OMS App Summary - Didacticos Jugando y Educando

Evidence source: repository files only (README, src, server, supabase).

What it is

A React + TypeScript order management system (OMS) that centralizes Mercado Libre and Wix orders in one operational dashboard. It lets staff sync orders, monitor status, filter/search, inspect details, and keep status-history audit records.

Who it's for

Primary persona: order operations staff at Didacticos Jugando y Educando (inferred from dashboard labels, manual sync buttons, and status workflow). Explicit persona definition: Not found in repo.

What it does

- Shows a unified orders table for channels mercadolibre and wix.
- Syncs Mercado Libre and Wix orders via dedicated actions in the dashboard.
- Normalizes channel-specific payloads into one common order schema.
- Supports order filtering by status, channel, and text search (ID, nickname, email).
- Supports order status transitions: nuevo, preparando, listo, enviado, cancelado.
- Stores every status change in order_status_history for traceability.
- Handles Mercado Libre pack and shipping fields (pack_id and shipping_id) when present.

How it works (repo-evidenced architecture)

UI layer	React app with main entry, app container, and dashboard page using React Query state management.
Data hooks	useOrders reads from Supabase. useSyncML and useSyncWix call local proxy endpoints, then upsert into Supabase.
Proxy service	Express server exposes two sync endpoints and a health endpoint, and calls external APIs server-side to avoid browser CORS issues.
External APIs	Mercado Libre Orders API with OAuth token refresh, plus Wix eCommerce Orders search API.
Data store	Supabase PostgreSQL with orders and order_status_history tables, indexes, updated_at trigger, and permissive MVP RLS policies.
Flow	Operator clicks sync. UI hook calls local proxy. Proxy fetches external orders. Data is normalized and upserted. Orders query is invalidated and UI refreshes.
Gaps	Auth and user-role model, plus production deployment topology: Not found in repo.

How to run (minimal getting started)

- npm install
- Create .env.local from .env.example and fill Supabase, Mercado Libre, and Wix variables.
- Apply DB schema by running supabase/schema.sql in Supabase SQL Editor.
- Start proxy server for sync endpoints: npm run server
- Start frontend: npm run dev and open http://localhost:5173