

Sistemas de Computação Móvel e Ubíqua

2º teste – 18/6/2022

Duração: 1h30

Nº _____ nome _____

1. Indique se cada uma das afirmações seguintes é verdadeira (V) ou falsa (F):

___ O desenvolvimento para dispositivos Android ou iOS usando tecnologias Web, permite tornar as aplicações independentes da plataforma, oferecendo interfaces nativas de melhor qualidade, com melhores tempos de resposta e menor uso de recursos.

___ Nos dispositivos móveis, a abordagem de desenvolvimento considerando "local-first for user data", permite, no limite, dispensar a necessidade de ligação rede/internet em algumas aplicações.

___ *Mobile Backend as a Service* (MBaaS) compreende as ofertas de serviços para auxiliar o desenvolvimento para móveis, que incluem normalmente serviços de autenticação de utilizadores, de base de dados, de notificações, entre outros.

___ Nos ambientes IoT, consideram-se as plataformas organizadas em camadas onde, tipicamente, temos uma "Perseption Layer" que se refere aos componentes como, por exemplo, sensores de temperatura ou luz, ou interruptores para ligar ou desligar outros aparelhos controlados pelo IoT.

2. Considere as aplicações do tipo *Progressive Web App* (PWA).

a) Quais as principais tecnologias usadas na sua plataforma de desenvolvimento?

b) Como é possível a estas aplicações terem dados persistentes (*storage*) no dispositivo cliente?

3. O *caching* de dados nos sistemas móveis, pode melhorar a resposta da aplicação e poupar no uso de energia.

a) Explique como é possível obter tais melhorias.

b) Indique possíveis desvantagens ou problemas relacionados com o uso dos dados em cache.

c) Apresente estratégias de gestão dos dados para aliviar os problemas anteriores.

4. Considere uma aplicação móvel que pretende ter uma cópia local de alguns dos dados residentes num serviço cloud para suportar algumas funcionalidades quando desconectada (sem rede). Considere que o utilizador necessita de informação relevante para a sua localização (por exemplo informação sobre o local onde este se encontra).

a) O utilizador sabe à partida qual o local onde vai precisar da informação e que nesse local não tem rede. Proponha como gerir estes dados na aplicação, usando a técnica de *pre-fetching*, indicando como obter os dados necessários, quando e quais os dados a obter.

b) Se o utilizador puder atualizar esses dados quando se encontra desconectado, será possível ter consistência forte entre a cópia no móvel e os dados no serviço cloud? E consistência fraca? Justifique.

5. Num sistema compreendendo dispositivos pervasivos IoT, podemos chegar a uma escala que coloca problemas à infraestrutura e aos recursos na cloud, assim como à experiência do utilizador.

a) Descreva esses problemas.

b) A *Edge/Fog Computing* pode responder a alguns desses problemas. Explique as principais características da arquitetura, quais as vantagens obtidas e como estas permitem minorar os problemas anteriores.

6. Os dados recolhidos nas plataformas IoT correspondem muitas vezes a séries temporais (*time series*) que estão continuamente a ser guardadas em bases de dados especializadas.

a) Indique as características das escritas e leituras para as quais estas bases de dados são otimizadas e porquê.

b) Os dados mais antigos podem perder a relevância para as análises efetuadas. Indique como este facto pode ser usado para reduzir o volume de dados nestas bases de dados.

7. Considere que num sistema de dados replicados se pretende uma CRDT do tipo conjunto ordenado (SortedSet). Este mantém um conjunto de elementos ordenado alfabeticamente e oferece as seguintes operações:

Add(*e*) – acrescentar *e* ao conjunto, na ordem correta;

Rm(*e*) – remover *e* do conjunto (se não existir não faz nada);

First() – obter o 1º elemento do conjunto

a) Este tipo de estrutura de dados oferece consistência forte ou fraca? Justifique.

b) Nesta CRDT não há problemas em suportar operações Add concorrentes. Justifique.

c) Num dado instante, a operação First pode devolver valores diferentes em diferentes réplicas? Justifique.

d) Assumindo uma semântica concorrente onde o Rm ganha (“Rm wins”), indique o conteúdo do conjunto S após a sincronização de todas as seguintes operações efetuadas, respetivamente, nas réplicas A e B (de início S é vazio):

A: S.Add(“baa”); S.Add(“zee”);

B: S.Add(“tee”); S.Rm(“zee”); S.Add(“baa”);