



OpenWebinars certifica que

SUILEN HERNANDEZ ALVARADO

Ha superado con éxito

**Clean Code: Mejorando tu
eficiencia y productividad**

Duración de la ruta

26 horas

Fecha de expedición

15 octubre 2024

A handwritten signature in black ink, likely belonging to Manuel Agudo, the CEO of OpenWebinars.

CEO de OpenWebinars

Manuel Agudo

Contenido

Clean Code: Mejorando tu eficiencia y productividad

1. Curso de Clean Code

1. Herramientas

Presentación del curso y del profesor

GIT: Control de versiones

Entornos de desarrollo

Pruebas unitarias - JUnit

2. Clean Code

Introducción

Comentarios

Nombres

Métodos

Clases

Arquitectura

3. Refactorización

Introducción

Code Smells I



Code Smells II

Renombrar variables y métodos

Ocultar propiedades y métodos

Magic Numbers

Extraer método

Método Inline

Convertir parámetros en objetos

Replace Temp with Query

Introducir variable explicativa

Separar variable temporal

Borrar las asignaciones a parámetros

Reemplazar método con objeto de método

Descomponer condicional

Consolidar expresión condicional

Consolidar fragmentos de condicional duplicados

Eliminar bandera de control

Reemplazar condicionales anidadas con cláusulas guard

Reemplazar condicional con polimorfismo

Introducir objeto nulo

Separar consulta de modificación

Parametrizar método

Reemplazar parámetro con métodos explícito

Sustituir algoritmo

Extraer clase

Inline Class

Ocultar delegado

Introducir método externo

Introducir extension local

Reemplazar un dato con un objeto

Encapsular colección



Pull Up

Push Down

Reemplazar array con objeto

4. Patrones de Software

Introducción

Creacionales singleton

Creacionales factory

Creacionales abstract factory

Creacionales builder

Creacionales Prototype

Estructurales adapter

Estructurales bridge

Estructurales composite

Estructurales decorator

Estructurales Façade

Estructurales flyweight

Estructurales proxy

Comportamiento Chain Of Responsibility

Comportamiento Command

Comportamiento interpreter

Comportamiento iterator

Comportamiento mediator

Comportamiento memento

Comportamiento observer

Comportamiento State

Comportamiento Strategy

Comportamiento Template Method

Comportamiento Visitor

5. Principios SOLID

Introducción



SRP: Single Responsibility Principle

OCP: Open/Closed Principle

LSP: Liskov Substitution Principle

ISP: Interface Segregation Principle

DIP: Dependency Inversion Principle

Principios LEAN

2. 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. Refactoring 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. Refactoring 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. Refactoring 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

3. Curso de Extreme Programming (XP): Herramientas para mejorar las entregas de los equipos ágiles

1. Introducción

Presentación

¿Por qué Extreme Programming?

¿Qué es Extreme Programming?

Valores

Principios

2. Artefactos en XP

Historias de usuario

Pruebas de aceptación

Estimación

Pruebas unitarias y de integración

Plan de entrega

Plan de iteración

3. Pair Programming

¿Qué es Pair Programming?

Mejores prácticas para realizar Pair Programming

4. TDD

¿Qué es TDD?

Mejores prácticas para realizar TDD

5. Integración Continua

¿Qué es Integración Continua?

Mejores prácticas para realizar Integración Continua

6. Refactoring

¿Qué es Refactoring?

Mejores prácticas para realizar Refactoring

7. Feedback Loops

Reuniones Stand Up

Iteraciones

Releases

Tipos de planificación

La importancia de la Mejora Continua

4. Clean Commits en proyectos NPM

1. Taller

Clean Commits en proyectos NPM

5. Aumenta tu productividad con Visual Studio Code

1. Taller

Aumenta tu productividad con Visual Studio Code

6. Aumenta tu productividad con IntelliJ

1. Taller

Aumenta tu productividad con IntelliJ

