

Sistemas Distribuídos

Projeto TP2 (versão: 16/Maio)

Engenharia Informática

Ano lectivo: 2018/2019, 2º Semestre

Prazo

2º Trabalho - 5 de Junho, 23h59 (online - código + relatório/formulário)

Contexto

No primeiro trabalho algumas premissas subjacentes ao desenvolvimento da solução foram deliberadamente simplificadas. Em particular, o modelo de falhas dos componentes adoptado não reflete a realidade pois, além da rede, é expectável que os próprios componentes falhem. Por outro lado, foram ignorados as questões relativas à persistência da informação e os requisitos de segurança da plataforma.

Objetivo

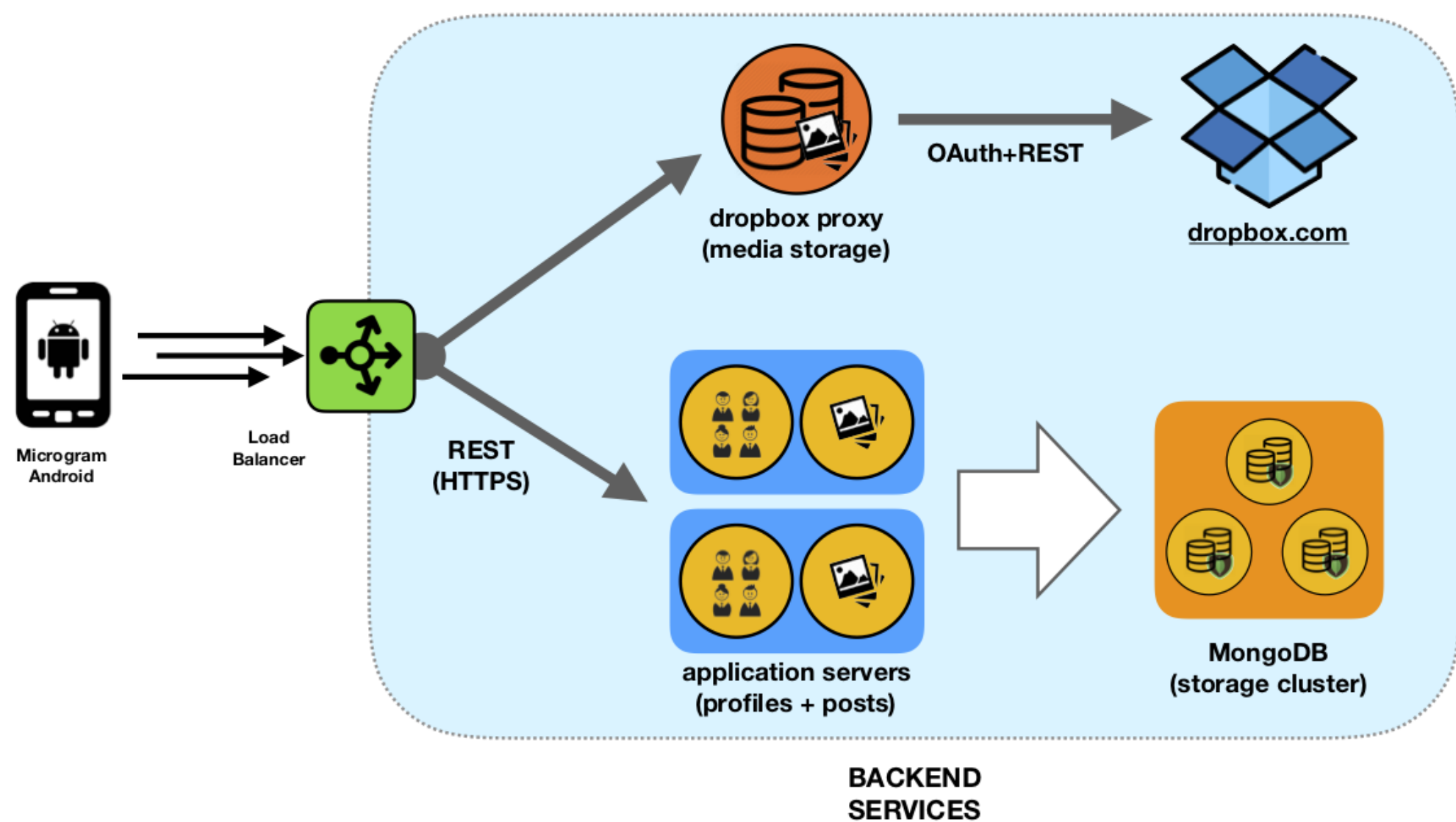
O trabalho 2 tem como objetivo tornar o sistema desenvolvido, anteriormente, num sistema mais confiável (*dependable*).

