

Ejercicio Semanal 02

Emiliano Galeana Araujo
314032324

Facultad de ciencias
Universidad Nacional Autónoma de México

07 Septiembre de 2017

1. Cualidades del Software

1.1. Internas

- Corrección: Sí lo hace, pero no como debería.
- Verificalidad: Está bien, ya que los atributos, se entienden.
- Sustentabilidad: Malo, porque no sé C.
- Reutilizable: No creo que se pueda usar, ya que todo está dentro del main, solamente el `multiplica.c`
- Transportable: No lo creo, no sé C, pero no sé si funcione de la misma manera en otras plataformas.

1.2. Externas

- Confiabilidad: Sí hace lo que debe, dado un número te regresa el impuesto íntegro.
- Robustez: No, si se rompe si no le pasas como parámetro lo que se indica, el problema es que solo se sabe qué se indica si lees el código de cómo funciona.

- Eficiencia: Creo que está bien, da los resultados rápido.
- Amigabilidad: A menos que el usuario sepa exactamente qué meter, y cómo hacerlo se podría decir vagamente que si es amigable, pero para alguien nuevo, no.
- Interoperatividad: Quizá si, si usáramos un programa dedicado a sacar impuestos y que vaya recibiendo más impuestos.

2. Cuantificación Del Programa Actual

d_i	3
d_o	2
c_i	2
c_o	1
g_d	5
g_c	3
f_o	4
f_i	0
Total	0.9615

3. Cualidades Del Programa Modificado

3.1. Internas

Corrección	✓
Verificabilidad	✓
Sustentabilidad	×
Reutilizable	✓
Transportable	×

- Corrección: Sí, se espera que de el impuesto sobre un monto, y es justo lo que hace, claro, dependiendo del país.
- Verificabilidad: Sí, no hay atributos ambiguos y se puede inferir para qué se usan.

- Sustentabilidad: Está comentado, pero muy vagamente, no es muy difícil de entender, pero aún así, puede que quien le quiera dar mantenimiento, tenga que leerlo al menos una vez.
- Reutilizable: Sí, puede ser usado, al menos algunas funciones, siempre y cuando se quiera algo muy parecido a lo que se hizo en este ejercicio.
- Transportable: No sé hacer muchas cosas en C, así que lo más probable es que en otra plataforma pueda morir.

3.2. Externas

Confiabilidad	✓
Robustez	×
Eficiencia	✓
Amigabilidad	✓
Interoperatividad	✓

- Confiabilidad: Sí, ya que si el país y el monto están bien, entonces va a regresar el impuesto sobre el monto, dependiendo del país.
- Robustez: No muy bien, ya que si das otros valores no numéricos en el monto, lo toma como 0 y entonces no se debe nada, pero si capta bien los parámetros.
- Eficiencia: Bien, ya que hace todo en un buen tiempo, y sobre el espacio, no se usa mucha memoria, eso debería de ser bueno.
- Amigabilidad: Sí, si se ingresan datos erróneos, explica cómo debe ser usado el programa.
- Interoperatividad: Es muy probable que si, obviamente tiene que ser un programa relacionado y se podría mandar a llamar o implementar en este mismo.

4. Cuantificación Del Programa Modificado

d_i	2
d_o	2
c_i	2
c_o	0
g_d	0
g_c	0
f_o	3
f_i	0
Total	0.9090

Lo que pasó en este punto, es que creo que cuantificar el cumplimiento del programa, tenía que ser con cada módulo, ya que sino, es muy complicado ir bajando el total, lo pensé un rato y creo que es por cada módulo del programa, sin embargo, al tener este varios módulos, no me sonaba lógico hacer un promedio, así que decidí hacerlo como general.

Lo que se hizo fue eliminar las variables globales, hacer funciones para no tener todo en el `main` y mandarlas a llamar, eliminar también las constantes, pasar todo a un archivo, para no tener que depender de otros módulos y arreglar la entrada de parámetros del programa.