

Last indexed: 11 August 2025 (172577)

Overview

Repository Structure
Getting Started
Project 1: Product Management (Proy1)
Database & Configuration
Product Management System
Development Environment
Project 2: Course Platform (Proy2_Cursos)
Data Models & Database
Authentication & Middleware
Development & Testing
Project 3: Food Delivery Platform (Proy3_Pedidos)
Architecture & Dependencies
Database Models & Schema
Environment & Configuration
Development Environment
VS Code & Debugging
Dependency Management

Repository Structure

> Relevant source files

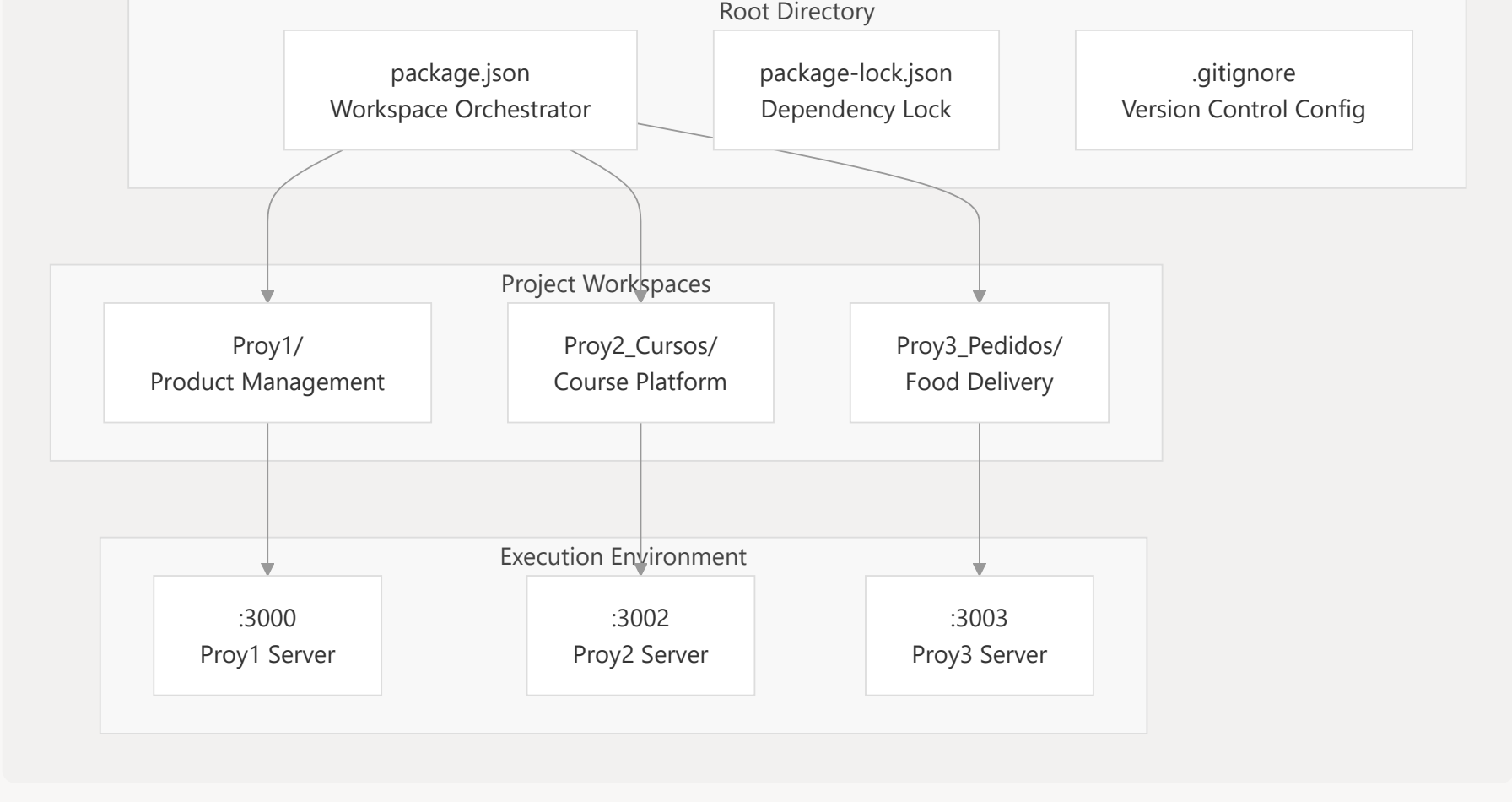
This document describes the organizational structure of the IronHack Course 2 monorepo, including its npm workspaces configuration, project layout, and orchestration mechanisms. The repository contains three independent web applications managed as a unified development environment.



