

# Monitorização e Gestão de API<sup>\*</sup>

## Um caso prático na E-goi

Catarina Sousa<sup>1</sup>, Ricardo Almeida<sup>1</sup>[0000–0002–3278–9021], Paulo Proença<sup>1</sup>[0000–0001–6314–5969], and Ivo Pereira<sup>2</sup>[0000–0001–5440–3225]

<sup>1</sup> Instituto Superior de Engenharia do Porto, Portugal  
{1150767,ral,prp}@isep.ipp.pt

<sup>2</sup> E-goi ipereira@e-goi.com

**Resumo:** Nos últimos anos, o mercado de API's tem vindo a crescer exponencialmente, fazendo com que cada vez mais as empresas apostem nesta tecnologia para garantir a continuidade dos seus negócios [1]. Contudo, com o crescimento da popularidade de uma empresa, a sua API começa a ter um aumento no numero de utilizações, o que torna mais complexa a identificação dos erros e aumenta a criticidade da sua resolução. Se esse problema não for colmatado com brevidade pode por em causa os negócios da empresa. Para estes casos existem ferramentas de gestão e monitorização de API que permitem combater esta adversidade. Contudo, são muitas vezes difíceis, complexas de utilizar e muitas vezes não respondem diretamente ao que é pretendido. Neste documento foi estudado o caso prático da empresa E-goi, que se encontrava na situação mencionada, onde foi desenvolvida uma ferramenta de gestão e monitorização de API simples e única. Contrariamente às ferramentas existentes, esta contém funcionalidades únicas (como análises por pedido e por cliente), mas relevantes para resolução do problema.

**Palavras-chave:** API, monitorização, gestão, análises por cliente, análises por pedido.

## 1 Introdução

Atualmente as API's representa uma tecnologia com um mercado cada vez mais lucrativo e popular [1]. Esta tecnologia acompanha o crescimento das empresas e para muitas constitui um elemento crucial para a sua estratégia empresarial. Segundo um estudo realizado pela Cloud Elements, empresa de produtos de integração com API [3], 55% dos inquiridos afirmaram que a integração com interfaces de programação é de importância vital para a estratégia empresarial e 29% afirmaram que é importante [4].

Com o crescimento das empresas, aumentam os pedidos às suas API o que torna torna difícil detetar e solucionar erros relacionados com estes pedidos. Se os erros não forem solucionados com brevidade podem ter impacto direto na

---

<sup>\*</sup> Supported by E-goi.

performance das tecnologias que estão dependentes da API em questão, causando problemas com os negócios das empresas. De forma a conceber uma solução que melhor resolve-se o problema foi utilizado como caso prático a situação da empresa E-goi.

A plataforma de automação de marketing multicanal da E-goi envia milhares de campanhas todos os dias, repartidas por vários canais de comunicação, nos quais se destacam e-mail, sms, push, webpush e voz. Em outubro de 2020, a E-goi contava com mais de meio milhão de utilizadores dispersos em mais de 40 países [2].

Todo o seu tráfego é efetuado através de uma interface de programação de aplicações (API) de acesso público. Devido ao rápido crescimento da plataforma torna-se cada vez mais difícil detetar e solucionar problemas na mesma, pelo que este facto constitui um grande problema para a empresa.

A monitorização permite saber o estado da API e consequentemente obter uma resposta rápida a problemas por parte dos developers [5]. A gestão permite facilitar, para a equipa de desenvolvimento, os processos relacionados com a API como a publicação, a segurança e documentação [6]. A eficiência na resolução de erros e execução de processos, proporciona diminuição do tempo de inatividade do produto. Esta solução, permite aos clientes, realizarem todas as suas tarefas nos produtos sem correrem o risco de não as conseguirem terminar ou mesmo começar. As empresas obtêm assim satisfação dos clientes e evitam a perda dos mesmos.

Com a implementação de tarefas de monitorização e de gestão é possível responder ao problema do aumento do uso da API.

## 2 Solução

Após uma pesquisa de mercado, não foi encontrada nenhuma tecnologia que resolve-se o problema, por completo, por isso tornou-se necessário desenhar e criar uma solução proprietária.

