

# ESTUDIO DE ADOPCIÓN DE TÉCNICAS DE DESARROLLO DE SOFTWARE GUIADO POR LAS PRUEBAS

Calixto Maldonado, Iris Gastañaga, Claudia Inchaurredo, Pablo Vaca, Matías Bueno, María Soledad Romero, Juan Pablo Peretti

Grupo de Investigación, Desarrollo y Transferencia de Sistemas de Información (GIDTSI)/ Departamento Ing. Sistemas de Información / Facultad Regional Córdoba / Universidad Tecnológica Nacional  
Maestro López y Cruz Roja

[calixtomaldonado@hotmail.com](mailto:calixtomaldonado@hotmail.com), [vacapablo72@gmail.com](mailto:vacapablo72@gmail.com), [matiasbueno@gmail.com](mailto:matiasbueno@gmail.com),  
[cinchaurredo@hotmail.com](mailto:cinchaurredo@hotmail.com), [romeroma.soledad@gmail.com](mailto:romeroma.soledad@gmail.com), [peretti.juan@gmail.com](mailto:peretti.juan@gmail.com)

## Resumen

En el presente proyecto, dentro del Grupo de Investigación, Desarrollo y Transferencia de Sistemas de Información (GIDTSI), se estudiaron las implementaciones existentes de los frameworks aplicados en Test-Driven Development (TDD) y Behavior-Driven Development (BDD) y como resultado al final del mismo se prevé proponer una forma de integración entre distintas técnicas de (TDD) y (BDD). Para el estudio de campo entre las empresas de desarrollo de software se ha creado un formulario de encuesta con fecha próxima de publicación para ser completado por responsables de área de Desarrollo de software y de Testing.

**Palabras clave: TDD, BDD, Ingeniería de Software, testing**

## Contexto

En la Facultad Regional Córdoba de la Universidad Tecnológica Nacional el Grupo GIDTSI tiene como una de sus líneas de investigación a la Ingeniería en Sistemas de Información y de Software, incluyendo proyectos de investigación en

calidad, métricas, mantenimiento, trazabilidad, requerimientos, bases de datos, seguridad, relacionados con las áreas de conocimiento: Sistemas de Información, Gestión Ingenieril y Programación.

A inicios del 2013 se observó en el país una fuerte tendencia a aplicar esfuerzos en TDD (Test-Driven Development) y además, se estaba empezando a trabajar en BDD (Behavior-Driven Development). Se conocía que para tal fin existían diferentes frameworks tales como JUnit o SUnit, los cuales se encontraban en un avanzado grado de desarrollo e implementación.

Si bien todas las metodologías de desarrollo de software basadas en las pruebas son una disciplina relativamente nueva, ya que han cobrado auge en los últimos 10 años, podemos nombrar algunos referentes a nivel internacional como Kent Beck (1) y Martín Fowler (2). En el ámbito local, actualmente en la UTN Facultad Regional Córdoba no hay grupos trabajando en la materia, y en el resto del país solamente un grupo mixto

del LIFIA (UNLP) y de la Facultad de Ingeniería de la UBA están investigando la aplicación de estos enfoques al desarrollo de sistemas web.

## Introducción

El desarrollo de software se ha transformado a lo largo del tiempo de un arte a un proceso detallado tendiente a obtener software de alta calidad, mantenible y por sobre todo confiable a lo largo del tiempo (1) (2). Es conocido en el ámbito del desarrollo de software que el 70% del costo de un software a lo largo de la vida útil del mismo es consumido en la etapa de mantenimiento (3) (4), por lo tanto es necesario encontrar mecanismos, ya sean manuales o automáticos que nos permitan garantizar que lo que se encuentra funcionando en un software siga funcionando luego de la introducción de nuevas características.

El desarrollo de software guiado por el negocio (BDD y STDD por sus siglas en inglés) o el desarrollo de software guiado por las pruebas unitarias o de integración (TDD y ATDD por sus siglas en inglés y todas sus variantes) son dos alternativas que buscan garantizar que el desarrollo progresivo e incremental del software se hace de manera controlada y salvaguardando el software ya desarrollado, son dos técnicas que se pueden implementar en conjunto o por separado, en ambos casos son técnicas que permiten automatización logrando de esta manera la reducción en los tiempos de pruebas unitarias, pruebas de regresión y pruebas de integración (5).

