

Estudo de caso de um aplicativo offline-first de controle financeiro na gestão de veículos

Adriane Cristina Galvão Cunha¹

¹Instituto de Engenharia e Geociências – Universidade Federal do Oeste do Pará (UFOPA)

{adriane.cunha}@gmail.com

Abstract. This article aims to reflect on clock and data synchronization as one of the fundamental pillars for the reliable operation of distributed systems. The synchronization problems of applications involving offline-first software are presented. The application was developed using React Native and Expo.

Resumo. O artigo tem por objetivo refletir sobre a sincronização de relógios e dados com um dos pilares fundamentais para o funcionamento confiável de sistemas distribuídos. São apresentados os problemas de sincronização de aplicativo envolvendo software Offline first. O aplicativo foi desenvolvido com React Native e Expo.

1. Introdução

O controle de gastos com combustível é uma necessidade constante para motoristas e gestores de veículos, especialmente em cenários onde a eficiência operacional e o monitoramento financeiro são essenciais. Com esse propósito, foi desenvolvido um aplicativo capaz de analisar o consumo médio por litro, registrar percursos e organizar informações relevantes sobre o uso do veículo. Um dos principais desafios desse processo consistiu em criar um sistema que se mantivesse funcional mesmo diante de falhas no servidor ou falta de conectividade, garantindo ao usuário uma experiência contínua e confiável. Nesse contexto, a abordagem offline first tornou-se indispensável, considerando que muitas regiões apresentam acesso limitado ou instável à internet.

A adoção desse paradigma permite que o aplicativo opere de maneira autônoma na maior parte do tempo, utilizando sincronização apenas como complemento, e não como dependência. Assim, mesmo que o servidor esteja indisponível, os dados locais continuam acessíveis e o usuário não é prejudicado. Diante desse cenário, este estudo busca analisar os desafios de sincronização de dados e relógios em um aplicativo offline first, identificando como esses mecanismos contribuem para a consistência, usabilidade e confiabilidade do sistema. Além disso, discute-se como as decisões de arquitetura influenciam diretamente o desempenho do aplicativo no contexto de controle financeiro e gestão de veículos.

2. Referencial teórico

O desenvolvimento de sistemas distribuídos móveis envolve desafios relacionados à consistência de dados, sincronização e tolerância a falhas, especialmente em cenários onde a conectividade com a internet é limitada ou instável. Nesse contexto, a abordagem offline first e os mecanismos de sincronização de relógios e dados tornam-se fundamentais para garantir o funcionamento contínuo e confiável das aplicações.

Os sistemas distribuídos, conforme discutido por [Tanenbaum and Van Steen 2007], caracterizam-se pela existência de múltiplas réplicas de dados operando de forma cooperativa, o que exige mecanismos capazes de manter coerência entre dispositivos que podem, eventualmente, funcionar de forma desconectada. Em aplicações móveis, a possibilidade de desconexão é uma condição natural do ambiente operacional, tornando indispensáveis modelos que aceitem autonomia local, como ressaltam Saito e Shapiro (2005) ao tratarem das técnicas de replicação otimista.

Nesse cenário, a abordagem offline first surge como um paradigma que assume a desconectividade como regra e não como exceção. Segundo [Malaquias and Silva 2021]. Aplicações offline first são usadas quando se quer construir um aplicativo que consiga funcionar nos modos online e offline. Além disso, permitir uma transição suave tanto do offline para o online quanto no sentido contrário. Nesse sentido sistemas offline first são projetados para operar plenamente com dados locais, usando o servidor apenas como fonte de sincronização quando a conectividade é restabelecida. Tal abordagem reduz a dependência do tempo de resposta da rede, melhora a experiência do usuário e aumenta a resiliência do sistema, sendo particularmente eficaz para aplicações de controle financeiro e rotinas em movimento, como no caso da gestão de veículos.

A sincronização de dados entre cliente e servidor, porém, introduz o desafio da resolução de conflitos. Em ambientes distribuídos, a atualização concorrente de informações pode gerar versões divergentes de um mesmo registro.

Segundo [Kleppmann 2017], “Uma abordagem para alcançar a convergência eventual é declarar que cada réplica precisa armazenar apenas o valor mais recente e permitir que valores mais antigos sejam sobreescritos e descartados. Então, desde que tenhamos alguma maneira de determinar inequivocamente qual gravação é mais recente e que cada gravação seja eventualmente copiada para cada réplica, as réplicas eventualmente convergirão para o mesmo valor”.

Nesse ponto, a sincronização de relógios desempenha papel essencial. Os relógios físicos tendem a apresentar desvios naturais, o que pode comprometer a ordenação de eventos em sistemas distribuídos. Protocolos como o NTP reduzem esses desvios, porém, em ambientes móveis desconectados, marcas temporais locais são frequentemente utilizadas como solução prática. Assim, mesmo com eventuais discrepâncias temporais, combinações entre timestamps, versionamentos e identificadores exclusivos tornam possível manter coerência após a reconciliação.

