



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

**Escola Tècnica Superior d' Enginyeria
Informàtica**

PROYECTO FIN DE CARRERA /
PROJECTE FI DE CARRERA

**Integración Continua y su aplicacion
a gvHIDRA Utilizando Selenium y
Hudson**

Para optar a la titulación de / per a optar a
la titulació de

Ingeniero Informatico

presentado por / presentat per

Yazan Hijazi

Dirigido /tutorizado por

dirigit / tutoritzat per

Gaspar Quiles Gomis

València, 24 de Febrero de 2012

Índice

1. Introducción.....	3
1.1. Definición de la Integración Continua.....	3
1.2. Estado del arte y Estrategias actuales.....	3
1.3. Motivación del proyecto.....	6
1.4. Objetivos.....	6
2. El Framework gvHIDRA y herramientas utilizadas.....	8
2.1. El Framework gvHIDRA.....	8
2.2. Tecnologías utilizadas.....	8
2.2.1. PHPUnit.....	9
2.2.2. Selenium.....	9
2.2.3. Hudson.....	10
2.3. Metodología de Desarrollo.....	10
3. Arquitectura y componentes del sistema	12
4. Análisis.....	14
4.1. La aplicación de personal de la CIT.....	14
4.2. Casos de Uso.....	14
4.2.1. Mantenimiento de clases de permisos.....	14
4.2.2. Solicitud de permisos y licencias.....	15
5. Diseño de los Tests.....	19
5.1. El Concepto de las Pruebas Unitarias.....	19
5.2. Generación de los Tests necesarios.....	19
5.2.1. Tests para el “Mantenimiento de clases de permisos”.....	19
5.2.2. Tests para “Solicitud de permisos y licencias”.....	31
6. Pruebas y Resultados.....	68
6.1. Lanzamiento de los tests.....	68

6.2. <i>Los Resultados</i>	69
7. Conclusiones	71
7.1. <i>Aportaciones realizadas</i>	71
7.2. <i>Problemas Encontrados y Mantenimiento Futuro</i>	71
8. Bibliografía	73
9. Apéndice: Patrones para la adaptación de los tests	74
10. Anexo. Instalación de PHPUnit, Selenium y Hudson	76
10.1. <i>PHPUnit</i>	76
10.2. <i>Selenium</i>	76
10.3. <i>Hudson</i>	77
10.4. <i>Ejemplo de uso</i> :.....	83

1. Introducción

Los proyectos de software implican a una gran cantidad de ficheros que deben ser organizados para construir una aplicación. Hacer el seguimiento de todos estos ficheros es costoso, sobre todo cuando hay varias personas involucradas. Para facilitar esta tarea han surgido las herramientas de gestión del código fuente o control de versiones, como el repositorio Subversion o CVS.

Un servidor de integración continua vigila el repositorio y, cada vez que se hace una entrega para el servidor, recoge el código fuente del repositorio, lo construye, lo prueba y notifica al responsable de la entrega del resultado.

La Integración Continua es integrar un proyecto de una forma incremental y frecuente, permitiendo así encontrar problemas de integración en etapas tempranas y resolverlos rápidamente, reduciendo drásticamente las fases de integración y prueba.

1.1. Definición de la Integración Continua

El término de integración continua (CI de sus siglas en inglés) fue creado por Martin Fowler en un patrón de diseño:

“La integración continua es una práctica de desarrollo de software en la cuál los miembros de un equipo integran su trabajo frecuentemente, como mínimo de forma diaria. Cada integración se verifica mediante una herramienta de construcción automática para detectar los errores de integración tan pronto como sea posible. Muchos equipos creen que este enfoque lleva a una reducción significativa de los problemas de integración y permite a un equipo desarrollar software cohesivo de forma más rápida.”[\[1\]](#)

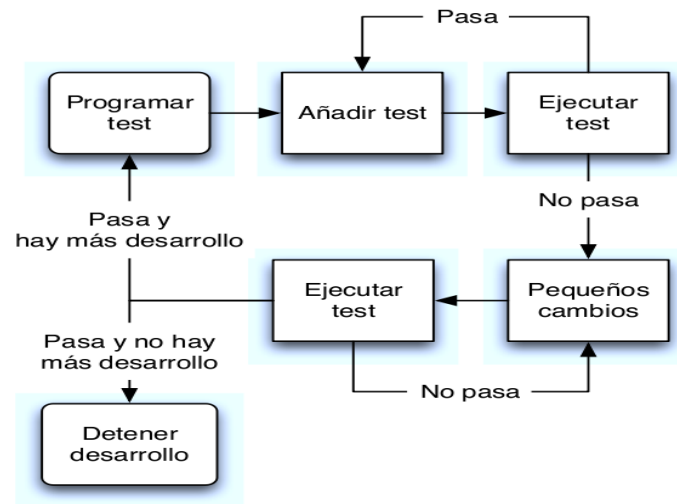
La idea principal es que con un potente conjunto de pruebas, y un sistema que permita asegurar estas pruebas de forma continua y automática, se puede asegurar que cualquier problema introducido en el sistema puede ser identificado en la próxima compilación.[\[2\]](#)

1.2. Estado del arte y Estrategias actuales

Durante todo el tiempo de la evolución de la computación y del software hasta ahora, cada vez ha ido tomando más peso lo que se considera como calidad del desarrollo, del código y del software en general.

Para ello, la calidad es una parte esencial en el producto, desde la tarea más básica hasta la más compleja.

Actualmente, para una mejora de la calidad en la programación, se suelen utilizar metodologías de Integración Continua que se basan en unos procesos continuos de desarrollo controlados permanentemente mediante tests de código. Un esquema común de este proceso se puede observar en la siguiente figura:



Las prácticas a llevar a cabo son:

- Mantener un repositorio de código.
- Automatizar la compilación e integración.
- El programador hace sus propios tests y los añade al directorio correspondiente.
- Hacer Commits diarios.
- Lanzamiento automático y periódico de los tests añadidos.
- Manejar los resultados y enviar informes a todas las partes implicadas.

Estrategias actuales:

Cuando se aborda la Integración Continua, se suelen tomar una serie de decisiones que afectan a la manera en la que se generan y se aplican los tests y al nivel de abstracción que se desea conseguir. Las resumimos en las siguientes estrategias básicas:

•Caja negra

No esta basada en el conocimiento del código o diseño interno, determina la funcionalidad del sistema.

Esta prueba implica una variada selección de los datos de prueba así como una buena interpretación de los resultados para determinar el nivel de optimización de la funcionalidad del sistema.

Se ha determinado con diferentes estudios estadísticos, que el software no debería ser probado por el creador o grupo de creadores del sistema ya que el extenso conocimiento de la estructura interna del programa limita la variedad de datos probados o el encaminamiento de las pruebas es hacia ciertos rasgos del programa, olvidando otras partes del software poco valoradas por su simpleza en la creación.

El principal objetivo es determinar la funcionalidad del software y parte de tratar al programa como si fuera una función matemática, estudiando si las respuestas o salidas son “codominio” de los datos entrantes “dominio”.

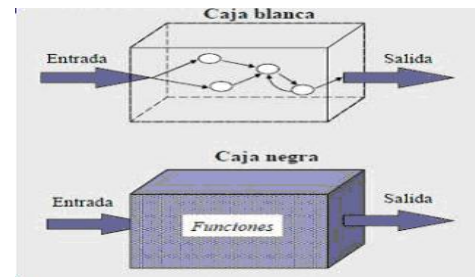
La prueba de caja negra tiene otras metas, determinar la eficiencia del programa desde el desempeño en el equipo, el tiempo de retardo de las salidas hasta el nivel de recuperación del sistema luego de fallas o caídas sean estas producidas por manejo incorrecto de datos, equipo, o producidas externamente como cortes de energía.

•Caja blanca

Esta basada en la lógica interna de la aplicación y el código. Hace una cobertura de declaraciones del código, ramas, caminos y condiciones.

Para esta prueba se consideran tres importantes puntos.

- I) Conocer el desarrollo interno del programa, determinante en el análisis de coherencia y consistencia del código.
- II) Considerar las reglas predefinidas por cada algoritmo.
- III) Comparar el desarrollo del programa en su código con la documentación pertinente.



La primera parte de esta prueba es el análisis estático.

- Análisis estático Manual.
- Inspección: Determina si el código está completo y correcto, como también las especificaciones.
- Walkthrough: Interrelación informal entre testers, creadores y usuarios del sistema.
- Verificación estática: Compara los valores generados por el programa con los rangos de valores predefinidos haciendo una descripción del funcionamiento de los procedimientos en términos booleanos determinando los puntos de falla.

La selección de datos de prueba es una de las más importantes disciplinas dentro de la caja blanca. Usualmente se generaban en forma aleatoria y hacían un acercamiento a una sofisticada prueba estructural determinando el desempeño de los módulos con dichos valores.

•Unidad de testeo o prueba

Es la escala más pequeña de la prueba, esta basada en la funcionalidad de los módulos del programa, como funciones, procedimientos, módulos de clase, etc... En ciertos sistemas también se verifican o se prueban los drivers y el diseño de la arquitectura.

•Integración incremental

Cuando nuevas funciones son incorporadas al sistema se hace la prueba basándose en la funcionalidad, la dependencia con otros módulos y la integración con el programa completo.

•Prueba de integración

Se basa en las pruebas de conexiones y comunicaciones entre diferentes módulos. Es esencial en sistemas de cliente-servidor o red.

