

Desenvolvimento de uma aplicação para dois ambientes em nuvem e comparação das duas soluções tecnológicas

João Brito, M9984
Department of Computer Science
University of Beira Interior
Covilhã, Portugal
joao.pedro.brito@ubi.pt

André Martins, M10157
Department of Computer Science
University of Beira Interior
Covilhã, Portugal
andremar1000@gmail.com

Abstract—Este documento foi escrito com o objectivo de comparar duas plataformas de computação em *cloud* em termos de eficiência, eficácia e ocupação de tempo e recursos, face à mesma aplicação, desenvolvida para Android. As plataformas a serem utilizadas são a IBM Cloud e a Firebase. Serão feitas todas as comparações necessárias às duas plataformas. Tais comparações incluem os tempos de resposta entre a plataforma e a aplicação. A escolha da melhor plataforma para o caso de estudo considerado será, assim, empiricamente fundamentada.

I. INTRODUÇÃO

Com este projeto pretende-se analisar as semelhanças e as diferenças entre dois sistemas de computação em *cloud*, sendo estes a IBM Cloud e o Firebase. Para tal, foi desenvolvida uma aplicação para dispositivos Android, usando o Android Studio, que terá acesso a vários serviços de cada um dos sistemas anteriormente referidos. Assim, a aplicação em questão apresenta a capacidade de utilizar, alternadamente, o sistema de *cloud* Firebase ou IBM Cloud. Esta dualidade manifesta-se também nos resultados obtidos, sendo que a conclusão derivada apontará uma solução como sendo mais adequada para o caso de uso em questão.

II. VISÃO GERAL DA APLICAÇÃO

No âmbito do presente documento, foi desenvolvida uma aplicação móvel (denominada CloudIt), com as devidas adaptações aos dois ambientes em nuvem considerados. De forma geral, a referida aplicação propõe-se a oferecer funcionalidades e serviços semelhantes aos populares sistemas de armazenamento em nuvem (e.g. Google Drive, Dropbox, etc...). Como tal, permite ao utilizador a partilha de ficheiros, entre o armazenamento local e remoto, bem como a sua listagem (com alguma meta informação).

O layout da aplicação caracteriza-se por um ecrã principal onde são apresentados os nomes dos ficheiros guardados, flanqueados por um ícone sugestivo (no caso específico das imagens, surge uma prévia das mesmas) e um botão de *download*.

Junto à base do ecrã, apresenta-se o botão destinado ao *upload* de ficheiros, sendo despoletada uma rotina que permite

ao utilizador escolher um ficheiro do armazenamento local. Na secção dedicada à IBM Cloud, serão destacadas algumas dificuldades em relação à implementação da funcionalidade de *upload* (o *download* funciona bem e conforme esperado).

Um dos grandes propósitos da aplicação desenvolvida (e tendo em conta as adaptações necessárias a cada plataforma estudada) é o de atribuir ao dispositivo móvel a responsabilidade de interação com o utilizador, apresentação dos documentos e registo das ações despoletadas. Todas as tarefas de manuseamento e armazenamento são executadas na nuvem, sendo o resultado devolvido às rotinas locais.

Abaixo é apresentado o aspeto gráfico da aplicação descrita:

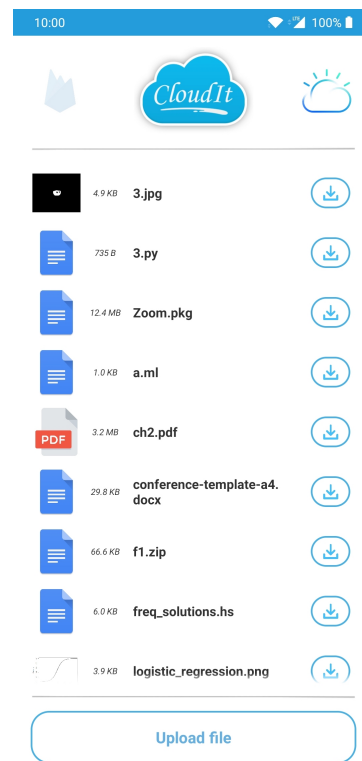


Fig. 1: Visão geral da aplicação proposta

No *layout* acima (fig. 1), é possível observar os aspetos anteriormente mencionados, bem como, os dois ícones de seleção da plataforma em uso. Neste caso, estava seleccionada a IBM Cloud, mas o comportamento é espelhado para a Firebase. A imagem pequena (prévia) que surge associada a imagens é um detalhe de design que permite uma rápida identificação do ficheiro. Ao clicar no botão de *download*, o ficheiro correspondente é descarregado da nuvem para o armazenamento local (pasta de transferências do dispositivo Android).