Sin embargo, varios investigadores y practicantes (5) han alertado que la proliferación de enfoques al desarrollo de software basado en pruebas disminuye su aplicabilidad en la industria. Es por esto que durante el proyecto de investigación se propuso desarrollar las siguientes actividades:

- Investigar los procesos de desarrollos basados en la generación de las pruebas como primer paso en el desarrollo de software (BDD, TDD, ATDD, STDD, NDD, DDD) comparando los diversos enfoques según la óptica de diversos autores y agregando nuestro punto de vista basado en la experiencia profesional en el ámbito privado utilizando estas metodologías.

- Investigar los distintos software y arquitecturas existentes para la automatización de las pruebas, comparando los mismos, instalando y probando cada uno de ellos, generando conclusiones y exponiendo las mismas en documentación con el objetivo de compartir con la comunidad académica nuestras conclusiones. A modo de ejemplo de los software a investigar se citan los siguiente: Jbehave (6), Selenium (7), JUnit (8) (9), NUnit, easyb (10) (11), RSpec(12), HttpUnit. Las pruebas se ejecutaro sobre dos plataformas de sistemas operativos una en Linux y otra sobre Windows, y como lenguajes de programación se utilizarán Java, Visual C++, Visual Basic, HTML, Javascript, en el caso de servidores web se utilizará Apache Tomcat.

- Investigar los software y arquitecturas asociadas al proceso, el cual se utiliza para medir la calidad de las pruebas y la cobertura que tienen los test sobre el código que se está probando, tales como Maven(14) (15), Visual Studio (16) (17), SimpleCov(18) (19). Se instaló el software y se ejecutaron pruebas del mismo.

- Como resultado de la investigación se generará una propuesta de integración de varios enfoques y se estructurará una propuesta de desarrollo de un framework (software y arquitectura) que permita la integración automática, facilitando la aplicación de estos métodos de desarrollo en el ambiente industrial.

- A modo de transferencia de los conocimientos adquiridos y generados en los pasos anteriores, se trabajará en la difusión de los resultados en cuatro aspectos:

1. Se buscará interesar a las cátedras de análisis, diseño y programación para realizar intercambio de transferencia al aula de los avances logrados en este trabajo.

2. Se buscará interesar a los laboratorios de desarrollo de software existentes en la universidad para el intercambio de experiencias y realizar transferencia de los avances en el proyecto.

3. Se enviaron artículos de difusión y científicos a los siguientes congresos WICC, JAIIO.[1] [2].

## **Líneas de Investigación, Desarrollo e Innovación**

La investigación completa se basó en procesos de investigación y se utilizarán técnicas deductivas y exploratorias, pero siempre apuntando a una investigación aplicada.

Uno de los objetivos del presente proyecto es realizar los contactos con grupos que se encuentren trabajando en la materia, tanto a nivel nacional como en el extranjero, con el fin de crear vínculos y lograr intercambio de conocimientos. Por otra parte, esta metodología está cobrando importancia dentro de las empresas que desarrollan software y se está volviendo una práctica normal y obligatoria durante el ciclo de vida de los proyectos de software. Por ejemplo, la empresa en la cual trabaja el Ing Pablo Vaca (Globant) es una de las que está dando mucho impulso al uso de estas prácticas.

Actualmente los más utilizados son los basados en BDD y TDD. Integrantes del equipo de investigación trabajan habitualmente en el ámbito privado con ambas técnicas. Se disponen de conocimientos sobre ambas técnicas pero con implementaciones separadas, mediante el presente proyecto se buscará profundizar en la integración entre distintas técnicas.

Por otra parte los resultados obtenidos del proyecto son la base para la formación de recursos humanos capacitados para desenvolverse en el ámbito privado y público, así como la aplicación de la gama de servicio que ofrece el laboratorio

de desarrollo de Software, generando de esta forma puestos de trabajo de alta nivel y requeridos a nivel mundial.

Las actividades fueron: una etapa exploratoria, en la cual se hizo un estudio bibliográfico de fuentes conocidas e investigar nuevos trabajos publicados sobre la materia, a la cual le siguió una etapa de investigación aplicada, probando productos de código libre y abierto disponible y comparando sus funcionalidades.

Para la elaboración de papers se trabajó planteando hipótesis y luego mediante investigación bibliográfica, intercambio con otros grupos y pruebas desarrolladas sobre el software instalado se buscará demostrar la misma o explicar por que dicha hipótesis no es correcta.

Para la instalación de software y entornos de ejecución de pruebas se utilizaron técnicas deductivas, partiendo de la lectura de la documentación de cada software y de foros que traten el tema, documentado cada paso y situación a resolver, con la finalidad de armar una base de conocimiento sobre el tema, que luego sirva para la elaboración de documentación de base en las transferencias que se realicen.

