

02/08/2025 - Lambda para salvar capturas de pokémons

Dono	@Evandro Douglas Capovilla Junior (Unlicensed)
Status	EM REVISÃO
Data Limite	10/08/2025
Refinamento Técnico	<Link do Refinamento Técnico>
Revisores	<div><input checked="" type="checkbox"/> @Evandro Douglas Capovilla Junior (Unlicensed)</div> <div><input type="checkbox"/> @Evandro Douglas Capovilla Junior (Unlicensed)</div> <div><input type="checkbox"/></div>

[Contexto](#)

[Suposições](#)

[Possíveis Soluções](#)

[Decisão](#)

[Criação de uma lambda para cadastro de API](#)

[Detalhes](#)

[Custos](#)

Contexto

Uma nova funcionalidade será implementada para permitir que os jogadores (Treinadores) registrem os Pokémons capturados em suas aventuras. As requisições de cadastro serão processadas de forma assíncrona na nuvem da AWS, por meio de um novo endpoint exposto pelo AWS API Gateway, utilizando o Lambda para o processamento dos dados.

Suposições

- O tráfego do site aumentará nos periodos das 12:00 ~ 16:00.
- Os usuários costumam ver seus pokemons no final do dia.

Possíveis Soluções

<Outras possíveis soluções>

<Outras possíveis soluções>







<Outras possíveis soluções>

Decisão

Criação de uma lambda para cadastro de API

A solução foi projetada com uma arquitetura **serverless** utilizando **AWS Lambda** para processar as informações. Esta escolha garante alta escalabilidade e eficiência de custos, pois o pagamento é baseado no tempo de execução, eliminando a necessidade de manter um serviço ativo 24 horas por dia. Para otimizar o desempenho, o **Amazon ElastiCache (Redis)** foi implementado para armazenar dados da Pokedex-API, reduzindo a latência e a carga desnecessária sobre a API de origem.

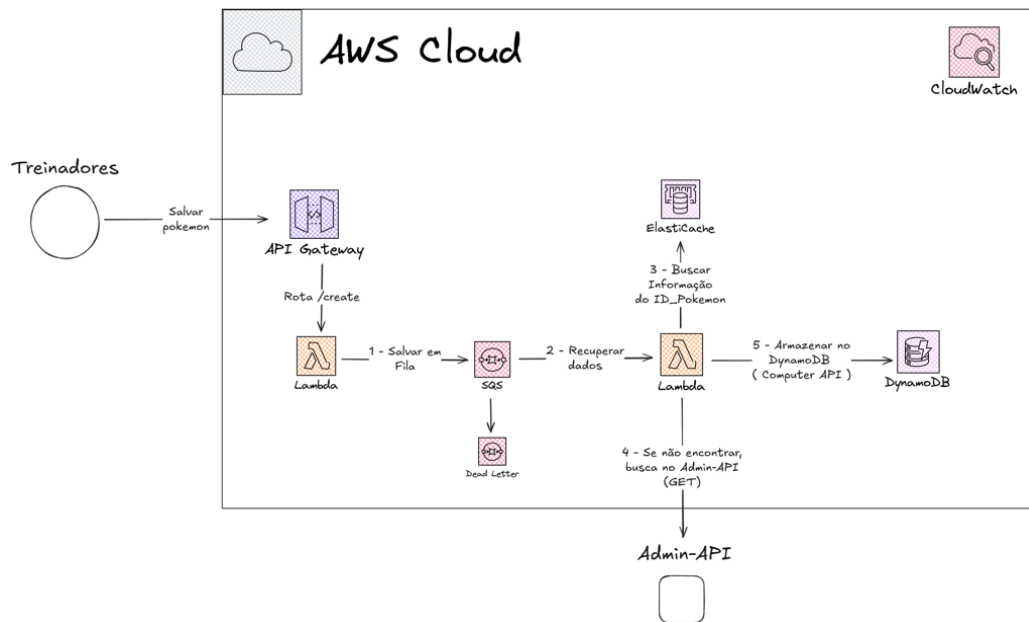
Trade-offs:

-  Velocidade de processamento dos dados.
-  Garantia que os dados serão salvos.
-  Redução da carga no banco de dados.
-  Aumento na complexidade da infraestrutura.
-  Custo para manter um sistema robusto na cloud.
-  Coldstart para as primeiras requisições.

Detalhes

Para essa solução foi pensado em utilizar a cloud AWS com os seguintes serviços:

- API Gateway - Para redirecionamento das rotas.
- Lambda - Para armazenar as requisições em filas.
- SQS e Dead Letter - Utilizados para gerenciamento das filas.
- Lambda - Para a consulta e cadastro das informações no banco de dados.
- DynamoDB - Para salvar os dados e garantir uptime e escalabilidade.



Custos

A implementação do cache distribuído resultará em:

Descrição	Calculo	Valor
Custo adicional de infraestrutura	<ul style="list-style-type: none"> \$X00/mês VPC, CloudWatch e ... 	+\$X00/mês
Custo das Lambdas	<ul style="list-style-type: none"> \$X00/tempo para as lambdas 2 Lambdas 	+\$X00/mês
Custo dos bancos de dados	<ul style="list-style-type: none"> \$X00/mês para o DynamoDB \$X00/mês para o ElastiCache 	+\$X00/mês
Custo da mensageria	<ul style="list-style-type: none"> \$X00/mês para o SQS 	+\$X00/mês

Custo líquido: \$Z00/mês

O aumento nos custos é justificado pela melhoria esperada no desempenho e na experiência do usuário.