Implementação de Backend com Notificações Agendadas em Android – Ensaio Teórico

Conceitos Base

As aplicações móveis que dependem apenas da interface gráfica enfrentam limitações quando precisam de executar ações programadas para o futuro.

No Android, o componente **Activity** é controlado pelo sistema operativo, o que significa que pode ser destruído ou recriado a qualquer momento — por exemplo, quando o utilizador sai da aplicação, roda o ecrã ou quando o sistema liberta memória.
Por essa razão**,** a **Activity não é um local fiável** para guardar lógica de execução futura, como lembretes ou notificações agendadas.

Problema

Se o código responsável pela notificação estivesse dentro da Activity, o lembrete deixaria de existir assim que a aplicação fosse encerrada.

O objetivo deste trabalho era garantir que o utilizador recebesse uma notificação à hora exata, independentemente do estado da aplicação**.**

Para alcançar este comportamento, a solução passou por delegar o controlo do tempo e da execução ao próprio sistema operativo, utilizando os componentes de backend do Android:

- AlarmManager
- PendingIntent
- Service
- BroadcastReceiver**.**

Componentes de Backend Utilizados

A aplicação foi estruturada segundo o princípio da responsabilidade única, onde cada componente cumpre uma função específica:

- **MainActivity** – recolhe as informações do utilizador, como data e hora, e envia o pedido de agendamento.
- **Service (VisitSchedulerService)** – comunica com o sistema operativo para registar o alarme e termina logo após o agendamento, evitando consumo de energia desnecessário.
- **BroadcastReceiver (VisitReminderReceiver)** – é chamado automaticamente pelo Android no momento certo e constrói a notificação que o utilizador vê.

- **Application (MyApp)** – prepara o ambiente da aplicação, criando o canal de notificações antes de qualquer tentativa de exibição.

Esta divisão assegura que cada parte do sistema atua no momento adequado do ciclo de vida da aplicação e de forma independente.

Agendamento e Precisão

O agendamento dos lembretes é feito através do **AlarmManager**, um serviço interno do Android responsável por executar tarefas futuras, mesmo que a aplicação esteja encerrada.

O método utilizado permite precisão total, garantindo que o lembrete é disparado exatamente no momento definido, mesmo durante o modo de poupança de energia (Doze Mode)**.** Este mecanismo é essencial em aplicações que exigem fiabilidade temporal, como alarmes, lembretes de visitas ou notificações de medicamentos.

Além disso, para cumprir os requisitos das versões mais recentes do Android, a aplicação declara a permissão **SCHEDULE_EXACT_ALARM**, que autoriza o uso de alarmes exatos e garante compatibilidade com as políticas de energia mais restritivas.

Autorização e Segurança

A execução da tarefa futura é assegurada através de um **PendingIntent**, um objeto que funciona como uma autorização formal dada ao sistema operativo para executar uma ação em nome da aplicação. Sem esta autorização, o Android não executa instruções agendadas por questões de segurança, já que cada aplicação é isolada do restante sistema.

O PendingIntent é, portanto, a "prova" de que a execução futura foi solicitada legitimamente pela aplicação, garantindo o disparo seguro e controlado da notificação.

Canal de Notificações

Desde o Android 8, as notificações precisam de estar associadas a um **NotificationChannel**. O canal define o nível de importância, o som, a vibração e a visibilidade no ecrã de bloqueio.

Na aplicação desenvolvida, o canal de notificações denominado "visitas" é criado na classe **Application**, assegurando que existe antes de qualquer tentativa de mostrar uma notificação. Esta escolha garante que a notificação é exibida corretamente, com som e prioridade alta, mesmo que a aplicação não esteja aberta.

Eficiência Energética

O Android introduziu mecanismos de gestão de energia, como o **Doze Mode**, que adiam ou agrupam tarefas para poupar bateria. Apesar de benéfico para o utilizador, este comportamento pode comprometer a precisão de notificações agendadas.

Ao recorrer a métodos que ignoram as restrições do Doze Mode, a aplicação assegura que as notificações são disparadas na hora exata**,** mantendo a eficiência e sem recorrer a processos em execução contínua.

Cenários de Falha Prevenidos

A arquitetura adotada previne diversos tipos de falhas comuns:

- **Aplicação fechada:** o alarme permanece ativo no sistema operativo e é executado no momento correto.
- **Activity destruída:** o lembrete não depende da interface gráfica, mantendo-se funcional.
- **Modo de poupança de energia:** o agendamento é executado mesmo em Doze Mode.
- **Ausência de canal de notificações:** o canal é recriado automaticamente no arranque da aplicação.
- **Questões de segurança:** o PendingIntent assegura autorização para o disparo futuro.
- **Reinício do dispositivo:** a arquitetura está preparada para reagendar lembretes após o arranque, através do evento de sistema *BOOT_COMPLETED*.

Escalabilidade e Evolução

A estrutura modular permite que a aplicação cresça sem comprometer o seu funcionamento base.
Entre as possibilidades de expansão destacam-se:

- Reagendamento automático após reinício do telefone.
- Registo dos lembretes numa base de dados local para persistência e histórico.
- Criação de múltiplas notificações simultâneas, cada uma com um identificador próprio.
- Inclusão de ações interativas na notificação, como “Adiar 10 minutos” ou “Abrir aplicação”.

Estas extensões demonstram que a arquitetura foi pensada para evoluir sem reescrever a lógica principal**.**

Conclusão

A aplicação demonstra uma separação clara entre a interface e o backend, delegando ao sistema operativo a responsabilidade pelo agendamento e execução das notificações. Esta abordagem garante que os lembretes são fiáveis, energeticamente eficientes e independentes do estado da aplicação.

A solução cumpre as boas práticas de desenvolvimento Android, assegura compatibilidade com diferentes versões do sistema e cria uma base sólida para futuras funcionalidades.

Em suma, trata-se de uma implementação robusta, precisa e extensível, capaz de garantir notificações fiáveis mesmo em contextos de restrição de energia ou ausência de interface ativa.