Metodología de la programación II
Practica VII

Emiliano Salvatori

Octubre 2019

1. Practica VII

Reflexión e Integración Continua

¿Qué es la Reflexión?

En informática, reflexión (o reflexión computacional) es *la capacidad que tiene un programa para observar y opcionalmente modificar su estructura de alto nivel.*

En un sentido más amplio, la reflexión es una actividad computacional que razona sobre su propia computación. [1]

¿La Reflexión es estática o dinámica?

Normalmente, la reflexión es dinámica o en tiempo de ejecución, aunque algunos lenguajes de programación permiten reflexión estática o en tiempo de compilación. Es más común en lenguajes de programación de alto nivel ejecutándose sobre una máquina virtual, como Smalltalk o Java, y menos común en lenguajes como C. [2]

¿Qué proporciona un lenguaje de reflexión?

Un lenguaje con reflexión proporciona un conjunto de funcionalidades disponibles en tiempo de ejecución que serían muy difícilmente realizables en un lenguaje de más bajo nivel.

Funcionalidades típicas:

- Descubrir y modificar construcciones de código fuente (bloques, clases, métodos, protocolos).
- Convertir una cadena que corresponde al nombre simbólico de una clase o función en una referencia o invocación a esa clase o función.
- Evaluar una cadena como si fuera una sentencia de código fuente.

¿Qué es la integración Continua?

La integración continua es un modelo informático que consiste en hacer la compilación y ejecución de pruebas de todo un proyecto de manera automática lo más a menudo posible, para poder detectar fallos cuanto antes.

Implica la fusión, varias veces al día, de todas las copias de los trabajos de los desarrolladores con una línea principal compartida.

¿En qué consiste el proceso?

El proceso consiste normalmente en descargar cada X cantidad de horas el código fuente desde el control de versiones, compilarlo, ejecutar las pruebas y generar informes.

Se usan aplicaciones como Anthill, Bamboo, Continuum, Hudson o Jenkins para proyectos Java, y CruiseControl.Net o Team Foundation Build para .Net, que se encargan de controlar las ejecuciones y se apoyan en herramientas como Ant o Maven (para Java) y Nant o MSBUILD (para .Net) que se encargan de las compilaciones, pruebas e informes.

[1] https://es.wikipedia.org/wiki/Reflexi%C3%B3n_(inform%C3%A1tica)
[2] https://es.wikipedia.org/wiki/Reflexi%C3%B3n_(inform%C3%A1tica)

¿Por qué la Integración Continua está más vinculada a las metodologías ágiles que a las tradicionales?

La integración continua está estrechamente asociada con el desarrollo ágil de software, como la metodología de programación extrema.

El modelo de Integración Continua está más vinculado a la comunicación cara a cara con el cliente por sobre el soporte técnico, esto fue establecido por Kent Beck para su metodología XP.

Describa algunas ventajas de la Integración Continua

Algunas de las ventajas son:

- Se pueden detectar y solucionar los problemas de integración de manera continua, evitando el tradicional caos que aparece cuando se acerca la fecha de entrega.
- Permite disponer constantemente de una versión para pruebas, demos o lanzamientos anticipados. Con la Integración Continua se pueden monitorear las métricas de calidad del proyecto prácticamente en tiempo real.