Ou seja, o foco do trabalho será melhorar a solução realizada, de modo a que o utilizador possa confiar que a plataforma continuará operacional e disponível, mesmo havendo falhas nos componentes. Adicionalmente, pretende-se que os clientes da plataforma estejam protegidos da indiscrição de agentes externos e que possam certificar-se que dialogam com entidades reconhecidas. Finalmente, os dados dos utilizadores serão guardados de forma persistente.

Arquitetura

A infra-estrutura da aplicação móvel Microgram, estará organizada numa arquitetura em camadas. Considerando a aplicação móvel a *camada de apresentação*, o trabalho terá como alvo, exclusivamente, as duas camadas seguintes: a camada aplicacional e a camada de armazenamento - o denominado *backend* da aplicação móvel, constituído pelos serviços que dão suporte às funcionalidades da aplicação.

A figura seguinte fornece uma visão global de alto-nível da arquitetura da aplicação reformulada.



Alterações à arquitetura

- MediaStorage
 - Passa a ser um *proxy* para a Dropbox. As imagens serão carregadas para uma conta Dropbox, em vez de armazenadas localmente pelo serviço;
- MongoDB - serviço replicado de armazenamento;
 - Usado para persistir os dados da aplicação (perfis de utilizadores e publicações). As imagens continuam a ser guardadas via o serviço MediaStorage.
- Profiles + Posts
 - Instanciados **conjuntamente** em cada servidor backend, continuando com endpoints REST individualizados;
 - Sem estado próprio. Recorrem ao MongoDB para as suas necessidades de armazenamento.
 - Interação entre os serviços Profiles e Posts **índireta** através da camada de armazenamento, ou por **partilha de memória**;
- Comunicações seguras
 - Invocações REST sobre HTTPS, com autenticação dos servidores via certificados.

Interfaces de programação

Os serviços mantêm a mesma especificação, mas a API SOAP **não será utilizada**.

A especificação dos serviços de *backend* continua disponível no [repositório](#) externo.

- [microgram.api](#) - Definições de Profile e Post;
- [microgram.api.java](#) - Documenta de forma genérica a semântica das operações e os erros a reportar;
- [microgram.api.rest](#) - Documenta a API REST dos serviços de backend;

Auto-configuração

Sem alterações face à versão original.

Os serviços deverão continuar a anunciar os seus endpoints via o serviço de descoberta.

Requisitos da solução

Premissas gerais

A componente relativa à segurança do segundo trabalho centra-se na utilização invocações REST seguras e no acesso autenticado à Dropbox. Todos os serviços deverão oferecer endpoints REST HTTPS. O serviço MediaStorage deverá oferecer a sua API REST em HTTP e HTTPS.

Os serviços Post e Profiles deixam de usar máquinas dedicadas, passando a ser instanciados em conjunto na mesma máquina (ie., partilhando o mesmo endereço IP). Poderão partilhar o mesmo porto e executar no mesmo processo. Alternativamente, poderão executar em processos separados (ie., com portos distintos).

Os serviços Post e Profiles poderão interagir indiretamente por via da camada de armazenamento. Ou seja, **podem manipular os mesmos dados** armazenados na base de dados, a qual será partilhada. Se executarem no mesmo processo (ie., na mesma JVM), poderão também comunicar diretamente por invocação de métodos locais e partilha de memória.

Os servidores poderão falhar (fail-stop). Havendo redundância, o sistema deverá continuar disponível.

