**Informe Taller 2 Ingeniería del Software**

Gabriel Freites Carnet: 08-10383

Daniel Leones Carnet: 09-10977

**Índice**

Introducción . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. 3

Capitulo 1: Programación . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

Capitulo 2: Egit . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

Capitulo 3: Unittest . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

Conclusión . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

**Introducción**

Este trabajo se hace con el propósito de tener un algoritmo eficiente, que nos facilite la realización de la aplicación para control de estacionamientos llamada “SAGE”

También en este informe resaltaremos algunos aspectos de la programación en parejas, el uso de las herramientas recomendadas y la utilización de las pruebas por casos a la hora de realizar el proyecto.

Seremos lo suficientemente específicos y concisos como para hacer de este informe un diario claro de la experiencia sin extendernos demasiado, describiéndola en tres capítulos, el de programación por pares, uso de git y uso de casos de prueba en Python.

**Capítulo 1: Programación**

Bueno la programación fluyo sin muchos contratiempos, nos vimos obligados a utilizar solo una maquina porqué era la única que podíamos llevar a la universidad sin inconvenientes. Es muy interesante la experiencia, hubo momentos en que de tanto programar y realizar casos de prueba nos bloqueábamos, y cambiar de “conductor” ayudaba bastante, tanto a no frenar el progreso, como al desbloqueo de la otra persona.

Después de tres rotaciones fue que logramos entender bien el algoritmo y manipularlo hasta que nos dio finalmente para los casos de prueba que teníamos planteados. Python es un lenguaje muy versátil y potente que también facilita la experiencia, sería bueno aplicar este método en algún lenguaje más trillado donde seguir el código del compañero sea más difícil a ver qué tal nos va.

**Capítulo 2: Git**

A pesar de que EGit facilita las cosas, un par de veces tuvimos que borrar y bajar el branch en el que estábamos trabajando por errores muy extraños a la hora de subir y hacer merge de lo que teníamos, sigo pensando que usar git desde el terminal es una manera mejor de aprenderlo y aplicarlo, el egit no es perfecto y a veces puede ser hasta limitante dentro de sus opciones.

Para la realización creamos 2 branches de usuarios (Gabo y Daniel), un branch de pre-producción para subir cualquier cambio no funcional o completo y el branch master donde subimos la mayor parte del trabajo gracias a que trabajábamos bastante tiempo seguido como para concluir las partes de código que íbamos planteándonos hacer. En resumen git es una herramienta muy importante y poderosa que agiliza mucho el trabajo a distancia y en grupo, no pudimos aprovecharla al máximo solo cuando trabajábamos desde lugares separados.

**Capítulo 3: Unittest**

El unittest es una herramienta muy útil de la que no teníamos conocimientos previos a la hora de hacer un programa desde cero y de la forma más eficiente posible, nos ayudó a encontrar errores desde un principio, ya con la entrada de datos donde teníamos un error de guardias, que no habíamos probado correctamente.

Personalmente nos parece un poco tedioso que se tenga que ser tan detallista como de probar una función antes de crearla, pero de resto los casos me parecen muy útiles.

Tuvimos algunos errores extraños, por lo menos, cuando no se pone test en los casos no se ejecutan, cuando los nombre de los casos tiene algún numero tienen un comportamiento poco predecible, y hay un caso que creo que subí que si corres por separado corre perfectamente, pero al correrlo en conjunto, la lista de tuplas parece solaparse lo que causa un error que no pude corregir, intentare actualizar mi Python al último e investigar un poco más del tema pero no pude encontrarlo antes de la entrega.

Esta herramienta nos pareció muy útil para agilizar el proceso de prueba de una función, así como reducir el de acoplamiento, y esperamos en un futuro perfeccionar su uso para aplicarlo a futuros proyectos que tengamos.

**Conclusión**

En conclusión se logra el objetivo de hacer una función correcta siguiendo el esquema de TDD, se aprende a programar guiándose por los casos de prueba, no de la forma natural que es al contrario y a pesar de algunos errores raros se logra probar bien la función, también se mejora en el manejo de git y Egit concretando así satisfactoriamente con los objetivos de la tarea.

Se obviara la Bibliografía ya que no la considero necesaria.