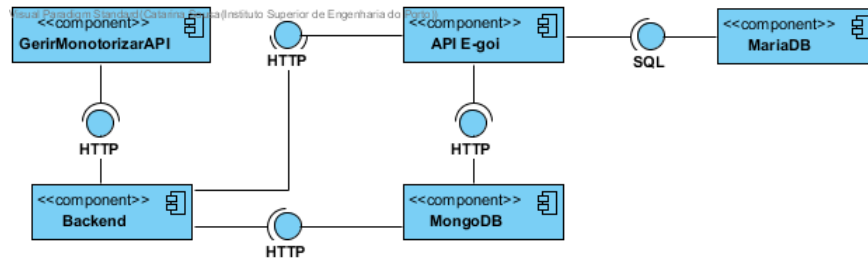
### 2.1 Análise

No processo de análise foram definidos o administrador, o utilizador e o utilizador não registado, como os atores do projeto. Para estes identificaram-se os seguintes casos de uso:

- **Criar registo** - o utilizador que não está registado cria conta na plataforma;
- **Consultar análises por pedido** – o utilizador e o administrador conseguem visualizar as análises dos pedidos da API;
- **Consultar análises de clientes** – o utilizador e o administrador conseguem ver as análises de pedidos que os clientes fizeram;
- **Adicionar API** – o administrador pode adicionar outras API para monitorizar e gerir;
- **Criar teste** – o utilizador pode criar testes;
- **Consultar testes** – o utilizador pode consultar testes passados;

## 2.2 Arquitetura

A arquitetura de componentes da solução, é a apresentada na Figura 1.



**Fig. 1.** Diagrama de componentes

Como é possível observar pelo diagrama da Figura 1, o componente GerirMonitorizarAPI comunica com o Backend. Estes comunicam com os componentes MongoDB e API E-goi. E por fim o componente API E-goi comunica com a base de dados da E-goi e com MongoDB. O componente GerirMonitorizarAPI, desenvolvida na linguagem de programação Angular, é onde se encontra a lógica da interface, que permite aos seus utilizadores criar e visualizar testes à API e consultar análises por pedido ou por cliente. A Figura 2 apresenta a arquitetura deste componente.

O componente divide-se em três subcomponentes: templates, components e services. Visto que o componente GerirMonitorizarAPI apenas inclui lógica para a interface, é necessária uma componente que tenha a lógica da persistência de dados, neste caso é Backend. Esta trata-se de uma API REST, desenvolvida na linguagem de programação Node.js, que contém a lógica de negócio e do tratamento de dados. O componente Backend comunica com a API da E-goi e com o MongoDB, por protocolo Hyper Text Transfer Protocol(HTTP). A Figura 3 apresenta a arquitetura do componente Backend.

O componente Backend apresenta os subcomponentes: rotas, controlador, serviços e modelos. O Backend e GerirMonitorizarAPI, comunicam por protocolo HTTP. Pela análise da Figura 1 é ainda possível visualizar que a componente API E-goi comunica com duas componentes de bases de dados diferentes, MongoDB e MariaDB. O componente MongoDB, é responsável por a persistência de dados da ferramenta e guarda os dados dos pedidos feitos à API E-goi. Utiliza a tecnologia MongoDB [7]. A componente MariaDB, é responsável pela persistência de dados do produto E-goi. A forma como estão organizados os componentes facilita adicionar ou alterar funcionalidades à ferramenta, uma vez que se encontra repartida por diferentes componentes, oferecendo assim escalabilidade e adaptabilidade (recurso não funcional).

### 3 Avaliação

Após finalizado o processo de desenvolvimento da ferramenta, procedeu-se à avaliação da ferramenta relativamente à resposta ao problema.