De notar que a informação relativa ao tamanho dos ficheiros foi obtida diretamente através do pedido HTTP de *download*. Tal procedimento irá ser descrito com maior formalidade na secção III-B.

III. IBM CLOUD

A. Principais Caraterísticas

A plataforma de computação *cloud* da IBM (nome adotado pela Bluemix a partir de 2017) apresenta vários serviços, dos quais se destacam os seguintes:

- Armazenamento de dados, através de tecnologias como o Cloud Object Storage [2] (que foi explorado no âmbito deste projeto);
- Inteligência Artificial, destacando-se o Watson Studio [3]
- Poder computacional, abrangendo uma ampla gama de necessidades e casos de uso.

Os serviços oferecidos podem ser acedidos e controlados através do painel de controlo (*dashboard*) intuitivo que existe na área pessoal de cada cliente. No caso específico da aplicação desenvolvida, optou-se por ativar o serviço Cloud Object Storage (COS) na sua configuração Lite [4]. Este plano permite, por exemplo, armazenar até 25 Gigabytes por mês e realizar, no máximo, 20 000 pedidos GET e 2000 pedidos PUT por HTTP.

A figura seguinte apresenta uma visão geral da interface específica ao Cloud Object Storage:

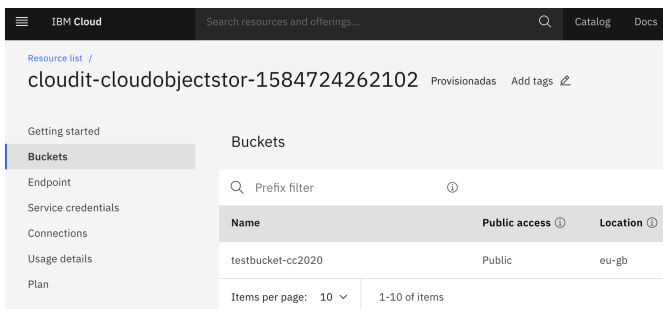


Fig. 2: Menus principais do COS

Esta página em específico apresenta um menu lateral onde se podem ver informações sobre os *endpoints* (servidores principais) em uso, o plano escolhido e, talvez a parte mais importante, os *buckets* (contentores) criados. Neles, encontram-se os ficheiros guardados, numa lista com alguma meta-informação, tal como o tamanho e a data da última modificação:

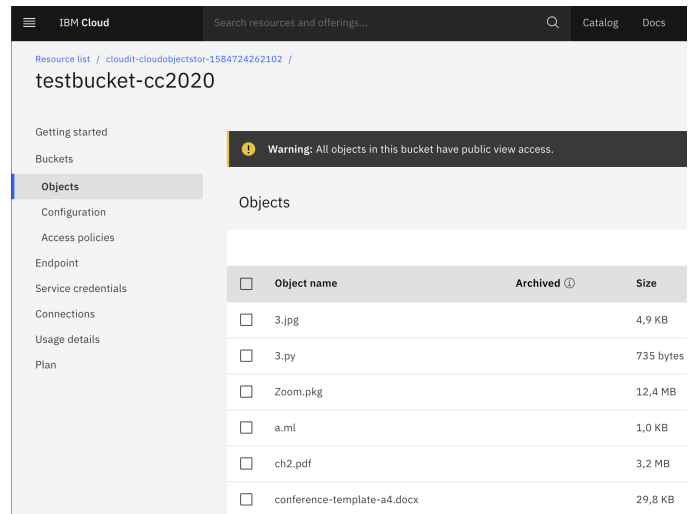


Fig. 3: Listagem dos ficheiros guardados no *bucket* "testbucket-cc2020"

Tal como destacado pela própria interface de utilizador, os ficheiros dentro do *bucket* estão marcados como sendo públicos. A seguinte política de privacidade permite dotar os ficheiros desta propriedade:

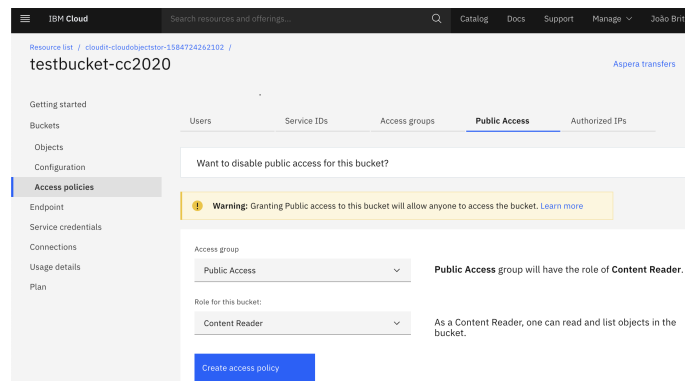


Fig. 4: Definição de uma nova política de acesso público

A caraterística de acesso público foi necessária para a rotina de descarregamento de ficheiros (feita com recurso a um pedido HTTP do tipo GET). A secção seguinte aborda a implementação desta rotina em maior detalhe.