•Prueba funcional

La caja negra hace la prueba funcional de los requerimientos de la aplicación y generalmente es realizada por el programador, en cambio, la prueba funcional es realizada por los testers.

•Prueba de sistema

Es una prueba de caja negra incluyendo todos los componentes del sistema desde el hardware a la documentación.

•Prueba de sanidad

Determina si la nueva versión de un software esta bien realizada y si necesita un nuevo esfuerzo en la prueba de software. Por ejemplo la nueva versión de un programa cumple con casi todos los requisitos pero destruye la base de datos al leerla, por lo tanto se dice que este software no esta en una condición sana.

•Prueba de carga

Esta basada en las aplicaciones bajo cargas pesadas, generalmente usadas en sitios web y en servidores con gran cantidad de datos donde se determina en cuales puntos existen degradaciones del sistema.

•Prueba de seguridad

Evalúa en qué medida el sistema se protege contra accesos, internos o externos, no autorizados. Esta prueba requiere sofisticadas técnicas y herramientas.

•Prueba de compatibilidad

Evalúa el desempeño del software en diferentes arquitecturas, plataformas, sistemas operativos, etc...

Esas eran las estrategias más comunes que se aplican hoy en día dentro del proceso de integración continua, aunque en realidad hay otras varias, las cuales no incluimos en este texto ya que este proyecto se concentrará en gran parte de su contenido en la aplicación de las “Pruebas Unitarias” sobre una plataforma concreta “gvHIDRA” estudiando las herramientas y los métodos utilizados para llevar a cabo esta tarea.

1.3. Motivación del proyecto

Los motivos para el planteamiento de este proyecto fin de carrera, se deducen de las muchas ventajas de la Integración Continua que resumiremos a continuación, y la necesidad de aprovecharlas en la construcción y la puesta en marcha de un sistema de Integración Continua para las aplicaciones realizadas en la Conselleria de Infraestructuras y Transporte (C.I.T). Estas razones son:

- Mejorar el proceso de desarrollo de software con el framework gvHIDRA. Es interesante realizar un esfuerzo en limitar el tiempo de desarrollo y de integración de los nuevos proyectos. Para ello, se estima que con un entorno de integración continua se reducirá el impacto de estas fases.
- Detectar de forma más eficiente errores que puedan afectar al comportamiento general de las aplicaciones.
- Independizar las pruebas del Sistema Operativo o del navegador sobre el que se ejecuten. En un entorno Web, la influencia de la tecnología que soporta nuestras aplicaciones hace que pueda funcionar en unos entornos si y en otros no. Por ello, nos interesa tener una serie de validaciones que nos garanticen el óptimo funcionamiento de forma independiente a la plataforma.

1.4. Objetivos

Los objetivos principales de este proyecto fin de carrera las podemos resumir en los siguientes puntos:

1. Hacer una introducción a los conceptos de la Integración Continua y de las pruebas unitarias, definiendolas y haciendo una mención de algunas estrategias utilizadas actualmente, además de destacar

las ventajas más importantes de llevar a cabo esta práctica.

2. Presentar las herramientas de PHPUnit, Selenium y Hudson que vamos a utilizar como soluciones que nos ayudaran a implementar las pruebas unitarias (que forman una parte esencial dentro del proceso de Integración Continua).
3. Proponer una arquitectura o esquema que nos ayudará a combinar las herramientas anteriores además de los recursos disponibles (repositorios, servidores, etc...), para la generación de los test de prueba de las aplicaciones en cuestión y para el lanzamiento periódico de esos tests y su mantenimiento.
4. Presentar la aplicación del personal de la CIT, y explicar la manera de generación de los tests de prueba necesarios para un conjunto concreto de casos de uso de esta aplicación.
5. Implantar un entorno de integración. Montar la infraestructura tecnológica (creando una guía de instalación), realizar varios tests y planificar su ejecución.
6. Resolver los problemas surgidos a la hora de probar los tests anteriores, y explicar los patrones y las modificaciones más importantes a seguir para adaptar correctamente el código de los tests generados para aplicaciones gvHIDRA en general, para que puedan lanzarse sin problemas.

En resumen, el objetivo general de este proyecto es emplear las soluciones propuestas para implementar el desarrollo de tests como una acción cotidiana y fácil en la programación y el mantenimiento de código por parte de los desarrolladores que trabajan con las aplicaciones realizadas con el framework gvHIDRA. También puede ser de ayuda para cualquier programador o tester interesado en iniciarse en el mundo de la integración continua independientemente de los métodos y de las herramientas que vaya a utilizar.

2. El Framework gvHIDRA y herramientas utilizadas

2.1. El Framework gvHIDRA

gvHIDRA: Herramienta Integral de Desarrollo Rápido de Aplicaciones, y como está definido en la página web del proyecto[3], es un proyecto OpenSource (GPL v.2) cuyo objetivo es simplificar el desarrollo de aplicaciones de gestión en entornos Web. Es un framework que, siguiendo una guía de estilo (unifica criterios de aspecto y usabilidad), sirve de base para las aplicaciones.

Trabaja sobre Web usando la arquitectura modelo vista controlador (MVC) con Phrame, Smarty y Pear. Utiliza la potencia del PHP (rapidez, sencillez en los despliegues, robustez, multiplataforma) en su versión 5 (orientación a objetos, interfaces, iteradores, soporte SOAP...). Proporciona independencia de Base de Datos usando Pear::MDB2 más una capa de abstracción que trata otros aspectos como el manejo de fechas, números con decimales, control de transacciones, etc... Actualmente probado con PostgreSQL, Mysql y Oracle.

Incorpora un generador de formularios que permite crear de forma rápida y sencilla mantenimientos funcionales (búsqueda, alta, baja y modificación). Para ello, únicamente requiere de la especificación de la conexión, la tabla de la BBDD y la forma de visualización (patrón de interfaz).

Funcionalidades:

- Alta productividad.
- Estandarización de los desarrollos.
- Simplifica el entorno de trabajo Web (no necesitamos conocer HTML o Javascript).
- Uniformidad en el aspecto y usabilidad de las aplicaciones.
- Separación de la lógica de negocio y de la presentación.
- Independencia del gestor de base de datos empleado
- Control de acceso por usuarios y grupos.
- Componentes complejos: Ventanas de selección, Mensajes modales, listas enlazadas,...
- Patrones de interfaz: tabular, registro, maestro detalle, maestro n-detalles, árbol.
- Integración con otros módulos.
- Autenticación de usuarios y control de permisos.
- Listados PDF, ODT, CSV, ...

En la página web del proyecto está toda la documentación y tutoriales necesarios para el aprendizaje y la comprensión de la manera de funcionamiento del framework, además de ejemplos completos en los que se detallan todos los pasos para construir aplicaciones web completas y robustas con un esfuerzo menor, ya que el framework se encarga de realizar gran parte del trabajo tanto para la generación de las distintas pantallas de la aplicación como para la implementación de varias funcionalidades comunes a nivel de negocio, como por ejemplo el código para el mantenimiento de la base de datos (acciones CRUD), y también la gestión de eventos y estímulos generados por los componentes gráficos de las diferentes pantallas, etc...

2.2. Tecnologías utilizadas

Aquí hacemos una pequeña introducción a las herramientas que hemos elegido en nuestro proyecto para la elaboración y el lanzamiento de los ficheros de tests.

Los pasos concretos de instalación y ejecución de cada una de ellas están detallados en el anexo: Instalación y manejo de PHPUnit, Selenium y Hudson.

2.2.1. **PHPUnit**

PHPUnit[4] es una librería o marco de trabajo que puede ser usada en la fase de pruebas de aplicaciones PHP. Está hecha para ejecutar pruebas y analizar resultados de manera sencilla. PHPUnit es una migración del popular JUnit utilizado en desarrollos Java, integra casos de prueba básicos, pruebas de bases de datos, cálculo de métricas de software, documentación ágil, entre muchas otras características. Algunas de éstas son:

- Marco de trabajo para las pruebas unitarias específicas a PHP.
- Se encuentra bajo licencia BSD.
- Forma parte del grupo de marcos de trabajo de xUnit
- Almacena los resultados en una base de datos de pruebas.
- Se integra con varias aplicaciones de pruebas.
- Facilita la creación de pequeños scripts que ayudan a probar las aplicaciones y analizar los resultados.
- Fácil manejo y comunicación con Selenium (que explicamos a continuación).

2.2.2. **Selenium**

Selenium[5] (o SeleniumHQ) es, en pocas palabras, un juego de herramientas que permite probar sistemas web de manera automatizada. Las pruebas que realiza son como las que haría cualquier usuario desde un navegador, con la ventaja de que las realiza de forma automática, y nos ahorra el trabajo repetitivo de probar una y otra vez lo mismo “a mano”.

Este conjunto de herramientas permiten desarrollar scripts para pruebas de aplicaciones Web en diversos lenguajes como Java, Ruby, Python, Perl, .Net o PHP. Es un producto Open Source que está permanentemente en evolución.

Existen 3 tipos de Herramientas: Selenium IDE, Selenium RC, y Selenium Grids.

Selenium IDE

Es un plugin de Firefox que pertenece a SeleniumHQ, permite realizar juegos de pruebas sobre aplicaciones web. Para ello realiza la grabación de la acción seleccionada (navegación por una página) en un "script", el cual se puede editar y parametrizar para adaptarse a los diferentes casos.

