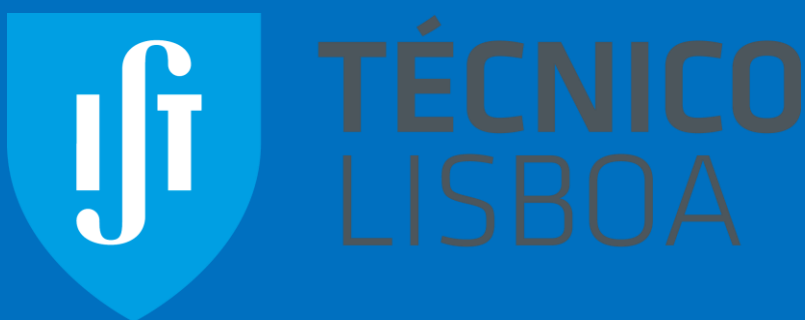


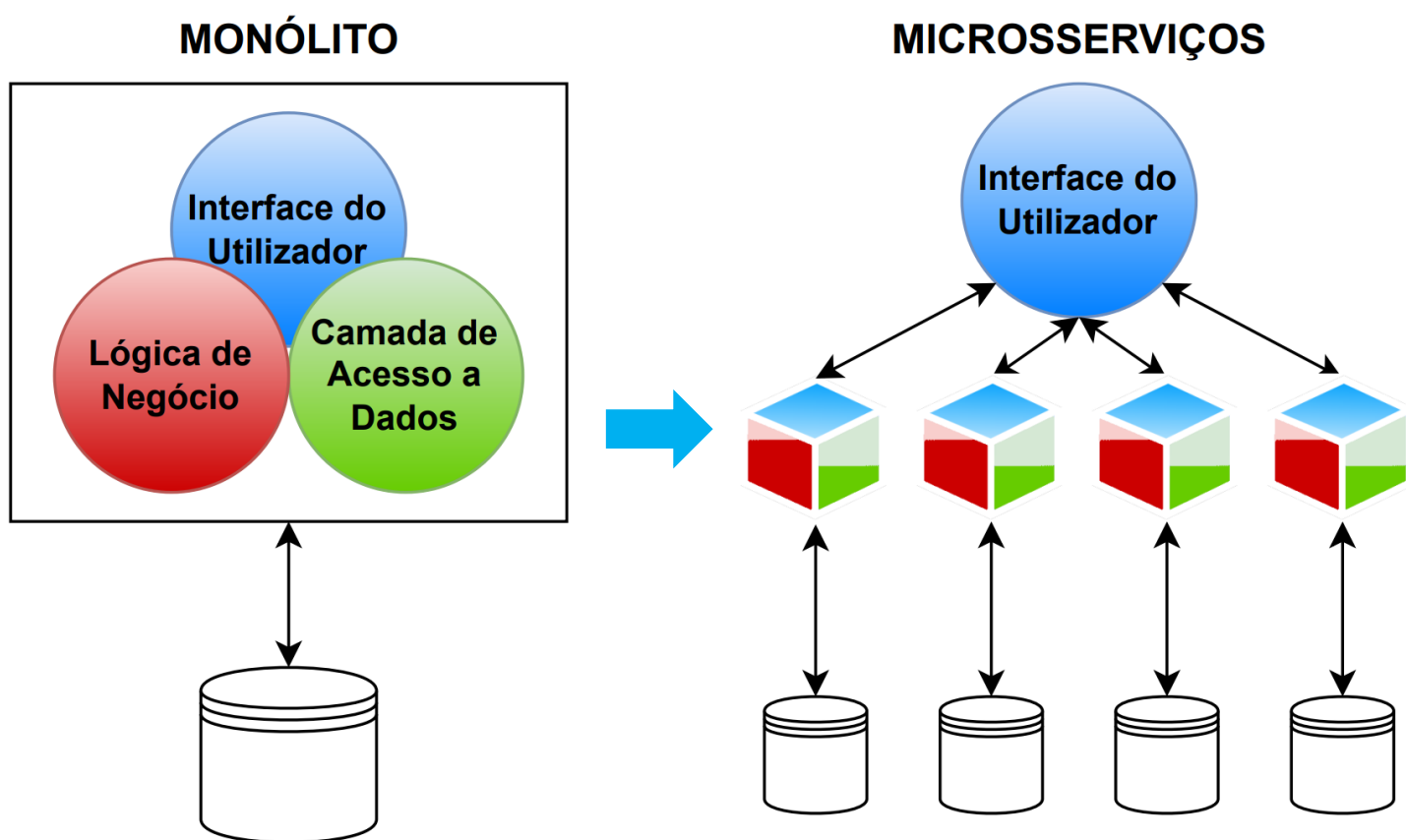
# Deteção Automática de Anomalias em Arquiteturas de Microserviços

Valentim Romão, Rafael Soares, Vasco Manquinho, Luís Rodrigues



## 1. Motivação

- A **migração** de monólito para microserviços pode introduzir **anomalias**;
- Estas anomalias são **difíceis de detetar**.

