



Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-3725 Ingeniería de Software - Laboratorio
Septiembre-Diciembre 2017

Tarea #2

Profesores: José Alfonso Reinoza Contreras

Integrantes:

Andrés Buelvas 13-10184
Erick Fleján 12- 11555

Sartenejas, 5 de octubre de 2017

Introducción

Hoy en día, en el mundo de la informática, muchas de las compañías y empresas contratan equipo de desarrolladores con la finalidad de crear aplicaciones o sistemas que les permita competir en el mercado. Normalmente, este equipo de desarrollares tiende a ser de una cantidad de personas considerable, dependiendo del trabajo solicitado. La iniciativa por parte de las empresas de usar métodos mas modernos y tecnológicos busca agilizar la realización de procesos y procedimientos en un periodo de tiempo reducido comparado a la usabilidad de herramientas manuales. Por lo tanto, es deber del equipo de desarrolladores cumplir con las expectativas y requerimientos de los clientes, sin embargo, esta tarea se vuelve difícil de implementar si no se tiene la preparación y organización necesaria, por eso, llevar un plan de trabajo coordinado y aprobado por todos los miembros del equipo de desarrolladores hará que la realización del proyecto se pueda concretar exitosamente. Este plan de trabajo se puede presentar en formas de tareas repartidas, reuniones con los clientes, discusión de temas de riesgo y duración de la realización del proyecto, tipo de herramientas se usarán, etc. Este ultimo punto, el tipo de herramientas, es importante pues, permite a los miembros adaptar el trabajo dividido entre los miembros del equipo en una sola dirección que es, la finalización del proyecto.

Durante todo proyecto es posible encontrar problemas técnicos que ralenticen la realización de la tarea solicitada, el mas común de ellos, es que lo implementado no cumpla con “todas” las necesidades del cliente, y por ende, su funcionalidad se ve reducida, es por esto, que hacer un buen levantamiento de los requerimientos y la realización de pruebas de casos por cada área trabajada por cada miembro del equipo desarrollador es importante en el desarrollo de un proyecto. El primer punto es resuelto a través de una comunicación eficaz con el cliente, el segundo punto se resuelve a través de la discusión y análisis de los requerimientos del cliente, aquí se verifica que lo pedido cumpla con lo solicitado. Los casos de prueba permiten al desarrollador de una aplicación web o sistema poder hacer una revisión exhaustiva para verificar su funcionalidad.

Para esta ocasión se pide al equipo de desarrollo hacer un programa que calcule el precio de un servicio, de acuerdo a una tarifa determinada, las horas trabajadas y los días de semana contemplados. Además se busca realizar un test del programa para determinar si cumple con la totalidad de los requerimientos solicitados, paralelo a esto, se busca evaluar la forma en como el equipo de desarrollo realiza el programa a través de un gestor de versiones que permita actualizar el programa todas las veces que sean necesarios hasta conseguir el programa final.

Este informe contemplará la forma en como el equipo de desarrollo se organizó y preparó para la realización de esta tarea y como fueron los resultados obtenidos.

El informe abarcará en la sección 2 información como las horas dedicadas, experiencias con las herramientas aplicadas, como fue dividido el trabajo, un análisis del dominio del programa y experiencias con el modelo de ramas de Git/GitHub y en la ultima sección se hablará de la conclusión del informe y se puntualizará aspectos como el objetivo del trabajo, el trabajo realizado y recomendaciones necesarias.

2. Desarrollo

En este punto se abarcarán detalles que fueron descritos en la introducción y que haremos mención a continuación:

2.1 Especificación y análisis del dominio del trabajo:

Se ha solicitado al equipo de desarrollo realizar un programa que de acuerdo a la fecha y hora de inicio de un servicio y la fecha y hora de finalización del mismo, arrojar el precio que una persona debería pagar por tal servicio, de acuerdo a una tarifa prefijada y a los días que fue prestado el servicio.

El programa debe cumplir algunos requisitos: la duración del servicio debe ser entre 15 minutos y 1 semana, estos últimos dos valores incluidos. La función que ejecuta el cálculo del precio `calcularPrecio()` del servicio solo debe aceptar el tiempo medido en horas y la tarifa prefijada. Se obvian minutos y segundos dentro de esta función. La tasa de la tarifa prefijada debe ser distinta si se presta el servicio en días de semana o fines de semana. Las tasas por hora que serán fijadas en la tarifa deben ser introducidos en bolívares y céntimos, esta cantidad debe ser no negativa. Finalmente la función `tiempoDeTrabajo()` calcula la cantidad de horas prestada en servicio, este debe aceptar fecha y hora del inicio y fin del servicio, este debe retornar las horas invertidas en días de semana o fines de semana. El servicio es prestado en horas completas, es decir, una (1) hora y un (1) minuto es considerado como 2 horas del servicio.

También se ha solicitado la inclusión de un test del programa que permita verificar todos los casos pruebas del mismo, es decir, verificar que el programa cumpla con lo solicitado. Además, se pide colocar todo lo referente al programa en el gestor de versiones GitHub a través del sistema de ramas.

