

Universidad de Las Américas
Facultad de Ingenierías y Ciencias Agropecuarias
Ingeniería de ...
Informe de laboratorio

1. DATOS DEL ALUMNO:

Dylan Clerque

2. TEMA DE LA LECTURA:

Código limpio y Nombres con sentido

3. FUENTE:

Martin, R. C. (2008). Clean code: A handbook of agile software craftsmanship. Prentice Hall.

4. RESUMEN (PALABRAS PROPIAS -EVITAR COPIAS- INCLUIR GRÁFICOS SI ES NECESARIO):

Capítulo 1

Los programadores y el código nunca desaparecerán, debido a la razón de que existen requisitos muy específicos que probablemente no se puedan cumplir mediante programas predeterminados. Básicamente el código es el lenguaje en donde se determinan los requisitos en última instancia, de hecho, el ser humano podría crear máquinas y programas que mezclen contextos, combinen requisitos en estructuras formales y los analicen, pero nunca llegará a la precisión necesaria.

¿Relevancia del código correcto? Para se debe tener en cuenta porque es importante un código correcto, por ejemplo, según el autor del libro, dice que existía una empresa que creó una gran aplicación, lastimosamente en sus ciclos de programación no se corrigieron errores, los tiempos de carga aumentaron al igual que muchos fallos, por lo que esta desapareció. El motivo principal y la causa de este suceso fue que la empresa comercializó el producto mucho antes de que este estuviera listo para ser lanzado, por lo que contenía errores que al tratar de solucionarlos en un futuro fue prácticamente imposible, cada vez que se añadían nuevas funciones este empeoraba, por lo que básicamente el motivo fue el código incorrecto inicial. ¿Por qué ocurre el código incorrecto y por qué lo hacemos? Pueden existir algunos motivos, como por ejemplo la falta de tiempo o tal vez la falta de ganas o simplemente el hecho de pensar que con que funcione está bien, a pesar de tener un código no limpio e incorrecto, el típico, *“Después lo arreglo”* y muchas veces el después se vuelve nunca.

Cada cambio en un código es muy importante debido a que este puede afectar a otras dos o tres partes del código, por lo que, para realizarlos, hay que tomar en cuenta los detalles, los efectos y las consecuencias que este puede ocasionar para añadirlos y que formen parte del código. Si es que no se pone atención a estos detalles con el paso del tiempo el desastre va en aumento, por lo que la productividad del equipo tiende a caer, esto simplemente genera más y cada vez más problemas, que a largo plazo son irremediables.

Se crea un nuevo diseño, algo desde cero, algo que pueda realizar de forma correcta, para de esta forma, no tener los mismos errores del anterior, errores que no se pueden corregir o que son muy complicados de hacerlo, por ende, se genera una competencia entre el antiguo equipo y el nuevo, a pesar de poder empezar sin errores el nuevo equipo tiene el reto de hacer que su creación haga lo mismo que la anterior, sin mejoras y sin errores. Según el libro, existen casos donde pasan mucho tiempo tratando de hacerlo y al final no lo logran y simplemente se esfuman, dejan al equipo antiguo exigiendo un cambio de diseño.

Al final ¿De quién es la culpa de todos estos desastres? ¿De los directores que son unos incompetentes? ¿De los usuarios los cuales no son tolerantes? Pues no, la culpa es del programador, pero ¿Cómo? Bueno al final del día son ellos quienes programan ¿No? Son quienes realizan el código incorrecto, los fallos, etc. Es por esto que el código correcto es tan importante, sin importar si el jefe requiere rápido el código, es preferible demorarse un poquito más a entregar al mal hecho.

El texto aborda el concepto de código limpio en el desarrollo de software, presentando diversas opiniones de expertos en programación. Se destaca la importancia de mantener un código claro, legible y eficiente para facilitar su comprensión y mantenimiento a lo largo del tiempo. Se mencionan principios como la elegancia, eficacia, legibilidad, expresividad y la importancia de las pruebas en la creación de código limpio. También se discute la necesidad de mejorar continuamente el código existente, siguiendo la regla del "Boy Scout" *"Deja el campamento más limpio de lo que lo encontraste"*.