For information about individual project implementations, see [Project 1: Product Management](#), [Project 2: Course Platform](#), and [Project 3: Food Delivery Platform](#). For development environment setup details, see [Development Environment](#).

Monorepo Architecture

The repository uses npm workspaces to manage multiple projects within a single repository structure. This approach enables shared dependency management while maintaining project independence.

Repository Organization Diagram



Sources:  package.json 13-16  .gitignore 1-2

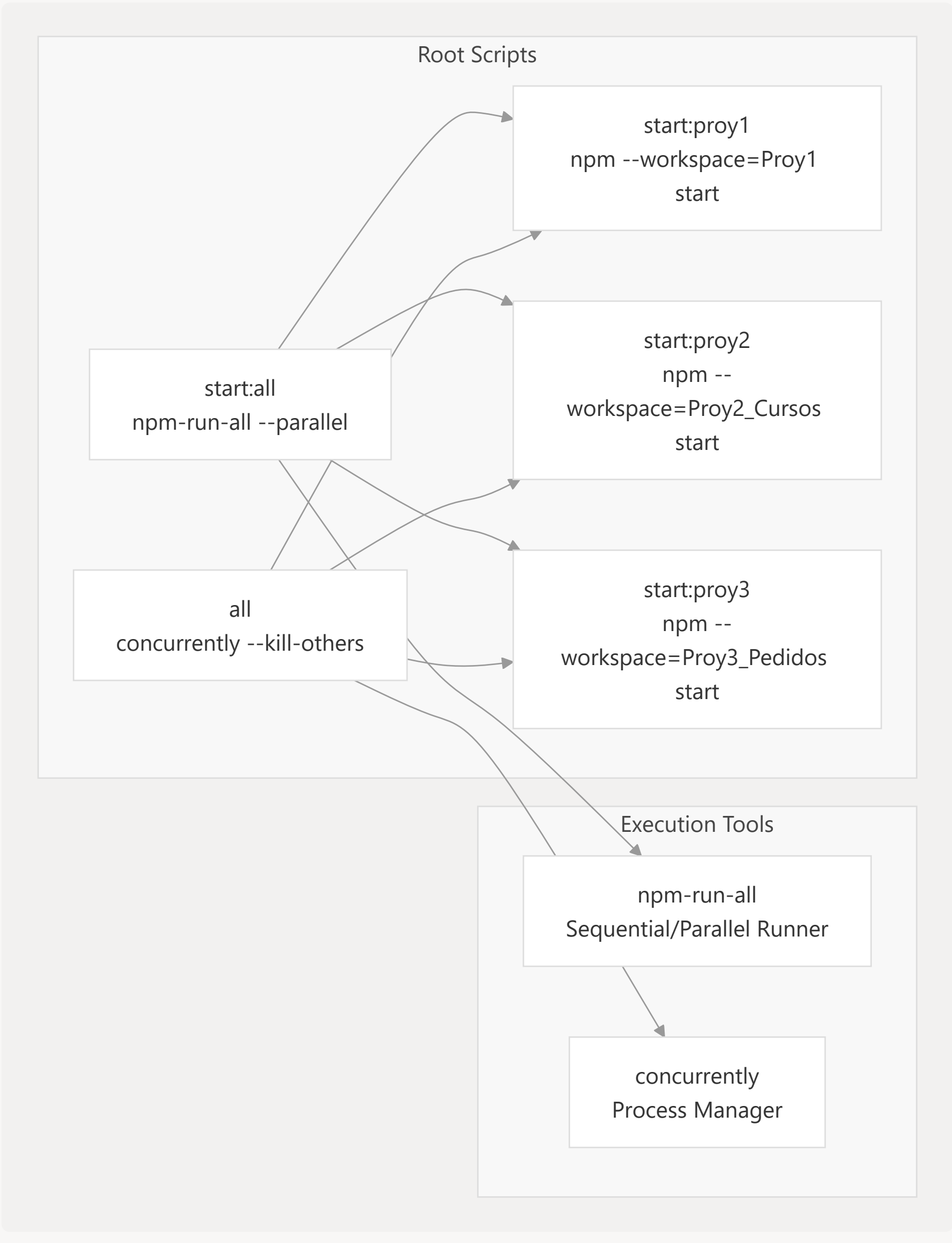
Workspace Configuration


The root `package.json` defines the workspace structure and provides orchestration scripts for managing all projects.

Workspace Definition

Property	Value	Purpose
workspaces	["Proy1", "Proy2_Cursos", "Proy3_Pedidos"]	Defines child project directories
name	"ironhack-course-2"	Root package identifier
version	"1.0.0"	Semantic version

