



Facultad de Ingeniería  
Escuela de Ingeniería Civil en Informática

# **DESARROLLO DE UNA PLATAFORMA PARA LA SOLICITUD Y GESTIÓN DE REQUERIMIENTOS Y SCM**

Por

**Alejandro Alvarez Ahumada**

Trabajo realizado para optar al Título de  
**INGENIERO CIVIL EN INFORMÁTICA**  
Prof. Guía: Carlos Becerra Castro  
Prof. Co-Referente: Nombre Profesor Correferente  
Septiembre 2012

Certifico que he leído este documento y que, en mi opinión, es adecuado en ámbito y calidad como trabajo para optar al título de Ingeniero Civil en Informática.

---

Carlos Becerra Castro    Profesor Guía

Certifico que he leído este documento y que, en mi opinión, es adecuado en ámbito y calidad como trabajo para optar al título de Ingeniero Civil en Informática.

---

Nombre Profesor Correferente    Profesor Co-Referente

Certifico que he leído este documento y que, en mi opinión, es adecuado en ámbito y calidad como trabajo para optar al título de Ingeniero Civil en Informática.

---

Nombre Profesor Informante 1    Profesor Informante

Aprobado por la Escuela de Ingeniería Civil en Informática, UNIVERSIDAD DE VALPARAÍSO.

# Resumen

La Dirección de Servicios de Información y Computación (DISICO) de la Universidad de Valparaíso durante los últimos años ha estado en constante crecimiento y en busca de mejoras que le permitan brindar un mejor servicio. Aunque en este poco tiempo son muchas las mejoras que se han hecho, aún quedan aspectos por mejorar, algunos de estos son los procesos relacionados a las solicitudes de requerimientos y solicitudes de cambios, para las cuales ya se han diseñado procedimientos y metodologías, sin embargo se carece de una herramienta que permita la automatización de estas. El propósito de este trabajo de título es dar solución a dicho problema mediante el desarrollo de una plataforma que permita automatizar los procedimientos actuales de solicitud de requerimientos y SCM. Los principales resultados que se esperan son disminuir el tiempo y esfuerzo invertido en la aplicación de las metodologías que existen actualmente.

# Agradecimientos

Aquí pueden colocar sus agradecimientos. Si han estudiado con becas es recomendable colocar los agradecimientos a las instituciones que les otorgaron las becas.

# Índice general

<b>Resumen</b>	<b>III</b>
<b>Agradecimientos</b>	<b>IV</b>
<b>1. Pruebas</b>	<b>1</b>
1.1. Pruebas Unitarias . . . . .	1
<b>Bibliografía</b>	<b>8</b>

# Índice de tablas

1.1. Test unitarios . . . . . 7

# Índice de figuras

# Capítulo 1

## Pruebas

En este capítulo se detallan las pruebas realizadas, junto con los resultados obtenidos durante la realización de estas. Las pruebas realizadas se dividen en:

- Pruebas Unitarias.
- Pruebas de Integración.
- Pruebas de Rendimiento.
- Pruebas de Aceptación.
- Pruebas Beta.

El principal enfoque de las pruebas es la detección de errores.

### 1.1. Pruebas Unitarias

Las pruebas unitarias están enfocadas en probar unidades pequeñas de código. Para llevarlas a cabo se diseñó un set de pruebas automatizados, las cuales fueron implementadas con el framework de pruebas JUnit y de Glassfish Embedded, dentro del cual se despliegan los EJB para ser utilizados durante la ejecución de las pruebas.

El procedimiento para ejecutar los test dentro del contenedor de Glassfish Embedded es el siguiente:

- Se copian los archivos del directorio build/classes a un directorio temporal para el despliegue de los EJB en build/embedded.
- Además se copia el archivo test-persistence.xml (que contiene una configuración diferente para la ejecución de las pruebas en una base de datos diferente a la de producción) dentro del directorio build/embedded/META-INF/ y se renombra como persistence.xml.