Requisitos mínimos (max: 9.5 valores)

- Comunicações seguras** - O serviços expõem a sua API REST sobre HTTPS, com certificados dedicados;
- MediaStorage Dropbox** - O serviço MediaStorage implementado por recurso à Dropbox;
- Dados Persistentes** - A informação de um dos serviços (Profiles ou Posts) guardada de forma persistente na camada de armazenamento (MongoDb).

Requisitos base (max: 14 valores)

- Funcionalidade completa** - Serviços de *backend* REST funcionais e integrados, suportados pela camada de armazenamento partilhadae replicada;

A percepção dos utilizadores da aplicação Microgram é a de um sistema coerente e integrado. As ações dos utilizadores repercutem-se no sistema de acordo com as expetativas, incluindo na evolução das estatísticas associadas aos dados (*Posts* e *Profiles*)

O sistema tolera falhas dos servidores aplicacionais, quando instanciados em múltiplas máquinas e partilhando a camada de armazenamento replicada.

Nota: Os requisitos base têm como pressuposto os requisitos mínimos;

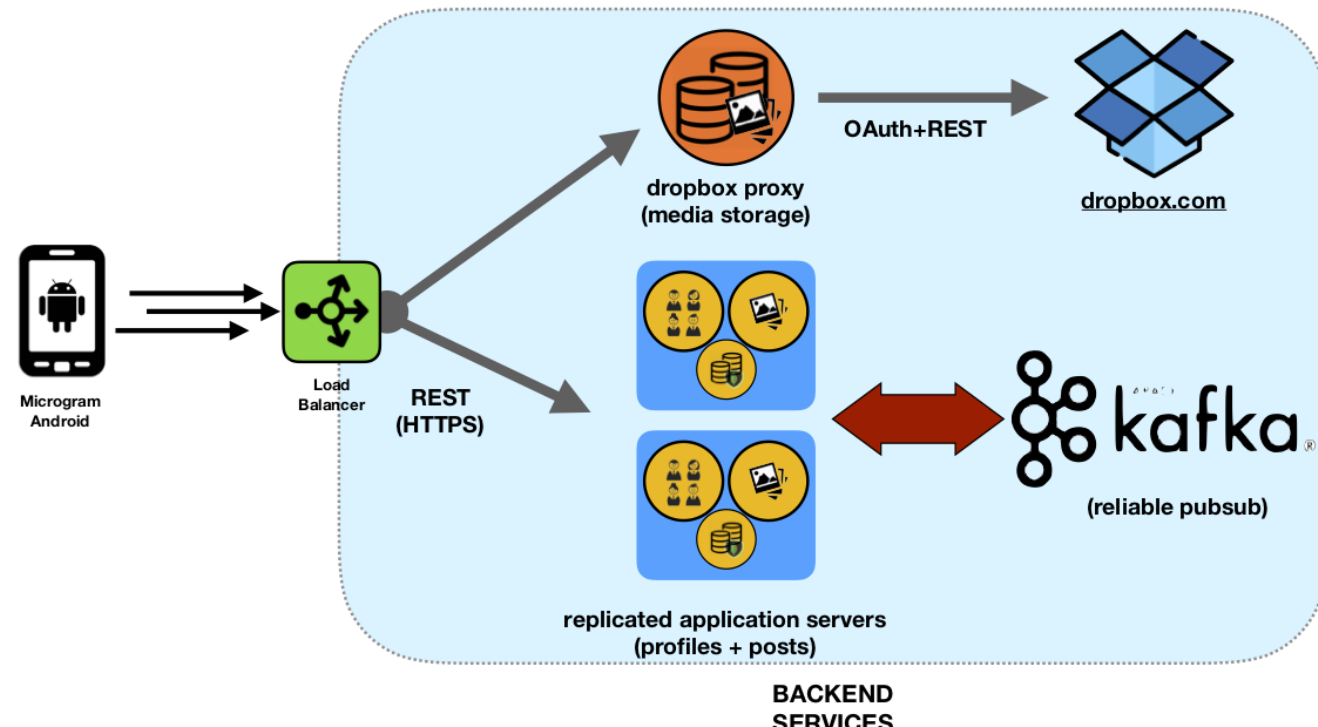
Elementos valorativos (max: 20 valores)

Serviços Replicados

Os serviços **não recorrem ao armazenamento externo replicado**. Implementam entre eles uma solução de replicação, suportada por armazenamento persistente local (MongoDB), como se indica na figura abaixo.

