

Document Tècnic: Projecte MindSpeak

1. Presentació del projecte MindSpeak

MindSpeak és una aplicació mòbil desenvolupada per a ajudar persones amb Trastorn de l'Espectre Autista (TEA) a expressar les seves emocions d'una manera còmoda i estructurada, alhora que proporciona eines de seguiment per als seus supervisors i familiars.

Objectius principals:

- Facilitar l'expressió emocional en persones amb TEA mitjançant una interfície intuïtiva i accessible
- Proporcionar un sistema de seguiment emocional per a supervisors i familiars
- Oferir recursos i exercicis per gestionar situacions d'estrès o d'ansietat
- Crear un registre històric d'emocions que permeti detectar patrons i possibles problemes

Rols d'usuari:

1. **Usuari (persona amb TEA):** Accedeix a una interfície simplificada per expressar emocions i utilitzar exercicis de relaxació
2. **Supervisor:** Pot crear nous usuaris, visualitzar alertes, consultar gràfics emocionals i assignar exercicis
3. **Familiar:** Similar al supervisor, pot monitoritzar l'estat emocional i assignar exercicis

El projecte respon a una necessitat social concreta: proporcionar un canal alternatiu de comunicació emocional per a persones amb TEA, que sovint tenen dificultats per expressar els seus sentiments verbalment o cara a cara. Aquesta eina no només facilita la comunicació, sinó que també permet la detecció precoç de problemes com ara depressió o ansietat.

2. Arquitectura tècnica

L'arquitectura de MindSpeak s'ha desenvolupat amb tecnologies modernes orientades a aplicacions mòbils natives d'Android.

Backend:

- **Firestore:** Base de dades principal en temps real, utilitzada per a:
 - Autenticació d'usuaris (email/contrasenya)
 - Emmagatzematge de dades d'usuaris (perfils, relacions supervisor-usuari)
 - Registre d'emocions i metadades associades
- **Storage:** Utilitzat específicament per a l'emmagatzematge d'arxius multimèdia:
 - Imatges associades a les emocions
 - Recursos visuals, vídeos i àudios dels exercicis de relaxació

Frontend:

- **Kotlin:** Llenguatge de programació principal, recomanat oficialment per Google per al desenvolupament d'aplicacions Android
- **Jetpack Compose:** Framework modern de UI declarativa que substitueix el tradicional sistema basat en XML
- **MVVM (Model-View-ViewModel):** Patró d'arquitectura que separa la lògica de negoci de la interfície d'usuari

Organització del codi:

```
MindSpeak/
├── app/
│   ├── src/
│   │   ├── main/
│   │   │   ├── java/cat/dam/mindspeak/
│   │   │   │   ├── firebase/      # Gestió de la connexió amb Firebase
│   │   │   │   ├── model/        # Models de dades i ViewModels
│   │   │   │   ├── ui/           # Components d'interfície d'usuari
│   │   │   │   │   ├── components/ # Components reutilitzables
│   │   │   │   │   ├── navigation/ # Sistema de navegació
│   │   │   │   │   ├── screens/    # Pantalles de l'aplicació
│   │   │   │   │   │   ├── familiar/ # Pantalles específiques per a familiars
│   │   │   │   │   │   ├── professor/ # Pantalles específiques per a professors
│   │   │   │   │   │   ├── shared/   # Components compartits entre pantalles
│   │   │   │   │   │   ├── supervisor/ # Pantalles específiques per a supervisors
│   │   │   │   │   │   └── user/     # Pantalles específiques per a usuaris
│   │   │   │   │   └── theme/       # Definició de temes i estils
│   │   │   │   ├── utils/         # Utilitats i configuracions
│   │   │   │   ├── MainActivity.kt # Punt d'entrada de l'aplicació
│   │   │   ├── res/              # Recursos (imatges, strings, etc.)
│   │   │   └── AndroidManifest.xml # Configuració de l'aplicació
│   └── ...
```

El sistema de navegació implementa diferents fluxos segons el rol de l'usuari, redirigint automàticament després del login a la interfície corresponent. L'arquitectura segueix el principi de separació per funcionalitats i responsabilitats, facilitant el manteniment i les futures ampliacions.

3. Funcionalitats clau de l'aplicació

Sistema d'expressió emocional

El nucli de l'aplicació és el sistema que permet als usuaris amb TEA expressar les seves emocions:

1. **Selecció d'emocions:** Interfície visual amb botons grans i colors distintius per a cada emoció (enfadat, feliç, por, trist, ansiós), acompanyats d'icones representatives.
2. **Valoració d'intensitat:** Escala de 1 a 5 estrelles que permet indicar la intensitat de l'emoció sentida.
3. **Descripció opcional:** Camp de text on l'usuari pot explicar la situació o el motiu de l'emoció.
4. **Adjuntar imatge:** Possibilitat d'afegir una imatge des de la càmera o galeria que representi la situació.
5. **Historial d'emocions:** Pantalla cronològica que mostra totes les emocions registrades, ordenades per data i amb la informació completa.

Gestió d'exercicis de relaxació

Els supervisors i familiars poden assignar exercicis personalitzats:

1. **Creació d'exercicis:** Interfície per crear nous recursos:
 - Definició de títol i descripció
 - Selecció del tipus de recurs (imatge, vídeo, àudio)
 - Càrrega del contingut multimèdia
 - Assignació a un o múltiples usuaris
2. **Visualització d'exercicis:** Els usuaris accedeixen a una llista dels exercicis assignats amb una interfície simplificada i intuïtiva.