A utilização de identificadores únicos universais (UUIDs), conforme definido por Leach et al. (2005), é outro elemento relevante em sistemas offline first, pois evita colisões durante a criação de registros no modo desconectado. Esse mecanismo garante unicidade estatística, facilitando a posterior mesclagem dos dados com o servidor. Complementarmente, o uso de bancos de dados locais como SQLite, amplamente empregado em dispositivos móveis, assegura durabilidade e atomicidade das operações realizadas no cliente. [Allen and Owens 2010] ressalta que o SQLite é projetado para operar de forma eficiente em ambientes embarcados, reforçando sua adequação em arquiteturas que dependem fortemente de armazenamento local.

3. Metodologia

A metodologia deste estudo teve início com a identificação da necessidade de desenvolver um sistema capaz de operar de forma autônoma, sem depender diretamente do servidor para executar suas principais funcionalidades. Essa necessidade foi motivada pelo uso do aplicativo em ambientes com conectividade instável ou limitada, o que tornava inadequado um modelo que exigisse comunicação contínua com a rede. A partir desse diagnóstico inicial, foram conduzidas reuniões com a equipe para levantar requisitos, discutir alternativas técnicas e estruturar uma arquitetura que garantisse robustez, eficiência e autonomia ao aplicativo.

Adotou-se a abordagem offline first como princípio norteador do projeto, priorizando a persistência local dos dados e a sincronização apenas quando a conectividade estivesse disponível. Nesse processo de definição arquitetural foi planejado que o dispositivo do usuário teria capacidade de armazenar, consultar e manipular registros localmente, retornando ao servidor somente para operações de atualização ou reconciliação de dados. Essa orientação visou preservar a experiência do usuário mesmo diante de falhas externas, além de minimizar a dependência em relação à disponibilidade imediata do servidor.

Para a implementação do cliente móvel optou-se por React Native e Expo, fornecendo compatibilidade multiplataforma e agilidade no desenvolvimento. A biblioteca UUID foi utilizada para gerar identificadores únicos para cada objeto criado, reduzindo a probabilidade de colisões de IDs durante a operação offline e a posterior sincronização. O banco de dados no dispositivo é o Expo-SQLite, escolhido por permitir interação com um banco relacional SQL local. Importante destacar que as tabelas do aplicativo móvel foram mantidas idênticas às do servidor, ou seja, a estrutura do banco do cliente é a mesma do servidor, estratégia que facilita a conversão e a reconciliação de dados entre as duas instâncias.

A fim de tratar possíveis conflitos advindos de atualizações concorrentes entre cliente e servidor, a estratégia de resolução adotada foi Last Write Wins (LWW). Essa política estabelece que, em caso de conflito, prevalece o registro com a marca temporal mais recente, simplificando a lógica de reconciliação e reduzindo a complexidade do processo de sincronização. Conjugada à padronização das estruturas de dados e ao uso de identificadores únicos, a política LWW permitiu um mecanismo previsível e de implementação direta para manter a coerência entre as réplicas, ao mesmo tempo em que atende ao objetivo de garantir autonomia e integridade das informações no contexto offline first.

4. Visão do Aplicativo

O Kontado é um aplicativo que centraliza funcionalidade de controle financeiro na gestão de veículos, permitindo o usuário cadastrar e gerenciar seus veículos, despesas, abastecimentos, manutenções, lembretes, diário de bordo, comunidade de preços, relatórios e etc, o app também permite rastreamento rotas/caminhos enquanto o condutor dirige, para obter dados importantes sobre distância percorrida, consumo, duração, percurso e etc.

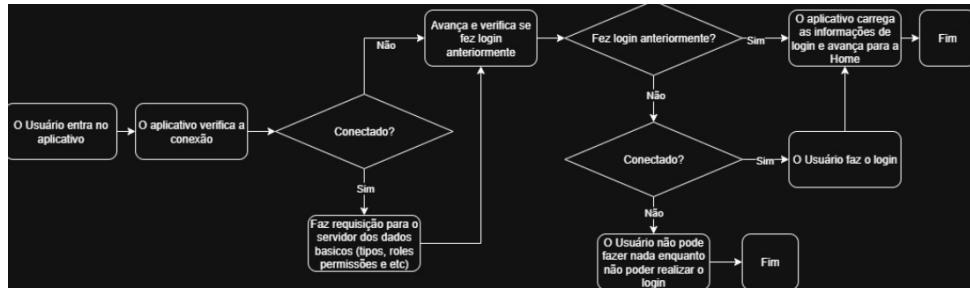


Figura 1. Autor: Fonte própria

O fluxograma descreve o funcionamento de um aplicativo desenvolvido sob a abordagem offline first, no qual o usuário pode operar a maior parte das funcionalidades mesmo sem conexão com a internet. Assim que o usuário abre o aplicativo, o sistema inicia verificando se há conectividade disponível. Quando a conexão existe, o aplicativo aproveita para sincronizar dados essenciais com o servidor, como tipos, papéis, permissões e demais informações necessárias para manter o funcionamento atualizado e seguro. Essa sincronização não impede o uso do aplicativo, pois seu design prioriza autonomia mesmo em cenários de instabilidade ou falta total de rede.

