

Facilidad de Comprension en Ingenieria de Software

Lindell Dennis Vilca Mamani

La facilidad de comprension en el contexto de la ingenieria de software se refiere a la capacidad de entender y asimilar rapidamente el codigo fuente o la documentacion de un programa. A continuacion, te proporciono informacion sobre este tema:

Definicion de Facilidad de Comprension:

La comprension de codigo fuente es fundamental para actividades como revision de codigo, correccion de defectos y refactorizacion. Se busca lograr un software que sea facil de entender y mantener [1].

Metricas de Facilidad de Comprension:

Algunas metricas evaluan la legibilidad y complejidad del codigo:

- Numero de lineas: Cuantas lineas de codigo tiene un programa.
- Metricas de Halstead: Evaluan la complejidad del programa basandose en operadores y operandos.
- Numero ciclomatico: Mide la complejidad estructural del codigo.
- Intervalo entre referencias a datos: Evalua la distancia entre las referencias a variables.
- Par de uso segmento-global: Mide la relacion entre variables locales y globales [2].

Aplicaciones:

La facilidad de comprension es crucial para:

- Revision de codigo: Facilita la deteccion de errores.
- Mantenibilidad: Un codigo comprensible es mas facil de mantener.

- Refactorizacion: Permite mejorar la calidad del codigo.
- Desarrollo colaborativo: Ayuda a los equipos a trabajar eficientemente [3].

Limitaciones:

- A pesar de las metricas, la comprension sigue siendo subjetiva y depende de la experiencia del desarrollador.
- No todas las metricas son aplicables en todos los contextos.
- La facilidad de comprension no garantiza automaticamente la calidad del software [4].

Ejemplo deCodigo:

A continuacion se muestra un ejemplo de un script sencillo en Python que ilustra como el estilo de codificacion puede influir en la facilidad de comprension:

Listing 1: Ejemplo de script en Python para calcular el area de un circulo

```
# Funcion para calcular el area de un circulo
def calcular_area_circulo(radio):
    area = 3.14159 * radio ** 2
    return area

# Programa principal
if __name__ == "__main__":
    r = float(input("Ingrese el radio del circulo: "))
    area_circulo = calcular_area_circulo(r)
    print(f"El area del circulo con radio {r} es: {area_circulo}")
```

En este ejemplo, el codigo esta estructurado de manera clara y utiliza nombres de variables descriptivos y comentarios adecuados, lo cual facilita su comprension y mantenimiento.

Ejemplo de Metrica:

A continuacion se muestra un ejemplo de como se podria calcular el numero de lineas de codigo y el numero de lineas comentadas en un script sencillo en Python:

Listing 2: Ejemplo de calculo del numero de lineas de codigo y lineas comentadas en Python

```
# Funcion para contar lineas de codigo y lineas comentadas en un archivo
def contar_lineas_codigo_y_comentarios(archivo):
    with open(archivo, 'r') as file:
```

```

        lineas = file.readlines()
        num_lineas = len(lineas)
        num_comentarios = sum(1 for linea in lineas if linea.strip().startswith('#'))
        return num_lineas, num_comentarios

# Programa principal
if __name__ == "__main__":
    archivo_ejemplo = 'codigo_ejemplo.py'
    num_lineas, num_comentarios = contar_lineas_codigo_y_comentarios(archivo_ejemplo)
    print(f"El archivo '{archivo_ejemplo}' tiene {num_lineas} lineas de codigo y {num_comentarios} comentarios")

```

En este ejemplo, la funcion `contar_lineas_codigo_y_comentarios` calcula tanto el numero de lineas de codigo como el numero de lineas comentadas en un archivo especificado. Esto es util para evaluar la complejidad y la facilidad de comprension del codigo basandose en estas metricas especificas.

References

- [1] Titulo: “Influencia del uso de recomendaciones de legibilidad en la facilidad de comprension en codigo fuente”. Disponible en: <https://repositorio.unal.edu.co/bitstream/handle/unal/75524/Tesis%20Oct%202019%20v1.pdf>
- [2] “Software Metrics: A Rigorous and Practical Approach”. Disponible en: https://sedici.unlp.edu.ar/bitstream/handle/10915/19748/Documento_completo.pdf?sequence=1
- [3] Titulo: “diseño e implantacion de una aplicacion web que permita el analisis de la facilidad de comprension en duagramas UML de inraciona traves de una replica experimental y sintesis de estudios previos”. Disponible en: <https://repositorio.espe.edu.ec/bitstream/21000/12434/2/ESPEL-MAS-0028-P.pdf>
- [4] “Comprension de Programas”. Disponible en: <https://repositorio.espe.edu.ec/bitstream/21000/12434/2/ESPEL-MAS-0028-P.pdf>