



NOVA SCHOOL OF  
SCIENCE & TECHNOLOGY

## **Sistemas de Computação em Cloud**

**2024/2025**

Relatório do 1º Projeto

**“Azure Tukano”**

**Diogo Lemos, nº 56837**

**José Trigueiro, nº 58119**

## 1. Introdução

Este relatório visa demonstrar e analisar as decisões tomadas em torno da incorporação dos serviços de *Platform as a Service* da Azure sobre o projeto previamente concebido, Tukano. Ao longo do documento, são descritas as abordagens tomadas e como estas impactam positivamente a performance do sistema. Para estas conclusões positivas, são usadas e comparadas métricas garantidas pela ferramenta de testes sugerida, Artillery.

## 2. Solução

O sistema Tukano foi aprimorado de maneira a incorporar serviços de PaaS da Azure, nomeadamente, CosmosDB, uma base de dados NoSQL e relacional, Blob Storage, também uma base de dados focada em dados não estruturados, e Redis, um sistema de armazenamento de dados com maior velocidade de acesso.

Estes serviços permitem ao sistema usufruir das vantagens de desenvolvimento em Cloud, isto é, permitem um maior grau de escalabilidade, acessibilidade e um custo de produção que acompanha a necessidade do sistema.

### 2.1 CosmosDB

O serviço do CosmosDB foi principalmente utilizado para o armazenamento tanto dos utilizadores do sistema, como dos Shorts, ou seja das publicações, dos mesmos. Foi também utilizado para guardar informação relativa aos utilizadores seguidos (Follows) e aos gostos de cada Short (Likes). A incorporação do CosmosDB veio a substituir a utilização do sistema Hibernate que guardava os dados em memória.

### 2.2 Blob Storage

O serviço do Blob Storage foi utilizado exclusivamente para o armazenamento dos ficheiros de média relativos aos Shorts, Blobs. Sendo este serviço focado no armazenamento de dados não estruturados, foi uma excelente escolha para guardar conjuntos de bytes relativos a imagens/vídeos que fossem carregados para o sistema. Os Blobs estavam originalmente a ser escritos em ficheiros presentes na memória local.

### 2.3 Redis

Por fim o serviço do Redis foi incorporado de maneira a servir de cache, e a manter dados recentes, nomeadamente informação em volta de utilizadores ou shorts em memória para que o acesso a estes fosse realizado de uma forma crucialmente mais rápida, tendo um enorme impacto na performance do sistema. Não existia um serviço de cache no sistema original do Tukano.

### 3. Performance

Para efeitos de avaliação de performance é utilizado o mesmo conjunto de testes sobre três versões diferentes do sistema. Começando pelo sistema original do Tukano, servindo de padrão, seguido de duas versões, com e sem cache, de maneira a testar a influência do Redis e do CosmosDB e Blob Storage, respectivamente, sobre o sistema.

As métricas utilizadas para comparação são essencialmente os tempos de resposta aos pedidos.

O conjunto de testes divide-se em três fases, sendo estas focadas em operações em volta de inserção e atualização de utilizadores (User Register), deleção dos mesmos (User Delete), publicação de shorts (Upload Shorts).

#### 3.1 Tukano (Original)

##### 3.1.1 User Register

Resultados:

**http.codes.200:** 600

**http.downloaded\_bytes:** 103279

**http.request\_rate:** 6/sec

**http.requests:** 600

Tempos de Resposta (ms):

**min:** 58

**max:** 1107

**mean:** 521

**median:** 450

##### 3.1.2. User Delete

Resultados:

**http.codes.200:** 200

**http.downloaded\_bytes:** 43410

**http.request\_rate:** 2/sec

**http.requests:** 200

Tempos de Resposta (ms):

**min:** 90

**max:** 1128

**mean:** 501

**median:** 442

##### 3.1.3 Upload Shorts

Resultados:

**errors.TIMEOUT:** 288

**http.codes.200:** 24

**http.downloaded\_bytes:** 5260

**http.request\_rate:** 7/sec

**http.requests:** 312

Tempos de Resposta 2xx (ms):