#### 3.1 Critérios de aceitação

Para este projeto consideraram-se os seguintes critérios de aceitação:

- 100 % de sucesso nos testes funcionais
- 60 % de satisfação dos utilizadores da plataforma

O critério de aceitação dos testes funcionais permite garantir a qualidade do código da solução. E o critério relativo à satisfação dos utilizadores garante que os requisitos definidos foram cumpridos, uma vez que a solução foi concebida para ajudar com a necessidade sentida pelos programadores, mencionada na secção 1.

#### 3.2 Metodologia de avaliação

As metodologias de avaliação, implementadas para suportar os critérios foram os testes funcionais e testes alfa.

Os testes funcionais foram realizados ao longo da implementação. Estes incluíram testes unitários, integração e aceitação. Desta forma foi possível validar o projeto em relação aos requisitos de negócio.

Os testes alfa foram efetuados na fase de término do projeto, onde os utilizadores da ferramenta, que neste caso são a equipa de desenvolvimento do produto, da empresa E-goí, apontaram os erros e os problemas de uso. Conseguindo dessa forma a aceitação do produto.

#### 3.3 Resultados

No final do desenvolvimento do projeto e perante os critérios da secção 3.1 obteve-se 100 por cento nos testes funcionais e 70 por cento de satisfação dos utilizadores.

Para os testes alfa, a amostra consistiu em 4 utilizadores, do sexo masculino, com experiência na área de informática entre 1 a 20 anos e com idades entre os 23 aos 40. Foi pedido aos intervenientes para avaliarem a solução de 0 a 10 tendo em conta a concordância com os requisitos estabelecidos e a usabilidade.

Os resultados da avaliação foram: 7, 7, 8 e 6. Em comum todos os intervenientes afirmaram que a ferramenta é simples e intuitiva de usar, contudo têm necessidade de análises relativas não só à API pública, como para a API interna e isso representa um ponto negativo da solução. Esse ponto era uma funcionalidade extra e algo que irá ser feito no futuro.

### 3.4 Discussão

Olhando de uma forma critica para os resultados apresentados na secção 4, a amostra é pequena. Porém os individuos que a constituem representam as pessoas que irão dar uso à ferramenta e apresentam diferentes níveis de experiência na área de informática.

A solução apresentada satisfaz os critérios de aceitação, logo resolve o problema identificado, na secção 1.

## 4 Conclusão

Em suma, o projeto descrito neste artigo passou nos testes efetuados, cumpriu os requisitos propostos e resolveu o problema. Contudo a solução e requiere ainda trabalho no futuro, como por exemplo, disponibilizar análises para outras API.

Comparativamente com as ferramentas de monitorização e gestão de API do mercado que são complexas e com demasiadas funcionalidades, esta abordagem representa uma forma simples e eficaz de resolver um problema muito comum em empresas que verificam um crescimento exponencial nas chamadas a API.

## Referências

1. API Economy: Is It The Next Big Thing?. Forbes <https://www.forbes.com/sites/tomtaulli/2020/01/18/api-economy-is-it-the-next-big-thing/?sh=6f212d7842ff> Acedido Nov. 18, 2020.
2. Tecnológica Portuguesa E-goi oferece plano para reerguer PMEs. Pplware <https://pplware.sapo.pt/internet/tecnologica-portuguesa-e-goi-oferece-plano-para-reerguer-pmes/> Acedido Fev. 2, 2020.
3. Cloud Elements. <https://www.cloud-elements.com> Acedido Fev. 2, 2020.
4. C. Elements, «Study: 55 Percent Say API Integration Is “Critical” to Business Strategy», GlobeNewswire News Room, 04-Ago-2019. <http://www.globenewswire.com/news-release/2019/04/08/1798931/0/en/Study-55-Percent-Say-API-Integration-Is-Critical-to-Business-Strategy.html>. Acedido em Jan. 11, 2020.
5. What Is API Monitoring | API Basics | SmartBear. <https://smartbear.com/solutions/api-monitoring/>. Acedido em Jan. 11, 2020. Oct 2017
6. What is API Management? | MuleSoft. <https://www.mulesoft.com/resources/api/what-is-api-management> Acedido Jan. 11, 2020.
7. MongoDB. <https://www.mongodb.com> Acedido Fev. 2, 2020.