SoftBancada: uma bancada de ferramentas para programadores, toda no telemóvel

Publicado em 2026-11-27 19:58:55



Android · Ferramentas

Cliente HTTP e WebSocket, Git, SQLite, JWT, cron, SemVer, estruturas de dados e dados de teste numa só app Android, sem contas, sem nuvem e sem telemetria.

Há dias em que um programador abre seis separadores, duas apps e um terminal só para ver se um JWT expirou, se uma expressão cron dispara às 3 da manhã de domingo, ou que commit mudou aquele ficheiro. A SoftBancada junta

Este é o relato técnico de como foi feita: as doze ferramentas, as decisões menos óbvias (porque escrevi um cliente HTTP e um leitor de Git de raiz) e os limites que convém conhecer antes de confiar nela.

⬇ Obter a SoftBancada na loja de apps (Android 10 ou superior)

[LINK DA LOJA DE APPS]

O problema

As ferramentas de bolso para programadores costumam ter um de dois defeitos. Ou são sites que pedem para colar lá um token, um esquema de base de dados ou uma resposta de API com dados de clientes, ou são apps que querem uma conta, sincronizam tudo e enviam estatísticas de uso para algum lado. Para quem trabalha com dados reais, nenhuma das duas serve.

A regra de partida da SoftBancada foi: **tudo é calculado no aparelho, e a rede só é usada quando o utilizador a pede**, por exemplo para enviar um pedido HTTP ou abrir um repositório por SFTP. Não há IA, não há serviços

Ficha técnica

App
: SoftBancada 1.0.0 (SofteLabs)

Plataforma
: Android 10 ou superior, telemóvel e tablet

Ferramentas
: 12, em 4 grupos: API, Git, Estruturas, Dados

Permissões
: Só Internet, usada a pedido

Dados
: Locais, cifrados (AES-256-GCM)

Idiomas
: Português e inglês

Contas e telemetria
: Nenhumas

Doze ferramentas, quatro grupos

API

- **Cliente HTTP / REST.** GET, POST, PUT, PATCH, DELETE, HEAD e OPTIONS, com cabeçalhos, corpo JSON/XML/texto/formulário validado antes de enviar, resposta formatada e **tempos por fase**: DNS, ligação, TLS e primeiro byte. Ambientes com `{{variáveis}}`, coleções exportáveis em JSON e histórico.

com descrição em linguagem corrente e próximas execuções no fuso escolhido; modelos JSON/SQL com variáveis e filtros.
- **JWT · Base64 · URL · tempo.** Descodificação local de JWT e verificação de assinaturas HS256/384/512, codificações, hashes e conversão de timestamps Unix/ISO-8601 entre fusos.

Git

- **Explorador Git**, só de leitura, sobre uma pasta do aparelho ou um servidor SFTP: histórico, ramos, etiquetas, diffs por ficheiro, estado da árvore de trabalho e navegação nos ficheiros.
- **.gitignore · .editorconfig · licenças** para Kotlin/Android, Python, C/C++, Rust, Shell e SQLite, com verificação de consistência entre vários repositórios.
- **SemVer · changelog.** Próxima versão, intervalos ao estilo npm e Maven, e `CHANGELOG.md` no formato «Keep a Changelog» a partir de mensagens Conventional Commits ou de entradas manuais.

Estruturas

- **Estruturas de dados e grafos** passo a passo: árvores BST e AVL, heaps, tabelas de hash, filas, pilhas, BFS, DFS, Dijkstra e ordenação topológica.
- **Máquinas de estados** escritas com setas, JSON ou YAML, simuladas com guardas e variáveis, com

WCAG, px/dp/sp por densidade de ecrã e matrizes de transformação.

Dados

- **Explorador SQLite:** esquema, índices, triggers, consultas só de leitura e exportação para CSV, JSON ou Markdown.
- **Analisador estático:** TODO/FIXME/NOTE, linhas, funções, complexidade, blocos duplicados e cabeçalhos de licença, com relatório em Markdown.
- **Dados sintéticos** com semente fixa, casos limite e NIF português válido, em CSV, JSON, JSONL ou SQL.

Por dentro: decisões que não são as óbvias

Um cliente HTTP escrito sobre sockets

O `HttpURLConnection` do Android não aceita PATCH sem truques e não diz quanto tempo demorou cada fase do pedido. Por isso a SoftBancada tem um cliente HTTP/1.1 próprio, sobre sockets: resolve o nome, liga, faz o aperto de mão TLS e lê a resposta, medindo cada passo. Trata chunked, gzip, redirecionamentos e respostas grandes (que são cortadas a partir de um limite, com aviso).

nenhum botão «ignorar todos os erros de TLS». O tráfego `http://` em claro é permitido, porque muitas APIs de desenvolvimento correm em `localhost` ou na rede local.

