

Universidad Simón Bolívar

Departamento de Computación y Tecnología de la Información

Asignatura: Ingeniería de Software – CI-3715

Profesor (es): Betzaida Romero,

Alfonso Reinoza.

Tarea 2.

Autores:

Lalezka Duque, 12-10613

Marcos Jota, 12-10909

Sartenejas, Enero del 2017.

Índice General

1. Introducción.....	3
2. Desarrollo.....	4
i. Repartición de Trabajo.....	4
ii. Dominio de Datos.....	4
iii. Experiencias con las Herramientas.....	5
iv. Modelo de Ramas.....	5
3. Conclusiones.....	6

Introducción

Al momento de programar en equipo es necesario seguir un esquema o metodología de desarrollo. Esto se debe a que para escribir, probar o ejecutar el código de un programa se requiere de la práctica de un plan de desarrollo ágil, que exige a su vez los conocimientos necesarios de las herramientas que faciliten el progreso del software que se quiere programar. Además, es necesario analizar de manera detallista todos los posibles casos que puedan afectar al programa desarrollado. En consideración a lo anterior, en la tarea 2 de Ingeniería de Software se busca aprender el conocimiento de las herramientas de apoyo a la Ingeniería de Software y, además, ejercitar las técnicas de verificación de programas.

Desarrollo

Repartición de Trabajo

Con respecto a la repartición de trabajo, fue hecha de forma consistente para ambos miembros del grupo. Esto en base a la complejidad del programa, su análisis y desarrollo de casos de uso. Luego, a medida que se trabajaba en la tarea, se subieron las versiones modificadas al nodo Master del repositorio de trabajo.

En particular, Lalezka se encargó de realizar algunos casos de prueba, el análisis del dominio de los datos, el desarrollo de parte del script `calcularServicio.py` y la escritura del informe. Por otro lado, Marcos culminó los casos de prueba planteados, el análisis de esquinas y fronteras, y la revisión del código montado en el repositorio de trabajo.

Finalmente, las horas de trabajo que cada miembro invirtió en el desarrollo de la tarea es de aproximadamente 5 horas en total.

Dominio de Datos

El dominio de los elementos de tiempo que se utilizaron en el desarrollo del programa `calcularPrecio.py`, los cuales pertenecen a la librería `DateTime`, es el siguiente:

- $\text{MIN_YEAR} \leq \text{year} \leq \text{MAX_YEAR}$, donde `MIN_YEAR` y `MAX_YEAR` corresponden al mínimo y al máximo año posible respectivamente.
- $1 \leq \text{month} \leq 12$
- $1 \leq \text{day} \leq N$, donde `N` representa al número de días en el año o mes dado.
- $0 \leq \text{hour} < 24$
- $0 \leq \text{minute} < 60$
- $0 \leq \text{second} < 60$
- $0 \leq \text{microsecond} < 1000000$

Con respecto a la clase `Tarifa`, su dominio corresponde con el conjunto de los números reales positivos. Por otro lado, el dominio de la duración del servicio comprende el intervalo de $1 \leq \text{días_totales_servicio} \leq 7$.

Experiencias con las Herramientas

Con respecto a las herramientas utilizadas para la programación, en particular Eclipse con la extensión PyDev, se puede destacar la facilidad que brinda este entorno de desarrollo en cuanto a las variables y las funciones. Por otro lado, al momento de utilizar Eclipse con la extensión egit no resultó tan intuitivo. Sin embargo, después de indagar por la interfaz de Eclipse y consultar la documentación de dicha herramienta, pudimos utilizarla de forma correcta sin generar conflictos en las versiones.

Modelo de Ramas

El modelo de ramas utilizado fue un árbol con un nodo padre (rama Master) y un hijo para resguardar la versión de la rama Master. Al culminar la tarea se hizo un merge hacia el nodo Master.

Conclusiones

El desarrollo de un software, es un proceso complejo que requiere de mucho tiempo y organización para ser logrado de manera exitosa, así como la cooperación y la colaboración de los desarrolladores. De igual forma, el buen manejo de las herramientas ayuda al progreso de un proyecto, sin embargo, no es suficiente para que el proceso de desarrollo sea eficiente. En este sentido, es importante resaltar que para lograr un trabajo de calidad se deben desarrollar métodos de distribución del trabajo, realizar análisis de pruebas y casos borde y, además, utilizar un buen modelo de ramas para programar el código.

A manera de conclusión, el análisis de pruebas y casos borde es uno de los pasos más importantes al momento de desarrollar un software. Esto se debe a que se asegura la estabilidad, fiabilidad y el buen funcionamiento del mismo.