



OpenWebinars certifica que

JORDY IVAN SAN MARTIN PONCE

Ha superado con éxito

**Curso de técnicas de Refactoring
y Clean Code**

Duración del curso

10 horas

Fecha de expedición

28 abril 2023

A handwritten signature in black ink, appearing to read 'Manuel Agudo', positioned over a large, light gray watermark of the OpenWebinars logo.

CEO de OpenWebinars

Manuel Agudo

Contenido

Curso de técnicas de Refactoring y Clean Code

1. INTRODUCCIÓN

Presentación

Refactoring

Clean Code

Technical Debt

¿Cuándo refactorizar?

¿Cómo refactorizar?

2. CATALOG

Code Smells to Refactor

Refactoring techniques

3. CODE SMELLS

Bloaters

Object Orientation Abusers

Change Preventers

Dispensables

Couplers

Other Smells

4. REFACTORING COMPOSING METHODS

Extract Method

Inline Method

Extract Variable

Inline Temp



Replace Temp with Query

Split Temporary Variable

Remove Assignments to Parameters

Replace Method with Method Object

Substitute Algorithm

5. REFACTORING MOVING FEATURES BETWEEN OBJECTS

Move Method

Move Field

Extract Class

Inline Class

Hide Delegate

Remove Middle Man

Introduce Foreign Method

Introduce Local Extension

6. REFACTORING ORGANIZING DATA

Self Encapsulate Field

Replace Data Value with Object

Change Value to Reference

Change Reference to Value

Replace Array with Object

Duplicate Observed Data

Change Unidirectional Association to Bidirectional

Change Bidirectional Association to Unidirectional

Replace Magic Number with Symbolic Constant

Encapsulate Field

Encapsulate Collection

Replace Type Code with Class

Replace Type Code with Subclasses

Replace Type Code with State-Strategy

Replace Subclass with Fields

7. REFACTORIZING SIMPLIFYING CONDITIONAL EXPRESSIONS

Decompose Conditional

Consolidate Conditional Expression

Consolidate Duplicate Conditional Fragments

Remove Control Flag

Replace Nested Conditional with Guard Clauses

Replace Conditional with Polymorphism

Introduce Null Object

Introduce Assertion

8. REFACTORIZING SIMPLIFYING METHOD CALLS

Rename Method

Add Parameter

Remove Parameter

Separate Query from Modifier

Parameterize Method

Replace Parameter with Explicit Methods

Preserve Whole Object

Replace Parameter with Method Call

Introduce Parameter Object

Remove Setting Method

Hide Method

Replace Constructor with Factory Method

Replace Error Code with Exception

Replace Exception with Test

9. REFACTORIZING DEALING WITH GENERALIZATION

Pull up Field

Pull up Method

Pull Up Constructor Body

Push Down Method

Push Down Field

Extract Subclass

Extract Superclass

Extract Interface

Collapse Hierarchy

Form Template Method

Replace Inheritance with Delegation

Replace Delegation with Inheritance

10. CLEAN CODE

Naming

Variables

Functions

Objects, Data Structures and Classes

Testing

Concurrency

Error Handling

Comments

