



UNIVERSIDAD SIMÓN BOLÍVAR

Departamento de Computación y T.I

Ingeniería de Software I

Abril Julio 2013

Informe – Tarea 4

Patrón Estrategia y Patrón Decorador

Integrantes:

Gamar Azuaje

Rosangelis García

José Luis Jiménez

Reinaldo Verdugo

Rebeca Machado

Gustavo El Khoury

Leopoldo Pimentel

Tabla de contenidos

1	Introducción	3
2	GIT/GitHub.....	4
2.1	Política del Commit	4
2.2	Integración.....	4
2.2.1	Proceso para resolver conflictos de integración	4
2.3	Proceso de mayor dificultad en Git/GitHub.	5
3	Aprendizaje de las últimas semanas.	5
4	Organización del Equipo	6
4.1	Coordinador de la gestión de configuraciones	6
4.2	Repartición de las tareas	6
4.3	Organización por parejas	6
4.4	Responsable de las integraciones	7
4.5	Responsable de actas/acuerdos de las reuniones (físicas y virtuales)	7
5	Conclusiones	8
6	Bibliografía.....	9

1 Introducción

Una vez realizado el levantamiento de información. Viene la hora de desarrollar un software o sistema que sea capaz de resolver el problema planteado.

Para el desarrollo de este software existe una gran cantidad de técnicas que permiten al programador comprender y facilitar su labor.

Para empezar cuando se habla de un patrón, estamos hablando de un modelo que sirve de muestra para realizar copias¹. Es decir, no importa las veces que se use siempre tendrá el mismo resultado. Y cuando se refiere a un patrón de diseño, se dice de una técnica de programación con la cual una vez encontrado un evento repetitivo en sistema, usando una misma solución podemos resolver todas las instancias del problema.

De esta estrategia de diseño, se desprenden varios casos. En este trabajo se realizó una parte con el patrón Decorador y otra parte con el patrón Estrategia.

Del patrón Decorador se puede decir que su *“propósito [es] agregar dinámicamente responsabilidades adicionales a un objeto. Los decoradores proporcionan una alternativa a la herencia (subclases) para extender funcionalidades”*.²

Mientras que del patrón Estrategia su propósito es el de desengranar el código. Convertir sus algoritmos en partes intercambiables de forma de usar el algoritmo que mejor cumpla con las necesidades en cada momento.

¹ (Real Academia Española, 2013)

² (Martínez, 2013)

2 GIT/GitHub

2.1 Política del Commit

Para la realización de este trabajo se usó la herramienta de control de versiones Git con el servidor GitHub. A través de la comunicación escrita por el canal de comunicación WhatsApp (aplicación para dispositivos móviles inteligentes) se crearon una serie de normas para la integración y manipulación del código que nos guiaron y facilitaron el trabajo en equipo. Cabe destacar que como grupo ya estábamos usando esta herramienta para la asignación pasada.

A la hora de realizar commits o push, mediante esta herramienta instantánea todos los integrantes del grupo estaban informados de quién usaba el repositorio y qué archivos iba a modificar, disminuyendo así los conflictos de merge.

El primer commit del repositorio incluía la tarea 3, sobre la que trabajaría el resto del equipo para realizar la tarea 4.

2.2 Integración

La integración del código fue una vez realizada y culminada cada parte. Es decir que al repositorio solo se subieron programas funcionales. De esta forma se redujo en gran medida la necesidad de acoplar el código cada cierto tiempo y adicionalmente las pruebas sobre el sistema no daban falsos errores por falta de culminación de algún módulo. Para lograr esta independencia del código se realizaron varias ramas o branches donde cada sub-equipo trabajo hasta la culminación de su módulo.

2.2.1 Proceso para resolver conflictos de integración

Para el proceso de la resolución de conflictos, usamos las herramientas que el control de versiones proporciona. Entre ellos el algoritmo de integración interno que resuelve una gran cantidad de conflictos automáticamente. Para casos más específicos,

se recorrió a la utilidad git merge que permite ver los problemas de integración y arreglarlos manualmente.

2.3 Proceso de mayor dificultad en Git/GitHub.

En este proyecto una de las cosas que más se nos dificultó fue el trabajo por separado de las branch en el que cada subgrupo trabajó de forma independiente. Aunque existe mucha información al respecto en internet, el trabajo de hacer un correcto merge es algo bastante complicado.

3 Aprendizaje de las últimas semanas.

Una de las cosas que más le sacamos provecho a la realización del código de la tarea 3 durante el verano fue la incomodidad para hacer reuniones físicas donde estuviesen todos los integrantes del grupo. Debido a esta limitante, nos vimos en la necesidad de trabajar en forma remota pero comunicada. Haciendo un uso eficiente del tiempo y de las responsabilidades de cada quien se ideó un plan de acción. Gracias a esta forma de trabajar pudimos avanzar y superar este obstáculo.