- Antes de levantar Glassfish Embedded se le indica que se crea un properties que le indica a Glassfish donde se encuentran los módulos a desplegar y el archivo de configuración domain.xml que debe utilizar (en este caso es un archivo aparte que contiene la configuración necesaria para la ejecución del test).
- Luego se inicia el Glassfish Embedded.
- Se ejecutan los test necesarios.
- Por ultimo se cierra Glassfish Embedded y se borra el directorio temporal build/embedded.

Para esto gestionar todo ese proceso se implemento una nueva clase denominada BaseTestEJB de la cual extienden todos los test que hacen uso de los EJB. En el caso de clases java comunes no se requiere de ninguna configuración especial para ejecutar los test, basta simplemente ejecutarlos.

En la Tabla 1.1 se encuentran documentadas todas las pruebas unitarias diseñadas, con su nombre y propósito.

Clase	Test	Propósito
Resources	testGetValue	Verificar que el método getValue es capaz de recuperar la cadena "ABCD" desde un archivo de propiedades.
	testGetValueCon Espacios	Verificar que el método getValue es capaz de recuperar la cadena "A B C D" desde un archivo de propiedades sin verse afectado por la cantidad de espacios entre los caracteres.
	testGetValueShort	Verificar que el método getValueShort es capaz de recuperar cadena desde un archivo de propiedades y convertirla a Short siempre que cumpla con el formato de este.
	testGetValueShort Negativo	Verificar que el método getValueShort es capaz de recuperar cadena desde un archivo de propiedades y convertirla a Short aunque este sea negativo.
	testGetValueShort ErrorEnString	Verificar que el método getValueShort dispara la excepción NumberFormatException al leer un String desde el archivo de propiedades.
	testGetValueShort ErrorValorMayor AShort	Verificar que el método getValueShort dispara la excepción NumberFormatException al leer un número entero que excede el valor máximo de un Short.
	testGetValueShort ErrorValorDecima	Verificar que el método getValueShort dispara la excepción NumberFormatException al leer un valor con decimales desde el archivo de propiedades.
	testGetValueInteger	Verificar que el método getValueInteger es capaz de recuperar cadena desde un archivo de propiedades y convertirla a Integer siempre que cumpla con el formato de este.
	testGetValueInteger Negativo	Verificar que el método getValueInteger es capaz de recuperar cadena desde un archivo de propiedades y convertirla a Integer aunque este sea negativo.
	testGetValueInteger ErrorValorMayorAInteger	Verificar que el método getValueInteger dispara la excepción NumberFormatException al leer un número entero que excede el valor máximo de un Integer.

Clase	Test	Propósito
Resources	testGetValueInteger ErrorValorMayorInteger	Verificar que el método getValueInteger dispara la excepción NumberFormatException al leer un numero entero que excede el valor máximo de un Integer.
	testGetValueInteger ErrorValorDecimal	Verificar que el método getValueInteger dispara la excepción NumberFormatException al leer un valor con decimales desde el archivo de propiedades.
	testGetValueLong	Verificar que el método getValueLong es capaz de recuperar cadena desde un archivo de propiedades y convertirla a Long siempre que cumpla con el formato de este.
	testGetValueLong Negativo	Verificar que el método getValueLong es capaz de recuperar cadena desde un archivo de propiedades y convertirla a Long aunque este sea negativo.
	testGetValueLong ErrorValorMayorALong	Verificar que el método getValueLong dispara la excepción NumberFormatException al leer un numero entero que excede el valor máximo de un Long.
	testGetValueLong ErrorValorDecimal	Verificar que el método getValueLong dispara la excepción NumberFormatException al leer un valor con decimales desde el archivo de propiedades.
	testGetPropertiesPath	Verifica que el metodo getPropertiesPath obtenga la ruta de un archivo properties correctamente.
	testGetPropertiesPath NotFound	Verifica que el metodo getPropertiesPath dispare la excepcion MissingResourceException al intentar obtener la ruta de un propertie que no existe.
	testGetPageList	Verifica que el metodo getPageList obtenga una lista con todas las propiedades contenidas en el archivo propertie especificado.
	testGetMapPageList	Verifica que el metodo getMapPageList obtenga un Map con todas las propiedades contenidas en el archivo propertie especificado.

