Development Environment

> Relevant source files

This document covers the comprehensive development environment setup for the IronHack Course 2 monorepo, including Docker containerization, VS Code configuration, and workspace management. The development environment is designed to support all three projects (Proy1, Proy2_Cursos, and Proy3_Pedidos) within a unified workspace while maintaining individual project configurations.

For project-specific setup instructions and running individual applications, see Getting Started. For VS Code debugging and workspace configuration details, see VS Code & Debugging. For npm workspace dependency coordination, see Dependency Management.

DevContainer Architecture

The repository provides a Docker-based development environment using Visual Studio Code DevContainers, ensuring consistent development setup across different machines and operating systems.

Container Infrastructure Overview

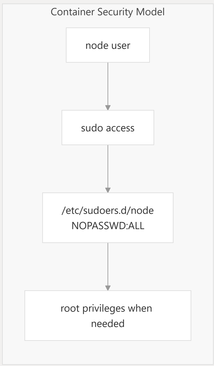
Sources: `.devcontainer/devcontainer.json` `1-8` `.devcontainer/Dockerfile` `1-13`

Docker Configuration Details

The DevContainer is built on `node:20` base image with essential development tools and proper user permissions configured for seamless development experience.

Component	Configuration	Purpose
Base Image	`node:20`	Latest stable Node.js runtime
Dev Tools	`less`, `man-db`, `sudo`	Command-line utilities for development
User Setup	`node` user with `NOPASSWD:ALL` sudo	Secure container access without password prompts
Environment	`DEVCONTAINER=true`	Environment detection for development-specific configurations

The container configuration ensures that the `node` user has appropriate permissions while maintaining security:

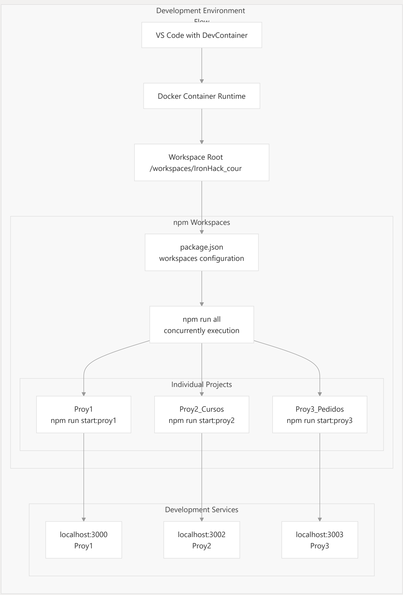


Sources: `.devcontainer/Dockerfile` `6-9`

Workspace Structure Integration

The development environment integrates with the npm workspaces structure to provide unified dependency management and project execution capabilities.

Development Workflow Integration



Sources: `.devcontainer/devcontainer.json` `3` `.devcontainer/Dockerfile` `11-12`

Environment Variables and Configuration

The development environment supports environment-specific configurations through the `DEVCONTAINER` environment variable, which can be used by applications to detect containerized development environments.

Container Environment Detection

The development environment sets `DEVCONTAINER=true` to enable development-specific behaviors:

Environment Variable	Value	Usage
`DEVCONTAINER`	`true`	Indicates running in development container
`NODE_ENV`	(inherited/configurable)	Node.js environment mode
`USERNAME`	`node`	Container user for development

This environment detection allows applications to:

- Enable development-specific logging
- Use different database configurations for development
- Activate hot-reloading and debugging features
- Configure different security settings for development

Sources: `.devcontainer/Dockerfile` `11-12` `.devcontainer/Dockerfile` `7-9`