El principal objetivo de este plugin es crear pruebas funcionales, aunque no se puede pasar por alto que este tipo de herramientas permiten automatizar tareas que requieren un cierto "procesamiento" mental básico:

- Rellenar formularios (autenticación o cualquier otro tipo)
- Navegación web
- Acciones de gestión (CRUD de comentarios blog / correos / noticias / etc...)
- ...

Esta herramienta permite al desarrollador web ahorrarse mucho esfuerzo cada vez que se resuelve alguna incidencia o se genera una versión nueva ya que permite automatizar la realización de las pruebas, ya sean o bien pruebas específicas (una acción en particular) o bien juegos de pruebas (un conjunto de acciones).

Características:

- Facilidad de registro y ejecución de los test.
- Referencia a objetos DOM en base al ID, nombre o a través de XPath.
- Autocompletado para todos los comandos.

- Las acciones pueden ser ejecutadas paso a paso.
- Herramientas de depuración y puntos de ruptura (breakpoints).
- Los test pueden ser almacenados como HTML y scripts Ruby, entre otros formatos.
- Soporte para Selenium user-extensions.js (definir métodos javascript propios que se pueden llamar desde los tests).
- Ejecución en varios navegadores.

Selenium RC

Es una biblioteca y un servidor escrito en lenguaje Java que permite ejecutar scripts de forma local o remota a través de comandos.

Esta solución está pensada para permitir la ejecución de tests de Selenium bajo un sistema de integración continua.

Los tests se integran dentro del código fuente de una plataforma tecnológica concreta como Java o PHP. El código fuente de estos tests puede ser obtenido mediante Selenium IDE.

Los tests se ejecutan en una única máquina que puede ser la misma máquina que en la que se generan o una máquina distinta. Para ello, la máquina que ejecuta los tests tiene un control remoto que permite la comunicación entre el proceso que lanza los tests y el que los ejecuta. La máquina en la que se ejecutan los tests puede utilizar uno o varios navegadores para dar soporte a la navegación.

2.2.3. Hudson

Hudson[6] es una herramienta de integración continua escrita en Java. Trabaja con herramientas de control de versiones como CVS, Subversion, ... y puede ejecutar proyectos basados en Apache Ant y Apache Maven, así como también shell scripts o scripts bat de Windows.

Tiene una interfaz muy sencilla de utilizar. A diferencia de otros servidores de integración, tanto la creación como la configuración de los jobs se hacen íntegramente desde la interfaz web.

Es una herramienta que automatiza la construcción de proyectos en lo que denomina jobs, así pues, un mismo proyecto software puede tener diferentes jobs. Por ejemplo, tener configurado un job que se ejecute cada vez que se haga un cambio en el repositorio y que lo único que compruebe es que el proyecto compila y que se pasan las pruebas unitarias, y tener otro job programado para ejecutarse tres veces al día para correr pruebas de aceptación o de integración más pesados.

Por estas características y ventajas de Hudson mencionadas anteriormente, lo utilizaremos para lanzar la ejecución de los tests construidos en las diferentes maquinas (básicamente pc's linux y windows), y con distintos navegadores según las necesidades, pudiendo configurar también parámetros como la periodicidad de las ejecuciones, e-mails para informes de errores en el caso de fallos, etc...

En noviembre de 2010 la comunidad que desarrolla el proyecto se ha dividido y ha creado un nuevo proyecto llamado Jenkins[7], por lo que la funcionalidad es similar y se puede usar de igual forma.

2.3. Metodología de Desarrollo

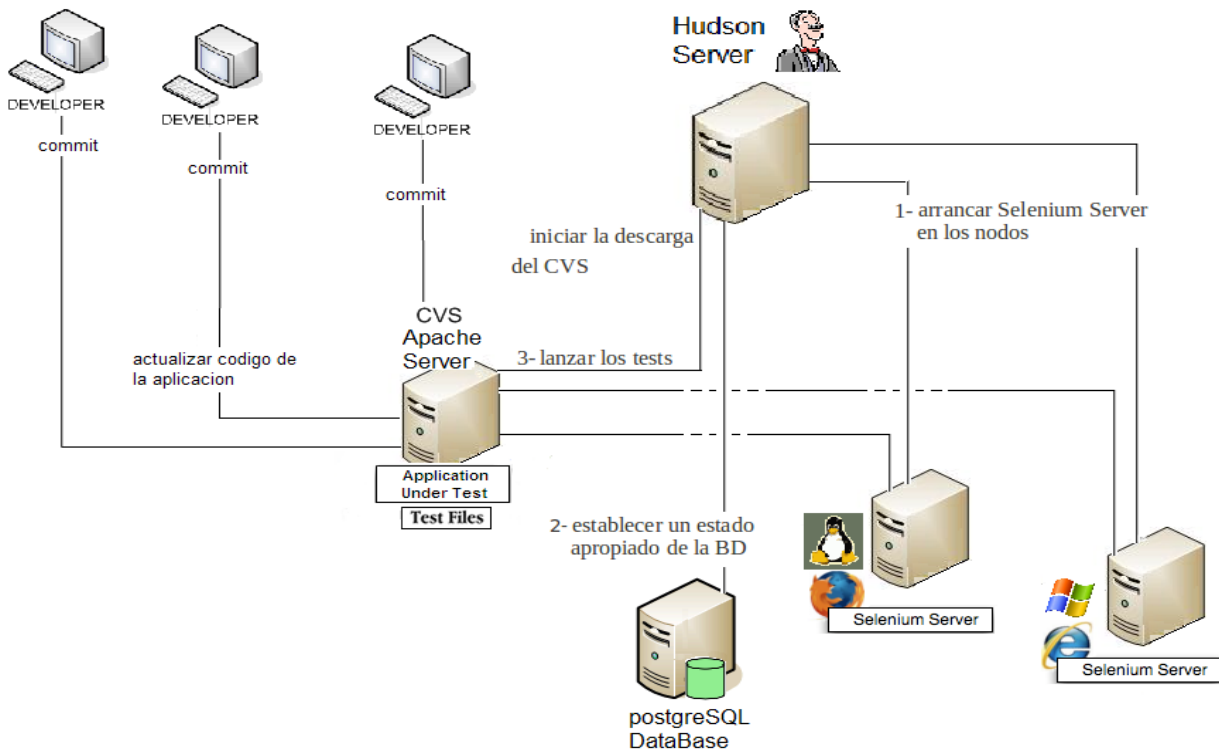
Una vez seleccionadas las herramientas que vamos a utilizar (PHPUnit, Selenium y Hudson), las tareas que iremos abordando a lo largo de las siguientes secciones son:

- 1) En primer lugar proponer una arquitectura global del sistema en la que figurarán los recursos que necesitamos utilizar y la distribución de las herramientas utilizadas e interacción entre ellas.
- 2) Escoger un repositorio para almacenar el código fuente, donde se lleve el historial de todos los cambios del código y de donde sea posible obtener la última versión del proyecto.

- 3) Explicar la aplicación de personal de la CIT, y analizar los diferentes casos de uso que necesitamos testear.
- 4) Elaborar para cada caso de uso los ficheros de tests necesarios, explicando como generarlos y modificarlos para que cumplan con su función. Cada uno de esos ficheros contendrá una serie de acciones (pruebas unitarias) ordenadas apropiadamente para que el caso de uso sea ejecutado correctamente.
- 5) Explicar la preparación del entorno (arrancando las herramientas necesarias en las diferentes maquinas), y crear y configurar las tareas en Hudson indicando los comandos que queremos lanzar, y los instantes en los que serán lanzados estos comandos.
- 6) Analizar los resultados obtenidos, y tratarlos de la manera adecuada (registrarlos, mandar informes y resúmenes por correo a los responsables, etc...).

3. Arquitectura y componentes del sistema

La arquitectura que vamos a utilizar, y la interconexión entre los diferentes componentes la representamos en la siguiente figura. Después comentaremos los pasos a seguir para llevar a cabo uno o varios lanzamientos de los tests con el fin de comprobar el código y la funcionalidad de la aplicación en cuestión:



- Como podemos destacar de la figura anterior, los componentes que necesitaremos son:
- Una máquina que soporta el servidor de Apache que ejecuta la aplicación, y el repositorio de código CVS en el que tenemos la última versión actualizada del código.
- El servidor de la base de datos PostgreSQL. Es posible que se opte por soportar este servidor por la misma máquina de Apache, pero para no incrementar la carga de trabajo y ralentizar el proceso, hemos representado esta máquina como un componente independiente. En todo caso, la decisión de incluir esos servidores en una sola máquina o dejarlos en máquinas separadas se deja a criterio del administrador responsable.
- Un conjunto de máquinas terminales que contengan la combinación deseada de sistemas operativos y navegadores con las que probar la aplicación. En nuestro caso tenemos dos ordenadores, uno con S.O Windows y Internet Explorer como navegador, y el otro con Linux y Mozilla Firefox.
- Cabe destacar que Selenium Server es básicamente un jar de Java que no consume muchos recursos del sistema, y es posible utilizar cualquier ordenador (dedicado o no) para soportarlo.
- Un servidor de Hudson, que es el responsable de iniciar todos los procesos en los momentos y periodicidad adecuados. Actualmente en la CIT ya está implantado, configurado y mantenido el servidor de Hudson.