Clase	Test	Propósito
MathUtil	testCalcularPorcentajeRedondeado	Verifica que el calculo del porcentaje entero sin decimal.
	testCalcularPorcentajeRedondeadoParaArriba	Verifica que el porcentaje sea redondeado hacia arriba cuando el decimal es mayor o igual a 5.
	testCalcularPorcentajeRedondeadoParaAbajo	Verifica que el porcentaje sea redondeado hacia abajo cuando el decimal es menor a 5.
	testCalcularReglaDeTres	Verifica el calculo de una regla de tres.
	testCalcularReglaDeTresParaDenominadorCero	Verifica que se devuelva un 0 en caso que el denominador sea 0.
	testCalcularReglaDeTresParaNumeradorCero	Verifica que se devuelva un 0 en caso que el numerador sea 0.
	testCalcularReglaDeTresParaCien	Verifica el caso en que se debe devolver cien.
	testCalcularReglaDeTresParaPorcentajeMayorACien	Verifica que funcione cuando el numerador es mayor a cien.
	testRedondearCero	Verifica la funcion redondear cuando el numero es cero.
	testRedondearParaArribaSinDecimal	Verifica que el numero sea redondeado hacia arriba cuando el decimal es mayor o igual a 5 y el resultado no debe tener ningún decimal.
	testRedondearParaAbajoSinDecimal	Verifica que el numero sea redondeado hacia abajo cuando el decimal es menor a 5 y el resultado no debe tener ningún decimal.
	testRedondearParaArribaConUnDecimal	Verifica que el numero sea redondeado hacia arriba cuando el decimal es mayor o igual a 5 y el resultado no debe tener un decimal.
	testRedondearParaAbajoConUnDecimal	Verifica que el numero sea redondeado hacia abajo cuando el decimal es menor a 5 y el resultado no debe tener un decimal.
	testRedondearParaArribaConDecimal5	Verifica el caso en que el decimal es 5.
	testRedondearFloatCero	Verifica la función redondear cuando el numero es cero. Cuando el resultado es convertido de BigDecimal a Float.

Clase	Test	Propósito
MathUtils	testRedondearFloatParaArribaSinDecimal	Verifica que el numero sea redondeado hacia arriba cuando el decimal es mayor o igual a 5 y el resultado no debe tener ningún decimal. Cuando el resultado es convertido de BigDecimal a Float.
	testRedondearFloatParaAbajoSinDecimal	Verifica que el numero sea redondeado hacia abajo cuando el decimal es menor a 5 y el resultado no debe tener ningún decimal. Cuando el resultado es convertido de BigDecimal a Float.
	testRedondearFloatParaArribaConUnDecimal	Verifica que el numero sea redondeado hacia arriba cuando el decimal es mayor o igual a 5 y el resultado no debe tener un decimal. Cuando el resultado es convertido de BigDecimal a Float.
	testRedondearFloatParaAbajoConUnDecimal	Verifica que el numero sea redondeado hacia abajo cuando el decimal es menor a 5 y el resultado no debe tener un decimal. Cuando el resultado es convertido de BigDecimal a Float.
	testRedondearFloatParaArribaConDecimal5	Verifica el caso en que el decimal es 5. Cuando el resultado es convertido de BigDecimal a Float.
TimerSolicitudRequerimientosUpdateTest	testBuscarSolicitudesVencidasVerificarCambio	Verifica que el método buscarSolicitudesVencidas haya cambiado el estado de una solicitud enviada cuya fecha de vencimiento ya fue superada.
	testBuscarSolicitudesVencidasVerificarNoCambioVencida	Verifica que el método buscarSolicitudesVencidas no haya cambiado el estado de solicitudes que ya se encontraban en estado vencida.
	testBuscarSolicitudesVencidasVerificarNoCambioCerrada	Verifica que el método buscarSolicitudesVencidas no haya cambiado el estado de solicitudes que se Cerraron después de la fecha de vencimiento.
	testBuscarSolicitudesVencidasVerificarNoCambioFinalizadaSinRespuesta	Verifica que el método buscarSolicitudesVencidas no haya cambiado el estado de solicitudes que se Finalizaron sin respuesta después de la fecha de vencimiento.
	testBuscarSolicitudesVencidasVerificarNoCambioSolicitudSinFechaVencimiento	Verifica que el método buscarSolicitudesVencidas no haya cambiado el estado de solicitudes que no poseen fecha de vencimiento