Las tareas de formación de recursos humanos se basaron en un plan específico de formación, previendo se realizar al menos dos presentaciones abiertas a la comunidad en general y con invitación a las cátedras de programación e Ingeniería de Software de la Carrera de Ingeniería en Sistemas de Información de la UTN-FRC

de los temas principales del proyecto. Se prevé publicar en la web la presentación electrónica, y si los medios disponibles lo permiten obtener un registro en video de la misma.

### **Objetivos Generales del Proyecto:**

- Investigar la posible integración de las distintas metodologías de desarrollo guiado por pruebas.
- Proponer un marco teórico para la integración de algunas de estas metodologías y establecer los requerimientos de un framework que permita automatizar la aplicación integrada de dichas metodologías.

### **Objetivos Particulares del Proyecto:**

- Estudiar las metodologías de desarrollo guiado por pruebas y establecer un foco de transferencia de resultados hacia la universidad y la industria.
- Instalar un entorno real de pruebas que le permitan al equipo experimentar y poner en práctica las conclusiones alcanzadas durante la investigación.
- Formar recursos humanos especializados en desarrollo de software guiado por pruebas.
- Realizar talleres de capacitación en los laboratorios CIDS y LIDICaSo de la FRC
- Asistir técnicamente en las mejores prácticas a grupos de Investigación de la

disciplina Homologados en la SCyT de la FRC – UTN Exponer los resultados alcanzados y los objetivos en curso/futuros de la línea de I/D/I presentada.

## Formación de Recursos Humanos

Un punto importante es la formación de profesionales capacitados en el desarrollo de software guiado por las pruebas.

Esta formación debe incluir desde los aspectos teóricos hasta los de implementación práctica. Pero también se busca que los integrantes, becarios y tesistas adquieran experiencia en la investigación y en el futuro puedan replicar y llevar adelante otros proyectos de investigación.

La experiencia adquirida en este proyecto es aplicada dentro de la universidad en grupos de investigación, cátedras de grado y en el ámbito privado.

Con estas habilidades se propiciaron que los participantes accedan a la categoría inicial de investigador dentro de la docencia investigativa.

Otro aporte a la formación de recursos humanos es la elaboración de tesis de maestría de integrantes del proyecto, con lo cual se podrá alcanzar mayores niveles académicos que permitan el progreso en el ámbito docente y profesional.

## Referencias

- (1) - Ingeniería de Software - Somerville - Addison Wesley.
- (2) - Ingeniería de Software - Pfleeger - Prentice Hall.

(3) - SW Maintenance, Concepts and Practices - Takang, Grubb - World Scientific 2005

(4) - SW Engineering Body of Knowledge (IEEE 2004 version).

(5) - Estado del arte y tendencias Test-Driven Development - Moisés, Carlos Fontela - 2011 - Trabajo Final Integrador de la Especialización en Ingeniería de Software - Universidad de La Plata - Facultad de Informática -

(6) - <http://jbehave.org/>

(7) - <http://seleniumhq.org/>

(8) - <http://www.junit.org/>

(9) - <http://es.wikipedia.org/wiki/JUnit>

(10) - <http://code.google.com/p/easyb/>

(11) - <http://www.ibm.com/developerworks/java/tutorial/s/j-easyb/>

(12) - <http://rspec.info/>

(13) - <http://en.wikipedia.org/wiki/RSpec>

(14) - <http://maven.apache.org/>

(15) - <http://en.wikipedia.org/wiki/Maven>

(16) - [http://es.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://es.wikipedia.org/wiki/Microsoft_Visual_Studio)

(17) - <http://msdn.microsoft.com/es-ar/vstudio/aa718325.aspx>

(18) - <http://rubygems.org/gems/simplecov>

(19) - <http://www.becodemyfriend.com/tag/simplecov/>  
Una exposición general sobre el tema de I/D/I con las referencias bibliográficas relacionadas. Se sugiere no más de 3 páginas.

Citas de los autores

[1] Vaca, P. A., Maldonado, C., Inchaurredo, C., Peretti, J., Romero, M. S., & Bueno, M. (2013). Test-Driven Development-Una aproximación para entender su utilidad en el proceso de desarrollo de Software. *Conaiisi-Cordoba, Octubre de*, 2347-0372.

[2] Vaca, P. A., Maldonado, C., Inchaurredo, C., Peretti, J., Romero, M. S., Bueno, M., & Cagliolo, M. Test-Driven Development-Beneficios y Desafíos para el Desarrollo de Software.