



SECRETARÍA DE
INNOVACIÓN



CURSO CLEAN CODE

Duración:
20 horas

Facilitador: Carlos Martínez Zaragoza



DESCRIPCIÓN DEL CURSO

101

La importancia del Clean Code radica en la legibilidad del código. Nuestro código debe ser simple y leerse con la misma facilidad que un texto bien escrito. Un buen programador debe escribir un código que pueda ser leído por cualquier otro programador, no sólo que los ordenadores puedan entender

OBJETIVOS DEL CURSO

102

- Aprender a definir variables, constantes, funciones y objetos asignando un nombre representativo
- Conocer los principios SRP y OCP para trabajar con funciones

INDICADORES DE LOGRO

103

- Comprensión de la importancia de programas escribiendo código que sea leído por cualquier otro programador
- Capacidad de definir variables y constantes utilizando nombres representativos
- Determinar los buenos formatos para escribir código

METODOLOGÍA

104

Desarrollo de evento en manera asíncrona, de acuerdo con los contenidos establecidos de manera cronológica, con presentaciones con videos y fundamentando los contenidos con guías instruccionales, con ejercicios para fundamentar los conceptos

ACTIVIDADES DE APRENDIZAJE

105

- Exposición magistral y dialogada
- Lectura de guías y resolución de discusión de los aspectos claves analizados en el curso
- Se desarrollan preguntas y respuestas que abran el espacio para interacción y comunicación
- Trabajo práctico para aplicar técnicas de comunicación intergrupal
- Análisis de casos para aplicar técnicas de comunicación intergrupal

CONTENIDO GENERAL

106

- Introducción a Clean Code
- La importancia de los nombres
- Funciones
- Escribe comentarios como un profesional
- Formatear código para facilitar la lectura
- Los objetos y estructura de datos
- Mejorar la gestión de errores
- Las fronteras del código

REQUISITOS DE APROBACIÓN DEL CURSO

- Participación en plataforma LMS
- Elaboración de las guías instruccionales de trabajo

Obtener una nota mayor de

6.0

en la evaluación final

CRONOLOGÍA DEL CURSO

107

Sesión	Contenido	Descripción	Recursos
Sesión 1 Duración 4 horas	Examen de conocimientos previos Tema 1: Importancia del Clean Code - Hágase el código - Código Incorrecto - El costo total de un desastre - El gran cambio de diseño - El arte del código limpio - Concepto de código limpio - Somos autores	- Codificación de los sistemas - Código sin tanta vuelta - El costo de repetición de códigos incorrecto - Hágalo bien desde la primera vez y limpio	- Plataforma para cuestionario en línea - Presentación en PowerPoint - Guías instruccionales - Videos de complemento
	Tema 2: La importancia de los nombres - Usar nombre que revelen la intención - Evitar la desinformación - Usar nombre que se puedan pronunciar - Usar nombres que se puedan buscar - Evitar codificaciones - Evitar asignaciones mentales - Nombres de clase, métodos - No haga juego de palabras - Añadir contexto con sentido	- Utilizar nombres fáciles y sin tanta vuelta - No acortar o abreviar nombres - Nombres fáciles de comprender e identificar de que se trata - Nombre que sean de acuerdo con el aplicativo que desarrolla - Dejar de utilizar nombres mentales propios	- Presentación en PowerPoint - Guías instruccionales - Videos de complemento
Sesión 2 Duración 4 horas	Tema 3: Funciones - Tamaño reducido - Hacer una cosa - Un nivel de abstracción por función - Instrucciones switch - Argumento de funciones - Sin efectos secundarios - Separación de consultas de comandos - Mejor excepción que devolver código error - No repetirse - Programación estructurada	- Tipos de funciones - Aplicación de bloque y sangrados - Secciones en funciones - Reglas descendentes - Aplicar formato monódicas habituales - Triadas - Objetos y listas Argumentos - Verbos y palabras claves	- Presentación en PowerPoint - Guías instruccionales - Videos de complemento
	Tema 4: Escribe comentarios como un profesional - Los comentarios no compensan el código correcto - Comentarios de calidad - Comentarios Incorrectos - Configurar zonas de búsqueda directa e inversa	- Explicarse en el código - Comentarios legales - Comentarios informativos - Comentarios obligatorios - Comentarios periódicos - Diferentes tipos de comentarios	- Presentación en PowerPoint - Guías instruccionales - Videos de complemento

Sesión	Contenido	Descripción	Recursos
Sesión 3 Duración 4 horas	Tema 5: Formatear código para facilitar la lectura - Las funciones del formato - Formato vertical - Formato horizontal	- Apertura vertical entre conceptos - Densidad y distancia vertical - Apertura y densidad horizontal - Sangrado	- Presentación en PowerPoint - Guías instruccionales - Videos de complemento
	Tema 6: Los objetos y estructura de datos - Abstracción de datos - Antisimétrica de datos y objetos - La ley de Demeter - Objeto de transferencia	- Antisimétrica de datos y objetos - Choque de trenes - Híbridos - Ocultar estructuras - Registro activo	- Presentación en PowerPoint - Guías instruccionales - Videos de complemento
Sesión 4 Duración 4 horas	Tema 7: Mejorar la gestión de error - Usar excepciones en lugar de código devueltos - Crear primero la instrucción try-catch-finally - Usar excepciones sin comprobar - Definir clases de excepciones de acuerdo con las necesidades del invocador - Definir el flujo de normas - No devolver null - No pasar null	- Excepciones y sus tipos - Definición de clases de acuerdo con necesidades - Ofrecer contexto junto a excepciones	- Presentación en PowerPoint - Guías instruccionales - Videos de complemento
	Tema 8: Límites - Utilizar código de terceros - Explorar y aprender límites - Las pruebas de aprendizaje son algo más que gratuitas - Usar código que todavía no existe - Límites limpios	- Aprender log4j - Utilización de códigos de otros - Definir límites y alcances - Códigos que no controlamos	- Presentación en PowerPoint - Guías instruccionales - Videos de complemento
Sesión 5 Duración 4 horas	Tema 9: Pruebas de unidad - Las tres leyes del DGP - Realizar pruebas limpias - Una afirmación por prueba - F.I.R.S.T.	- Las pruebas propician posibilidades. - Lenguajes de pruebas específicos del dominio - Un solo concepto de pruebas	- Presentación en PowerPoint - Guías instruccionales - Videos de complemento
	Tema 10: Clases - Organización de clases - Las clases deben ser del tamaño reducido - Organizar los cambios - Proxies Java - Estructura AOP java puras - Pruebas de unidad de arquitectura de sistema	- Encapsulado - Cohesión - Aislarnos de los cambios - Separar Main - Factorías - Inyectar dependencias - Aspectos transversales	- Presentación en PowerPoint - Guías instruccionales - Videos de complemento



SECRETARÍA DE
INNOVACIÓN