Script Orchestration Diagram



Sources:  package.json 7-11  package.json 21-25

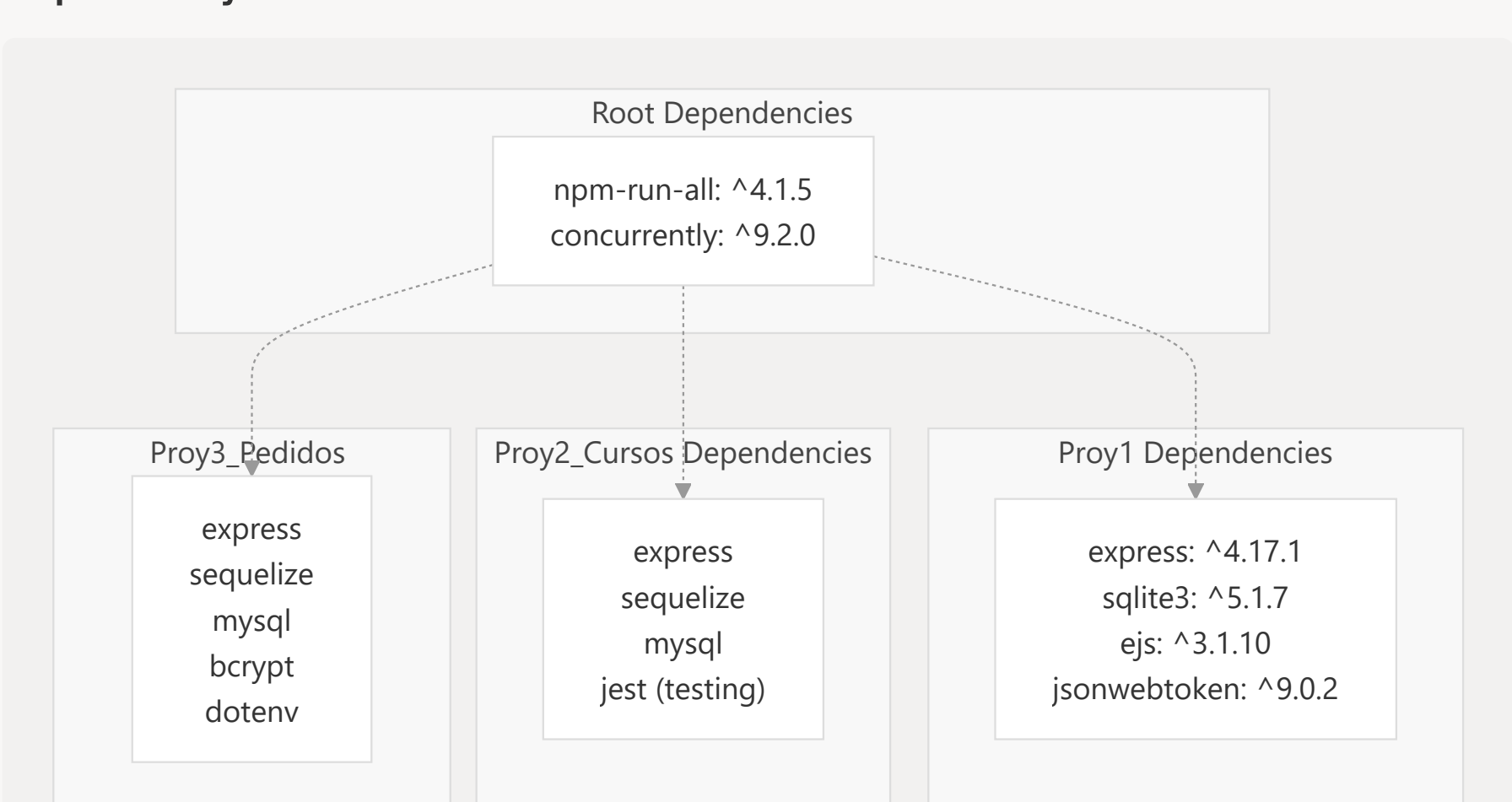
Project Directory Structure



Each workspace contains an independent Node.js application with its own `package.json` and dependency tree.

Project Layout Table

Directory	Project Name	Technology Stack	Port
Proy1/	Product Management	Express + SQLite + EJS	3000
Proy2_Cursos/	Course Platform	Express + MySQL + Sequelize	3002
Proy3_Pedidos/	Food Delivery (FoodExpress)	Express + MySQL + Sequelize + Multi-Role Auth	3003

Dependency Isolation



Sources:  package-lock.json 11-16  package-lock.json 2233-2251

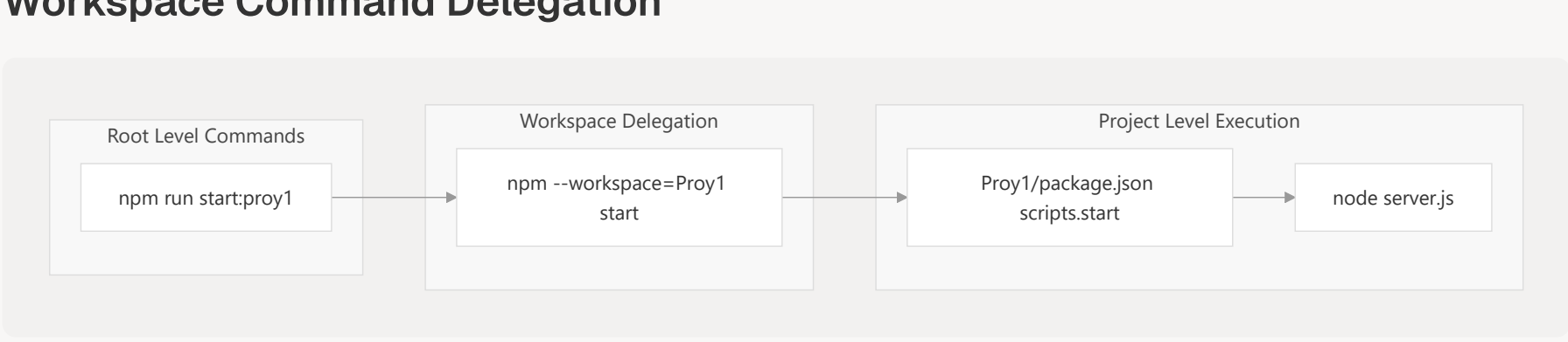
Orchestration Mechanisms


The repository provides multiple execution strategies for development and testing workflows.

Script Execution Patterns

- Individual Project Execution**
 - `npm run start:proy1` - Runs only Product Management system
 - `npm run start:proy2` - Runs only Course Platform
 - `npm run start:proy3` - Runs only Food Delivery system
- Parallel Execution**
 - `npm run start:all` - Uses `npm-run-all --parallel` for coordinated startup
 - `npm run all` - Uses `concurrently --kill-others` for process management

Workspace Command Delegation



Sources:  package.json 7

Version Control Configuration

The `.gitignore` file manages version control exclusions at the repository level.

