

## Registro de Prompts y Soluciones IA

### 1 Inyección de Dependencias con Dagger-Hilt en ViewModel

#### ◆ Prompt enviado a IA:

"¿Cómo puedo inyectar Room Database en un ViewModel usando Dagger-Hilt en Android? Actualmente, estoy llamando a `getDatabase()` en el ViewModel en lugar de inyectar `CiudadDao`. ¿Cuál es la mejor manera de hacer esto?"

#### ◆ Respuesta de IA:

"Para inyectar `CiudadDao` en el ViewModel, debes asegurarte de que Hilt proporciona una instancia del DAO en un `@Module`. Luego, el ViewModel debe usar `@HiltViewModel` y recibir `CiudadDao` en el constructor con `@Inject`."

#### ◆ Implementación en Código:

##### Corrección en `ModuloBaseDatos.kt` para proveer `CiudadDao`:

```
@Module
@InstallIn(SingletonComponent::class)
object ModuloBaseDatos {
    @Provides
    @Singleton
    fun provideDatabase(@ApplicationContext context: Context): AppBBDD {
        return Room.databaseBuilder(context, AppBBDD::class.java,
            "app_database").build()
    }

    @Provides
    fun provideCiudadDao(db: AppBBDD): CiudadDao {
        return db.ciudadDao()
    }
}
```

##### Corrección en `ClimaViewModel.kt` para recibir `CiudadDao` inyectado:

```
@HiltViewModel
class ClimaViewModel @Inject constructor(
    private val ciudadDao: CiudadDao,
    private val repositorioClima: ClimaRepository
```

```
) : ViewModel() { ... }
```

---

## 2 Error "navigation destination pantallaClima/Madrid is not a direct child of this NavGraph"

### ◆ Prompt enviado a IA:

"Mi app de Jetpack Compose con Navigation Component da error al iniciar porque dice que la pantalla 'pantallaClima/Madrid' no es un hijo directo del NavGraph. ¿Cómo puedo solucionarlo?"

### ◆ Respuesta de IA:

"Tu problema es que la startDestination en NavHost usa una ruta con parámetro {ciudad}, pero debería apuntar a una ruta estática. Usa startDestination = "pantallaClima" y luego maneja el parámetro dentro de NavHost."

### ◆ Implementación en Código:

#### Corrección en Navegacion.kt:

```
NavHost(navController = navController, startDestination = "pantallaClima") {  
    composable(route = "pantallaClima") {  
        PantallaClima(navController, "Madrid") // Valor por defecto  
    }  
    composable(route = "pantallaClima/{ciudad}") { backStackEntry ->  
        val ciudad = backStackEntry.arguments?.getString("ciudad") ?: "Madrid"  
        PantallaClima(navController, ciudad)  
    }  
}
```

---

## 3 La lista de ciudades no se actualiza automáticamente después de agregar una nueva

### ◆ Prompt enviado a IA:

"Cuando guardo una nueva ciudad en la base de datos Room, no aparece inmediatamente en la lista de ciudades guardadas. Solo se actualiza si vuelvo a la pantalla manualmente. ¿Cómo puedo hacer que se actualice automáticamente?"

### ◆ Respuesta de IA:

"El problema es que la lista de ciudades en LiveData no se actualiza después de insertar una nueva ciudad. Debes llamar a cargarCiudades() después de cada inserción para que Room emita la nueva lista."

### ◆ Implementación en Código:

### Corrección en ClimaViewModel.kt para actualizar ciudadesGuardadas automáticamente:

```
fun guardarCiudad(ciudad: RespuestaClima) {  
    viewModelScope.launch(Dispatchers.IO) {  
        val ciudadExiste = ciudadDao.existeCiudad(ciudad.name) > 0  
        if (ciudadExiste) {  
            _mensajeConfirmacion.postValue("La ciudad ${ciudad.name} ya está  
guardada.")  
            return@launch  
        }  
  
        val nuevaCiudad = CiudadEntity(  
            id = 0,  
            nombre = ciudad.name,  
            temperatura = ciudad.main.temp,  
            icono = "https://openweathermap.org/img/wn/${ciudad.weather[0].icon}@2x.png"  
        )  
        ciudadDao.insertarCiudad(nuevaCiudad)  
  
        // ♦ Actualizar la lista de ciudades inmediatamente  
        cargarCiudades()  
  
        _mensajeConfirmacion.postValue("Ciudad ${ciudad.name} añadida  
correctamente.")  
    }  
}
```

---

#### 4. No se muestra un mensaje de confirmación al eliminar o actualizar una ciudad

##### ♦ Prompt enviado a IA:

"En mi app de Jetpack Compose con Room, cuando edito o elimino una ciudad, no aparece ningún mensaje de confirmación. ¿Cómo puedo hacer que el usuario reciba feedback visual?"

◆ **Respuesta de IA:**

"Usa un Snackbar para mostrar los mensajes de éxito o error al usuario. Puedes usar LaunchedEffect para que el mensaje desaparezca automáticamente después de unos segundos."

◆ **Implementación en Código:**

**Agregar Snackbar en ListaCiudadesScreen.kt para mostrar mensajes de confirmación:**

```
@Composable
fun ListaCiudadesScreen(navController: NavController, viewModel: ClimaViewModel =
hiltViewModel()) {
    val ciudadesGuardadas by
viewModel.ciudadesGuardadas.observeAsState(emptyList())
    val mensajeConfirmacion by viewModel.mensajeConfirmacion.observeAsState("")
    val snackbarHostState = remember { SnackbarHostState() }

    LaunchedEffect(mensajeConfirmacion) {
        if (mensajeConfirmacion.isNotEmpty()) {
            snackbarHostState.showSnackbar(mensajeConfirmacion)
        }
    }
}
```

---

## **5. Evitar Duplicados en la Base de Datos Room.**

◆ **Pregunta enviada a IA:**

En mi aplicación de Android con Room, al guardar una ciudad en la base de datos, se duplica si ya existe. ¿Cómo puedo evitar esto y mostrar un mensaje de error si la ciudad ya está guardada?

◆ **Respuesta de IA:**

Para evitar duplicados, puedes usar una consulta @Query para verificar si la ciudad ya existe antes de insertarla. También puedes modificar la base de datos para que nombre sea una clave única.

◆ **Implementación en Código:**

```

@Query('SELECT COUNT(*) FROM ciudades WHERE nombre = :ciudadNombre')
suspend fun existeCiudad(ciudadNombre: String): Int

fun guardarCiudad(ciudad: RespuestaClima) {
    viewModelScope.launch {
        val ciudadExiste = ciudadDao.existeCiudad(ciudad.name) > 0
        if (ciudadExiste) {
            _mensajeError.postValue('La ciudad ${ciudad.name} ya está guardada.')
            return@launch
        }
        val nuevaCiudad = CiudadEntity(
            id = 0,
            nombre = ciudad.name,
            temperatura = ciudad.main.temp,
            icono = 'https://openweathermap.org/img/wn/${ciudad.weather[0].icon}@2x.png'
        )
        ciudadDao.insertarCiudad(nuevaCiudad)
        cargarCiudades()
    }
}

```