**Polly (Design Pattern de Resiliência)**

Fornece estratégias de resiliência para a aplicação com padrões como:

* Retry
* WaitAndRetry
* CircuitBreaker
* Fallback

**Retry (primeira camada de resiliência)**

Ele serve para tentar novamente caso uma chamada falhe. Ele pode tentar novamente das seguintes formas:

* Imediatamente (Retry)
  + Usado quando ocorre um erro raro (retorno 500)
* Após um tempo (Wait And Retry)
  + Usando quando um serviço está indisponível (timeout)
* Cancelar a retentativa
  + Erro conhecido, por exemplo “senha inválida”

E quando parar de repetir? O responsável por isso é o circuit breaker.

**Circuit Breaker (segunda camada de resiliência)**

Ele serve para evitar sobrecarga de chamadas.

* Closed = API funcionando normalmente, porém sendo monitorada para eventuais falhas.
* Open = Após uma determinada quantia de falhas dentro de um intervalo de tempo, ele irá retornar erro para todas as chamadas por um período.
* Half-Open = Após o período, ele irá fazer uma tentativa de execução e determinar se fecha o circuito ou se mantem ele aberto.

E quando o circuito ficar aberto, o que fazer?

Bom você pode esperar ele voltar (Half-Open) ou dar um Fallback

**Fallback (Última camada de resiliência)**

O fallback será executado quando todas as tentativas falharem, como se fosse o último recurso.

* **Custom fallback:** Retorna um valor padrão
* **Fail silent:** Retorna null
* **Fail fast:** Retorna um erro (código 500)