

---

## Computação Móvel 2014/2015

---

### **METI - TAGUS** **grupo 14**

Patrícia Semedo

79428

patricia1990@live.com.pt

Pedro Braz

73991

pbraz.93@gmail.com

Rui Pereira

70600

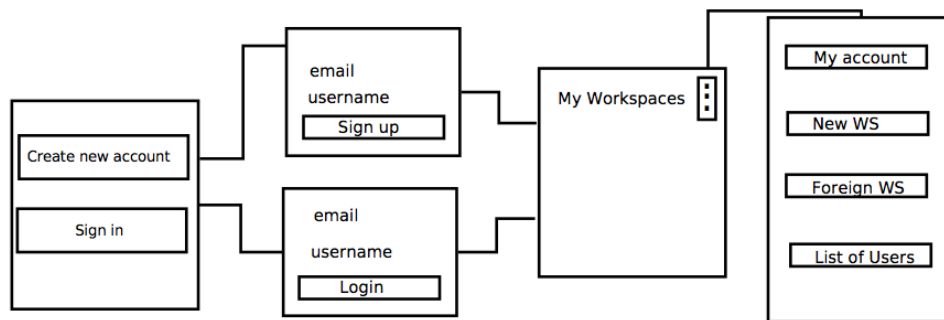
ruijosemangas@gmail.com

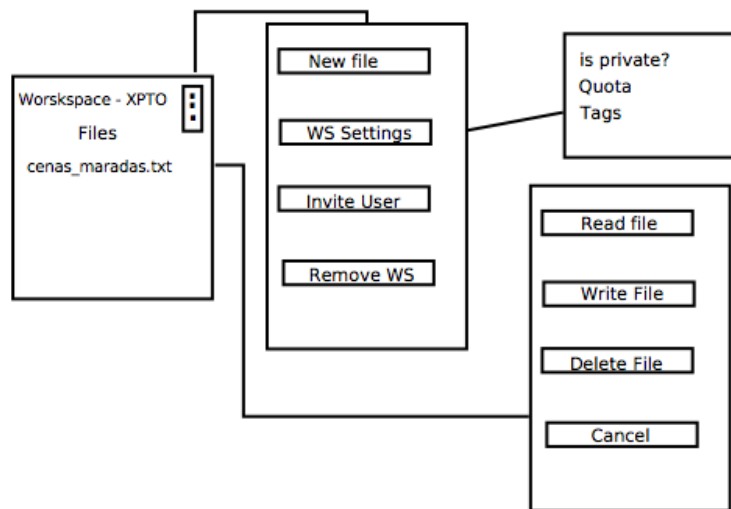
## 1 OBJECTIVOS ALCANÇADOS

Version	Feature	Fully / Partially / Not implemented?
S-Version	Maintain locally owned workspaces	Fully
	Manage workspace files	Fully
	Enforce quota restrictions	Fully
	Invite local user for owned workspace (mirroring)	Fully
N-Version	Set up WiFi Direct networks	Fully
	Workspace sharing by invitation	Fully
	Workspace sharing by subscription	Fully
	Shared operations for file and workspace management	Fully
A-Version	A1. Server-backed storage	Not Implemented
	A2. Offline operation	Not Implemented
	A3. Security	Partially
	Other	Not Implemented

Em relação à parte avançada da implementação do projecto, desenvolvemos um mecanismo de segurança durante as escritas nos ficheiros. Este mecanismo permite que um utilizador, fazendo-se passar por outro, não consiga escrever no ficheiro.

## 2 ESPECIFICAÇÕES TÉCNICAS





### 3 DESENHO DA APLICAÇÃO

#### 3.1 PLATAFORMA ESCOLHIDA

O projecto, foi desenvolvido e testado para correr em dispositivos reais pois correr o projecto em simuladores é uma tarefa muito intensiva.

#### 3.2 ARQUITECTURA

Todo do estado da aplicação, é guardado de forma persistente numa base de dados relacional *SQLite*. Os ficheiros são guardados internamente no dispositivo móvel. A aplicação funciona de maneira distribuída usando o *Wi-Fi Direct* com o objectivo de haver partilha de informação entre diversos utilizadores na mesma rede. Na camada superior ao *Wi-Fi Direct* é corrido um servidor que aceita vários pedidos em simultâneo.

