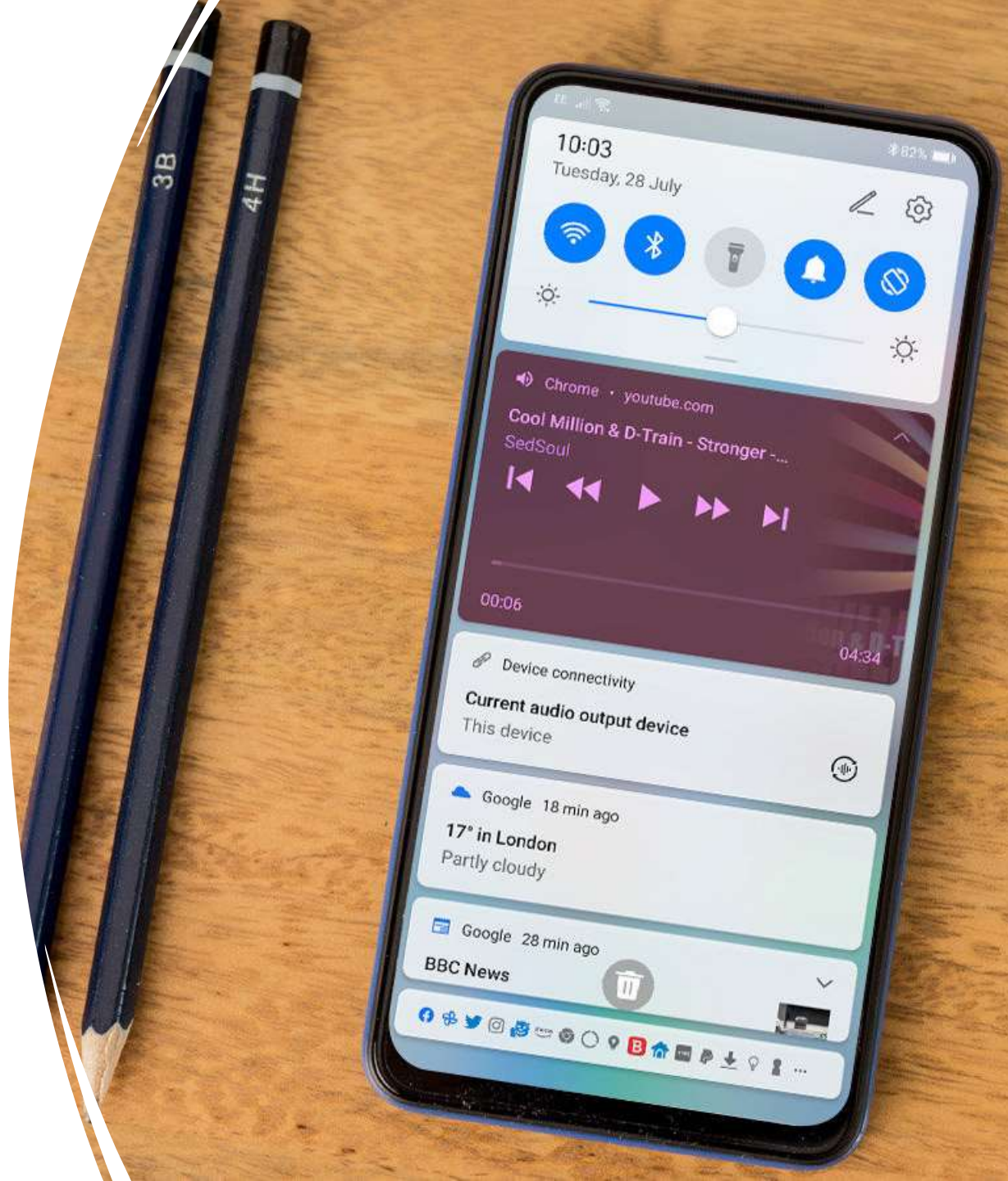


Processi/Tasks in background in Android

Approfondimento per la
materia Programmazione di
Sistemi Mobile



Definizione di lavoro in background

Un'app è in esecuzione in *background* quando:

- **Nessuna delle attività dell'app è attualmente visibile all'utente.**
- **L'app non esegue alcun servizio in primo piano avviato mentre un'attività dell'app era visibile all'utente.**

In caso contrario, l'app è in esecuzione in *foreground* (primo piano).