Exclusion Patterns

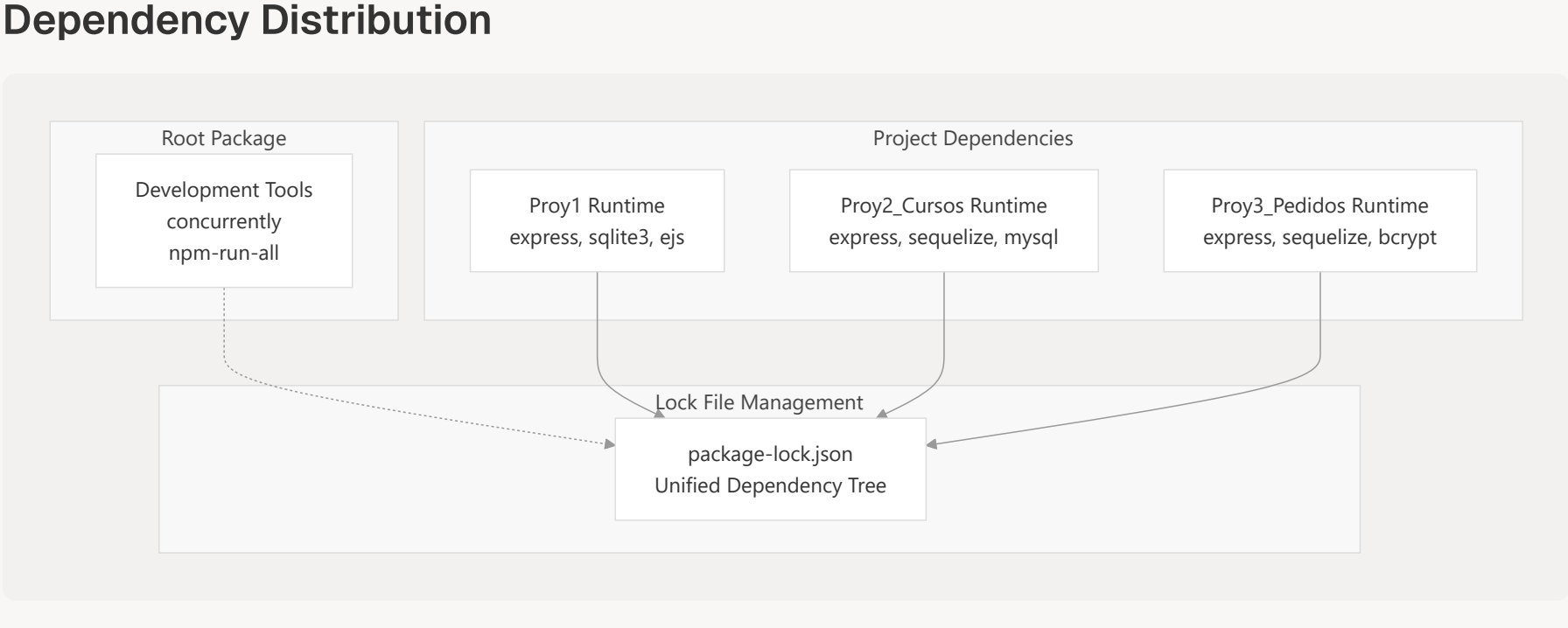
Pattern	Scope	Purpose
/node_modules	Global	Excludes all npm dependencies
/Proy1/db/SQLite_editor_win	Proy1-specific	Excludes SQLite GUI tools


Sources:  .gitignore 1-2

Dependency Management Strategy

The monorepo uses npm workspaces for dependency isolation while sharing common development tools at the root level.

Dependency Distribution



Sources:  package.json 20-25  package-lock.json 1-26

This structure enables independent development of each project while maintaining coordinated dependency management and simplified execution workflows through the root-level orchestration scripts.