

ESCUELA POLITECNICA NACIONAL

NOMBRE: Juan Mateo Quisilema

CURSO: GR2SW

FECHA: 12 de diciembre de 2025

Taller: Aplicando Principios de Código Limpio en Proyectos Reales

Objetivo

Que los estudiantes comprendan los principios del código limpio y los apliquen analizando código real de repositorios públicos, identificando mejoras y proponiendo refactorizaciones.

Estructura del Taller

1. Introducción

- **¿Qué es Código Limpio y por qué es importante?**
El código limpio es fácil de entender, modificar y mantener. Se basa en principios de diseño que mejoran la legibilidad y reducen la complejidad del código.
- **Impacto en mantenibilidad, escalabilidad y trabajo en equipo:**
 - El código limpio mejora la mantenibilidad al hacerlo más comprensible.
 - Facilita la escalabilidad mediante una estructura modular y reutilizable.
 - Promueve el trabajo en equipo, ya que todos los miembros pueden entender rápidamente el código.
- **Ejemplos de “código sucio” vs “código limpio”**
 - **Código sucio:** Largo, desordenado, sin comentarios claros, con funciones largas y nombres ambiguos.
 - **Código limpio:** Funciones pequeñas y claras, nombres descriptivos, y ausencia de código redundante.

Principios clave

- **Reglas generales de diseño:**
 - La simplicidad es clave. El código debe ser lo más sencillo posible para que se pueda mantener fácilmente.
- **Comprendibilidad y nombres:**
 - Usa nombres descriptivos para variables y funciones.
 - Evita abreviaciones innecesarias.

- **Funciones y comentarios:**
 - Las funciones deben ser pequeñas y hacer solo una cosa.
 - Los comentarios deben explicar el “por qué” del código, no el “cómo”.
- **Organización de código, objetos y estructuras:**
 - Organiza el código de forma que tenga sentido y siga un patrón lógico.
- **Pruebas y olores de código:**
 - Realiza pruebas unitarias para asegurarte de que el código funciona como se espera.
 - Identifica olores de código, como la duplicación, funciones largas o variables ambiguas.

2. Actividad Principal

“Detectives de Código”

Paso 1: Elección del Repositorio

Cada grupo (2-3 estudiantes) elige un repositorio público en GitHub/GitLab.

Ejemplos de repositorios:

Lenguaje	Repositorio	Descripción	Plataforma
Python	awesome-python	Lista curada de librerías y recursos útiles en Python.	GitHub
	tensorflow	Framework de aprendizaje automático desarrollado por Google.	GitHub
JavaScript	react	Biblioteca para construir interfaces de usuario.	GitHub
	javascript-algorithms	Algoritmos y estructuras de datos en JS.	GitHub
Java	developer-roadmap	Hoja de ruta para desarrolladores en varios lenguajes.	GitHub
C++	recursos-programacion	Recursos educativos para C++ y otros lenguajes.	GitHub
Go	recursos-programacion	Tutoriales y documentación para aprender Go.	GitHub
Rust	Rust_by_Example	Ejercicios prácticos en Rust.	GitLab

Lenguaje	Repository	Descripción	Plataforma
PHP	afip.php	Librería para usar los Web Services de AFIP.	GitHub
Kotlin	one-day-one-language	Introducción rápida a varios lenguajes, incluido Kotlin.	GitHub
Swift	recursos-programacion	Recursos para aprender Swift.	GitHub
SQL	recursos-programacion	Libros y tutoriales para aprender SQL.	GitHub
Multi-lenguaje	freeCodeCamp	Plataforma educativa con certificaciones en varios lenguajes.	GitHub