

ARQUITETURA ORIENTADA A EVENTOS COM AWS LAMBDA: CONCEITOS, PADRÕES E BOAS PRÁTICAS

**Cainã Antunes Silva¹, Glauco Todesco², Deivison Shindi Takatu³, André Cassulino
Araújo Souza⁴, Gabriel Claro da Silva⁵, Lucas Miguel Leal da Silva⁶**

¹ Professor Me. no Centro Universitário SENAI São Paulo – UniSENAI-SP, Campus Sorocaba – Santa Rosália. E-mail: caina.silva@sp.senai.br

² Professor Dr. no Centro Universitário SENAI São Paulo – UniSENAI-SP, Campus Sorocaba – Santa Rosália. E-mail: glauco.todesco@sp.senai.br

³ Professor Me. no Centro Universitário SENAI São Paulo – UniSENAI-SP, Campus Sorocaba – Santa Rosália. E-mail: deivison.takatu@sp.senai.br

⁴ Professor Esp. no Centro Universitário SENAI São Paulo – UniSENAI-SP, Campus Sorocaba – Santa Rosália. E-mail: andre.souza@sp.senai.br

⁵ Professor Esp. no Centro Universitário SENAI São Paulo – UniSENAI-SP, Campus Sorocaba – Santa Rosália. E-mail: gabriel.csilva@sp.senai.br

⁶ Coordenador Esp. no Centro Universitário SENAI São Paulo – UniSENAI-SP, Campus Sorocaba – Santa Rosália. E-mail: lucas.msilva@sp.senai.br

Resumo: Este artigo apresenta uma análise conceitual e aplicada da Arquitetura Orientada a Eventos, com foco no uso do AWS Lambda como componente central em soluções serverless modernas. O estudo discute os fundamentos desse modelo arquitetural, destacando o papel de eventos, produtores, consumidores e intermediadores na construção de sistemas distribuídos fracamente acoplados. Também são abordados os principais mecanismos de integração do Lambda com serviços da AWS, como API Gateway, Amazon S3, Amazon SNS, Amazon SQS, DynamoDB Streams, Kinesis e EventBridge. A pesquisa examina ainda aspectos de implantação, desempenho, custo, observabilidade, padrões, antipadrões e boas práticas, evidenciando como funções serverless podem favorecer escalabilidade, resiliência, flexibilidade e redução de custos em cargas de trabalho variáveis. Conclui-se que a adoção de arquiteturas orientadas a eventos com AWS Lambda exige planejamento cuidadoso, especialmente quanto ao desacoplamento, monitoramento, controle de concorrência e mitigação de cold starts, mas oferece uma alternativa eficiente para o desenvolvimento de aplicações cloud native.

Palavras-chave: Arquitetura orientada a eventos; AWS Lambda; Serverless; Computação em nuvem;

INTRODUÇÃO

À medida que os sistemas distribuídos se tornam mais complexos, arquiteturas baseadas exclusivamente no modelo tradicional de requisição e resposta passam a apresentar limitações relacionadas à escalabilidade, manutenção e capacidade de resposta em tempo real. Em ambientes cloud native, nos quais aplicações precisam reagir rapidamente a alterações de estado, a Arquitetura Orientada a Eventos surge como uma alternativa relevante para organizar a comunicação entre componentes de forma assíncrona, desacoplada e escalável.

A Arquitetura Orientada a Eventos, também conhecida como Event-Driven Architecture, estrutura sistemas a partir da publicação, propagação e consumo de eventos. Nesse modelo, produtores geram eventos

quando uma mudança relevante ocorre, enquanto consumidores processam esses eventos sem depender de comunicação direta com os produtores. Essa separação reduz o acoplamento, aumenta a autonomia dos serviços e favorece a construção de soluções resilientes e flexíveis.

No ecossistema da Amazon Web Services, o AWS Lambda destaca-se como serviço serverless baseado no modelo Function as a Service. Por meio dele, funções são executadas em resposta a eventos, sem que o desenvolvedor precise gerenciar servidores, provisionamento, escalabilidade ou manutenção da infraestrutura. Essa característica torna o Lambda um componente central em arquiteturas orientadas a eventos, especialmente quando integrado a serviços como Amazon S3, API Gateway, Amazon SNS,



Amazon SQS, DynamoDB Streams, Amazon Kinesis e Amazon EventBridge.

O objetivo deste artigo é apresentar os conceitos fundamentais da Arquitetura Orientada a Eventos e analisar como o AWS Lambda pode ser utilizado na construção de sistemas modernos, escaláveis e econômicos. Para isso, são discutidos princípios arquiteturais, componentes do Lambda, fontes de eventos, aspectos de custo e desempenho, casos de uso, boas práticas, padrões e antipadrões associados ao desenvolvimento de aplicações serverless.

REVISÃO DE LITERATURA

A literatura sobre arquiteturas distribuídas evidencia uma transição progressiva de sistemas fortemente acoplados para modelos baseados em eventos, microsserviços e comunicação assíncrona. Nesse contexto, um evento pode ser compreendido como uma alteração relevante de estado ocorrida em um sistema, como a criação de um pedido, o envio de um arquivo, a atualização de um registro ou a chegada de uma mensagem em uma fila.