Los pasos que se realizan son los siguientes:

- 1) El servidor de Hudson, ejecuta primero la tarea que inicia el arranque de las máquinas que soportan las diferentes estancias de “Selenium Server”. Una alternativa sería que se arranquen automáticamente (estableciendo los responsables de esas máquinas la configuración adecuada en sus sistemas operativos para este fin).

```
java -jar selenium/selenium-remote-control-1.0.3/selenium-server-1.0.3/selenium-server.jar -port 4445
```

la salida sería algo parecido a:

```
14:06:23.799 INFO - Java: Sun Microsystems Inc. 20.1-b02
14:06:23.838 INFO - OS: Linux 2.6.32-22-generic i386
14:06:24.029 INFO - v2.0 [a2], with Core v2.0 [a2]
14:06:24.792 INFO - RemoteWebDriver instances should connect to: http://192.168.43.1:4445/wd/hub
14:06:24.796 INFO - Version Jetty/5.1.x
14:06:24.799 INFO - Started HttpContext[/selenium-server/driver,/selenium-server/driver]
14:06:24.803 INFO - Started HttpContext[/selenium-server,/selenium-server]
14:06:24.804 INFO - Started HttpContext[/,/]
14:06:24.941 INFO - Started org.openqa.jetty.jetty.servlet.ServletHandler@14fe5c
14:06:24.941 INFO - Started HttpContext[/wd,/wd]
14:06:24.949 INFO - Started SocketListener on 0.0.0.0:4445
14:06:24.949 INFO - Started org.openqa.jetty.jetty.Server@14ed9ff
```

- 2) Lanzar un proceso en el servidor de la base de datos para establecer un entorno o un estado de la base de datos inicial adecuado para llevar a cabo la funcionalidad que se desea testear en este momento. Esto va a ser básicamente un script de SQL que realice las comprobaciones y modificaciones oportunas de los datos.

- 3) Por último, lanzar en Hudson la tarea responsable de ejecutar los tests de pruebas unitarias en el servidor de apache utilizando PHPUnit. Los tests normalmente residen en la carpeta “personal/tests” dentro de la aplicación, y los comandos que ejecuta esta tarea los veremos más adelante en la sección de Pruebas.

4. Análisis

4.1. La aplicación de personal de la CIT

La aplicación de personal de la CIT es un programa software que permite al servicio de Personal gestionar todos los permisos y licencias que solicitan los funcionarios de la Conselleria.

El objetivo de la aplicación "Personal: Solicitud de permisos y licencias" es que cada usuario (funcionarios, laborales, eventuales e internos) pueda, desde la INTRANET, dar de alta sus solicitudes de permisos y licencias y su responsable las validará. A continuación, pasarán al Servicio de personal todas las solicitudes para ser validadas por el tramitador, el revisor y el responsable de personal, antes de pasar a la firma del Subsecretario. Tan sólo se grabarán en la base de datos de personal aquellas solicitudes que hayan sido autorizadas por el Subsecretario.

Además, el usuario podrá anular y borrar sus solicitudes, dependiendo del estado en que se encuentren, es decir, si su solicitud todavía no ha sido validada por su responsable, sólo podrá borrarla. En caso contrario, deberá anularla.

La aplicación tiene alrededor de 600 usuarios potenciales y su puesta en funcionamiento fue hace aproximadamente dos años, y está en constante remodelación debido a los sucesivos cambios de requerimientos (generalmente, cambios en la legislación).

4.2. Casos de Uso

Tenemos dos casos de uso de la aplicación del Personal que requieren ser testeados para comprobar el correcto funcionamiento del código.

4.2.1. Mantenimiento de clases de permisos

Es el caso más sencillo, y nos servirá de ayuda para familiarizarnos con el entorno de gvHIDRA y también con el uso de la herramienta Selenium Ide con lo cual, conviene echar un vistazo al anexo de Instalación de esta herramienta entre otras. Una vez finalizada la generación del test se procederá a su modificación según los patrones explicados en el apéndice.

Este caso consiste en añadir, modificar y borrar una clase de permisos como se detalla en la siguiente tabla:

Nº	Contexto	Descripción	Pasos a seguir para realizar la prueba	Resultado esperado
1	Alta de clases de permisos		- Seleccionar opción "Mantenimientos" - Seleccionar opción "Clases de permisos" - Pulsar en botón de insertar registros en esquina superior derecha	
2		Se da de alta una nueva clase de permiso correctamente	- Introducir los valores obligatorios (clase y descripción) para una clase de permiso. - Pulsar en botón "Guardar"	Se inserta el permiso correctamente.
3		Se da de alta una nueva clase de permiso incorrectamente	- No introducir alguno de los valores obligatorios (clase o descripción) para una clase de permiso. - Pulsar en botón "Guardar"	Aparece el mensaje: "Faltan campos por rellenar" y no se inserta el permiso
	Consulta y/o		- Seleccionar opción "Mantenimientos"	

Nº	Contexto	Descripción	Pasos a seguir para realizar la prueba	Resultado esperado
	modificación de clases de permisos		- Seleccionar opción “Clases de permisos” - Seleccionar una clase de permiso marcando el checkbox de la izquierda - Pulsar en el botón de modificar registros de la esquina superior derecha.	
4		Se consulta el detalle de una clase de permiso	- Tras visualizar los datos, y sin haber modificado los datos se pulsa en cancelar	Se muestran los datos del permiso
5		Se modifican los datos de una clase de permiso y se graban	- Se modifican los datos - Se pulsa en el botón “Guardar”	Los datos modificados de la clase de permiso se almacenan en el sistema
6		Se modifican los datos de una clase de permiso pero no se graban	- Se modifican los datos - Se pulsa en el botón “Cancelar”	Los datos modificados de la clase de permiso no se almacenan en el sistema
7	Borrado de clases de permisos	Se borra una clase de permiso	- Seleccionar opción “Mantenimientos” - Seleccionar opción “Clases de permisos” - Seleccionar una clase de permiso marcando el checkbox de la izquierda - Pulsar en el botón de borrar registros de la esquina superior derecha - Pulsar en el botón de “Guardar”	El registro se borra del sistema

4.2.2. Solicitud de permisos y licencias

Este es un caso de uso más complejo que el anterior y bastante frecuente. En la siguiente tabla se describen los pasos que debemos seguir en orden para realizarlo, y además para ciertas pruebas, se especifican las otras precedentes que debían haber sido ejecutadas para que la prueba en cuestión pueda efectuarse correctamente y producir el resultado esperado. Por motivo de permisos y perfil de usuario las pruebas A5 y A14 no se realizaran.

Nº	Contexto	Descripción	Ej. Datos de entrada	Resultado esperado
A1	Alta de solicitud de PRUEBAS SELECTIVAS Y EXÁMENES con error	Se da de alta una solicitud de PRUEBAS SELECTIVAS Y EXÁMENES de más de 1 día, por lo que se genera un mensaje de error	04/11/2010 – 05/11/2010	La solicitud no se puede dar de alta porque aparece un mensaje de error indicando “Ha solicitado 2 días de PRUEBAS SELECTIVAS Y EXÁMENES. No se pueden solicitar más de 1 días hábiles.”
A2	Alta de solicitud de PRUEBAS SELECTIVAS Y EXÁMENES	Se da de alta una solicitud de PRUEBAS SELECTIVAS Y EXÁMENES que requiere justificante y se pulsa en el botón “Guardar e imprimir”	04/11/2010 – 04/11/2010	La solicitud se graba correctamente en el sistema tras pulsar en los botones "Continuar" y “Guardar e imprimir”. Se muestra un pdf con la solicitud para imprimir y adjuntar al justificante y un mensaje indicando que la solicitud se ha guardado: "La solicitud de FULANITO, con DNI ????????, se ha generado con el nº ?”.
A3	Alta de solicitud de VACACIONES con error	Se intenta dar de alta una solicitud de VACACIONES superando el máximo permitido.	02/08/2010 - 03/09/2010	La solicitud no se puede dar de alta porque aparece un mensaje de error indicando “Superado el máximo de días de VACACIONES. Ha intentado solicitar 25 días en el periodo 2010, pero sólo dispone de un total de 22 días de VACACIONES.”
A4	Alta de solicitud de VACACIONES	Se da de alta una solicitud de VACACIONES de 10 días hábiles	27/09/2010 – 08/10/2010	La solicitud se graba correctamente en el sistema tras pulsar en los botones “Continuar” y “Guardar” con el siguiente mensaje: “La solicitud de FULANITO, con DNI ????????, se ha generado con el nº ?”.

