

Projeto 1: Aplicativo Móvel (Scanner e Envio):

- **Objetivo:** Ser um aplicativo simples para o celular que:
 - Permita ao usuário fazer login com a mesma conta usada no site.
 - Utilize a câmera do celular para ler códigos de barras.
 - Envie o número do código de barras para o backend (sua API) associado à conta logada.
- **Foco:** Leitura eficiente do código de barras e comunicação segura com o backend, mantendo a identidade do usuário.

Projeto 2: Sistema Web (Seu Site/API de Compras):

- **Objetivo:** Ser a plataforma principal onde o usuário navega pelos produtos, visualiza o carrinho de compras, finaliza a compra e gerencia sua conta.
- **Funcionalidades:**
 - Autenticação e gerenciamento de usuários (login com a mesma conta do app).
 - Recepção do código de barras enviado pelo aplicativo móvel.
 - Consulta do produto no banco de dados pelo código de barras.
 - Adição do produto ao carrinho de compras do usuário logado.
 - Visualização e modificação do carrinho de compras.
 - Processamento de pedidos e pagamentos (funcionalidades normais de um programa de mercado online).

Dois Projetos em Um?

Sim, essa abordagem pode ser vista como **dois projetos distintos, mas fortemente integrados**. Cada um tem sua própria interface (aplicativo móvel e site) e seu próprio conjunto de tecnologias e desafios de desenvolvimento. A chave para o sucesso será a comunicação eficiente entre eles através da sua API.

Vantagens dessa abordagem:

- **Melhor Experiência do Usuário:** A leitura via aplicativo nativo tende a ser mais rápida e confiável do que via navegador. A conexão com a conta já logada no site torna o processo de adicionar itens mais direto.
- **Separação de Responsabilidades:** Cada projeto pode ser desenvolvido e mantido de forma mais independente. O aplicativo foca na leitura, e o site/API foca na lógica de negócios e na interface de compra.
- **Escalabilidade:** Você pode escalar o aplicativo móvel e o backend da API separadamente, conforme a necessidade.
- **Aproveitamento das Capacidades Nativas:** O aplicativo móvel pode aproveitar

ao máximo os recursos do celular (câmera, notificações, etc.).

Próximos Passos:

1. Backend (API Spring Boot):

- Implementar a autenticação para que o aplicativo móvel possa se conectar de forma segura com a conta do usuário logada no site. Você pode usar tokens (JWT, por exemplo).
- Criar um endpoint na API para receber o código de barras enviado pelo aplicativo móvel, juntamente com a identificação do usuário.
- Implementar a lógica para buscar o produto no banco de dados pelo código de barras e adicioná-lo ao carrinho de compras do usuário.
- Continuar desenvolvendo as funcionalidades normais do carrinho de compras e do processo de checkout no seu site.

2. Aplicativo Móvel (Android/iOS):

- Desenvolver a interface de login que se conecte ao mesmo sistema de autenticação do seu site.
- Implementar a funcionalidade de acesso à câmera e leitura do código de barras (usando bibliotecas nativas ou cross-platform).
- Implementar a lógica para enviar o número do código de barras para o endpoint da sua API, juntamente com o token de autenticação do usuário.

Considerações:

- **Autenticação:** Garantir que o aplicativo móvel se comunique de forma segura com a API e que a identificação do usuário seja confiável é crucial.
- **Sincronização:** Pensar em como o carrinho de compras será sincronizado entre o aplicativo (se o usuário adicionar itens por lá) e o site (se o usuário estiver navegando e adicionando itens pelo navegador). Uma abordagem comum é ter o carrinho centralizado no backend (na conta do usuário).
- **Desenvolvimento Mobile:** Você precisará aprender desenvolvimento mobile (Android, iOS ou um framework cross-platform como React Native ou Flutter).

Essa abordagem é mais ambiciosa do que apenas integrar um leitor de código de barras ao site, mas oferece uma experiência de usuário muito melhor para a funcionalidade de "scanner" no supermercado. Sim, você estará essencialmente trabalhando em dois projetos distintos que precisam se comunicar de forma eficaz. Comece focando em estabelecer a comunicação segura entre o aplicativo (mesmo que um protótipo simples de leitura e envio) e a sua API backend.