Os principais elementos de uma arquitetura orientada a eventos são os produtores, os consumidores e os intermediadores de eventos. Os produtores publicam eventos sem precisar conhecer os consumidores. Os consumidores reagem aos eventos recebidos e executam processos específicos. Já os intermediadores, como filas, tópicos, barramentos ou brokers, são responsáveis por receber, filtrar, armazenar e encaminhar eventos às partes interessadas.

Entre as abordagens mais comuns de EDA destacam-se o modelo de publicação e assinatura e o streaming de eventos. No primeiro, eventos são publicados em tópicos e entregues a assinantes interessados. No segundo, os eventos são armazenados em um fluxo ou registro consultável, permitindo maior controle sobre processamento, reproprocessamento e análise histórica. Ambas as abordagens contribuem para desacoplamento, escalabilidade e tolerância a falhas.

METODOLOGIA

Este estudo possui natureza aplicada, descritiva e qualitativa, fundamentada na análise de conceitos, padrões arquiteturais e práticas recomendadas para o desenvolvimento de soluções orientadas a eventos com AWS Lambda. A pesquisa foi estruturada a partir da organização do conteúdo em quatro eixos: fundamentos da EDA, funcionamento do AWS Lambda, integração com fontes de eventos e boas práticas para construção de sistemas serverless.

A análise considera serviços AWS recorrentes em arquiteturas orientadas a eventos, como API Gateway, Amazon S3, Amazon SNS, Amazon SQS, DynamoDB Streams, Amazon Kinesis, Amazon

EventBridge, Amazon CloudWatch e AWS X-Ray. Também são discutidos aspectos de implantação por arquivo compactado ou imagem de contêiner, além de fatores relacionados a desempenho, custo, cold start, observabilidade, resiliência e controle de concorrência.

RESULTADOS E DISCUSSÕES

O AWS Lambda permite executar pequenos blocos de código em resposta a eventos, abstraindo grande parte da complexidade de infraestrutura. Seus principais componentes incluem o ambiente de execução, responsável por disponibilizar recursos como memória, CPU e armazenamento temporário; o runtime, que define a linguagem e o ambiente de execução; e o handler, que representa o ponto de entrada da função e recebe o evento processado.

As funções Lambda podem ser implantadas por meio de arquivos compactados ou imagens de contêiner. O uso de arquivos .zip é adequado para funções menores e dependências simples, enquanto imagens de contêiner oferecem maior controle sobre dependências, bibliotecas e configurações específicas. Essa flexibilidade permite adaptar o modelo de implantação ao perfil técnico da aplicação e aos requisitos de governança do ambiente.

No que se refere às fontes de eventos, o Lambda pode ser acionado diretamente por serviços como API Gateway, Amazon S3 e Amazon SNS, ou por meio de mapeamentos de origem de evento associados a serviços baseados em fila ou fluxo, como Amazon SQS, DynamoDB Streams e Kinesis. Essa diversidade de integrações possibilita a criação de aplicações que reagem a requisições HTTP, uploads de arquivos, mensagens, registros de banco de dados e fluxos de dados em tempo quase real.

Em relação a desempenho e custo, o modelo de cobrança do Lambda considera principalmente o número de requisições e o tempo de execução associado à memória configurada. Esse formato favorece cargas de trabalho variáveis, pois o custo acompanha o uso efetivo. Em comparação com servidores sempre ativos, como instâncias EC2, ou containers mantidos continuamente em execução, funções serverless podem representar economia significativa quando há períodos de ociosidade ou demanda imprevisível.

Os principais casos de uso incluem backends de APIs, microsserviços, automação operacional, tarefas agendadas, processamento de arquivos, integração entre serviços, tratamento de eventos em streaming e execução de rotinas administrativas. Em arquiteturas orientadas a eventos, o Lambda pode atuar tanto como consumidor de eventos quanto como componente de transformação, validação, roteamento ou

enriquecimento de dados, conforme exemplificado na Figura 1.

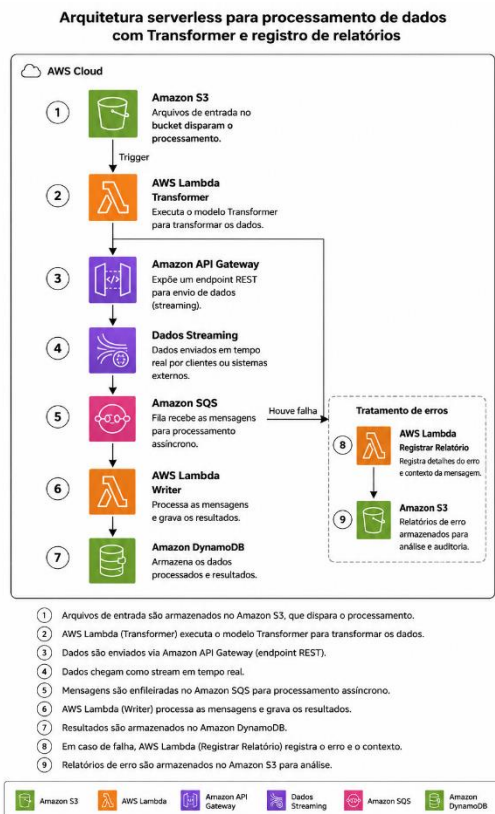


Figura 1 – Diagrama da arquitetura de um sistema montado na AWS para aprendizado dos conceitos fundamentais da arquitetura orientada a eventos e AWS Lambda. Fonte: Elaboração do autor.