N°	Contexto	Descripción	Ej. Datos de entrada	Resultado esperado
A4.1	Alta de solicitud de VACACIONES	Se da de alta una solicitud de VACACIONES de 10 días hábiles	18/10/2010 – 29/10/2010	La solicitud se graba correctamente en el sistema tras pulsar en los botones “Continuar” y “Guardar” con el siguiente mensaje: “La solicitud de FULANITO, con DNI ????????, se ha generado con el nº ?”.
A4.2	Alta de solicitud de VACACIONES con aviso	Se da de alta una solicitud que genera un mensaje de aviso (se piden menos de 7 días naturales, incluyendo sábados y domingos, o bien, menos de 5 días naturales sin incluir sábados y domingos) en la pantalla de información de la solicitud pedida.	17/11/2010 - 18/11/2010	La solicitud se puede dar de alta tras pulsar en el botón “Guardar”. Los avisos que se muestran en la pantalla, antes de pulsar dicho botón es el siguiente: “Deberá solicitar a continuación los días adicionales de vacaciones para que el total solicitado no sea inferior a 7 días naturales consecutivos.” y “Deberá solicitar los días adicionales de vacaciones en fechas consecutivas a las vacaciones, salvo que su solicitud, de días adicionales de vacaciones, tenga una duración de al menos 7 días naturales. “
A5	Alta de solicitud de DÍAS ADICIONALES VACACIONES con error	El usuario solicita DÍAS ADICIONALES DE VACACIONES cuando no tiene ninguno asignado (SELECT diasadi_act FROM TPER_PERSONAS WHERE nregpgv = '?????????'; Resultado diasadi_act = 0)	19/11/2010 – 19/11/2010	La solicitud no se puede dar de alta porque aparece un mensaje de error: “No le corresponden DIAS ADICIONALES VACACIONES, por lo que no puede solicitarlos.”
A6	Alta de solicitud de DÍAS ADICIONALES VACACIONES	El usuario solicita DÍAS ADICIONALES DE VACACIONES cuando sí que tiene días adicionales asignados y no ha solicitado todas las VACACIONES (SELECT diasadi_act FROM TPER_PERSONAS WHERE nregpgv = '?????????'; Resultado diasadi_act = 2)	02/11/2010 – 03/11/2010	La solicitud se graba correctamente en el sistema tras pulsar en los botones “Continuar” y “Guardar” con el siguiente mensaje: “La solicitud de FULANITO, con DNI ????????, se ha generado con el nº ?”.
A6.1	Alta de solicitud de DÍAS ADICIONALES VACACIONES con error	Se solicitan más DÍAS ADICIONALES VACACIONES de los que le corresponden al interesado cuando sí que tiene días adicionales asignados (SELECT diasadi_act FROM TPER_PERSONAS WHERE nregpgv = '?????????'; Resultado diasadi_act = 2)	05/11/2010 – 05/11/2010	La solicitud no se puede dar de alta y se generan dos mensajes de error de DIAS ADICIONALES VACACIONES: “Ha solicitado 3 DIAS ADICIONALES VACACIONES, y no puede solicitar más de 2 días hábiles. No se pueden solicitar permisos de DIAS ADICIONALES VACACIONES de menos de 7 días naturales”
A7	Clases de permisos disponibles según el perfil	Un usuario con perfil “Tramitador” entra en la ventana de alta de solicitudes y selecciona la clase de permiso a solicitar.	MATERNIDAD BIOLÓGICA	Las clases de permisos que también son posibles para el tramitador son: - FLEXIBILIDAD DE PERMANENCIA OBLIGATORIA - IT - MATERNIDAD BIOLÓGICA - REDUCC. HORARIO DE 9 A 14 - REDUCC. ENFERMEDAD MUY GRAVE - REDUCC. 1H. DIARIA POR ENFERMEDAD - REDUCC. DIARIA GUARDA LEGAL - REDUCC. 1H. DIARIA POR MINUSVALÍA - REDUCC. JORNADA DIARIA % - REDUCC. POR VIOLENCIA SOBRE MUJER
A8	Alta de solicitud de VACACIONES con control de fecha fin vacía	Un usuario entra en la ventana de alta de solicitudes, selecciona el motivo VACACIONES y deja la fecha fin vacía. Se pulsa en el botón “Continuar”.	a) 22/11/2010 – nulo b) 16/11/2010 – nulo	La solicitud de VACACIONES no se puede realizar porque se muestra uno de los siguientes mensajes de error: a) “Dado el motivo de la solicitud indicado, la fecha de finalización es obligatoria.”, porque no ha solicitado ningún otro permiso posteriormente a dicha solicitud b) “No puede solicitar el permiso porque ya ha realizado la solicitud VACACIONES entre el 17/11/2010 y el 18/11/2010 ”, porque ha solicitado otro permiso posteriormente a dicha solicitud.

N°	Contexto	Descripción	Ej. Datos de entrada	Resultado esperado
A9	Alta de solicitud de LIC. PERFECCIONAMIENTO PROFESIONAL con control de solapamiento no permitido	Un usuario entra en la ventana de alta de solicitudes y solicita una LIC. PERFECCIONAMIENTO PROFESIONAL en un periodo de tiempo en el que algún día coincide con los de una solicitud de "VACACIONES"	12/11/2010 – 17/11/2010	No se permite dar de alta la nueva solicitud de LIC. PERFECCIONAMIENTO PROFESIONAL porque se solapa con otra ya solicitada, mostrando el siguiente mensaje: "No puede solicitar el permiso porque ya ha realizado la solicitud VACACIONES entre el 17/11/2010 y el 18/11/2010".
A10	Alta de solicitud REDUCC. JORNADA DIARIA %	Un usuario entra en la ventana de alta de solicitudes y selecciona el motivo "REDUCC. JORNADA DIARIA %"	22/11/2010 – 30/11/2010; Tiempo reducción: 1:30	El usuario deberá indicar el tiempo de reducción, calculándose automáticamente el porcentaje de reducción (19) en el campo no editable "Porcentaje reducción". Además, se indicará automáticamente el tiempo de reducción en el campo "Observaciones" (Reducción de 1 horas y 30 minutos.) tras abandonar el campo "Tiempo reducción". A continuación, pulsará el botón "Continuar" y "Guardar e imprimir", mostrando un pdf y el mensaje: "Se ha generado el permiso solicitado por FULANITO, con DNI ????????. Ha sido registrada con éxito la incidencia, para la aplicación de nóminas, del permiso RJP entre el 22/11/2010 y el 30/11/2010."
A11	Alta de solicitud de LIC. ASUNTOS PROPIOS con control de solapamiento permitido	Un usuario entra en la ventana de alta de solicitudes y solicita una LIC. ASUNTOS PROPIOS en un periodo de tiempo en el que algún día coincide con los de otra solicitud de REDUCC. JORNADA DIARIA %	25/11/2010 – 02/12/2010	Tras pulsar en los botones de "Continuar" y "Grabar", la solicitud se graba en el sistema correctamente sacando el siguiente mensaje: "La solicitud de FULANITO, con DNI ????????, se ha generado con el nº ?".
A12	Campos visibles al intentar dar de alta una solicitud de IT	Un usuario entra en la ventana de alta de solicitudes y selecciona el motivo IT.	Generar incidencia: checkbox Causa: lista desplegable de valores	El campo "Interesado" se queda vacío. Se muestran en pantalla los siguientes campos: - "Generar incidencia" para poder generar o no incidencia para la aplicación de nóminas, - "Causa" para indicar la causa de la IT. Los posibles valores son: Enfermedad común – Seguridad Social Enfermedad común – Muface, Accidente trabajo/Enf. profesional – Seguridad Social Accidente trabajo/Enf. profesional – Muface Accidente no laboral – Seguridad social Accidente no laboral – Muface
A13	Campo visible al intentar dar de alta una solicitud de REDUCC. 1H. DIARIA POR MINUSVALÍA	Un usuario entra en la ventana de alta de solicitudes y selecciona el motivo REDUCC. 1H. DIARIA POR MINUSVALÍA	Tipo de minusvalía: lista desplegable de valores	El campo "Tipo de minusvalía" se muestra en pantalla con los siguientes valores en la lista: Padres, Hijos, Propios y Otros.
A14	Envío de correo al responsable	Un usuario con el perfil "Responsable" (P_RESPONSA), siendo el responsable del usuario que ha dado de alta la solicitud de VACACIONES, recibirá un correo con la información de dicha solicitud		El responsable del usuario que ha dado de alta la solicitud de VACACIONES, recibirá un correo con la información del alta.
A15	Alta de solicitud de LIC. SIN RETRIB. POR INTERÉS PARTICULAR de más de 30 días	Se da de alta una solicitud de LIC. SIN RETRIB. POR INTERÉS PARTICULAR de 2 meses.	01/09/2009 – 31/10/2009	La solicitud se graba correctamente en el sistema tras pulsar en los botones "Continuar" y "Guardar" con el siguiente mensaje: "La solicitud de FULANITO, con DNI ????????, se ha generado con el nº ?".
A16	Alta de solicitud de	Se da de alta una solicitud de LIC.	01/04/2010 –	La solicitud se graba correctamente en el sistema tras

Nº	Contexto	Descripción	Ej. Datos de entrada	Resultado esperado
	LIC. SIN RETRIB. POR INTERÉS PARTICULAR de más de 30 días	SIN RETRIB. POR INTERÉS PARTICULAR de 3 meses.	31/05/2010	pulsar en los botones “Continuar” y “Guardar” con el siguiente mensaje: “La solicitud de FULANITO, con DNI ????????, se ha generado con el nº?”.
A17	Alta de solicitud de LIC. SIN RETRIB. POR INTERÉS PARTICULAR de más de 30 días con error	Se da de alta una solicitud de LIC. SIN RETRIB. POR INTERÉS PARTICULAR de 2 meses, habiendo solicitado 5 meses en los 3 últimos años.	01/07/2010 – 31/08/2010	La solicitud no se puede realizar porque se muestra el siguiente mensaje de error: “Superado el máximo de meses naturales de LIC. SIN RETRIB. POR INTERÉS PARTICULAR: el total solicitado en los tres últimos años es de 7 meses, siendo el máximo 6 meses cada tres años.”
A18	Alta de solicitud de LIC. ASUNTOS PROPIOS con error	Se solicitan más de 6 días de LIC. ASUNTOS PROPIOS para el periodo 2011, en el que aún no se han definido el número de días de LIC. ASUNTOS PROPIOS disponibles para el municipio de ALICANTE (select CASE WHEN max(maxdias) IS NULL THEN 0 ELSE max(maxdias) END as diasMaximo from TPER_PERMISOSMAXDIASLO CAL where cclase= 'VV6' and periodo = 2011 and (cmun = '014' or ('03' = '46' and cmun='250') or ('03' = '12' and cmun='040') or ('03' = '03' and cmun='014')) and Cpro = '03' ; Resultado diasMaximo = 0)	03/01/2011 – 12/01/2011	La solicitud no se puede dar de alta y tras pulsar el botón “Continuar”, aparece el siguiente mensaje de error: “No se ha introducido el número de días de LIC. ASUNTOS PROPIOS del periodo 2011, para su municipio, por lo que el máximo permitido son 6 días.”.