Um leitor de Git próprio, em vez de JGit

O JGit espera um sistema de ficheiros normal. Numa app Android moderna, as pastas do utilizador chegam pelo Storage Access Framework (SAF), e um repositório num servidor chega por SFTP. Em vez de copiar repositórios inteiros, escrevi um leitor só de leitura sobre uma interface mínima (`stat`, `list`, `read` com posição e tamanho), que funciona da mesma forma nas duas origens.

Lê objetos soltos e packfiles (índices v1 e v2, deltas `OFS_DELTA` e `REF_DELTA`), referências e `packed-refs`, o índice nas versões 2, 3 e 4, e repositórios SHA-1 ou SHA-256. Os diffs de linhas usam o algoritmo de Myers, com blocos ao estilo `git diff`. Uma cache LRU de objetos evita ler o mesmo pacote vezes sem conta por SFTP.

Deixar o SQLite validar o SQL

Escrever um analisador de SQL completo seria trabalho para meses e daria sempre respostas ligeiramente diferentes das do motor real. A solução é mais simples: a

guarda lexical garante que só passam instruções de leitura, e o ficheiro é aberto a partir de uma cópia, para nunca tocar no original.

Dados de teste que se repetem

Os dados sintéticos usam um gerador SplitMix64 com uma semente por campo, derivada da semente global e do nome do campo. Assim, acrescentar uma coluna nova não muda os valores das outras, e a mesma semente gera exatamente as mesmas linhas hoje, amanhã e noutro aparelho. É o tipo de pormenor que torna um teste reproduzível.

Motores em Kotlin puro

Toda a lógica (Git, HTTP, WebSocket, cron, SemVer, máquinas de estados, estruturas, dados sintéticos, cifra) vive em Kotlin puro, sem depender do Android, e é testada numa JVM com **47 testes**. Os testes do Git comparam o leitor com o `git` real, e os da rede correm contra servidores HTTP, TLS, WebSocket e SFTP embebidos no próprio teste.

Rede	Só a permissão INTERNET, usada apenas quando o utilizador [illegible]
Dados guardados	Ficheiro JSON cifrado com AES-256-GCM, chave por ficheiro [illegible] HKDF a partir de um segredo protegido pelo Android K[illegible]
TLS	Só certificados do sistema ou instalados pelo utilizador, por S[illegible] entidade.
SFTP	Chave do servidor fixada na primeira ligação; se mudar, a app [illegible]
Bloqueio	No telemóvel, identificador ou elegível do aparelho ao abrir [illegible] segundo plano. No tablet, sem bloqueio próprio. Capturas des[illegible] em ecrãs.
Cópia entre aparelhos	Ficheiro .slb cifrado com palavra-passe (AES-256-GCM, PBKD[illegible] iterações). As palavras-passe SFTP nunca são expo[illegible]

Os dados ficam num ficheiro JSON cifrado em vez de numa base SQLite/Room. É um desvio consciente, comum a todas as apps SofteLabs: para o volume de dados de uma app destas, um ficheiro cifrado por inteiro é mais simples e não deixa nada em claro.

A pilha técnica

- Kotlin 2.0.21, Jetpack Compose com Material 3 (BOM 2025.04.01)

de terceiros
- Interface bilingue, português e inglês, segundo o idioma do sistema
- Documentação de utilizador e técnica dentro da app, nas duas línguas

Limites, ditos às claras

> A SoftBancada é uma bancada de consulta e verificação, não um IDE nem um cliente Git completo.

- O Git é só de leitura: não faz commits, push nem pull. Não lê o formato reftable nem aplica `autocrlf` ou filtros.
- O cliente HTTP fala apenas HTTP/1.1.
- O explorador SQLite lê uma cópia sem o ficheiro `-wal`, por isso alterações ainda não consolidadas não aparecem.
- O analisador estático é lexical: contagens de funções e complexidade são aproximações.
- As licenças longas (Apache, GPL, LGPL, MPL) aparecem como aviso com o endereço oficial, não com o texto completo.
- Pastas e repositórios vêm do aparelho ou de SFTP; SMB não é suportado.

só para fazer contas que um telemóvel faz sozinho. O resultado é uma app que não precisa de saber quem é o utilizador, não tem para onde enviar nada e dá sempre a mesma resposta à mesma pergunta.

Se experimentar e alguma ferramenta falhar, ou faltar uma que usa todos os dias, diga-me. É assim que a bancada cresce.

⬇ Obter a SoftBancada

[LINK DA LOJA DE APPS]

SoftBancada é uma app da SofteLabs. Usa JSch (licença BSD) para SFTP. Sem anúncios, contas nem recolha de dados.