

Roadmap Optiflow – MVP « ERP IA »

But : construire en 4 semaines un ERP de stock basique enrichi de 6 blocs IA (prévision, alertes, recommandations, chat, scoring, XAI). Chaque tâche précise l'outil principal, l'info utile et la durée cible.

Phase 0 : Mise en place (Jour 0-1)

ID	Tâche	Outil	Infos utiles	Durée
0.1	Créer repo stock-erp-ai sur GitHub (main + dev)	Git/ GitHub	Ajouter <code>.editorconfig</code> , licence MIT	1 h
0.2	Installer Cursor + activer <i>Agent Mode</i> (models : GPT-4o, Gemini 2.5, Claude Opus)	Cursor	Fichier <code>.cursor/rules.yaml</code> pour conventions code	1 h
0.3	Installer Supabase CLI & créer projet <code>optiflow_dev</code>	Supabase	Stockage Postgres + Edge Functions gratuits	1 h
0.4	Lancer Claude Code (Docker ou OrbStack)	Claude Code	Notebook Python 3.12 + libs : pandas, prophet, scikit-learn	1 h

Phase 1 : Vertical Slice ① – CRUD de base (Jour 1-3)

- **1.1 Schéma BDD** – Définir tables `products`, `stock_movements`, `suppliers`, `sales` \ *Outil* : Supabase SQL Editor via Cursor prompt \ *Livrable* : première migration commitée.
- **1.2 API** – Générer routes tRPC (+ Zod) pour CRUD complet \ *Outil* : Cursor Agent (« Generate tRPC routers for products ») \ *Durée* : 0,5j.
- **1.3 Interface** – Page React/Next.js avec shadcn/ui + TanStack Table \ *Outil* : Cursor UI generator \ *Durée* : 1j.
- **1.4 Seed & tests** – Script CSV → Supabase + tests Postman \ *Outil* : Claude Code pour le script, Postman \ *Durée* : 0,5j.

Phase 2 : Vertical Slice ② – Prévisions de demande (Jour 4-6)

- **2.1 Import données historiques** (CSV ventes) dans notebook Claude \ *Outil* : Claude Code \ *Info* : prévoir colonnes `product_id`, `date`, `quantity`.
- **2.2 Prototype modèle** – ARIMA & Prophet ; sélectionner celui avec MAPE < 15 % \ *Livrable* : `forecast_model.pkl` + README modèle.
- **2.3 Export modèle** dans Storage Supabase \ *Durée* : 0,5j.
- ****2.4 Edge Function **** – TypeScript wrapper qui charge le modèle \ *Outil* : Supabase Edge Functions via Cursor \ *Durée* : 0,5j.

- **2.5 UI graphique** – Recharts, courbe + intervalle confiance\ *Durée* : 0,5j.

Phase 3 : Vertical Slice ③ – Alertes & suggestions (Jour 7-9)

- **3.1 Cron Supabase** – Job nocturne qui calcule : rupture < 7j, sur-stock > 200 % forecast\ *Durée* : 0,5j.
- **3.2 Table** – stocker type, message, `is_read` \ *Durée* : 0,25j.
- **3.3 Fonction** – formule stock_sec + forecast – stock_actuel\ *Outil* : Cursor (TypeScript helper)\ *Durée* : 0,25j.
- **3.4 Badge UI + liste alertes** \ *Durée* : 0,5j.

Phase 4 : Vertical Slice ④ – Chat IA & XAI (Jour 10-12)

- **4.1 Vectoriser docs & descriptions** – pgvector extension\ *Outil* : Supabase SQL + Python embedding model\ *Durée* : 0,5j.
- **4.2 Endpoint `` (RAG)** – LangChain (Claude) pour mixer contexte + LLM\ *Durée* : 1j.
- **4.3 Prompt template** : « Explain reasoning, confidence score »\ *Outil* : Cursor\ *Durée* : 0,25j.
- **4.4 Front ChatBox** – composant Stream UI\ *Durée* : 0,25j.

Phase 5 : Finition & déploiement (Jour 13-14)

- **5.1 CI/CD** – GitHub Actions : lint, test, build, deploy Vercel\ *Durée* : 0,5j.
- **5.2 Migration Supabase prod** – `supabase db push` \ *Durée* : 0,25j.
- **5.3 Tests utilisateur** – checklist fonctionnelle + récolte feedback\ *Durée* : 0,5j.
- **5.4 Documentation finale** – Générée par Cursor (`docs/`) + export PDF\ *Durée* : 0,25j.

Phase 6 : Améliorations post-MVP (optionnel)

- Optimisation seuils dynamiques via Reinforcement Learning léger.
- Tableau de bord cash-flow prévisionnel.
- Scénarios alternatifs + simulateur de prix.

Synthèse calendrier (14 jours ouvrés)

J0-1	: Phase 0
J1-3	: Phase 1
J4-6	: Phase 2
J7-9	: Phase 3
J10-12	: Phase 4
J13-14	: Phase 5

Règle vibecoding : après **chaque** tâche, commit + test rapide + micro-documentation.