5. Diseño de los Tests

5.1. El Concepto de las Pruebas Unitarias

Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código.

Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego, con las Pruebas de Integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

Razones para realizar pruebas unitarias:

- Asegura la calidad del código entregado. Es la mejor forma de detectar errores en el desarrollo de una forma temprana. No obstante, esto no asegura detectar todos los errores, por tanto las pruebas de integración y aceptación siguen siendo necesarias.
- Ayuda a definir los requerimientos y responsabilidades de cada método en cada clase probada.
- Constituye una buena forma de ejecutar pruebas de concepto. Cuando es necesario hacer pruebas de conceptos sin integrar usar pruebas unitarias se convierte en un método efectivo.
- Permite encontrar errores o bugs tempranamente en el desarrollo. Y está demostrado que mientras más temprano se corrijan los errores, menos costará corregirlos.

Características:

Automatizable: no debería requerirse una intervención manual. Esto es especialmente útil para integración continua.

Completas: deben cubrir la mayor cantidad de código.

Repetibles o Reutilizables: no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez.

Independientes: la ejecución de una prueba no debe afectar a la ejecución de otra.

Profesionales: las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc...

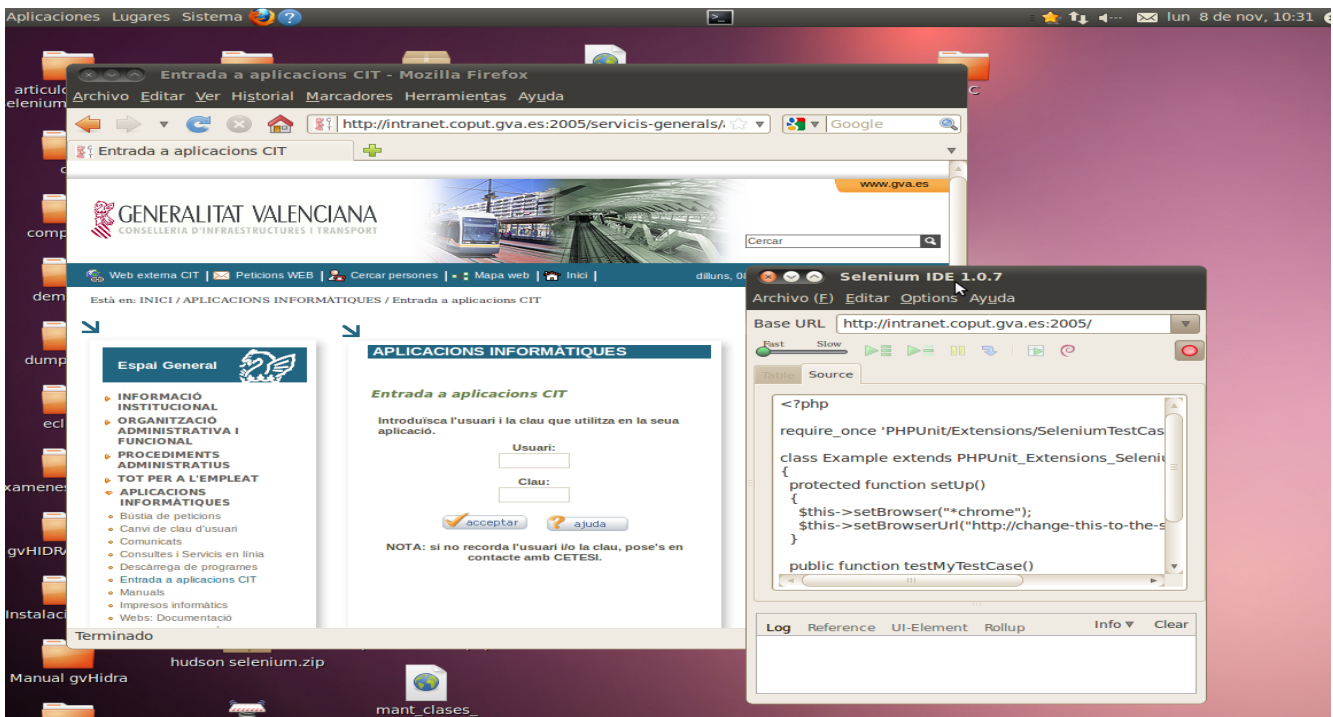
5.2. Generación de los Tests necesarios

5.2.1. Tests para el “Mantenimiento de clases de permisos”

Primero arrancamos el plugin de firefox “Selenium IDE” desde el menú de Herramientas de firefox y se procede a la grabación del test a partir de la pagina de autenticación del usuario de la aplicación. Vemos que hay un botón rojo, que debe estar pulsado para que se empiecen a grabar las acciones que iremos haciendo a partir de este momento.

Nota importante:

mientras navegamos por la aplicación procurar no pinchar con el ratón sobre ningún sitio del código en la ventanita de Selenium Ide, porque si lo hacemos, los siguientes pasos de navegación que haremos se intercalaran dentro del código que ya tenemos generado y no al final del mismo, lo cual hará que el test fallara.