Caso o aplicativo identifique conexão ausente no momento da abertura, ele não interrompe imediatamente o uso. Em vez disso, avança para verificar se existe um login previamente realizado no dispositivo. Esse comportamento é fundamental para preservar a experiência offline first, permitindo que o usuário continue acessando o aplicativo mesmo que o servidor esteja indisponível, desde que já tenha sido autenticado em algum momento anterior. Assim, o aplicativo consegue manter uma operação fluida sem depender continuamente da rede.

Quando a verificação encontra um login já registrado, o aplicativo carrega as informações locais de autenticação e direciona o usuário diretamente para a tela principal, garantindo acesso rápido e totalmente offline. Entretanto, se não houver registro de login anterior, o comportamento se torna dependente da conectividade: havendo internet, o usuário é conduzido ao processo de login tradicional; mas, sem conexão, não é possível autenticar pela primeira vez, o que impede o acesso até que uma rede esteja disponível. Essa restrição preserva a segurança e integridade dos dados, evitando liberar acesso sem credenciais válidas.

Dessa forma, o fluxograma reforça a lógica híbrida entre autonomia offline e sincronização inteligente. O aplicativo funciona plenamente sem internet, apenas recorrendo ao servidor para atualizar regras, permissões e dados auxiliares quando há rede disponível. Assim, o sistema concilia segurança, eficiência e usabilidade contínua, garantindo que o usuário tenha acesso mesmo em ambientes desconectados, sem comprometer os processos que realmente exigem autenticação e sincronização com o servidor.

4.1. Conclusão

O desenvolvimento do aplicativo Kontado demonstrou a relevância e a eficácia da abordagem offline first em sistemas distribuídos que necessitam funcionar de maneira confiável mesmo diante de instabilidade ou ausência de conexão com a internet. A análise dos processos de sincronização revelou que a capacidade de operar autonomamente, atualizando

dados apenas quando a conectividade está disponível, garante uma experiência contínua, segura e eficiente ao usuário. A arquitetura projetada, apoiada em tecnologias como React Native, Expo e UUID, mostrou-se adequada para lidar com os desafios de controle financeiro e gerenciamento de veículos, preservando a integridade das informações e mantendo o funcionamento fluido do sistema.

Além disso, o estudo evidenciou que a sincronização de dados e relógios é um elemento central para reduzir inconsistências e assegurar coerência entre ambiente local e servidor, sobretudo em aplicações que precisam registrar trajetos, abastecimentos e despesas em tempo real, independentemente de conexão. O fluxograma analisado reforçou que a combinação entre autenticação local, atualização incremental e checagem de conectividade contribui diretamente para a robustez do sistema, evitando bloqueios desnecessários e priorizando a autonomia do usuário.

A experiência obtida durante o desenvolvimento e a avaliação do aplicativo reforçam que soluções offline first são especialmente vantajosas em contextos onde o acesso à internet é limitado, como em viagens, áreas remotas ou situações emergenciais. Assim, o Kontado se apresenta como uma solução sólida e confiável para o controle financeiro de veículos, oferecendo recursos avançados sem abrir mão da usabilidade e da segurança.

Por fim, este estudo de caso confirma que a adoção de estratégias de sincronização inteligente e design centrado na autonomia do dispositivo são fundamentais para a construção de aplicativos modernos e resilientes. Os resultados apontam que a abordagem offline first não apenas melhora a experiência do usuário, mas também amplia a escalabilidade e a confiabilidade do sistema, tornando-se uma alternativa estratégica para o desenvolvimento de aplicações móveis que exigem disponibilidade contínua.

Referências

- Allen, G. and Owens, M. (2010). *The Definitive Guide to SQLite*. Apress, 2 edition.
- Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media, Sebastopol, 1 edition.
- Malaquias, E. J. and Silva, J. L. (2021). Aplicação mobile para gestão de aulas ead: utilizando as tecnologias offline first e optimistic user interface. *Fórum Gerencial*. Acesso em: 15 out. 2025.
- Tanenbaum, A. S. and Van Steen, M. (2007). *Sistemas Distribuídos: Princípios e Paradigmas*. Pearson.