Interfície adaptada per a usuaris amb TEA

L'aplicació incorpora principis d'accessibilitat específics:

- **Disseny previsible:** Estructura consistent i navegació intuïtiva
- **Contrast elevat:** Colors distintius per a cada emoció i funcionalitat
- **Elements visuals grans:** Botons i text de mida considerable
- **Navegació simplificada:** Menú inferior amb icones intuïtives
- **Instruccions clares:** Text directe i concís en totes les funcionalitats

Sistema d'alertes i seguiment

Per als supervisors i familiars, l'aplicació ofereix:

1. **Notificacions:** Alertes quan un usuari registra una nova emoció
2. **Historial detallat:** Accés a l'historial complet d'emocions de cada usuari
3. **Registres recents:** Visualització prioritzada de les entrades més recents

Gràfiques i estadístiques emocionals

El sistema analitza les dades emocionals i proporciona:

1. **Distribució d'emocions:** Gràfic de barres que mostra la freqüència de cada emoció en un període determinat
2. **Evolució temporal:** Visualització per setmanes que permet observar canvis en els patrons emocionals
3. **Detall d'intensitats:** Informació sobre la intensitat mitjana de cada emoció

4. Fragments de codi destacats

Autenticació i obtenció del rol d'usuari

```
// Iniciar sessió
suspend fun iniciarSessio(
    email: String,
    contrasenya: String,
    onSuccess: () -> Unit,
    onFailure: (String) -> Unit
) {
    try {
        // Connectar l'usuari via Firebase Auth
        auth.signInWithEmailAndPassword(email, contrasenya).await()
        onSuccess()
    } catch (e: Exception) {
        onFailure(e.message ?: "Error desconegut")
    }
}

// Obtindre el rol de l'usuari connectat
suspend fun obtenirRolUsuari(): String? {
    try {
        val userId = auth.currentUser?.uid ?: return null
        val documentSnapshot = db.collection(collectionPath: "Persona").document(userId).get().await()

        if (documentSnapshot.exists()) {
            return documentSnapshot.getString(field: "rol")
        }
    } catch (e: Exception) {
        println("Error al obtenir el rol: ${e.message}")
    }
    return null
}
```

Aquest codi gestiona l'autenticació d'usuaris mitjançant Firebase Authentication i recupera el rol assignat (usuari, supervisor o familiar) de la base de dades. La funció iniciarSessio utilitza corrutines de Kotlin (suspend) per gestionar les operacions asíncrones amb Firebase de manera elegant.

Sistema de navegació basat en rols

```
@Composable
fun NavigationHost(
    navController: NavHostController,
    emotionViewModel: EmotionViewModel,
    userViewModel: UserViewModel
) {
    NavHost(navController = navController, startDestination = "logo") {
        composable(route: "logo") { LogoPage(navController) }
        composable(route: "signup") { SignUp(navController) }
        composable(route: "homesupervis") { HomeSupervisorScreen(navController, userViewModel) }
        composable(route: "user_management") { SupervisorManagementScreen(navController, FirebaseManager() ) }
        composable(route: "user_assignment") { ... }
        composable(route: "upload"){ SupervisorResourceAssignmentScreen() }
        composable(route: "notis") { ... }
        composable(
            route = "user_emotions/{userId}",
            arguments = listOf(navArgument( name: "userId") { type = NavType.StringType } )
        ) { ... }
        composable(route: "problemas") { UserProblems(navController) }
        composable(route: "login") { Login(navController = navController, userViewModel = userViewModel, context = LocalContext) }
        composable(route: "homefamiliar") { HomeFamiliarScreen(navController, userViewModel) }
        composable(route: "homeprofessor") { HomeProfessorScreen(navController, userViewModel) }
        composable(route: "homeuser") { Inicio(navController, userViewModel) }
        composable(route: "exerciselist") { ExerciseList(navController, userViewModel) }
        composable(route: "resource_view/{resourceId}") { ... }
        composable(route: "emotions") { Emotions(navController) }
        composable(route: "exercise") { Exercises(navController) }
        composable(route: "settings") { SettingsUser(LocalCustomColors, navController, userViewModel) }
        composable(route: "history") { EmotionHistoryScreen(viewModel = emotionViewModel) }
        composable(route: "grafic") { EmotionStatsScreen() }
        composable(route: "emotionRating/{emotionType}") { ... }
    }
}
```

Aquest fragment mostra la implementació del sistema de navegació utilitzant Jetpack Compose Navigation, que defineix totes les pantalles de l'aplicació i les rutes corresponents. El codi gestiona la navegació condicional segons el rol d'usuari (mitjançant el `userRoleViewModel`) i també implementa la transferència de paràmetres entre pantalles, com es pot veure en la ruta `"emotionRating/{emotionType}"`.

Conclusió tècnica

El projecte MindSpeak demostra una implementació efectiva d'arquitectura moderna d'Android, amb decisions tècniques que prioritzen l'experiència d'usuari i l'accessibilitat per a persones amb TEA.

Entre els reptes tècnics principals destaquen:

1. La necessitat d'utilitzar dues plataformes de backend diferents (Firebase i Supabase) per superar les limitacions d'emmagatzematge d'imatges en el pla gratuït de Firebase.
2. El disseny d'una interfície que sigui alhora intuïtiva per a usuaris amb TEA i completa per a supervisors.
3. La implementació d'un sistema de navegació condicional basat en rols que mantingui la seguretat i privacitat de les dades.

Des d'una perspectiva de millora, el projecte podria evolucionar cap a:

- Implementació d'anàlisi predictiva per detectar tendències emocionals
- Integració amb assistents virtuals per proporcionar recomanacions personalitzades
- Desenvolupament d'una versió iOS utilitzant Kotlin Multiplatform
- Implementació d'exercicis interactius mitjançant realitat augmentada

MindSpeak representa un exemple excel·lent d'aplicació de la tecnologia per millorar la qualitat de vida de col·lectius específics, combinant tecnologies modernes amb un disseny centrat en l'usuari.