

Ciclo de Vida do Aplicativo

Letícia Vidal

O Ciclo de Vida Operacional

(Como o app se comporta no sistema)

O Processo de Instalação (O Lado do Sistema)

Nesta etapa, o foco é entender como o sistema operacional (Android ou iOS) recebe e organiza o aplicativo no hardware do dispositivo.

O Pacote do App: Entender que o aplicativo é baixado como um arquivo compactado único contendo todo o código, imagens e configurações (arquivos `.apk/.aab` no Android ou `.ipa` no iOS).

Descompactação e Permissões: O sistema operacional descompacta o arquivo, aloca um espaço isolado e seguro no armazenamento do celular (chamado de *Sandbox*) e registra as permissões básicas que o app exige para rodar (como acesso à internet).

Otimização Nativa: Os sistemas modernos realizam uma pré-compilação de partes do código durante a instalação para garantir que o app abra o mais rápido possível quando for clicado.

O Processo de Inicialização (O Lado da Execução)

Aqui estuda-se o que acontece nos milissegundos logo após o usuário tocar no ícone do aplicativo. Existem três tipos clássicos de inicialização que todo desenvolvedor precisa conhecer:

Inicialização Fria (*Cold Start*)

É o início do zero absoluto. Acontece quando o app é aberto pela primeira vez após a instalação, ou depois que o celular foi reiniciado, ou quando o app foi completamente fechado da memória RAM.

- **O que acontece por trás:** O sistema precisa criar o processo do app do zero, alocar memória RAM, carregar as telas e inicializar o motor do código (como a engine de JavaScript no caso do React Native).
- **Impacto:** É a inicialização mais lenta e demorada. É aqui que a famosa *Splash Screen* (tela de carregamento com a logo do app) se faz necessária para distrair o usuário enquanto as engrenagens se movem.

Inicialização Morna (*Warm Start*)

Acontece quando o aplicativo já está na memória RAM, mas a tela dele foi descartada pelo sistema para economizar recursos.

- **O que acontece por trás:** O processo principal ainda está vivo no sistema, mas o app precisa recriar a interface visual e reexibir os dados do zero.
- **Impacto:** É mais rápida que a inicialização fria, mas ainda exige algum esforço do processador.

Inicialização Quente (*Hot Start*)

É o cenário ideal e o mais comum no dia a dia. Ocorre quando o usuário minimiza o app por alguns instantes para ver uma notificação e volta para ele logo em seguida.

- **O que acontece por trás:** Tanto o processo quanto as telas do app continuam intactos e guardados na memória RAM. O sistema operacional apenas traz o app de volta para o topo da tela.
- **Impacto:** É instantânea. O usuário continua exatamente de onde parou, sem telas de carregamento.

Primeiro Plano (Foreground / Active):

O app consumindo recursos enquanto o usuário interage visualmente com ele.

Quando um app está em Primeiro Plano, ele entra em um estado de "privilégio e consumo". O foco desta aula teórica gira em torno de três pilares:

O Consumo de Recursos do Dispositivo

Como o usuário está interagindo visualmente e em tempo real, o sistema operacional abre as portas do hardware para o app. Isso envolve:

- **Poder de Processamento (CPU/GPU):** O processador e a placa gráfica trabalham intensamente para renderizar animações a 60 ou 120 quadros por segundo, garantindo que o app pareça fluido e rápido.
- **Consumo de Bateria:** Este é o estado que mais drena a bateria do aparelho, pois a tela está obrigatoriamente acesa e os componentes internos estão operando sem travas de economia.
- **Acesso Irrestrito a APIs:** Em primeiro plano, o app pode acessar livremente a câmera, o microfone, o teclado e os sensores de movimento, pois o usuário está vendo e autorizando aquela ação naquele exato momento.

A Interação Sensorial (UI/UX)

Estar em Primeiro Plano significa gerenciar a experiência direta. O aprendizado aqui foca em como o app responde aos estímulos do usuário:

- **Eventos de Toque e Gestos:** O aplicativo precisa capturar e responder instantaneamente a cliques, arrastes (*swipes*) e pinças de zoom.
- **Foco de Entrada:** Garantir que, se o usuário tocar em um campo de texto, o teclado virtual suba imediatamente e o app ajuste a tela para que o campo não fique escondido atrás do teclado.
- **Gerenciamento de Estado Visual:** O app precisa atualizar a tela dinamicamente conforme os dados chegam da internet, sem travar a navegação do usuário.

As Boas Práticas de Mercado

Neste ponto, o curso conecta o conceito técnico ao sucesso do produto:

O Gargalo da Performance: Se um aplicativo em primeiro plano começar a travar ou demorar para responder aos toques (causando os famosos *lags*), o usuário perde a paciência imediatamente.

- **Não travar a UI:** Todo processamento pesado (como baixar uma imagem grande ou processar um banco de dados) deve ser feito em "linhas de execução" secundárias (*background threads*) para que a tela principal nunca congele.
- **Otimização de Renderização:** Evitar criar interfaces excessivamente pesadas que façam o celular esquentar na mão do usuário. Se o celular esquenta em primeiro plano, o usuário desinstala o app por medo de danificar o aparelho.

