Carlos Guadalupe López Trejo

Diagramas

NAO ID: 3323

11/11/2025

Bécalos TechnoReady In-México 2025

Java and JavaScript. Programming Procedures

Version 1.0.0

Índice

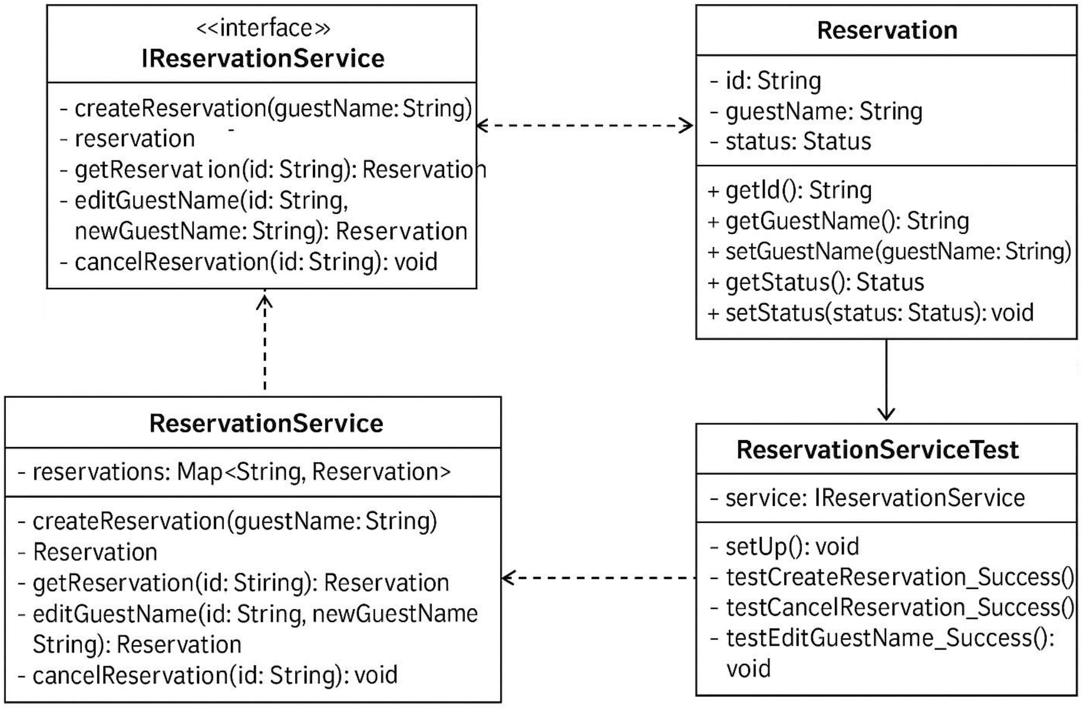
Diagramas 3

Diagramas

Java Module - Class Diagram

The diagram above illustrates the class structure of the Java reservations module, designed following Object-Oriented Programming (OOP) principles.

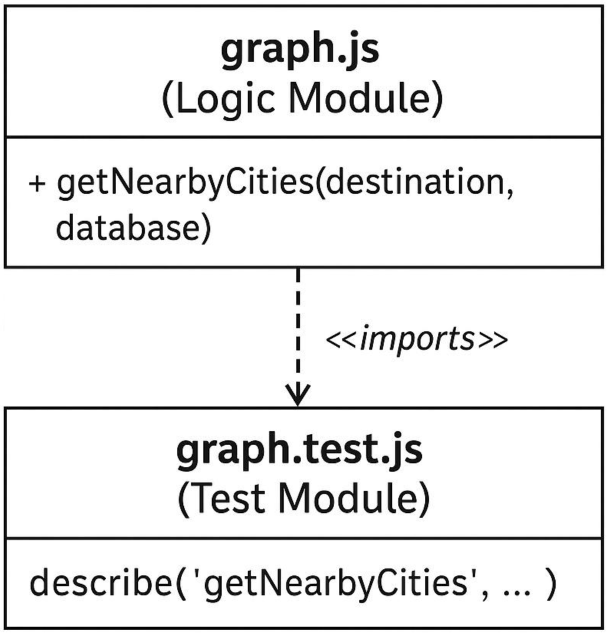
- The **IReservationService** interface defines the public contract, abstracting the business logic.
- The **ReservationService** class provides the concrete in-memory implementation of this contract.
- The **Reservation** class acts as the data model for the application.
- The **ReservationServiceTest** class depends on the IReservationService interface, not the concrete class, which is a best practice that facilitates decoupled and maintainable tests.



JavaScript Module - Module Interaction Diagram

This diagram shows the simple yet effective architecture of the JavaScript module. As this module follows a functional approach, a module interaction diagram is used instead of a class diagram.

- The **graph.js** file encapsulates the core business logic within the getNearbyCities function.
- The **graph.test.js** file contains the Jest test suite, which imports the logic from graph.js to validate its behavior, demonstrating a clear separation between the implementation and its tests.



Development and Testing Workflow

The flowchart above outlines the standardized development and quality assurance workflow followed throughout the project for both the Java and JavaScript modules.

This iterative process ensures that all code is validated against a comprehensive test suite and meets the required code coverage targets before being considered complete. The cycle of **Test -> Fail -> Fix -> Pass** was central to this workflow, especially during the bug discovery and resolution phase documented in the README.md file.