#### 3.3 PROTOCOLOS UTILIZADOS

Quando um utilizador entra na aplicação, liga-se a outros utilizadores que já estejam a usar a aplicação na tentativa de formar um grupo dentro da rede. Se o grupo já estiver formado, o novo utilizador contacta o *group owner* para obter informação acerca de todos os utilizadores do grupo. Na rede, os utilizadores partilham os nomes dos *workspaces* que possuem com todos os utilizadores do grupo.

Em relação ao *subscribe*, o utilizador tem duas opções: subscrever a todos os *workspaces* públicos fornecendo uma *query* vazia ou procurar por *workspaces* específicos com base em *tags*. O utilizador guarda o nome do *workspace* na sua base de dados. Se o dono do *workspace* estiver no grupo, ele pode aceder aos ficheiros. Se o dono apagar o *workspace*, esse deixa de ser propagado na rede e quem o tiver subscrito sabe que o *workspace* foi apagado.

Quando um *user* quer aceder a ficheiros dentro de um *workspace* remoto, é feito um pedido em *real time* ao dono do mesmo. Este responde com os nomes dos ficheiros disponíveis e a *view* da aplicação do utilizador inicial é actualizada. Todos estes pedidos são feitos em *background* na camada do servidor. Se um utilizador desejar aceder ao conteúdo de um ficheiro, é feito um pedido com a intenção do utilizador. Em ambos os casos (ler ou escrever), é necessária uma leitura da versão mais recente do ficheiro. A diferença é que se o utilizador quiser escrever, é criada uma sessão que impede escritas em simultâneo. Esta sessão termina quando o utilizador que escreve aceita ou cancela as alterações de escrita.

A sessão referida anteriormente, é bloqueada pelo dono do ficheiro que entrega uma chave utilizador para ser devolvida quando as alterações terminarem. Se a chave for a mesma que o dono entregou inicialmente, quer dizer que foi o *user* correto a submeter as alterações. Após esta verificação, o dono faz *commit* ao ficheiro.

Tentámos minimizar a latência na rede diminuindo a quantidade de informação que é transmitida. Os ficheiros são transferidos *on demand*.

## 4 IMPLEMENTAÇÃO

O *churn* pode provocar anomalias na comunicação entre utilizadores. Uma das limitações é ser preciso estar na actividade *MyWorkspacesActivity.java* para a ligação entre utilizadores começar. Se por algum motivo externo (e.g. falha na rede), a ligação entre utilizadores falhar é preciso voltar à actividade anteriormente mencionada para começar a ligação de novo. Outro dos problemas associados à implementação em dispositivos reais é a dificuldade em testar o projecto em dispositivos com diferentes versões do Android. Relativamente ao *Wi-Fi* o ideal era que a ligação fosse transparente. Contudo, não é possível detectar quais os dispositivos que estão a correr a nossa aplicação. Daí, a ligação ter de ser feita manualmente. Outra limitação do nosso projecto é que na parte distribuída não existe qualquer distinção entre *workspaces* públicos e privados.

A plataforma Android tem disponível diversos mecanismos de armazenamento persistente. Optámos por guardar a maioria dos dados numa base de dados *SQLite*. A base de dados permite-nos procurar registos eficientemente e alterá-los com facilidade. Além disso permite fazer *queries* por um campo específico da tabela.

Para manter o estado do último utilizador na aplicação, utilizámos *shared preferences* pois para esta operação um mecanismo *key -> value store* é o suficiente.

Os ficheiros são guardados no sistema de ficheiros do telemóvel (*internal storage*).

## 5 CONCLUSÃO

Em suma, o nosso projecto tem margem para algumas melhorias. Em relação às partes avançadas começamos a implementar a parte da segurança contudo ficou inacabada. Queríamos ter desenvolvido um protocolo mais coeso que fornecesse uma maior segurança dos dados trocados na rede. Para trabalho futuro podíamos desenvolver as outras componentes avançadas do projecto e uma interface gráfica melhor.

Em relação à componente prática da cadeira julgamos que o *Wi-Fi direct* devia ser introduzido mais cedo. Além disso o código desenvolvido na primeira parte do projecto foi muito alterado para conseguir suportar a componente distribuída. Na nossa opinião a segunda parte devia ser a implementação de um novo módulo que não entrasse em conflito com a primeira parte.