Segundo Plano (Background):

Como o sistema pausa o app quando o usuário abre outra ferramenta (ex: o app de música continua tocando, mas o jogo é pausado).

Quando o usuário arrasta o dedo para voltar à tela inicial ou abre outra ferramenta, o aplicativo perde o foco visual. A partir desse milissegundo, o sistema operacional (Android ou iOS) assume o controle e muda drasticamente as regras do jogo.

1. O Mecanismo de Pausa e Restrição

Atualmente, tanto o Android quanto o iOS são extremamente agressivos com apps em segundo plano para evitar abusos (como aplicativos maliciosos que ficam minerando dados ou gastando internet escondidos).

- **Notificações Visuais Obrigatórias:** No Android, para um app rodar em segundo plano ativamente (como o Spotify), ele é obrigado a exibir uma notificação fixa na barra do topo. O usuário *precisa* saber que o app está consumindo bateria.
- **Sistemas de IA e Economia:** Os sistemas operacionais modernos usam inteligência artificial para entender a rotina do usuário. Se o sistema nota que você nunca abre um aplicativo específico à noite, ele corta totalmente as permissões de segundo plano daquele app nesse horário para economizar energia.

A Grande Diferença: Apps "Passivos" vs. Apps "Ativos"

Como a própria descrição da sua ementa aponta, o sistema trata os aplicativos de formas diferentes dependendo da sua categoria profissional. É aqui que entram os **Serviços de Segundo Plano**:

Apps Pausados (O exemplo do Jogo)

Um jogo ou um app de rede social não precisa continuar rodando se você não está olhando para ele.

- **O comportamento:** O sistema pausa o código imediatamente. Se você estava no meio de uma corrida ou de uma partida single-player, o jogo fica "congelado no tempo". Quando você voltar, ele descongela exatamente de onde parou (se a memória RAM não tiver sido limpa nesse meio tempo).

Apps Autorizados a Rodar (O exemplo da Música ou GPS)

Alguns aplicativos *precisam* funcionar mesmo invisíveis. Os sistemas operacionais abrem exceções rígidas para categorias específicas através de APIs nativas:

- **Reprodução de Áudio:** Spotify ou reprodutores de podcast continuam decodificando arquivos de som e mandando para o fone de ouvido.
- **Geolocalização (GPS):** Apps de corrida (Strava) ou navegação (Waze, Google Maps) continuam rastreando as coordenadas do usuário em tempo real para falar as direções ou calcular a distância.

As Restrições Modernas (Android e iOS)

Atualmente, tanto o Android quanto o iOS são extremamente agressivos com apps em segundo plano para evitar abusos (como aplicativos maliciosos que ficam minerando dados ou gastando internet escondidos).

- **Notificações Visuais Obrigatórias:** No Android, para um app rodar em segundo plano ativamente (como o Spotify), ele é obrigado a exibir uma notificação fixa na barra do topo. O usuário *precisa* saber que o app está consumindo bateria.
- **Sistemas de IA e Economia:** Os sistemas operacionais modernos usam inteligência artificial para entender a rotina do usuário. Se o sistema nota que você nunca abre um aplicativo específico à noite, ele corta totalmente as permissões de segundo plano daquele app nesse horário para economizar energia.

Suspensão e Encerramento (Destroy):

Como o Android ou iOS "matam" o aplicativo da memória quando o celular precisa de performance para outra tarefa.

Por que o sistema precisa "matar" um aplicativo?

Ao contrário de um computador tradicional, que possui muita memória RAM e sistemas de paginação pesados em disco, os smartphones lidam com recursos de hardware altamente limitados e chips que precisam economizar energia térmica e de bateria.

A Linha de Prioridades: Quem morre primeiro?

Os sistemas operacionais não encerram os aplicativos aleatoriamente.

Primeiros a morrer (Apps Vazios/Inativos): Aplicativos que o usuário não mexe há muitas horas e que não possuem nenhuma atividade pendente.

Segundos a morrer (Apps em Background Padrão): Apps de redes sociais ou utilitários que foram minimizados. O sistema congela seus dados e os remove da RAM.

Difícilmente morrem (Apps em Background Ativo): Aplicativos que prestam serviços essenciais em tempo real, como o GPS que está guiando o usuário no trânsito ou o player de música. O sistema só vai fechá-los em caso de necessidade extrema de sobrevivência do aparelho.

Imortais (O App em Primeiro Plano e o Sistema): O aplicativo que o usuário está olhando e interagindo na tela e as funções vitais do próprio sistema operacional (como a recepção de sinal telefônico). Esses nunca são fechados pelo gerenciador de memória.

Suspensão vs. Encerramento Absoluto

Suspensão com Preservação de Estado (O "Susto"): O sistema operacional avisa o aplicativo milissegundos antes de fechá-lo: *"Vou te encerrar, salve o que puder"*. O app tem uma fração de segundo para salvar de forma leve onde o usuário estava (ex: a rolagem do feed ou o texto digitado).

Encerramento Abrupto (Crash ou Force Kill): Se o sistema precisar de memória desesperadamente ou se o aplicativo começar a se comportar mal (vazamento de memória ou travamento da interface), o sistema operacional simplesmente corta o processo do app instantaneamente.