En el programa se ha introducido como valor por default al valor cero (0) como valor que resulta de una incorrecta introducción de valores o violación de condiciones del mismo.

2.2 Herramientas utilizadas:

Hacemos mención del conocimiento previo que se tenía del lenguaje de programación Python, por lo que se redujo considerablemente, el tiempo invertido a lo largo de la realización del trabajo.

Es importante destacar que el uso de las herramientas es un mundo nuevo para los integrantes del equipo. PyUnit, GitHub y Eclipse son algunas de estas herramientas.

La experiencia con Eclipse fue buena, el uso de Eclipse nos permitió detectar errores de una manera mas rápida, a través del uso del depurador, también no permitió observar que librerías o variables habían sido usados o no. Además, nos permitió hacer uso de PyUnit y GitHub a través del mismo.

La experiencia con PyUnit fue normal, es una herramienta que no estamos acostumbrados a usar, por lo que se invirtió una cantidad de tiempo considerable para estudiar su función y aplicación en el programa.

La experiencia con GitHub fue buena, aunque en un principio no se conocía como trabajar con las ramas, sin embargo, fue posible hacer lo pedido. GitHub nos facilitó el trabajo de interacción con los programas del otro compañero, porque mientras uno de los integrantes del equipo hacía su programa y actualizaba en el repositorio, el otro estudiante podía ver que hizo y hacer correcciones o mejoras o detalles que pudieron haber pasado por alto por el otro compañero.

Se solicitó aplicar el mecanismo de ramas de GitHub/Git. En un principio no se tenía conocimiento de cómo implementar las ramas en GitHub para luego hacer un merge de ellas a la rama master, por lo que las primeras versiones del programa inicial fueron subidas al repositorio a través de la rama master. Posterior a eso el trabajo fue subido nuevamente en una carpeta, pero ya en una rama distinta a la cual se le hizo merge, lo mismo ocurrió con el informe de la tarea.

2.3 Desarrollo del trabajo:

Para realizar la tarea solicitada se dividió el mismo entre los miembros del equipo para abarcar todas las solicitudes de la tarea en el menor tiempo posible.

El programa que resuelve la solicitud de dar precio a un servicio ofrecido con respecto a una tarifa fue realizado por el estudiante Andrés Buelvas y posteriormente revisado por el estudiante Erick Fleján. Luego, ambos miembros del equipo discutieron cómo que tipo de casos borde podrían ocurrir, haciendo la distinción entre los tipos de casos de pruebas unitarias a desarrollar, posterior a eso, el estudiante Erick Fleján llevó a cabo el desarrollo del test del programa, que luego sería revisado por el estudiante Andrés Buelvas, paralelo a eso, el estudiante Andrés Buelvas comenzaba con el desarrollo del informe que posteriormente sería finalizado por Erick Fleján y revisado por ambos.

El tiempo invertido por cada uno de los miembros se menciona a continuación:

- 6 horas para el desarrollo del programa (Andrés Buelvas)
- 1 hora para analizar el dominio del programa y los casos de prueba (ambos miembros del equipo)
- 4 horas para el desarrollo de pruebas unitarias con PyUnit (Erick Fleján)
- 4 horas para el desarrollo del informe (ambos miembros del equipo)

3. Conclusión

Para finalizar, haciendo una recopilación de lo dicho anteriormente, se pedía al equipo de desarrolladores hacer un programa que devolviera el precio de un servicio, dado un fecha de inicio y una fecha de finalizado del servicio, este precio podía variar dependiendo de la duración del servicio y de si incluía o no días de semanas o fines de semana. El objetivo era realizar tal programa y ejecutar las pruebas que fueran necesarias para que el programa hecho sea de perfecta funcionalidad. Como puntos apartes del trabajo, se pedía hacer uso del gestor de versiones GitHub para colocar el trabajo realizado en la plataforma a través del mecanismo de ramas de Git. Para realizar la tarea solicitada se dividió el trabajo en dos partes, que serían realizada por personas diferentes, pero con revisiones mutuas, una persona realizaba el programa y el otro realizaba el test del programa. Es importante destacar que la utilización de la librería de Python para fechas y horas fue un reto, pues era necesario investigar que método de la librería permitía al desarrollador hacer lo que se pedía en el trabajo, y el uso de PyUnit también fue un reto debido a que era la primera vez que ambos miembros del equipo hacían uso del mismo. Otro punto que podríamos mencionar que fue de dificultad moderada, fue la de saber que debía incluir el test del programa, que se debía probar y por qué, por tal motivo ambos miembros del equipo dedicaron una (1) hora de trabajo para analizar y discutir que debería incluir el test del programa.

Por último, algunas recomendaciones que se podían describir son, por ejemplo: tratar de explicar con detalle más el trabajo solicitado, ya que podría ocurrir casos donde se asuman cosas o situaciones que no eran correctas, y especificar que metodología o características se pedía para la documentación del programa.

