

## **Reporte Programa 3.**

### **Error en la estimación de tiempo.**

Para el primer programa sobreestime el tiempo, dándome un margen de error del -9.7%. En el segundo programa también sobreestime un poco más en comparación con el primer programa, dándome un margen de error del -21.4% y en el tercer programa el tiempo estimado fue mucho menor que el tamaño real, dándome un margen de error del 57.3%.

La tendencia que tuve fue sobreestimar en los primeros dos programas siendo que en el segundo tuve un margen de error mayor que en el primer programa, y por último en el tercer programa tuve un gran error al subestimar.

### **Error en la estimación de tamaño.**

En el segundo programa sobreestime, dándome como margen de error un 42.7% de error, y en este tercer programa, como la herramienta me estima el tiempo, dio un error de subestimación del -32.9%.

Los errores al estimar el tamaño fueron grandes, pero uno fue sobreestimando y el otro subestimando, teniendo en cuenta que cada vez tengo un margen de error menor.

### **Diseño.**

Cada vez le he dedicado más tiempo a la fase de diseño por lo que el porcentaje de errores insertados en esta fase ha ido disminuyendo, al igual que el número de errores insertados en codificación y errores removidos en compilación y testing. Esto hace que el diseño impacte en gran medida en la estimación de tiempo y tamaño.

### **Tamaño por clases y su complejidad.**

El número de métodos y la complejidad de una clase están muy ligados entre sí, ya que entre más métodos más complejidad tendrá la clase, aunque también existen excepciones donde una clase con pocos métodos tengan una gran complejidad, pero esos métodos siempre tienden a ser grandes.

### **Tiempo consumido.**

El tiempo consumido en la fase de compilación resolviendo errores fue de 10 minutos, y en la fase de testing fue de 18.6 minutos con un total de 231 líneas de código. Si mi programa tuviera 1000 líneas de código, tardaría 43.2 minutos en la fase de compilación y en la fase de testing tardaría 1 hora 20 minutos.

**Proceso.**

El diseño es indispensable para que el proceso funcione, es por eso que mientras el diseño este bien detallado el resto del proceso debe resultar fácil de realizar, ya que con el diseño obtenemos como resolver el problema e implementar la solución, y con esto no debe haber un gran numero de errores, y el tiempo invertido en las fases que le siguen a diseño no debe ser mucho.

El proceso que se debe seguir para realizar los reportes de los programas seria primero comenzar por analizar los datos obtenidos y de ahí identificar porque salieron dichos datos, después se tienen que desglosar los datos en partes para poder plasmarlos en un reporte en el que se detallen los resultados.

Los procesos que se utilizan en ingeniería de software están representados por fases en las que en cada fase hay instrucciones que seguir para obtener un resultado y con esto seguir con las demás fases.

**Lecciones aprendidas.**

Con este proceso aprendí que requiero invertir una gran cantidad de atención y el tiempo necesario a la fase de diseño para así poder seguir el proceso con sin tantos errores y con un tiempo más optimo.