B. Integração com CloudIt

A documentação oficial da IBM Cloud do SDK para Android [5] aponta desde logo para a possibilidade de fazer *uploads* e *downloads*, através de um URL com a seguinte forma: `https://endpoint/bucket/file_name`. Relembra-se que o *bucket* (e por consequência, os ficheiros nele contidos) devem usufruir de acesso público. A IBM Cloud facilita a exploração desta via ao devolver um ficheiro XML, com uma listagem de ficheiros guardados e respetiva meta-informação, quando o pedido HTTP só contém os campos *endpoint* e *bucket* (`https://endpoint/bucket`):

Também foi criada uma função de download de um ficheiro, onde a aplicação faz o download do ficheiro da cloud do

Firebase para um ficheiro temporário no armazenamento do dispositivo.

Primeiro, foi criada uma referência à cloud onde esse ficheiro está presente, sendo criado logo de seguida um ficheiro temporário com o nome desse mesmo ficheiro.

Por último, é iniciado o download do ficheiro, onde poderão ocorrer dois casos. O primeiro, onde o download foi um sucesso, aparecerá uma mensagem a indicar que o download ocorreu sem problemas. O segundo caso, em que poderá ter ocorrido alguma falha, aparecerá uma mensagem a afirmar que ocorreu algum erro durante o download do ficheiro.

Este sistema permite, na consola do desenvolvedor, ver vários aspetos relacionados à aplicação, neste caso sendo os dois mais importantes o *bucket*, ou seja, a funcionalidade Firebase Storage e o *dashboard*, onde estão presentes alguns dados relativos à aplicação.

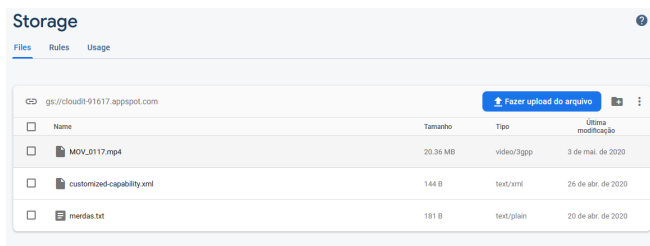


Fig. 6: Bucket da aplicação.

Na Figura 6 é visível todo o conteúdo presente no armazenamento da cloud, sendo que todos os ficheiros presentes foram carregados para a cloud usando a aplicação. Como também é visível, esta funcionalidade tem compatibilidade para os mais diversos tipos de ficheiros, podendo estes também ser do tipo ".mp4", ".txt" ou ".xml".

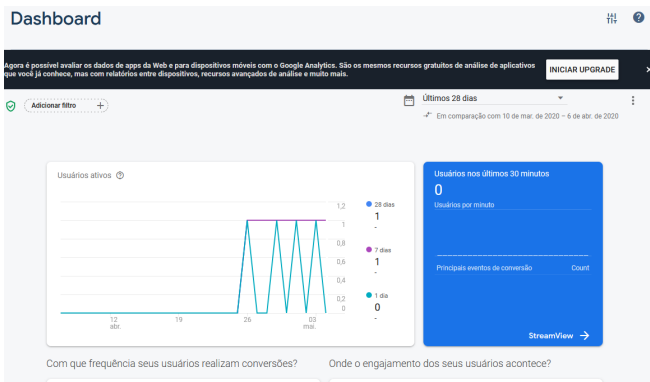


Fig. 7: Dashboard da aplicação.

Na Figura 7 é possível visualizar o *layout* do *dashboard* da aplicação, sendo que este poderá dar ao desenvolvedor uma visão geral da sua aplicação, sendo possível ver dados como o número de utilizadores que utilizaram a aplicação, quantos *downloads* ou *uploads* tiveram falhas, de onde são a maioria dos utilizadores, entre muitos outros dados que poderão ser úteis aquando da actualização da aplicação para outras fases.

V. RESULTADOS EXPERIMENTAIS

A. IBM Cloud

Os ficheiros carregados via *dashboard* para o armazenamento na nuvem compreendiam imagens, ficheiros comprimidos, PDFs, ficheiros com código de várias linguagens de programação e um documento Word. Esta panóplia variada de itens pretende representar os casos de uso típicos de utilizadores reais de um sistema como o que foi implementado. Os ficheiros mais leves (com alguns *bytes*) demoram menos que um segundo para serem descarregados, ao passo que ficheiros maiores (vários megabytes) podem demorar entre dois e três segundos. De forma mais teórica, o processo de *upload* representaria um tempo de espera da mesma ordem de grandeza.

