

# Sistemas Distribuídos — Trabalho Prático

## Alarme Covid

Engenharia Informática  
Universidade do Minho

2020/2021

### Informações gerais

- Cada grupo deve ser constituído por até 4 elementos.
- Deve ser entregue o código fonte e um relatório de até 6 páginas (A4, 11pt) no formato PDF.
- O trabalho deve ser entregue até às 23:59 do dia 25 de janeiro de 2021 no *eLearning*.
- A indicação da preferência para a data/hora da apresentação é feita também até às 23:59 do dia 25 de janeiro de 2021.
- No dia 26 de janeiro é publicado o calendário das apresentações.
- A apresentação do trabalho ocorrerá nos dias 27 a 29 de janeiro de 2021 por videoconferência e na hora marcada.

### Resumo

A CoViD-19 tem a particularidade de ser altamente contagiosa num período de alguns dias em que o doente é pré-sintomático (i.e., ainda não apresentou sintomas) e não suspeita que já está doente, pelo que mantém contactos sociais. Quando a doença se manifesta ou é identificada num teste, é pois importante que os contactos de risco do doente durante esses dias sejam identificados, no sentido de fazerem isolamento profilático ou serem testados. Ao mesmo tempo, de forma a evitar contactos desnecessários, um cidadão pode minimizar a eventualidade de uma contacto de risco evitando deslocar-se para locais em que há concentração de pessoas. Têm sido propostas variadas aplicações para telemóveis que auxiliam os cidadãos e os serviços de saúde nestas tarefas, tais como a STAYAWAY COVID.<sup>1</sup>

Neste projeto pede-se a implementação de uma plataforma inspirada no problema do rastreio de contactos e deteção de concentração de pessoas sob a forma de cliente/servidor em Java utilizando *sockets* e *threads*. De forma a simplificar esta tarefa não se considera importante a privacidade dos dados pessoais, que podem livremente ser comunicados e armazenados no servidor. O mapa é apenas uma grelha de  $N \times N$  locais, sendo as coordenadas geográficas pares discretos de índices. Considera-se também que quando um utilizador comunica uma infeção, poderá ter estado contagioso desde que fez o registo no sistema não sendo por isso necessário registar datas.

---

<sup>1</sup>O modo de funcionamento da STAYAWAY COVID é muito diferente do proposto neste trabalho de forma a garantir a segurança e privacidade dos dados pessoais e está descrito aqui: <https://stayawaycovid.pt/funcionamento/>.

## Funcionalidade Básica

Este serviço deverá suportar a seguinte funcionalidade básica:

- Autenticação e registo de utilizador, dado o seu nome e palavra-passe. Sempre que um utilizador desejar interagir com o serviço deverá estabelecer uma conexão e ser autenticado pelo servidor.
- Manter o servidor informado sobre a localização atual do utilizador.
- O utilizador saber o número de pessoas numa dada localização.
- O utilizador saber quando não houver ninguém numa dada localização, com o intuito de se vir a deslocar para lá.
- Comunicação ao servidor de que o utilizador está doente. Depois da comunicação este utilizador não poderá mais interagir com o servidor, pois assume-se que está em isolamento.

## Funcionalidades Adicionais

A resposta será valorizada pelas seguintes funcionalidades adicionais:

- O servidor deve notificar todos os utilizadores potencialmente contagiados por um doente: que tenham estado na mesma localização geográfica no passado, ou seja, desde que se tenham cruzado.
- Um utilizador com autorização especial poderá descarregar um mapa indicando quantos utilizadores e quantos doentes (i.e., utilizadores que mais tarde comunicaram uma infeção) visitaram cada localização.

## Cliente

Deverá ser disponibilizado um cliente que ofereça uma interface com o utilizador que permita suportar a funcionalidade descrita acima. Este cliente deverá ser escrito em Java usando *threads* e *sockets* TCP.

## Servidor

O servidor deverá ser escrito também em Java, usando *threads* e *sockets* TCP, mantendo em memória a informação relevante para suportar as funcionalidades acima descritas, receber conexões e input dos clientes, bem como fazer chegar a estes a informação pretendida. O protocolo entre cliente e servidor deverá ser num formato binário, através de código desenvolvido no trabalho, podendo recorrer apenas a `Data[Input|Output]Stream`. Para o serviço não ficar vulnerável a clientes lentos, não deverá ter *threads* do servidor a escrever em mais do que um socket, devendo as escritas serem feitas por threads associadas a esse socket.