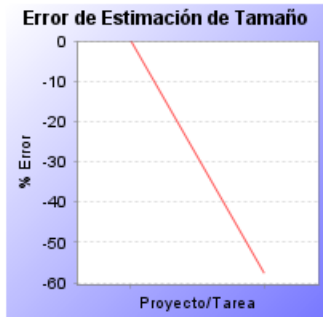
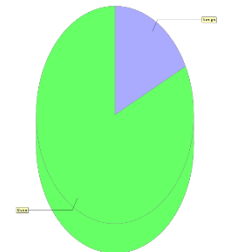


Estimación y LOC - El tiempo estimado del programa 1 fue de 360 minutos y el total en ese programa resultó de 840 minutos, donde se pudo ver una estimación muy mala, o que era buena pero tardé demasiado en el proceso de elaboración del programa, con un porcentaje de error del 59.686450168%, en el segundo programa trate de mejorar la estimación con respecto al trabajo que se iba a elaborar, mi estimación para el segundo programa fue de 360 minutos, con un valor real de 350 minutos con un porcentaje de error del 2.85714285714% lo que quiere decir que para este programa tuve una mejor estimación conforme al programa anterior, fui más optimista en la realización de este y por lo mismo creo que dejar la estimación en 6 horas o 360 minutos fue adecuado para poder medir dicha productividad con los programas.

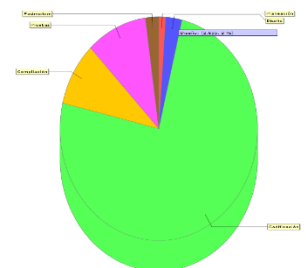


En este programa estimé 200 líneas en añadidas y modificadas, al inicio no tenía contemplado usar código del programa pasado por la situación del mal diseño que tuve en él pero si terminé usando algunas, que me quedaron como 81 líneas modificadas de las 197 de base, 115 eliminadas, 4 líneas añadidas y efectivamente no reutilicé ninguna, pero en añadidas y modificadas tuve 85 líneas. Dando un error en la estimación de tamaño según la herramienta de -57.5%, lo que nos dice que tenemos que mejorar en la estimación de LOC para el siguiente programa. Al tener esta cantidad de líneas nos damos cuenta de que nos falta mejorar en la parte de diseño, al no tener ninguna línea nueva reutilizables para los otros programas.

Defectos - El total de defectos fue de 17, 3 inyectados en la fase de diseño y 14 que se inyectaron en codificación, podemos ver la distribución de manera gráfica. Los defectos que se inyectaron en código son mayormente errores de sintaxis, asignación o funcionalidad, en el caso de los errores inyectados en diseño son errores de funcionalidad y de asignación. En la fase de codificación tuve que pasar alrededor de 45 minutos solucionando los defectos que se atraparon en la compilación y en diseño la solución tardó 1:21 minutos en solucionarse, dichos defectos podemos interpretarlos como los que más tiempo nos consumieron para ser solucionados.



Tiempo - Por fases tuvimos el registro de tiempo siguiente, 1.13% en planeación, 2.65% en diseño, un 74.9% en código, 9.17% compilación, 9.98% en pruebas y 2.17% en postmortem de los 350 minutos que registramos. Analizando dos tiempos en específico que son los tiempos de compile y testing, en compilación se tuvo un 13.7% de tiempo a comparación del 26.9% de testing, lo que indica que estamos pasando más tiempo en las pruebas que en la compilación, como vimos antes también se debe a los errores que se inyectaron en dichas fases y que al final nos cuesta ese porcentaje de tiempo del proceso general.



En particular hay dos métricas interesantes que están alejadas del valor real en la planeación, una de ellas es la codificación donde el valor planeado fue de 5:06 pero el real fue de tan solo 2:51 minutos lo que nos habla de la mejora al momento de hacer el código, la otra es la de las pruebas, donde el valor planeado era de 12 minutos y al final terminamos tardando 1:34 minutos, lo que nos dice que tenemos que mejorar en dicha fase.

Aprendizaje – La elaboración de este programa me permitió a comprender un poco mejor el proceso de PSP, como tal PSP 0.1, comprender las mejoras que se hacen en los scripts, seguir el proceso y mejorar en este, identificar cuales son los puntos más débiles en mi proceso y tenerlos presentes para la mejora, me ayudó a comprender mejor a la hora de diseñar y codificar para después tener un mejor desarrollo con el uso de la medición del programa según sus líneas de código y hacer el diseño conforme a esto.