Mobile Engineer - About the job

Hi there! Thanks for your curiosity
We're driven by ambition and guided by a bold vision for the years ahead. As we scale our profitable ecosystem of tools and services, we're seeking a standout **Mobile Engineer** to help us go to the next level.

Who we are

Zubale is a technology company that enables retailers to win in eCommerce. We serve a wide range of retailers from supermarkets, specialty stores, pharmacies, department stores and fashion brands across Mexico, Brazil, Chile, Costa Rica, Peru and Colombia (and soon the U.S!).
How? We have three key products:

- Flexifleet: Freelancer Marketplace to perform picking, packing, and delivery tasks for retailers and aggregators.
- Fulfillment Optimizer: Modular Software to increase productivity in eCommerce picking, packing and delivery.
- Consumer Connect: AI WhatsApp Conversational Commerce to boost sales and elevate customer experience.

We are passionate about helping brands, aggregators, and retailers optimize their eCommerce operations, lower costs, and boost profitability. Backed by over $70M USD in venture capital, we've delivered consistent growth over the past six years and completed more than 100 million orders through our platform. Today, we serve 100+ customers and engage thousands of freelancers across our network. We surpassed $100M in annualized revenue and we are a profitable company. And this is just the beginning. We're building a profitable, scalable ecosystem — and our vision is to become the #1 partner for retailers to grow their eCommerce business.

Who you are

- A **Mobile Developer specialized in React Native** to design, build, and scale mobile applications used by thousands of users.
- An AI-first developer who owns and delivers high-quality code
- You enjoy owning end-to-end mobile experiences, moving fast, and collaborating with cross-functional teams to deliver high-quality products.
- You play a key role in our mobile architecture and technical decision-making.
- You work closely with Product, Design, Backend, and QA Automation teams to ship high-impact features with speed, quality, and reliability.

What you will do

- Design and implement scalable React Native features for iOS and Android.
- Take architectural decisions for mobile platforms, including navigation, state management, performance patterns, and code reuse.
- Integrate with backend APIs, ensuring reliability, observability, and performance.
- Build reusable UI components and contribute to shared mobile libraries.
- Implement automated testing (unit, integration, end-to-end)
- Optimize application performance, memory usage, and startup times.
- Collaborate with QA Automation to ensure high test coverage.
- Partner with Product and Design from ideation to delivery to ensure feasibility and clarity.
- Participate in code reviews, architecture discussions, and sprint planning.
- Identify opportunities for technical improvements that increase speed, stability, or developer experience.
- Drive proofs-of-concept and contribute to strategic technical roadmaps.

The skills and experience you will bring

- Experience building and releasing **React-Native mobile apps**.
- Command of JavaScript/TypeScript and mobile design patterns.
- Experience publishing apps to the **App Store and Google Play**.
- Knowledge of **RN architecture** (navigation, state management, animations).
- Experience debugging, profiling, and optimizing RN apps.
- Familiarity with CI/CD for mobile (i.e. Bitrise.)
- Experience integrating with backend services
- Communication skills and ability to work cross-functionally.
- Good English communication skills.
- High adoption of AI IDEs, like Cursor

- Studies: Bachelor's degree or equivalent experience in a quantitative field (Computer Science, Engineering)

Is a plus

- Native development experience (Swift, Objective-C, Kotlin, Java).
- Experience with OTA updates (Expo Updates, CodePush).
- Experience in a startup or high-growth environment.
- Background in automated QA for mobile.
- Experience building shared UI kits or monorepo setups
- Experience building agentic-ai systems with Langchain or Google SDK

Technical Challenge

Introduction

Welcome to the selection process for the **Mobile Engineer** role at Zubale.

At Zubale, our mobile app is the office for thousands of gig workers ("Zubaleros") across Latin America. They work in basements with zero signal, use low-end Android devices, and rely on our app to earn their daily income. A crash, a frozen screen, or a failed data sync isn't just a bug—it means someone doesn't get paid.

We are looking for engineers who understand **Mobile Constraints**: Memory management, Battery consumption, Offline-First architecture, and Native Performance.

Instructions

1. **Choose ONE** of the three challenges below. Pick the one where you can show off your strongest skills.
2. **Tech Stack:** Use **React Native**. You can use Expo or CLI, but you must be able to justify your choice.
3. **State Management:** Use whatever you prefer (Redux, Zustard, Context, MobX, TanStack Query), but handle it cleanly.
4. **Timeframe:** Designed for a weekend. We prioritize quality and architectural decisions over feature quantity.

Evaluation Criteria (The Rubric)

We don't just want pixels on a screen. We evaluate:

1. Architecture & "Offline-First" Mindset (25%)

- **Resilience:** How does the app behave when the network dies unexpectedly? Does it crash? Does it show a generic error? Or does it queue the request?
- **Data Layer:** How do you persist data locally? (SQLite, WatermelonDB, MMKV, Async Storage).
- **Separation of Concerns:** Is your UI decoupled from your business logic?