A adoção de boas práticas é essencial para preservar os benefícios da abordagem serverless. Recomenda-se que cada função tenha responsabilidade única, reduzindo complexidade e facilitando testes, manutenção e escalabilidade independente. O ajuste adequado de memória e CPU, conhecido como power tuning, também contribui para equilibrar desempenho e custo, uma vez que mais memória pode reduzir o tempo de execução e, em alguns casos, diminuir o custo total.

A observabilidade representa outro requisito crítico. Ferramentas como Amazon CloudWatch e AWS X-Ray permitem acompanhar logs, métricas, rastreamentos distribuídos, latência, erros, volume de eventos, filas acumuladas e mensagens encaminhadas para Dead Letter Queues. Esse acompanhamento é indispensável em fluxos assíncronos, nos quais falhas podem ocorrer em diferentes pontos da cadeia de eventos.

Entre os antipadrões mais relevantes estão o Lambda monolito, as invocações recursivas e o encadeamento direto entre funções. O Lambda monolito ocorre

quando uma única função concentra responsabilidades excessivas, dificultando testes, manutenção e aplicação do princípio do menor privilégio. Invocações recursivas podem gerar loops de execução e custos inesperados. Já a chamada direta entre funções aumenta o acoplamento e reduz a resiliência. Como alternativas, recomenda-se decompor funções, utilizar filas como Amazon SQS e empregar serviços de orquestração, como AWS Step Functions, quando o fluxo exigir coordenação explícita.

Os padrões mais indicados para arquiteturas orientadas a eventos envolvem desacoplamento, roteamento de eventos, resiliência e observabilidade. Serviços como Amazon EventBridge e Amazon SQS contribuem para intermediar comunicações assíncronas, enquanto mecanismos de retry e Dead Letter Queues reduzem o risco de perda de mensagens. Assim, a qualidade da arquitetura depende menos do uso isolado de uma função Lambda e mais da composição adequada entre serviços, responsabilidades e estratégias de recuperação.

CONCLUSÃO

A Arquitetura Orientada a Eventos representa uma abordagem eficiente para o desenvolvimento de sistemas distribuídos modernos, especialmente em ambientes que exigem escalabilidade, resiliência e flexibilidade operacional. Ao permitir que componentes publiquem e consumam eventos de forma assíncrona e desacoplada, esse modelo favorece a evolução independente dos serviços e reduz dependências diretas entre partes do sistema.

O AWS Lambda consolida-se como um componente importante nesse paradigma ao possibilitar a execução de funções sob demanda, acionadas por eventos e cobradas conforme o uso. Sua integração com diversos serviços da AWS permite construir soluções serverless para APIs, automações, processamento de arquivos, filas, streams e fluxos orientados a eventos sem a necessidade de gerenciamento direto de servidores.

Entretanto, a adoção de Lambda e EDA não elimina a necessidade de planejamento arquitetural. Questões como cold start, observabilidade, controle de custos, concorrência, segurança, tratamento de falhas e prevenção de antipadrões devem ser consideradas desde o desenho da solução. Conclui-se que o uso adequado do AWS Lambda em arquiteturas orientadas a eventos pode gerar aplicações escaláveis, econômicas e resilientes, desde que apoiado por boas práticas, monitoramento contínuo e decisões coerentes com os requisitos do negócio.

REFERÊNCIAS



AMAZON WEB SERVICES. O que é arquitetura orientada a eventos (EDA)? Amazon Web Services, 2025.

AMAZON WEB SERVICES. AWS Lambda: documentação oficial. Amazon Web Services, 2025.

AMAZON WEB SERVICES. AWS Lambda: preços oficiais. Amazon Web Services, 2025.

AMAZON WEB SERVICES. Boas práticas: AWS Lambda. Amazon Web Services, 2025.

AMAZON WEB SERVICES. Best practices for implementing event-driven architectures in your organization. AWS Architecture Blog, 2025.

AMAZON WEB SERVICES. Operating Lambda: anti-patterns in event-driven architectures. AWS Compute Blog, 2025.

GOOGLE CLOUD. O que é arquitetura orientada a eventos? Google Cloud, 2025.

MICROSOFT. Event-driven architecture, pub/sub e fluxo. Microsoft Learn, 2025.

RED HAT. O que é arquitetura orientada a eventos? Red Hat, 2019.