



TRABALHO PRÁTICO

Relatório de Execução do Trabalho Prático

Licenciatura em Engenharia Informática
Arquiteturas Moveis
2023/2024

Trabalho realizado por:
Francisco Reis – 2019149992
João Castro – 2019128258
Quevin Moderno – 2019135563

1. Índice

Conteúdo

1. Índice	1
2. Introdução.....	2
3. Criação e gestao de Locais de interesses.....	3
4. Armazenamento de Localização	4
5. Amostra da Localização no maps.....	5
6. Funcionalidades Implementadas	6
7. Conclusão.....	7

2. Introdução

No âmbito da unidade curricular de Arquiteturas Móveis, foi-nos pedido o desenvolvimento de uma aplicação móvel Android que facilite as visitas turísticas a diversas localizações, disponibilizando informação sobre os locais de interesse existentes nesses locais. Para o desenvolvimento desta aplicação android recorreu-se à linguagem Kotlin, linguagem esta aprendida ao longo da unidade curricular.

As tecnologias utilizadas incluem o Firebase Firestore para o armazenamento e gerenciamento dos locais de interesse, o Jetpack Compose para a construção da interface do usuário de forma declarativa, o FusedLocationProviderClient para acessar a localização do dispositivo, e o MediaStore para o gerenciamento de imagens e mídia.

3. Criação e gestão de Locais de interesses

Esta atividade do aplicativo Android utiliza o Firestore, uma base de dados oferecida pelo Firebase. Exibe os Locais de Interesse armazenados nessa base de dados remota.

Ao ser iniciada, a atividade carrega os dados dos locais por meio da função `fetchPlaces()`, uma função assíncrona que utiliza a API do Firestore para recuperar documentos da coleção 'Place'. Esses documentos são mapeados para objetos do tipo `Place`, contendo informações como nome, descrição, localização geográfica, categoria e imagem associada.

A representação visual dos locais é realizada por meio da `Composable Place()`, exibindo cada local como um item em uma lista vertical usando o `LazyColumn`. Dentro de cada item, são exibidos detalhes como o nome, descrição e, quando disponível, a imagem do local, obtida através de um URL fornecido pelo Firebase Storage.

A interação do usuário com os locais é permitida por meio de ações específicas. Por exemplo, se o usuário é o responsável pelo local, é exibido um ícone para eliminar que, ao ser acionado, remove o local do Firestore. Além disso, é possível votar na aprovação de um local, onde a alteração de status é refletida localmente na lista por meio da função `setCliqueNoBotao()`, alterando o estado do local para aceite ou rejeitado.

A ordenação da lista de locais por ordem alfabética é apresentada através de botões na interface, mas essa funcionalidade ainda não foi implementada completamente no código fornecido.

A classe principal, `AddInterestPlacesActivity`, possibilita a adição de locais de interesse aos usuários no aplicativo. Isso envolve a captura ou seleção de imagens, fornecimento de informações como título, descrição, categoria e localização geográfica, além do armazenamento desses dados no Firestore para uso posterior.

Portanto, esta atividade oferece uma visualização dinâmica e interativa dos locais de interesse, proporcionando ao usuário a capacidade de visualizar, interagir e gerir os dados provenientes do Firestore.

```
data class Place (  
    val description: String = "",  
    val name: String = "",  
    val useradded:String = "",  
    val category:String = "",  
    val location:String= "",  
    val approvals:Int = 0,  
    val latitude:Double = 0.0,  
    val longitude:Double = 0.0,  
    val canVote:Boolean = true,  
    val image:String = ""): Serializable
```

4. Armazenamento de Localização

Neste ponto o nosso programa utiliza um botão na pagina de criação de uma localização para ir buscar a latitude e longitude atual do telemóvel.

```
private fun getCurrentLocation() {  
    if (ActivityCompat.checkSelfPermission(  
        context: this,  
        Manifest.permission.ACCESS_FINE_LOCATION  
    ) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(  
        context: this,  
        Manifest.permission.ACCESS_COARSE_LOCATION  
    ) != PackageManager.PERMISSION_GRANTED  
    ) {  
        return  
    }  
    fusedLocationClient.lastLocation  
        .addOnSuccessListener { location: Location? ->  
        {  
            location?.let {  
                currentLocation = it  
  
                latitude = it.latitude  
                longitude = it.longitude  
                Toast.makeText( context: this,  
                    "Latitude: "+"$latitude"+ " Longitude:"+"$longitude", Toast.LENGTH_SHORT).show()  
            }  
        }  
        .addOnFailureListener { e ->  
        {  
            Toast.makeText( context: this,  
                "Error getting location", Toast.LENGTH_SHORT).show()  
        }  
    }  
}
```

5. Amostra da Localização no maps

Esta parte é feita recorrendo á biblioteca “com.google.maps.android.compose.GoogleMap”, que irá buscar á base de dados em Place, a latitude e longitude do local e adiciona um marker nessa posição.

```
val localizacion= com.google.android.gms.maps.model.LatLng(  
    place.latitude,  
    place.longitude  
)  
val localizacionState= MarkerState(position = localizacion)  
val cameraPositionState= rememberCameraPositionState(  
    position= CameraPosition.fromLatLngZoom(localizacion, zoom: 10f)  
)  
GoogleMap( modifier = Modifier  
    .fillMaxWidth()  
    .height(300.dp)  
    .padding(8.dp),  
    cameraPositionState=cameraPositionState) {  
    Marker(  
        state=localizacionState,  
        title= stringResource("Localizacion"),  
    )  
}
```

6. Funcionalidades Implementadas

	Feito	Não feito
Pesquisas, filtros, ordenamentos e visualização da informação	X	
Utilização de mapas	X	
Criação e gestão de locais	X	
Criação e gestão de categorias	X	
Criação e gestão de locais de interesse	X	
Processo de confirmação de nova informação	X	
Processo de remoção de informações	X	
Armazenamento e consulta dos dados a partir de um servidor/repositório partilhado	X	
Registo e autenticação de utilizadores	X	
Suporte para diferentes línguas	X	
Suporte para diferentes orientações de ecrã	X	
Ecrã de créditos	X	

7. Conclusão

Este projeto de desenvolvimento de aplicativo Android que permite o acesso a informações relativas a locais turísticos, revelou-se desafiador e esclarecedor. Ao explorar uma interface intuitiva com Jetpack Compose e a integração com o Firebase, criamos uma plataforma robusta para acesso e gestão de informações sobre diferentes locais de interesse.