2. Performance & Optimization (25%)

- **Render Cycles:** Are you re-rendering the whole screen when only a button changes? (We look for useMemo, useCallback usage).
- **List Performance:** Can your list scroll at 60fps on a mid-range Android device?
- **Asset Management:** How do you handle image caching and memory footprint?

3. UX & Polish (25%)

- **Feedback:** Do you use optimistic updates? (The UI updates immediately before the server responds).
- **Native Feel:** Handling the keyboard avoiding view, safe areas, and platform-specific navigation behavior.

4. The "Seniority" Check (25%)

- Did you attempt the **"Bonus Points"** listed in the challenge? These are the hard problems we face daily. Did you choose the easier challenge? We also evaluate how you choose the challenges and how you tackle the problems.

The Challenges (Choose One)

Option 1: The "Offline-First" Auditor (Sync & Persistence)

Context: A Zubalero enters a Walmart to audit products. The signal is lost inside the store. They fill out forms and take photos. When they step outside, everything must sync automatically.

The Mission: Build a **Task Execution App** that works 100% offline.

Requirements:

1. **Local Database:** Use a robust local solution (e.g., WatermelonDB or SQLite) to store a list of tasks and their status.
2. **The Queue:** Implement a "Sync Manager" that queues requests when offline and processes them when online.
3. **UI Feedback:** Show visual indicators for tasks that are "Pending Sync," "Syncing," or "Synced."
4. **Persistency:** If I kill the app and restart it, my pending tasks must still be there.

Bonus Points:

- **Conflict Resolution:** What happens if the server says the task is "Cancelled" but the user locally marked it as "Done"? Implement a strategy to handle this collision.
- **Background Sync:** Try to sync the queue even if the app is in the background (using background fetch or similar).

Option 2: The "Infinite" Marketplace (Performance)

Context: We show thousands of available tasks (Zubales) on a map and list. Rendering 2,000 items with complex filtering kills the battery and frame rate if not optimized.

The Mission: Build a **High-Performance Feed** capable of rendering a dataset of 10,000 items smoothly.

Requirements:

1. **Virtualization:** Use FlashList or an optimized FlatList configuration to render the items (Card with Image + Title + Price + Distance).
2. **Heavy Filtering:** Implement client-side filtering (by Price, Distance, Category) that updates the list instantly without blocking the JS thread.
3. **Image Optimization:** Implement aggressive caching and lazy loading for images. The list should not flicker when scrolling fast.

Bonus Points:

- **Shared Element Transition:** When clicking a card, navigate to the detail screen using a smooth shared element animation.
- **Complex Layout:** Implement a "Pinterest-style" masonry layout or a sticky header implementation that is buttery smooth.

Option 3: The "Secure" Enforcer (Hardware & Context)

Context: We need to prove the user is physically present at a store and is taking a real photo, not uploading a screenshot to commit fraud.

The Mission: Build a **Secure Check-In Module** using Native capabilities.

Requirements:

1. **Geofencing:** The user can only press the "Check In" button if their GPS location is within 500 meters of a specific Target Location.
2. **Custom Camera:** Do not use the system picker. Build a custom camera view.
3. **Watermarking:** When the photo is taken, process the image to "burn" the current Date/Time and Lat/Long coordinates onto the image file (canvas manipulation) before displaying it.

Bonus Points:

- **Mock Location Detector:** Implement a check to warn if the user is using a "Fake GPS" application (Developer Settings enabled or specific libraries).
- **Compass/Orientation:** Show a compass arrow on the camera overlay pointing towards the store location.

Assets for the Challenge

Since this is mobile, you often need to generate your own state, but here are the structures we expect you to simulate.

For Option 1 & 2 (Generate a JSON array with this structure):

```
{
 "id": "task_uuid_v4",
 "title": "Audit Coca-Cola Shelf",
 "price": 50,
 "status": "available",
 "location": {
   "lat": 19.4326,
   "lng": -99.1332,
   "address": "Walmart Buenavista"
 },
 "image_url": "https://picsum.photos/200/300",
 "expires_at": "2025-12-31T23:59:59Z"
}
```

For Option 3 (Target Location to hardcode):

- **Target:** Pick a coordinate near your current location so you can test the Geofence logic easily (or allow the user to set the "Target" pin on a map first).

Deliverables

Please send us an email to eng-hiring@zubale.com with:

1. **Link to Repository:** (GitHub/GitLab).

2. **APK (Android) or TestFlight/Simulator Build:** We want to install it and feel the performance.

3. **Video Demo (Loom/YouTube):** Show us the offline mode (toggle airplane mode) or the scroll performance (FPS monitor on).

Instructions: A README.md explaining how to run it.

Good luck. We can't wait to see what you build.