No estado atual, a aplicação poderia servir de meio de comunicação entre um mestre (um utilizador com privilégios de acesso à *dashboard* online) e os servos/clientes (que só podem descarregar o conteúdo disponível).

B. Firebase

De todos os testes efetuados foram obtidos diversos resultados. Primeiramente, de todos os ficheiros carregados para a cloud, ou seja, cerca de 10 ficheiros, não houve qualquer tipo de falha ou ficheiro que não foi carregado.

Em relação ao tempo de *upload* dos ficheiros, foi calculado um tempo médio de 1.5 segundos, podendo então variar de acordo com o tipo e tamanho de um ficheiro. Para ficheiros de texto, com cerca de 100 KB, o *upload* demorava cerca de um segundo ou menos, enquanto que para imagens com cerca de dois MB poderia demorar entre os dois e três segundos.

A aplicação Android permite carregar qualquer tipo de ficheiro para a cloud, sejam imagens, ficheiros de texto, ficheiros binários ou ficheiros executáveis.

Em relação à funcionalidade de *download*, foram feitos vários testes ao download do ficheiro, onde em nenhuma das tentativas houve qualquer erro do *download*.

Quanto aos tempos obtidos, para um ficheiro de cerca de 20 MB, a aplicação demorou entre 15 a 20 segundos, podendo variar de acordo com a velocidade de acesso à internet.

Como não existem mecanismos de autenticação na aplicação, todos os utilizadores podem fazer *upload* e *download* de ficheiros de e para a mesma *cloud*, não havendo distinção entre si.

Por último, ao fim de um certo tempo sem utilizar a aplicação, a conexão à Firebase passa a um estado inactivo, o que leva a que haja uma menor carga de ligações de acesso à *cloud*.

C. Comparação das Tecnologias

O propósito da aplicação desenvolvida está bem patenteado no seu design. Os dois ícones no topo permitem escolher o protocolo Firebase ou o equivalente (com as devidas diferenças) da IBM. Ao longo do desenvolvimento deste projeto, uma diferença entre as duas tecnologias começou a materializar-se: o suporte ao ambiente Android e a comunidade de desenvolvedores. O primeiro ponto é particularmente inusitado dado

que existe suporte à linguagem Java na sua forma tradicional. Porém, o Android usa uma máquina virtual Java customizada, levando a que essa solução não seja exequível (leva a um erro de importação/inexistência de bibliotecas). O segundo ponto enunciado manifesta-se na falta de respostas a perguntas comuns em fóruns especializados e adesão/reação a discussões online. Em geral, é preciso mais tempo de pesquisa para se resolver um problema relacionado com o sistema da IBM no Android.

Por outro lado, o sistema Firebase é um sistema que está completamente adaptado para desenvolvimento Android, existe documentação em vários pontos da Internet, no próprio site da Firebase e, como aspeto realmente importante, apresenta documentação embutida no Android Studio, sendo este o ambiente de desenvolvimento usado para desenvolvimento de aplicações Android. Também apresenta uma comunidade bastante alargada, visto que é uma das tecnologias mais utilizadas para Android devido a todas as suas funcionalidades.

Assim sendo, e em jeito de conclusão, o sistema Firebase é um sistema que leva menos tempo ao desenvolvedor para procurar informação e começar a aplicar todas as funcionalidades à aplicação desenvolvida.

VI. CONCLUSÃO

Em suma, durante o desenvolvimento desta aplicação Android, pode-se concluir que o sistema Firebase é o melhor sistema a ser utilizado, tanto para os utilizadores, devido ao maior leque de funcionalidades que poderão estar presentes na aplicação, como para desenvolvedores, devido à facilidade de acesso à documentação como também do potencial de desenvolvimento das funcionalidades apresentadas.

REFERENCES

- [1] Documentação Firebase [Online] <https://firebase.google.com/docs?authuser=0>
- [2] IBM Cloud Object Storage [Online] <https://www.ibm.com/cloud/object-storage>
- [3] IBM Watson Studio [Online] <https://www.ibm.com/cloud/watson-studio>
- [4] Plano Lite Cloud Object Storage [Online] <https://cloud.ibm.com/catalog/services/cloud-object-storage>
- [5] SDK da IBM Cloud para Android [Online] https://cloud.ibm.com/docs/cli?topic=cloud-cli-sdk#android_sdk