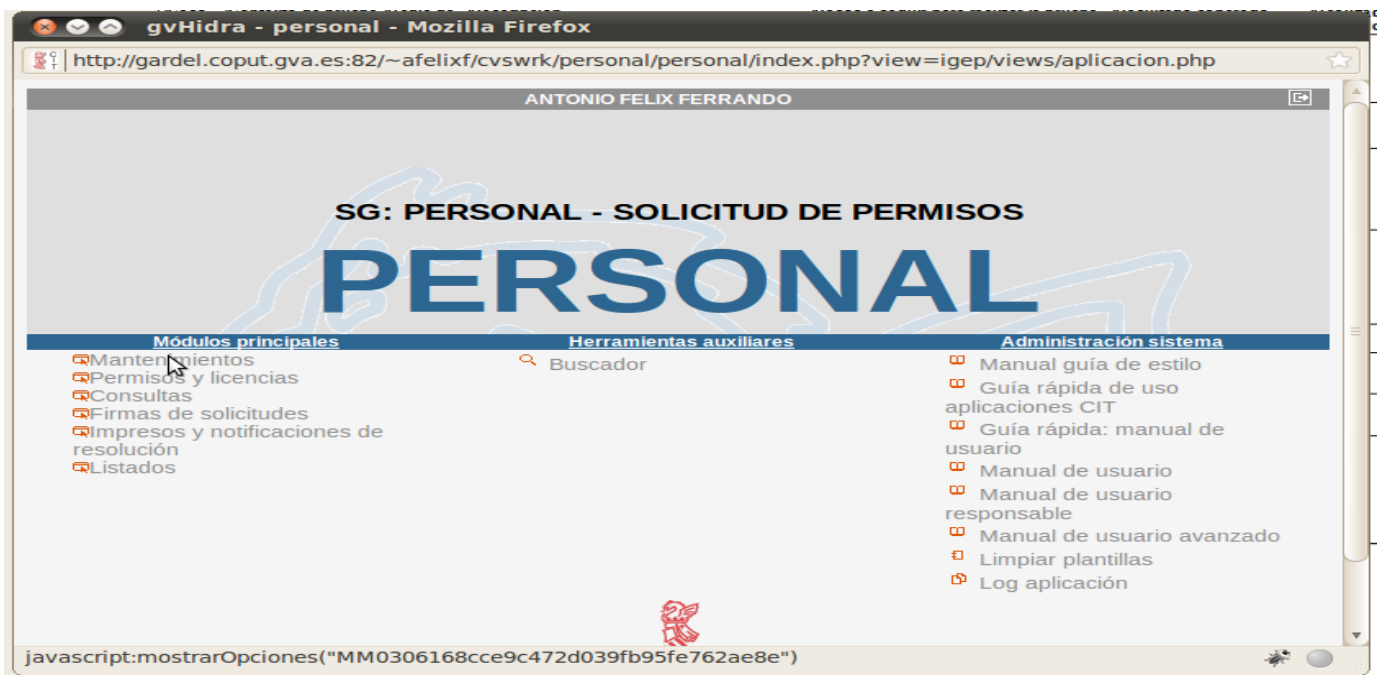


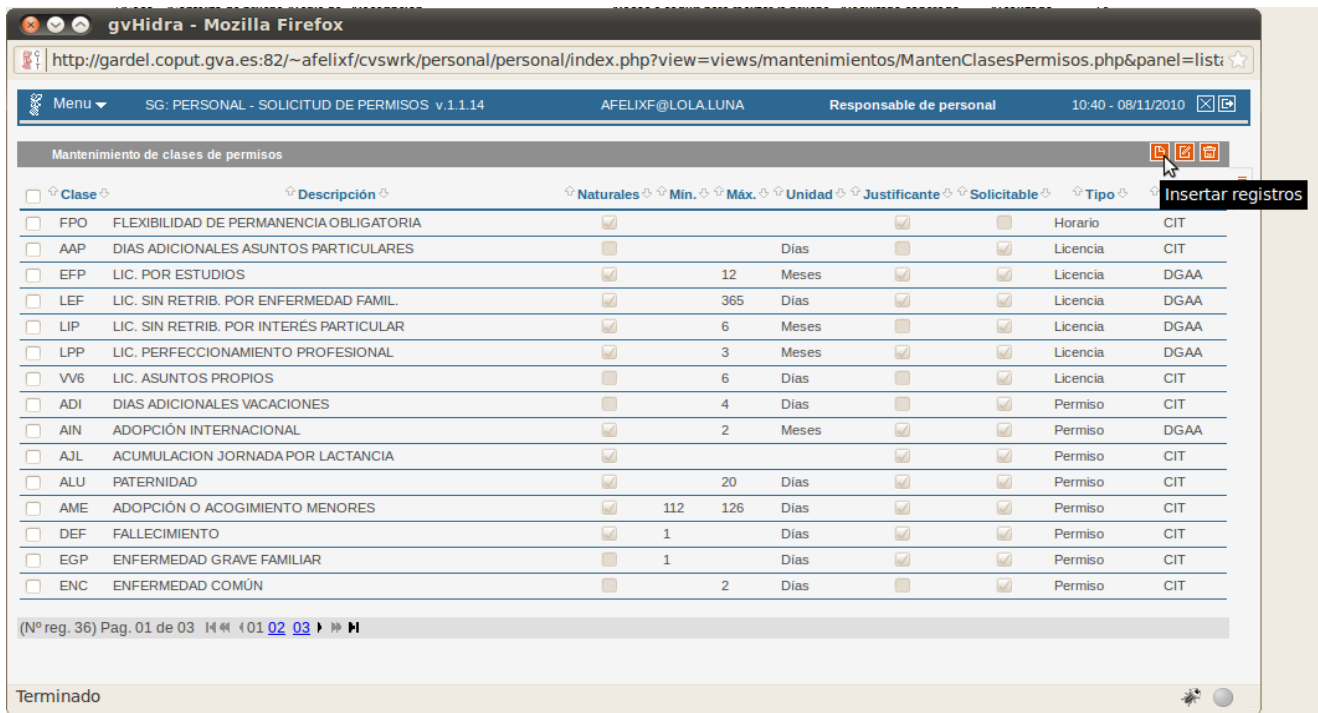
Minimizamos la ventana de Selenium Ide y seguimos utilizando la aplicación

Nota:

si es la primera vez que utilizamos Selenium Ide, por defecto las acciones que realizamos se generan en html y no en php como nos interesa, en este caso antes de comenzar con el test vamos al menú Options->Formate, y allí seleccionamos en lenguaje que nos interesa, y si no esta PHP(o PHPUnit) incluida, vamos al menú "Options->Options", y activamos allí en la pestaña "General" las casillas "activate developer tools "y "enable experimental features".

Prueba n°1:





el código generado hasta ahora (cambiar *user* y *password* por los datos correctos) es:

```

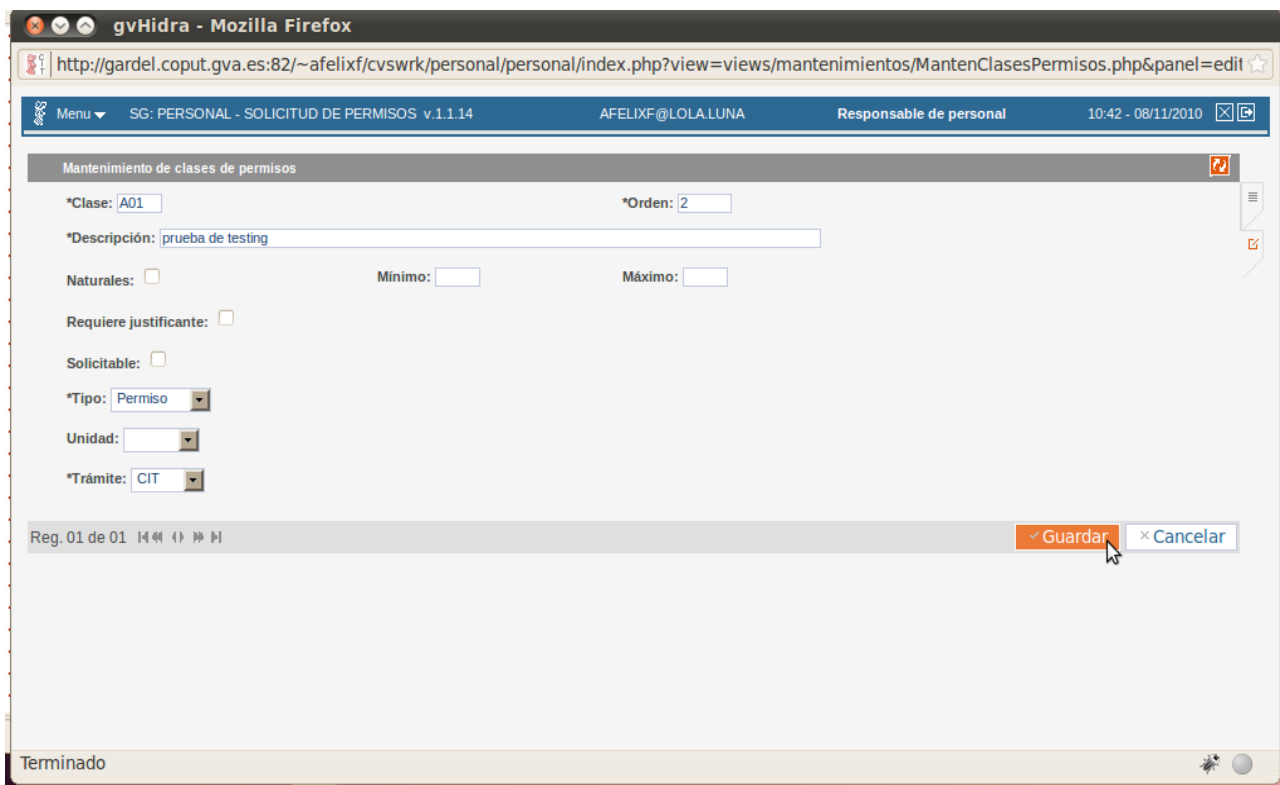
$this->open("/servicis-generals/aplicacions-informatiques/acces-personalitzat-a-aplicacions/");
$this->selectFrame("");
$this->selectFrame("contenido");
$this->type("as_usuario", "<user>");
$this->type("as_clave", "<password>");
$this->click("Aceptar");
$this->waitForPageToLoad("30000");
$this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
$this->selectFrame("relative=up");
$this->selectFrame("oculto");
$this->waitForPopUp("personal__W_PERDES", "30000");
$this->selectWindow("name=personal__W_PERDES");
$this->click("btnAceptar");
$this->click("link=Mantenimientos");
$this->click("link=Clases de permisos");

```

se abre el panel para introducir los datos del nuevo registro ,que serán en nuestro ejemplo :

clase : A01 , orden :2 , descripción : “prueba de testing” .

Le damos al botón guardar :



código generado para este paso :

```

$this->click("link=Clases de permisos");

$this->click("img_btllInsertar_lis");

$this->type("ins__cclase__FichaEdicion_0", "A01");

$this->type("ins__orden__FichaEdicion_0", "2");

$this->type("ins__dclase__FichaEdicion_0", "prueba de testing");

$this->click("bnedi_guardar");