Además, se hace referencia a la analogía con las artes marciales, donde diferentes escuelas de pensamiento pueden ofrecer enfoques distintos, pero igualmente válidos para el desarrollo de código limpio. Se resalta la responsabilidad del programador como "autor" del código, enfatizando que la lectura y comprensión del código son fundamentales para su escritura. El texto concluye señalando que, al igual que en cualquier arte, la práctica y el compromiso con la mejora continua son clave para convertirse en un buen programador.

Capítulo 2

La introducción del texto destaca la importancia crítica de elegir nombres efectivos en el desarrollo de software. Se señala que, aunque puede parecer una tarea simple, la selección de nombres adecuados puede tener un impacto significativo en la legibilidad, mantenibilidad y comprensión del código por parte de otros programadores. Se menciona que, en muchos casos, los nombres son una de las

formas más efectivas de comunicar la intención y el propósito de las variables, funciones, clases y otros elementos del código.

El texto continúa explicando que los nombres bien elegidos pueden actuar como una forma de documentación viva, proporcionando información instantánea sobre el propósito y el contexto de cada elemento del código. Esto puede resultar crucial en proyectos grandes y colaborativos, donde múltiples programadores pueden estar trabajando en el mismo código simultáneamente o en diferentes momentos. Se resalta la importancia de evitar la desinformación al elegir nombres, ya que un nombre incorrecto o ambiguo puede conducir a malentendidos y errores en el desarrollo y mantenimiento del software. También se advierte sobre los peligros de los nombres genéricos o abstractos, que pueden carecer de contexto y dificultar la comprensión del código para quienes no estén familiarizados con él.

Además, se subraya la necesidad de proporcionar contexto adecuado al elegir nombres, lo que implica tener en cuenta el dominio del problema o la solución que el software está abordando. Esto puede ayudar a los desarrolladores a comprender rápidamente el propósito y la funcionalidad de cada parte del código, incluso si no están familiarizados con todos los detalles técnicos.

En resumen, la introducción establece la importancia de elegir nombres efectivos en el desarrollo de software, destacando su papel en la comunicación de la intención y el propósito del código, así como en la facilitación de la comprensión y el mantenimiento del mismo.

5. OPINIÓN PERSONAL:

1. La importancia del código correcto en el desarrollo de software es crucial para el éxito del producto: El texto destaca el caso de una empresa que lanzó una gran aplicación sin corregir los errores presentes en el código. Esto llevó a un aumento en los tiempos de carga y una serie de fallos que eventualmente llevaron al fracaso del producto y, en última instancia, a la desaparición de la empresa. Esta afirmación resalta la importancia de tener un código limpio y correcto desde el principio para evitar problemas graves en el futuro.
2. La responsabilidad del programador en la calidad del código: Aunque se mencionan factores como la falta de tiempo o la falta de motivación como posibles razones para la creación de código incorrecto, el texto enfatiza que al final del día, la responsabilidad recae en los programadores. Se subraya que, a pesar de las presiones externas, es esencial dedicar el tiempo necesario para escribir un código limpio y de calidad, ya que esto tiene un impacto directo en el éxito del producto y, en última instancia, en la supervivencia de la empresa.

3. La elección de nombres descriptivos mejora la legibilidad del código: Tras analizar el impacto de los nombres en la comprensión del código, se encontró que utilizar nombres descriptivos y significativos facilita la lectura y comprensión del software, lo que contribuye a una mejor legibilidad y entendimiento por parte de otros desarrolladores.
4. Nombres apropiados reducen la probabilidad de errores y malentendidos: Al evaluar la incidencia de errores y malentendidos en el desarrollo de software, se observó que la utilización de nombres precisos y con contexto disminuye la probabilidad de interpretaciones erróneas o fallos en la implementación, lo que resulta en un código más confiable y robusto.

6. TRES PREGUNTAS Y RESPUESTA SOBRE LA LECTURA:

- a. **PREGUNTA 1:** ¿Por qué es importante tener un código limpio?

Respuesta: Es importante ya que de no hacerlo esto generaría muchos fallos y con el paso del tiempo estos fallos se vuelven más grandes al punto que son irreversibles.

- b. **PREGUNTA 2:** ¿Cuál es la importancia de tener nombres apropiados en el código?

Respuesta: La importancia radica en la reducción de la probabilidad de cometer errores y malas interpretaciones o malos entendidos.

- c. **PREGUNTA 3:** ¿Qué frase se usa para hacer referencia a los Boy scout comparándolo con el código?

Respuesta: *"Deja el campamento más limpio de lo que lo encontraste"*