- Replicação sobre comunicação em grupo, com ordem total (leituras e escritas) (max: 2 valores)
- Replicação sobre comunicação em grupo, com ordem total (apenas escritas) (max: 4 valores)
- Replicação sobre comunicação em grupo, sem ordem total (max: 6 valores)

Nota: Para execução dos elementos valorativos é fornecido um [esboço da solução replicada](#), implementada sobre o sistema Kafka. (Este esboço pode ser adaptado ao trabalho previamente realizado, ou pode ser utilizado para a execução integral do trabalho.)



Fatores depreciativos

- O código entregue deverá seguir boas práticas de programação. A repetição desnecessária de código, inclusão de constantes mágicas, o uso de estruturas de dados inadequadas, etc., poderá incorrer numa penalização.
(**max: 2 valores**)
- Falta de robustez e comportamentos erráticos da solução são motivo de penalização. (**max: variável**)

Execução

O trabalho pode ser executado em grupo, de 1 ou 2 alunos. Os alunos do mesmo grupo não precisam de pertencer ao mesmo turno prático.

A composição do grupo pode ser diferente da do trabalho anterior.

Avaliação

A avaliação do trabalho terá em conta os seguintes critérios:

- Funcionalidades desenvolvidas e a sua conformidade com a especificação, tendo como base os resultados da bateria de testes automáticos;
- Qualidade da solução, podendo incluir uma avaliação de desempenho;
- Qualidade do código desenvolvido.

A classificação final do aluno é individual e será menor ou igual à classificação do trabalho, em função dos resultados obtidos na discussão individual, a realizar no final do semestre.

Bateria de testes

Oportunamente, será disponibilizada uma bateria de testes destinada a verificar a conformidade da solução com a especificação.

A solução deverá ser compatível com Java 8 (1.8), sob pena de poder falhar os testes.

Aplicação Microgram + Proxy Load Balancer

Estes componentes não são necessários para resolver o trabalho em pleno.

Devido à política restritiva de gestão de tráfego da rede WIFI dos laboratórios, a aplicação android Microgram e o Proxy Load-balancer só serão disponibilizados por pedido expresso do grupo.

Material de apoio

A realização do trabalho poderá recorrer ao material de apoio fornecido disponível neste [repositório](#). A utilização deste código, no todo ou em parte, é opcional. Este repositório contém um esboço de uma solução de replicação dos serviços Posts e Profiles, mas pode ser usado para realizar o trabalho em todas as suas fases.

Soluções do primeiro trabalho:

- [Solução 1](#).
- [Solução 2](#), com serviços desacoplados via Kafka; Post e Profiles particionados, por utilizador.

O segundo trabalho é avaliado de forma independente do primeiro. Não há qualquer penalização em utilizar o material de apoio fornecido, em todo ou em parte. Não há valorização por executar o segundo trabalho sobre a solução do primeiro que foi entregue.

O trabalho pode recorrer a outros componentes e bibliotecas já existentes e evitar re-inventar a roda.

Ambiente de desenvolvimento

Todo o material de apoio fornecido pressupõe que o desenvolvimento será em ambiente Linux e Java 8. A validação do trabalho por via da bateria de testes automática fará uso de tecnologia [Docker](#).

Na falta de um ambiente Linux nativo, recomenda-se o recurso a uma solução de virtualização, em particular, aos utilizadores Windows. Para tal, poderão utilizar software *opensource* [VirtualBox](#).

Entrega

A entrega do trabalho será feita através do preenchimento de um formulário Google Forms, dentro do prazo previsto.

O código da solução deverá ser depositado num repositório **privado** na plataforma GitHub, indicando no formulário de entrega o identificador do *commit* da versão final.

Será ainda necessário adicionar como colaborador do repositório, o utilizador **smduarte**.

Experiência de utilização

O vídeo abaixo captura um exemplo da experiência do utilizado ao usar a aplicação Microgram.