**min:** 141

**max:** 665

**mean:** 520

**median:** 584

## 3.2 Azure Tukano (Sem Cache)

### 3.2.1 User Register

Resultados:

**http.codes.200:** 600

**http.downloaded\_bytes:** 113420

**http.request\_rate:** 6/sec

**http.requests:** 600

Tempos de Resposta (ms):

**min:** 58

**max:** 528

**mean:** 88.4

**median:** 74.4

### 3.2.2. User Delete

Resultados:

**http.codes.200:** 200

**http.downloaded\_bytes:** 43696

**http.request\_rate:** 2/sec

**http.requests:** 200

Tempos de Resposta (ms):

**min:** 80

**max:** 414

**mean:** 99.1

**median:** 90.9

### 3.2.3 Upload Shorts

Resultados:

**http.codes.200:** 299

**http.codes.403:** 1

**http.codes.404:** 300

**http.downloaded\_bytes:** 130516

**http.request\_rate:** 17/sec

**http.requests:** 600

Tempos de Resposta (ms):

**min:** 75

**max:** 1483

**mean:** 206.4

**median:** 169

### 3.3 Azure Tukano (Completo)

#### 3.3.1 User Register

Resultados:

**errors.TIMEOUT:** 126

**http.codes.200:** 222

**http.downloaded\_bytes:** 32366

**http.request\_rate:** 4/sec

**http.requests:** 348

Tempos de Resposta 2xx (**ms**):

**min:** 62

**max:** 308

**mean:** 88

**median:** 79

#### 3.3.2. User Delete

Resultados:

**http.codes.200:** 122

**http.codes.500:** 78

**http.downloaded\_bytes:** 33647

**http.request\_rate:** 2/sec

**http.requests:** 200

Tempos de Resposta 2xx (**ms**):

**min:** 76

**max:** 374

**mean:** 97

**median:** 87

#### 3.3.3 Upload Shorts

Resultados:

**http.codes.200:** 299

**http.codes.403:** 1

**http.codes.404:** 300

**http.downloaded\_bytes:** 130319

**http.request\_rate:** 17/sec

**http.requests:** 600

Tempos de Resposta 2xx (**ms**):

**min:** 57

**max:** 704

**mean:** 105

**median:** 79

### **3.4 Análise de Dados**

Analizando superficialmente os dados, é facilmente observável a diferença em tempo de execução entre a versão original do Tukano, com tempos médios em volta de 500 ms, e as versões aprimoradas do mesmo, onde os tempos médios rodeiam os 100 ms. Esta diferença em valores ilustra uma vantagem na utilização dos serviços de armazenamento da Azure, sobre o armazenamento com recurso ao Hibernate, utilizado no projeto original.

A implementação da Cache causou alguns erros sobre a execução, o que afetou diretamente os tempos. Para efeitos de isolar estes casos, foram considerados apenas os tempos das operações com resultados de sucesso (2xx). Tendo este facto em conta, a Cache revelou-se vantajosa nos vários testes declarados. Em particular, obteve resultados extremamente mais rápidos no teste Upload Shorts onde, apesar do melhor caso se ter mantido ligeiramente semelhante, os tempos médios desceram para metade do valor do sistema anterior, aumentando a performance do sistema de forma drástica.

Estas melhorias ilustram o impacto que as tecnologias de PaaS da Azure podem ter não apenas sobre o sistema Tukano, mas para qualquer projeto que vise a ser implementado em rede.

## **4. Conclusão**

Tendo em conta os dados analisados na fase anterior, observamos uma melhoria crucial sobre a performance do sistema original, tanto pela inclusão dos serviços de armazenamento (CosmosDB e Blob Storage) como pela implementação de um sistema de cache (Redis). Concluindo assim que para além das vantagens habituais do desenvolvimento em Cloud, a incorporação destes serviços proporcionou um impacto altamente positivo em termos do rendimento do sistema.