 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Live Data</p>
---	--

Tema: Live Data

Objetivos:

- Execução de tarefas em *background*
- Utilização da Internet para obtenção de recursos externos (ex. download de ficheiros)

PARTE 1

Exercício 1

Utilizando o IDE Android Studio deve criar um projeto baseado num clone do repositório <https://github.com/fasiPP/ViewModelDemonstration.git>. Utilizando a aplicação implementada no repositório como ponto inicial, deve fazer a ligação entre os diversos elementos utilizando os conceitos *View Model* e *Live Data* lecionados nas aulas teóricas.

O resultado final deve ser semelhante ao representado na figura 1.

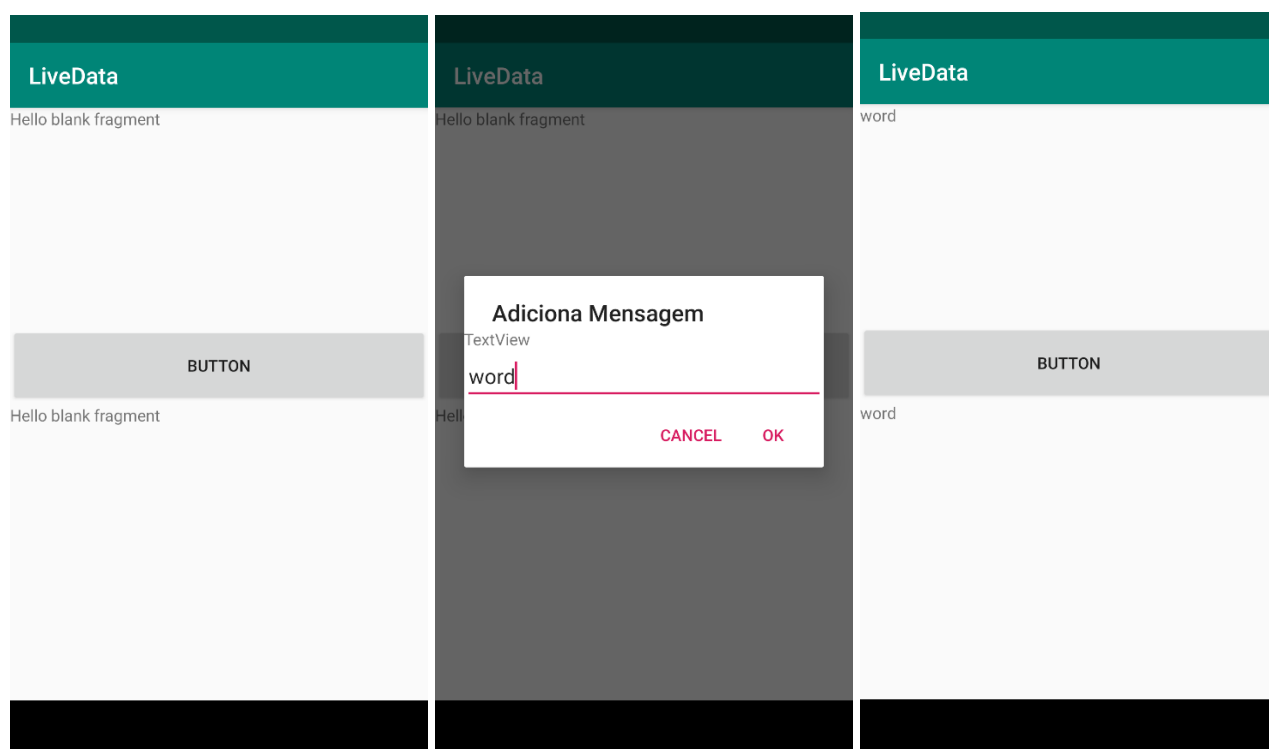


Figura 1 -Exemplo de funcionamento final da aplicação usando os conceitos de Live Data

Nota: A aplicação presente o repositório serve apenas para iniciar o desenvolvimento deste exercício.

<p>P.PORTO</p> <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Live Data</p>
---	--

Exercício 2

Utilizando o IDE Android Studio deve criar um projeto baseado num clone do repositório <https://github.com/fasIPP/NotesApp.git>. Esta aplicação, representa o ponto de partida para uma aplicação que guarda notas tiradas com um smartphone. As notas são persistidas numa base de dados baseada na biblioteca *Room* e apresentadas recorrendo a uma *list view*.

O objetivo deste exercício é atualizar a aplicação e aplicar o conceito de *Live Data* utilizando para o efeito *View Models*, o padrão observable. A aplicação deve no final ter cada uma das camadas representadas na figura 2.