Para la realización de esta tarea ya teníamos en gran parte adelantado nuestro plan de acción y nos funcionó en gran medida ya que para esta tarea solo nos reunimos una vez en donde realizamos una división de las partes que le tocaban a cada subgrupo. En esta reunión también se habló de la correcta utilización de los repositorios git. De esta forma todos estuvimos de acuerdo y se continuó con el trabajo.

Así pues se podría decir que una de las cosas más interesante fue el trabajo en equipo hecho sin la necesidad de compartir espacio físico, que fue algo nuevo para nosotros.

Otra cosa que se aprendimos esta semana y tuvo un importante repercusión a la hora de hacer la tarea, fue lo fácil que es programar con patrones de diseño. Como estos cuando se usan adecuadamente simplifican en gran medida el trabajo y el entendimiento del código.

De las herramientas de GitHub, una de las más interesantes para el equipo es su capacidad de llevar estadísticas de los contribuyentes y sus commits. Así siempre se puede mantener un record de la eficiencia y el trabajo de cada integrante del equipo.

4 Organización del Equipo

4.1 Coordinador de la gestión de configuraciones

Para la gestión de configuraciones usamos Github. El encargado del correcto uso de la herramienta y el encargado de “*asegurar la calidad de todo producto obtenido durante cualquiera de las etapas del desarrollo*”³ fue **Gustavo El Khoury**. Gustavo también se encargó de llevar las pruebas necesarias para confirmar la satisfacción de los requerimientos.

4.2 Repartición de las tareas

La repartición del trabajo para esta tarea fue propuesta en la reunión efectuada el día viernes de 4 Octubre. Donde Rebeca, Gustavo, José Luis y Leopoldo iban a trabajar el patrón estrategia y el patrón decorador iba a ser desarrollado por Gamar, Ronsangelis y Reinaldo.

4.3 Organización por parejas

Ese día (viernes 4 de octubre), fue decisión unánime desarrollar la tarea dividida en dos partes, para las cuales unas parejas iban a realizar el Patrón Decorador y las otras parejas iban a realizar el Patrón Estrategia. En la parte del patrón decorador el encargado fue Reinaldo y en la parte del patrón estrategia el encargado fue José Luis. Ellos se encargaron cada quien de que su parte estuviese bien documentada y cumpliera los requerimientos de la tarea.

³ (Fundación Wikimedia, Inc., 2013)

4.4 Responsable de las integraciones

El responsable de las integraciones para esta tarea fue Gamar, él fue el encargado de los “procesos utilizados [...] para asegurar que los elementos del proyecto están siendo correctamente coordinados.”⁴. Adicionalmente trabajó conjuntamente con Gustavo para asegurar el correcto funcionamiento del software una vez unidas las dos partes que los conformaban.

4.5 Responsable de actas/acuerdos de las reuniones (físicas y virtuales)

Para este trabajo se designó a Leopoldo. Fue él quien llevó el acta que se levantó el viernes. También se le asignó la ejecución de este informe.

⁴ (Burburaca Business Solutions)

5 Conclusiones

Esta tarea significó varios retos para el equipo; retos que asumimos y trabajamos sobre su base. Retos como el trabajar de modo remoto y trabajar con ramas (branches) en Github. Retos que nos ayudaron a madurar el código y a lograr una fuerte cohesión de este.

Después del taller sobre patrones de diseño, logramos entender y estuvimos mucho más claros de cuál era el alcance de la tarea, cuáles eran sus requerimientos y qué era lo que teníamos que hacer.

Durante la primera semana dividimos el trabajo en dos y cada sub equipo implementó uno de los patrones necesarios para la entrega de la tarea.

Se pudo observar que los patrones de diseño, lejos de complicar la implementación y el trabajo necesario, más bien facilitaban el entendimiento de los requerimientos y ayudaban en gran parte en la programación del código, dejando un código más complejo y más trabajado pero también mucho más entendible y más fácil de digerir por una persona externa al grupo. Así mismo, la distribución de las responsabilidades y la buena interacción como grupo hicieron posible la culminación de la tarea.

6 Bibliografía

Fundación Wikimedia, Inc. (12 de 03 de 2013). *Gestión de la configuración*. Recuperado el 5 de Octubre de 2013, de Wikipedia: http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_la_configuraci%C3%B3n

Burburaca Business Solutions. (s.f.). *Gestión de integración de proyectos*. Obtenido de Burburaca: <http://www.burburaca.com/gestion-proyectos/11-gestion-de-integracion-de-proyectos.html>

Martínez, P. I. (2013). PATRONES DE DISEÑO. 7-8.

Real Academia Española. (2013). Recuperado el 5 de Octubre de 2013, de Diccionario de la Real Academia Española: <http://lema.rae.es/drae/?val=patron>