Clase	Test	Propósito
SolicitudRequerimientoEJB	testGenerarCodigoNumCero	Verifica que el código es vacío cuando el número es cero.
	testGenerarCodigoNumNegativo	Verifica que el código es vacío cuando el número es negativo.
	testGenerarCodigoNumUno	Verifica que el código es "q" cuando el número es 1.
	testGenerarCodigoMaxLong	Verifica que el código es "2teCogGBXee" cuando el número es el máximo valor de un Long.
	testGenerarCodigoConsultaNoDuplicados	Verifica que el código no se repite aunque sea generado en instantes de tiempo casi iguales y con ruts similares.
	testValidarCodigoConsultaExistente	Verifica que un código de consulta sea invalido cuando ya existe.
	testValidarCodigoConsultaInexistente	Verifica que un código de consulta es invalido cuando no existe.
	testEnviarSolicitud	

Tabla 1.1: Test unitarios

# Bibliografía

- [1] Oracle, “The java ee 6 tutorial.” Último acceso: 1 Junio 2012, <http://docs.oracle.com/javase/6/tutorial/doc/>.
- [2] I. Sommerville, *Ingeniería de Software*. Pearson Addison Wesley, 2005. 7ma Edición.
- [3] Rectoría de la Universidad de Valparaíso, “Decreto exento nº 03301.” Último acceso: 27 Junio 2012, [http://uv.cl/universidad/descargas/archivos/decreto\\_03301.pdf](http://uv.cl/universidad/descargas/archivos/decreto_03301.pdf).
- [4] Dirección de Extensión y Comunicaciones, “Manual de normas gráficas de la universidad de valparaíso.” Último acceso: 27 Junio 2012, <http://uv.cl/universidad/descargas/archivos/manualdenormasUV.pdf>.
- [5] P. Méndez, “Desarrollo de Metodologías de SQA y SCM para la Dirección de Servicios de Información y Computación,” Trabajo de Título, Universidad de Valparaíso, 2011.
- [6] Roger Pressman, *Ingeniería del Software: Un enfoque practico*. McGraw-Hill, 2005.
- [7] C. Jurado, *Diseño Ágil con TDD*. Lulu, 2010.
- [8] Object Mentor, “JUnit.org resources for test driven development.” Último acceso: 24 Junio 2012, <http://www.junit.org/>.
- [9] JBoss Community, “Arquillian - write real tests!” Último acceso: 24 Junio 2012, <http://www.jboss.org/arquillian.html>.
- [10] The Apache Software Foundation, “Apache jmeter.” Último acceso: 27 Junio 2012, <http://jmeter.apache.org/>.
- [11] Oracle, “Visualvm all-in-one java troubleshooting tool.” Último acceso: 27 Junio 2012, <http://visualvm.java.net/>.

- [12] M. I. M. Estellés, M. C. Bria, P. L. Torres, and F. S. Grueso, “Gestión de requisitos basada en pruebas de aceptación: Test-driven en su máxima expresión.,” pp. 61–72, 2010.