Group 1 – Foundation & Project Skeleton (1–5)

1. Create Git repo and base project structure (FastAPI, main app package, folders)
2. Add settings/config module and `.env` files for dev/prod
3. Configure SQLAlchemy + Alembic and initial empty migration
4. Set up Jinja templates, static files and base Bootstrap layout
5. Implement health-check endpoint and simple home/login placeholder

Group 2 – Users, Roles & Branches (6–10)

6. Model `users` and `sucursales` tables with roles and states
7. Implement password hashing and authentication service
8. Implement session-based login/logout flow
9. Create basic admin UI to manage users and branches
10. Seed original super admin and initial branches

Group 3 – Materials & Price Tables (11–15)

11. Model `materiales` and `tablas_precios` with versioning
12. Implement CRUD endpoints and UI for materials
13. Implement CRUD for price tables (COMPRA/VENTA, regular/mayorista/menudeo)
14. Add permission checks and audit hooks for price changes
15. Seed a small catalog of materials and default price tables

Group 4 – Partners: Suppliers & Clients (16–20)

16. Model `proveedores` and `clientes` tables
17. Implement CRUD and search UI for partners

18. Add “quick create partner” from worker note screen

19. Validate uniqueness (plates/contact) and active/inactive status

20. Wire partner selection into new-note form (COMPRA/VENTA)

Group 5 – Weighing Notes Core (21–25)

21. Model `notas`, `nota_materiales` y `subpesajes` entities

22. Build worker UI to create draft purchase notes with sub-weighings

23. Implement transitions BORRADOR → EN_REVISION with snapshot `nota_original_trabajador`

24. Integrate S3 upload for sub-weighing photos and link to `fotos` table

25. Build admin and super admin listings/detail views for notes

Group 6 – Approval, Inventory & Cancellations (26–30)

26. Implement admin review UI for notes in EN_REVISION

27. Implement approval logic (APROBADA) with recalculated totals and final prices

28. Model `inventario_actual` and `movimientos_inventario` tables

29. Hook approval into inventory movements (ENTRADA/SALIDA) and update balances

30. Implement cancellation flow with reverse inventory movements and state rules

Group 7 – Accounting & Payments (31–35)

31. Model `cuentas_financieras` and admin UI to manage them

32. Model `movimientos_contables` and `pagos_movimiento` tables

33. Wire note approval into accounting movement creation (COMPRA_EXTERNA/VENTA_EXTERNA)

34. Implement cancellation reversals (REVERSO) linked to original movements

35. Implement super admin accounting report query with filters (sucursal, fechas, tipo)

Group 8 – Reports, Audit & Security (36–40)

36. Implement CSV/Excel export for accounting report
37. Model `audit_log` and hook into key actions (notas, inventario, precios, usuarios)
38. Harden permissions by role and branch, including session durations
39. Add logging, error handling and custom 403/404/500 pages
40. Build simple dashboards for admins/super admins with core KPIs

Group 9 – UX Polish, Mobile UI & Testing (41–45)

41. Optimize worker note UI for mobile (forms, buttons, validation, flows)
42. Add filters and status badges for notes lists (BORRADOR, EN_REVISION, APROBADA, CANCELADA)
43. Write unit tests for models, services and main business rules
44. Add integration/E2E tests for purchase, sale and cancellation flows
45. Draft quickstart guides for workers, admins and super admins

Group 10 – Ops, Backups & Deployment (46–50)

46. Configure S3 bucket, IAM, and lifecycle rules for photos in prod
47. Provision Heroku app, Heroku Postgres and config vars for prod
48. Add Alembic migration commands and startup scripts for Heroku
49. Configure automated DB backups and basic monitoring/logging
50. Deploy final MVP to Heroku, run smoke tests and tag v1.0 release