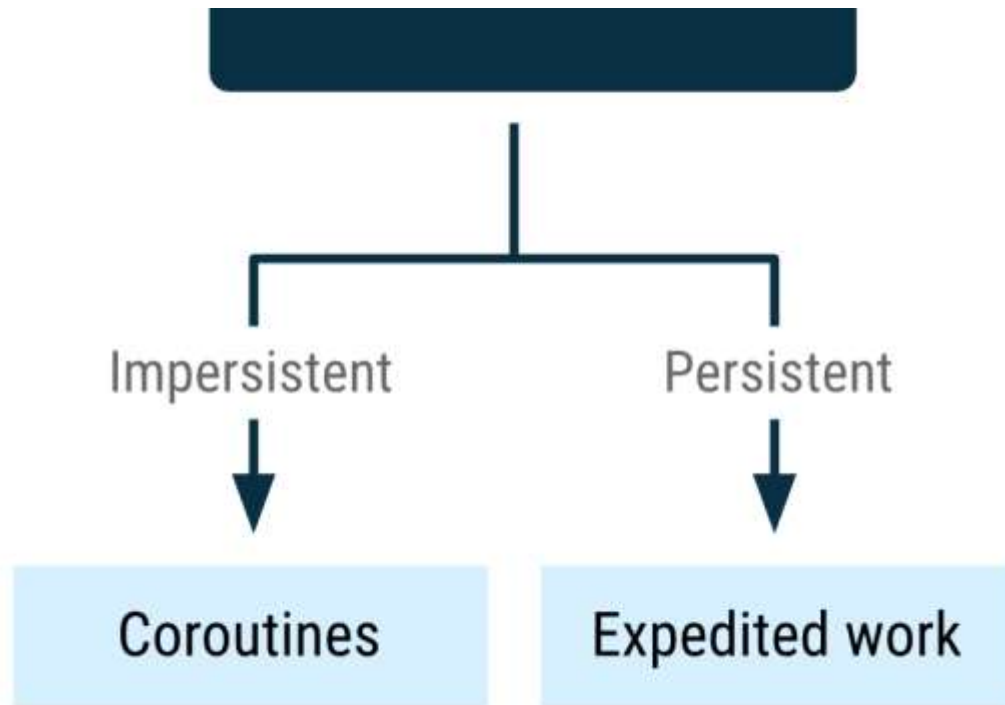
Tipi di lavoro in background

Esistono tre categorie principali in cui si divide il lavoro:

- **Immediato:** deve essere eseguito immediatamente e completato a breve.
- **Lunga durata:** il completamento potrebbe richiedere del tempo.
- **Rinviabile:** non ha bisogno di essere eseguito immediatamente.



Lavoro persistente e non persistente



Ogni categoria di lavoro si suddivide a sua volta in:

- **Lavoro persistente** : rimane programmato nonostante il riavvio dell'app e il riavvio del dispositivo.
- **Lavoro non persistente**: non rimane programmato al termine del processo.

Approcci al lavoro in background

per tutto il lavoro persistente: dovresti usare WorkManager, un componente della libreria Android Jetpack specializzato nel lavoro in background.

per i lavori immediati non persistenti: dovresti usare le coroutine Kotlin, un modello di progettazione di concorrenza per semplificare il codice che viene eseguito in modo asincrono.

per il lavoro non persistente di lunga durata e rinviabile : dovresti completare tali attività attraverso un lavoro persistente utilizzando WorkManager (non dovresti usare questo tipo di lavoro).

Lavoro immediato



Comprende compiti che devono essere eseguiti immediatamente. Si tratta di **attività importanti** per l'utente o che altrimenti **non è possibile pianificare** per l'esecuzione posticipata in un secondo momento. Rimangono programmati per l'esecuzione immediata anche se l'app si chiude o il dispositivo si riavvia.

Esempi:

- Non Persistente: un'app deve **caricare i dati da database**, effettuare una tale richiesta sul thread principale lo bloccherà e causerà un errore dell'interfaccia utente. L'app invece effettua la richiesta fuori dal thread principale in una coroutine.
- Persistente: un'app deve **inviare un messaggio** in un'app di chat. L'app crea un Worker e accoda l'attività come WorkRequest. Accelera la richiesta di lavoro con `setExpedited()`.

Lavoro di lunga durata



Il lavoro è di lunga durata se è probabile che siano **necessari più di dieci minuti** per essere completato. Dove possibile **dovresti suddividere i carichi di lavoro** e gestire le attività come lavoro differibile, altrimenti WorkManager consente di gestire tali attività utilizzando un lavoratore di lunga durata.

Esempio:

- *Un'app **deve scaricare un file di grandi dimensioni** che non puoi raggruppare. Crea un Worker di lunga durata e accoda il download. L'app scarica il file in background.*

Lavoro rinviabile



È qualsiasi lavoro che **non deve essere eseguito immediatamente**. La **pianificazione** del lavoro differito tramite WorkManager è il modo migliore per gestire le attività che non devono essere eseguite immediatamente ma che devono rimanere pianificate alla chiusura dell'app o al riavvio del dispositivo.

Esempio:

- *Un'app vuole **sincronizzare regolarmente i dati** con un back-end. L'utente non attiva la sincronizzazione e il lavoro dovrebbe aver luogo quando il dispositivo è inattivo. L'approccio consigliato consiste nell'usare `PeriodicWorkRequest` con un ruolo di lavoro personalizzato e vincoli per questi scenari.*



Allarmi

Sono un caso d'uso speciale che non fa parte del lavoro in background. Dovresti usare AlarmManager solo per programmare allarmi esatti come sveglie o eventi del calendario.

Quando si utilizza AlarmManager si riattiva il dispositivo dalla **Doze mode**, modalità dove il sistema tenta di risparmiare la batteria limitando l'accesso delle app alla rete e ai servizi ad alta intensità di CPU.

Questo comporta un impatto negativo sulla durata della batteria e sulla salute generale del sistema.



Utilizzo dei servizi in foreground

Android 12 limita l'avvio di servizi in primo piano in background. Utilizziamo i **servizi in foreground per eseguire attività di lunga durata** e che devono notificare all'utente che sono in corso.

Se utilizzi direttamente i servizi in primo piano, assicurati di chiudere correttamente il servizio per preservare l'efficienza delle risorse.

Alcuni casi d'uso:

- Riproduzione multimediale
- Monitoraggio delle attività
- Condivisione della posizione
- Chiamate vocali o videochiamate