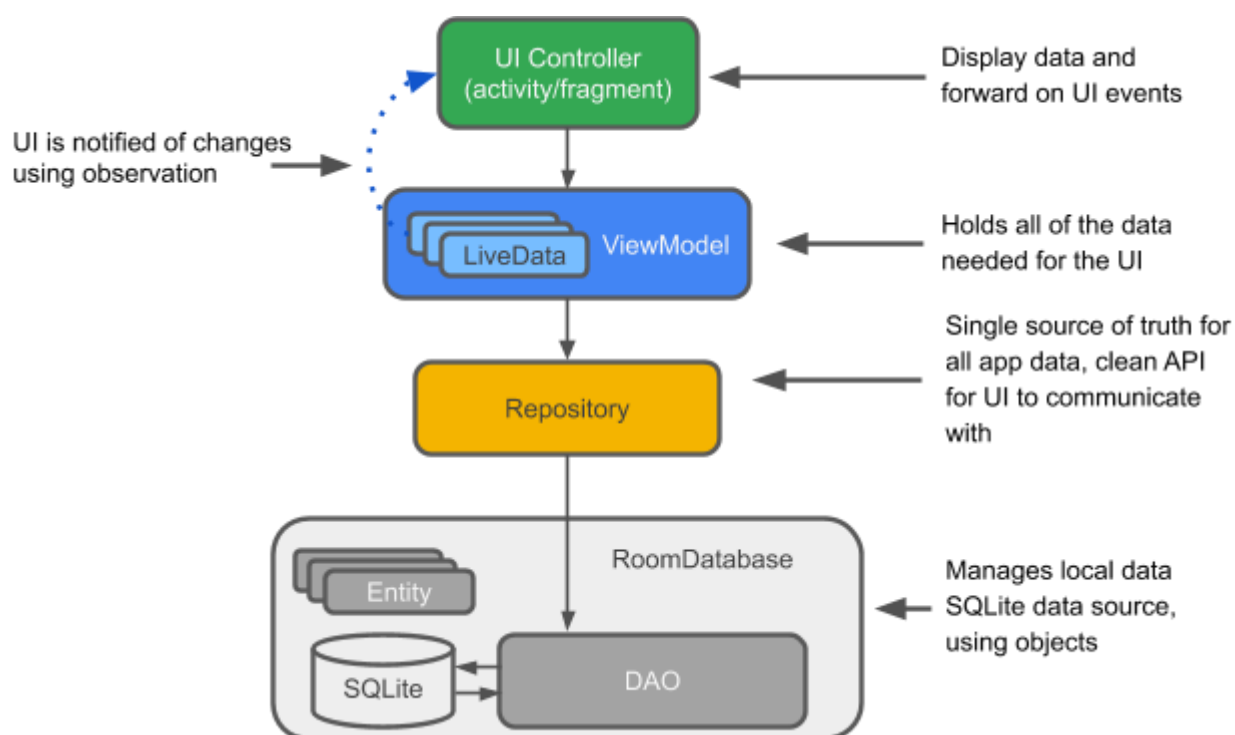



Figura 2- Arquitetura de uma aplicação com LiveData e acesso à base de dados (imagem retirada de <https://codelabs.developers.google.com/codelabs/android-room-with-a-view>)

A aplicação fornecida deve ser usada como exemplo para o desenvolvimento deste exercício, podendo ser necessário inclusive complementar as funcionalidades já implementadas.

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Live Data</p>
---	--

PARTE 2

Questão 1


A classe ViewModel:

1. É opcional quando queremos implementar o conceito de live data em android
2. É usada para encapsular objetos e criar observables que irão notificar consumers das alterações efetuadas aos objetos encapsulados
3. É usada para encapsular items numa recycler view
4. Nenhuma das anteriores

Questão 2

Usando a base de dados Room e o conceito de live data:

1. Devemos criar a noção de repositório e usar AndroidViewModel como a view models dos objetos monitorizados pela UI Thread
2. Temos de estender a classe viewmodel para cada query à base de dados
3. Não podemos usar o padrão observable com queries à base de dados
4. É útil para monitorizar alterações a queries na base de dados de forma automática através de callbacks via observables

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Live Data</p>
---	--

Questão 4

Que permissões é necessário fornecer à aplicação para que se conecte à Internet?

- android.permission.CONNECTIVITY
- android.permission.INTERNET
- Não é necessário nenhuma permissão especial, pois todas as aplicações Android têm permissão para se conectar à Internet.