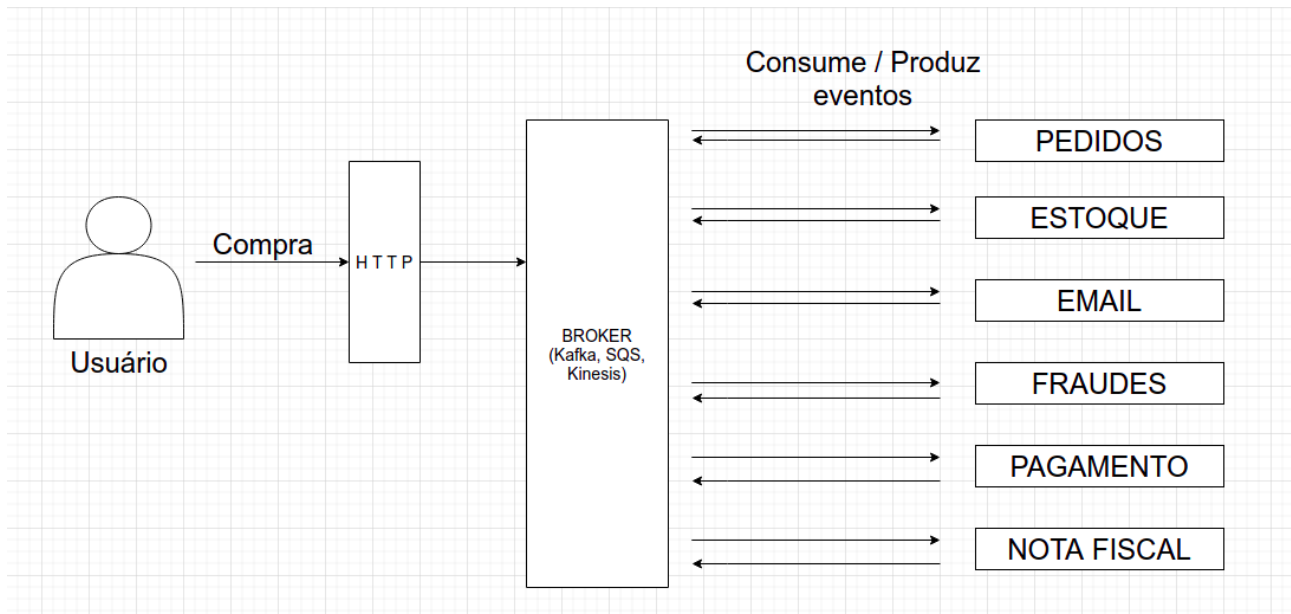
Depois de ser detectado, o evento é transmitido ao consumidor, que por meio de canais processa eventos de maneira assíncrona.

Assíncrona = que não ocorre nem se efetiva ao mesmo tempo.

Para deixar claro:

- **produtores de eventos:** enviam dados para canais sempre que um evento ocorre;
- **canais:** enviam dados recebidos dos produtores de eventos para os consumidores;
- **consumidores do evento:** assinam os canais e consomem os dados.





Benefícios da AOE (ou EDA):

- Ser altamente escalável; → Escalável por natureza, é capaz de ter uma elasticidade refinada aumentando e diminuindo no nível de eventos individuais. Ou seja, no pico de eventos, vai conseguir se comportar bem, e sem muitos eventos reduz os custos por não estar utilizando muito dos recursos..
- Serviços desacoplados e independentes; → As mensagens ficam armazenadas no buffer. Logo os consumidores e produtores são independentes, não precisam se conhecer para funcionar.
- Ser tolerante a falhas; → Kafka e outros sistemas de messengerias são tolerantes a falhas.
- Auditoria central em sistemas distribuídos. → É possível saber tudo que aconteceu se baseando nos eventos.

Pontos fracos da AOE:

- Complexidade ao implementar; → Ter um certo nível de experiencia em plataformas de messengerias seria interessante, pelo nível de complexidade.

Próximos passos:

Entender os Brokers / Middleware (meio de campo) do mercado e como se comportam:

Alguns nomes conhecidos no mercado são:

- Amazon Kinesis
- Amazon SQS
- Apache ActiveMQ
- Apache Kafka
- HornetQ (Red Hat)
- IBM MQ
- JBoss Messaging (JBoss)

- Microsoft Azure Service Bus (Microsoft)
- RabbitMQ (Mozilla Public License, written in Erlang)
- TIBCO Enterprise Message Service

Empresas de sucesso que utilizam da Arquitetura Orientada a Eventos:

Spotify, Twitter, LinkedIn, Ifood, Uber.

Todas essas empresas utilizam da arquitetura orientada a eventos, e o que elas tem em comum?

Todas lidam com milhares de requisições e grande volume de dados.

GitHub:

O git não armazena todo nosso código a cada novo commit, o que ele armazena na verdade são pequenos eventos com nossas alterações, os commits.

Imagine o problema que seria salvar todo o código a cada commit, pelo tamanho e etc.

Entre dois commits, ele não salva duas cópias do arquivo que foi alterado, ele armazena a diferença entre eles, o evento com as modificações.

Por isso, é possível navegar entre as modificações, revertendo um commit ou simplesmente entrando dentro de um evento/commit.

```
commit 78dcad72721b807a6256647585782fe4cc0d1a05 (HEAD -> main, origin/main, origin/HEAD)
Author: Vinicius Dias <carlosv775@gmail.com>
Date: Sat Aug 14 16:43:17 2021 -0300
```

Atualizando package-lock.json

```
commit b74350843ab59ee95756aa58ad6c1abbe61a3589
Author: Vinicius Dias <carlosv775@gmail.com>
Date: Sat Aug 14 15:54:22 2021 -0300
```

Atualizando .gitignore

```
commit 482e1fd2cf81e75c24fb8db44a2b6aae36bbca14
Author: Vinicius Dias <vinicius.dias@achievers.com>
Date: Thu Aug 12 23:01:34 2021 -0300
```

Área logada (lista de cursos) do front-end implementada

```
commit 5e6d6fc097ecf8d8ff09fe4b4e2b94c0147b33f1
Author: Vinicius Dias <vinicius.dias@achievers.com>
Date: Wed Aug 11 23:38:19 2021 -0300
```