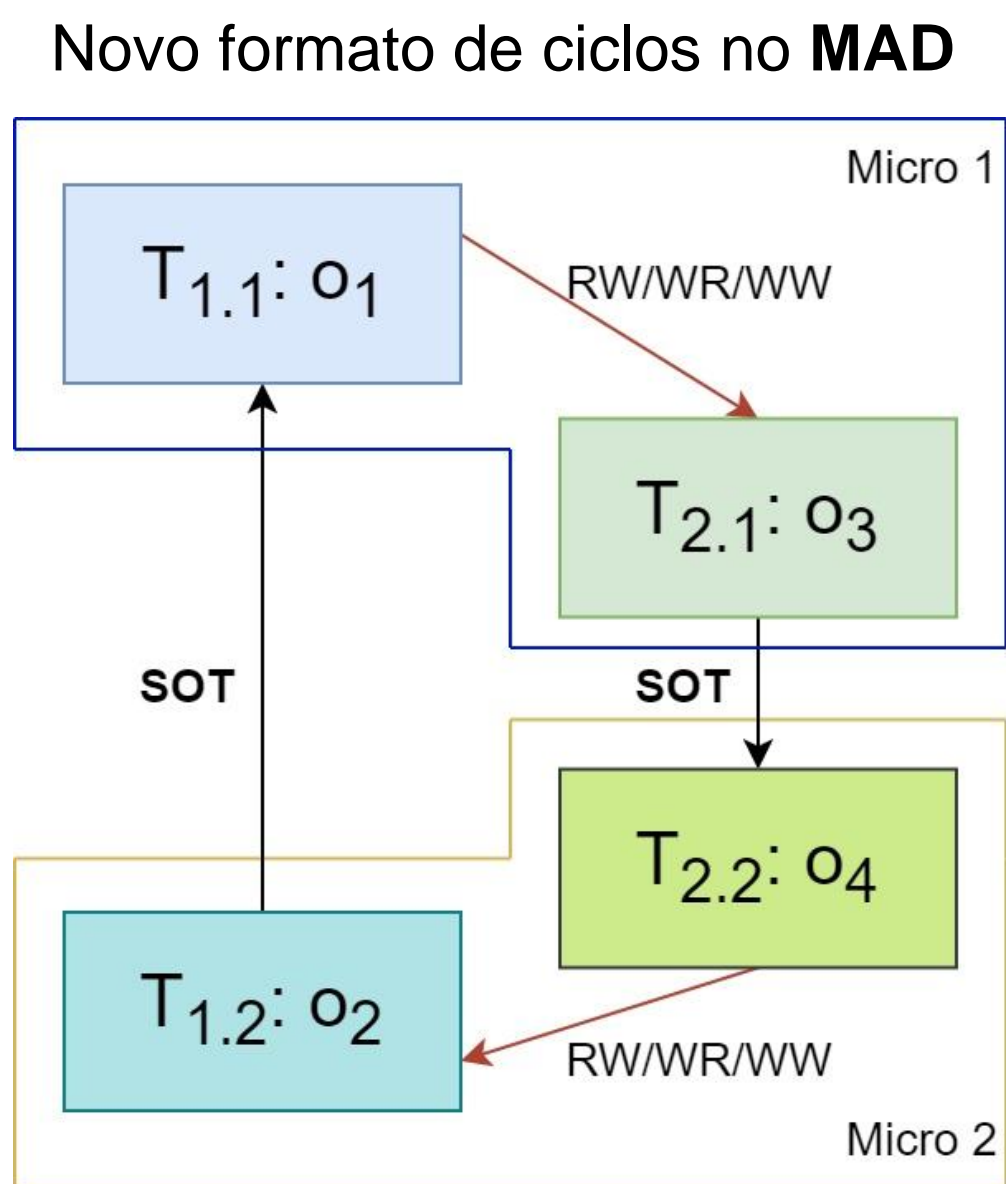


## 2. Objetivo

- Criar uma **ferramenta** capaz de **detetar** de forma automática estas **anomalias**;
- MAD: Microservices Anomaly Detector**.

## 3. Implementação do MAD

- O **MAD** foi desenvolvido como uma **extensão** de uma ferramenta anterior, o **CLOTHO**;
- O **MAD** constrói **grafos** que capturam os **padrões de acesso aos dados** feitos pelos **microserviços**;
- A **existência de um ciclo** no grafo sinaliza uma **anomalia**;
- O **MAD** introduz uma aresta, **SOT (Same Original Transaction)**, para indicar que duas **sub-transações** pertencem à mesma funcionalidade, e considera o **microserviço** em que cada operação irá ser executada.



## 4. Resultados

- Os nossos resultados focam-se na **comparação** entre os resultados obtidos através de heurísticas como a descrita no trabalho “A Complexity Metric for Microservices Architecture Migration” (**CMMAM**) e os resultados obtidos pelo **MAD**.

	#Transações	#Tabelas	#Microserviços	Ferramenta	Complexidade/ #Anomalias	Tempo de Execução [s]
TPC-C mono	5	9	1	CMMAM	0	≈ 0
				MAD	0	33
TPC-C decomposição-1	22	9	9	CMMAM	36	≈ 0
				MAD	68	13.241
TPC-C decomposição-2	12	9	3	CMMAM	20	≈ 0
				MAD	25	2.094

### Vantagens:

- O **MAD** é **mais preciso**, pois considera dependências entre escritas e as linhas das tabelas acedidas;
- O **MAD** pode também ser utilizado para **escolher a decomposição** com menos anomalias.

### Desvantagens:

- O **MAD** tem uma **maior complexidade temporal**.

## 5. Agradecimentos

- Este trabalho foi suportado pela FCT – Fundação para a Ciência e a Tecnologia, através dos projectos UIDB/50021/2020 e DACOMICO (financiado pelo OE com a ref. PTDC/CCI-COM/2156/2021).