



REPÚBLICA BOLIVARIANA DE VENEZUELA
MINISTERIO DEL PODER POPULAR PARA LA EDUCACIÓN SUPERIOR
UNIVERSIDAD SIMÓN BOLÍVAR
INGENIERÍA DE SOFTWARE I
PROF. TERUEL ALEJANDRO – PROF. JEAN CARLOS GUZMÁN –
PROF. ALFONSO REINOZA
CI-3715

Informe Tarea 3

ALUMNO:

Goudet David #08-10479 / davidgccs@gmail.com

Sánchez Jesús #10-10898 / jhsanchezz@gmail.com

CARACAS, ENERO DE 2015



INDICE GENERAL

1. Introducción	2
2. Resumen de la experiencia	3
3. Conclusiones Generales	4
4. Referencias	5



INTRODUCCIÓN

El presente informe se presenta tras asumir la resolución de un problema propuesto por el equipo docente de la materia. En él, se pretende brindar una solución precisa y adecuada al planteamiento de programar una aplicación simple asociada a un sistema de reservas para un estacionamiento, haciendo énfasis en los múltiples casos de prueba que podrían llegar a presentarse enfocados en una visión “maliciosa” de testeo y asumiendo como técnica de desarrollo principal la *programación por pares*.

El algoritmo de Marzullo, creado por Keith Marzullo, de uso expandido en redes, ha sido el seleccionado para hallar resolución al planteamiento. En este orden, como entrada se tienen una lista de tuplas que consisten en dos números: hora de entrada y de salida de una reservación de estacionamiento. Como salida se devuelve True si la lista de reservas es factible y False si no lo es.

El algoritmo de Marzullo en este caso contiene una variable de puestos actualmente ocupados que va aumentando o disminuyendo conforme se van reservando o vaciando puestos, de esta forma se puede verificar que el máximo de puestos (10) no sea superado en un mismo intervalo de tiempo determinado. Un subproducto de este proceso son las variables *beststart* y *bestend*, correspondientes a las horas de inicio y fin del mejor lapso de reservación.

El informe contempla aspectos importantes como:

- 1 – Resumen de la experiencia.
- 2 – Conclusiones generales.
- 3 – Referencias.



RESUMEN DE LA EXPERIENCIA

Nuestra experiencia con este tipo de programación fue un poco difícil de llevar debido a la disposición de ambas partes; nuestras clases, compromisos y zonas de residencia nos impidieron tener un acercamiento más enriquecedor con la técnica, sin embargo, el tiempo dispuesto para su ejecución mostró claras ventajas sobre el método de programación regular que veníamos aplicando en el pasado.

La técnica lleva a ambas partes a familiarizarse *si o si* con todos los aspectos y partes del código lo que permite la detección de errores de forma más efectiva, sin mencionar que el roce continuo con otro modo de programar supone un aprendizaje importante asociado al modo en el cual “el otro” desarrolla.

La estructura para el repositorio que decidimos mantener fue la de hacer *commit* tras avances importantes y tras cada vez que los roles se intercambiaban al tronco principal.



CONCLUSIONES GENERALES

El objetivo primordial de este trabajo fue el de brindar una solución adecuada y acorde al enunciado presentado por el equipo docente. En él se brindó una solución al problema de implementar un programa que gestionara las reservaciones para un estacionamiento determinado, haciendo especial énfasis en la técnica de desarrollo por pares y en la generación de casos borde donde dichas reservaciones podrían ocasionar problemas.

Se presentaron conflictos a la hora de manejarnos con la técnica antes mencionada debido a dificultades de tiempo, compromisos y lugares de residencia. Cuando las reuniones eran imposibles de hacerse personalmente se hizo uso de herramientas de comunicación web (*Skype*) para la conexión y entendimiento de las partes. Sin embargo, con lo que se pudo experimentar, la sensación que nos llevamos es sumamente satisfactoria, su relación con la producción de software de calidad y la mejora tácita en los tiempos de programación son verdaderamente destacables; sin lugar a dudas, se pretende usar este tipo de desarrollo en futuras asignaturas.



REFERENCIAS

[1] Javier Garzás, Beneficios de la programación por pares. En <http://www.javiergarzas.com/2012/06/beneficios-pair-programming.html>

[2] Leonardo Sotillo, Universidad de Oviedo – Departamento de Informática. Sincronización en los sistemas distribuidos. Lección 11 – ATC Distribuidas.