```

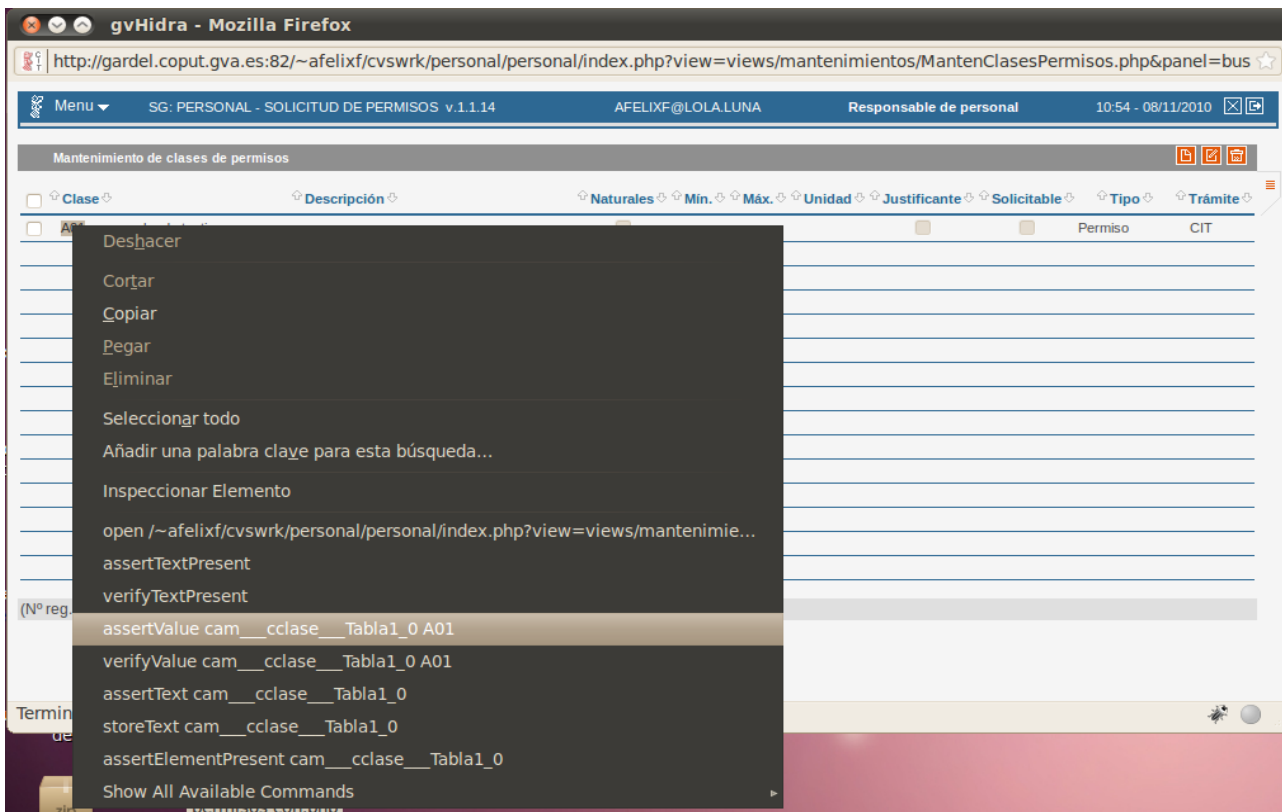
se nos abre una pagina que contiene una tabla con una única fila que corresponde al registro nuevo creado. Esta fila la utilizamos para hacer la comprobación (a nivel de interfaz gráfica) de que se inserto el registro correctamente, por ejemplo seleccionamos el texto del código de la clase “A01”, y como esta arrancado Selenium Ide, si hacemos clic con el botón derecho sobre la selección se nos muestran unas opciones extra (asserts, verify, etc...) en el menú contextual, de las cuales utilizamos las apropiadas para llevar acabo la comprobación que pretendemos hacer, concretamente en nuestro caso utilizamos el AssertEquals, tanto para la cadena de clase “A01”, como para la de descripción “prueba de testing”:

```

$this->assertEquals("A01", $this->getValue("cam__cclase__Tabla1_0"));

$this->assertEquals("prueba de testing", $this->getValue("cam__dclase__Tabla1_0"));

```



salimos de la ventana actual(un clic sobre el botón tooltip azul X en la esquina superior derecha).

Prueba nº2:

Accedemos a la tabla de permisos nuevamente (como hicimos al principio, menú mantenimiento > clases de permisos...).

```
$this->click("//img[@alt='X']");
$this->waitForPageToLoad("30000");
$this->click("link=Mantenimientos");
$this->click("link=Clases de permisos");
```

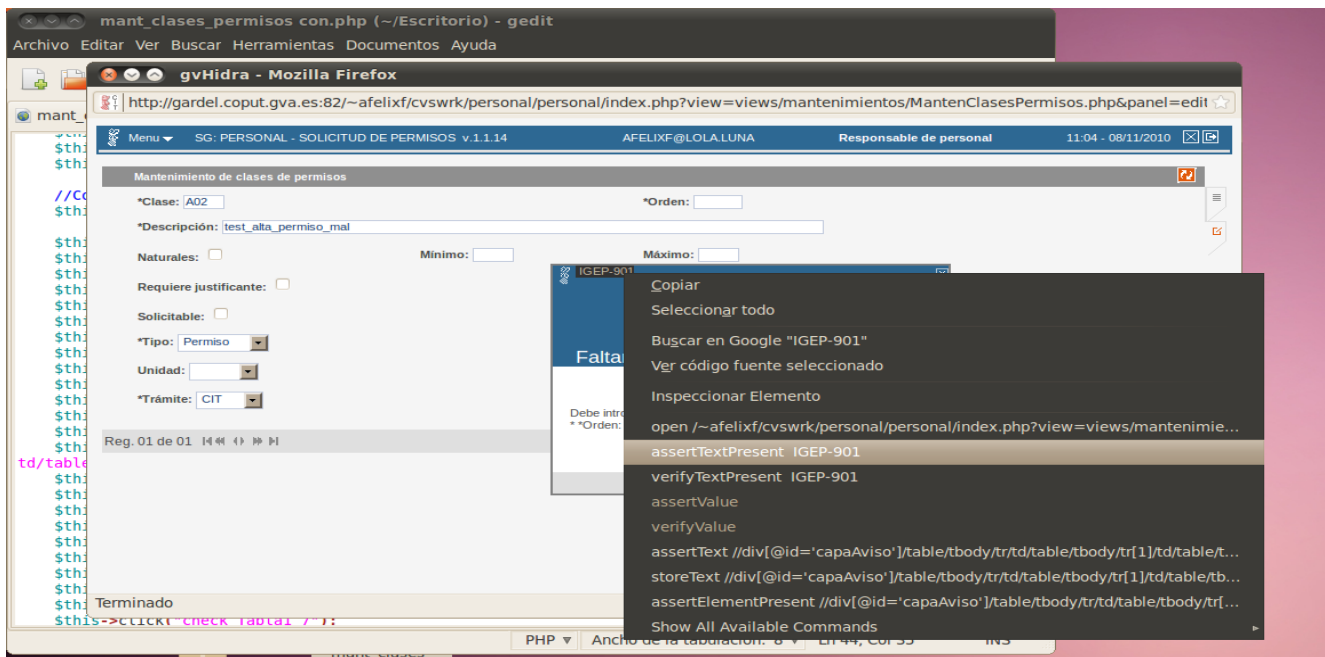
Ahora probamos insertar incorrectamente un nuevo registro :

- Introducir los valores obligatorios (clase y descripción) para una clase de permiso (no introducimos orden).
- Pulsar en botón “Guardar”

```
$this->type("ins__cclase__FichaEdicion_0", "A02");
$this->type("ins__dclase__FichaEdicion_0", "test_alta_permiso_mal");
$this->type("ins__orden__FichaEdicion_0", "");
$this->click("ins__dclase__FichaEdicion_0");
$this->click("bnedi_guardar");
```

y vemos que se nos sale una ventana de aviso (faltan campos obligatorios).

Utilizamos esta ventana para generar el código de comprobación dentro de nuestro test, por ejemplo comparando con una assert (como hicimos antes, seleccionar texto > botón derecho, etc...), que el código del aviso y su texto están presentes en la ventana realmente:



código generado:

```
$this->assertTrue($this->isTextPresent("IGEP-901"));
$this->assertTrue($this->isTextPresent("Faltan campos por rellenar"));
$this->click("btnAceptar");
```

le damos al botón cancelar para volver a la ventana de la tabla de permisos.

```
$this->click("bnedi_cancelar");
```

Prueba nº3

Procedemos ahora a realizar el paso de “consulta y modificación” del caso de uso:

- Seleccionar una clase de permiso marcando el checkbox de la izquierda
- Pulsar en el botón de modificar registros de la esquina superior derecha

Clase	Descripción	Naturales	Min.	Máx.	Unidad	Justificante	Solicitable	Tipo	Trá
<input type="checkbox"/>	FPO FLEXIBILIDAD DE PERMANENCIA OBLIGATORIA	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Horario	CIT
<input type="checkbox"/>	AAP DIAS ADICIONALES ASUNTOS PARTICULARES	<input checked="" type="checkbox"/>			Días	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Licencia	CIT
<input type="checkbox"/>	EFP LIC. POR ESTUDIOS	<input checked="" type="checkbox"/>	12		Meses	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Licencia	DGAA
<input type="checkbox"/>	LEF LIC. SIN RETRIB. POR ENFERMEDAD FAMIL.	<input checked="" type="checkbox"/>	365		Días	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Licencia	DGAA
<input type="checkbox"/>	LIP LIC. SIN RETRIB. POR INTERES PARTICULAR	<input checked="" type="checkbox"/>	6		Meses	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Licencia	DGAA
<input type="checkbox"/>	LPP LIC. PERFECCIONAMIENTO PROFESIONAL	<input checked="" type="checkbox"/>	3		Meses	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Licencia	DGAA
<input type="checkbox"/>	VV6 LIC. ASUNTOS PROPIOS	<input checked="" type="checkbox"/>	6		Días	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Licencia	CIT
<input checked="" type="checkbox"/>	A01 prueba de testing	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Permiso	CIT
<input type="checkbox"/>	ADI DIAS ADICIONALES VACACIONES	<input checked="" type="checkbox"/>	4		Días	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Permiso	CIT
<input type="checkbox"/>	AIN ADOPCION INTERNACIONAL	<input checked="" type="checkbox"/>	2		Meses	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Permiso	DGAA
<input type="checkbox"/>	AJL ACUMULACION JORNADA POR LACTANCIA	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Permiso	CIT
<input type="checkbox"/>	ALU PATERNIDAD	<input checked="" type="checkbox"/>	20		Días	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Permiso	CIT
<input type="checkbox"/>	AME ADOPCION O ACOGIMIENTO MENORES	<input checked="" type="checkbox"/>	112	126	Días	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Permiso	CIT
<input type="checkbox"/>	DEF FALLECIMIENTO	<input checked="" type="checkbox"/>	1		Días	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Permiso	CIT
<input type="checkbox"/>	EGP ENFERMEDAD GRAVE FAMILIAR	<input checked="" type="checkbox"/>	1		Días	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Permiso	CIT

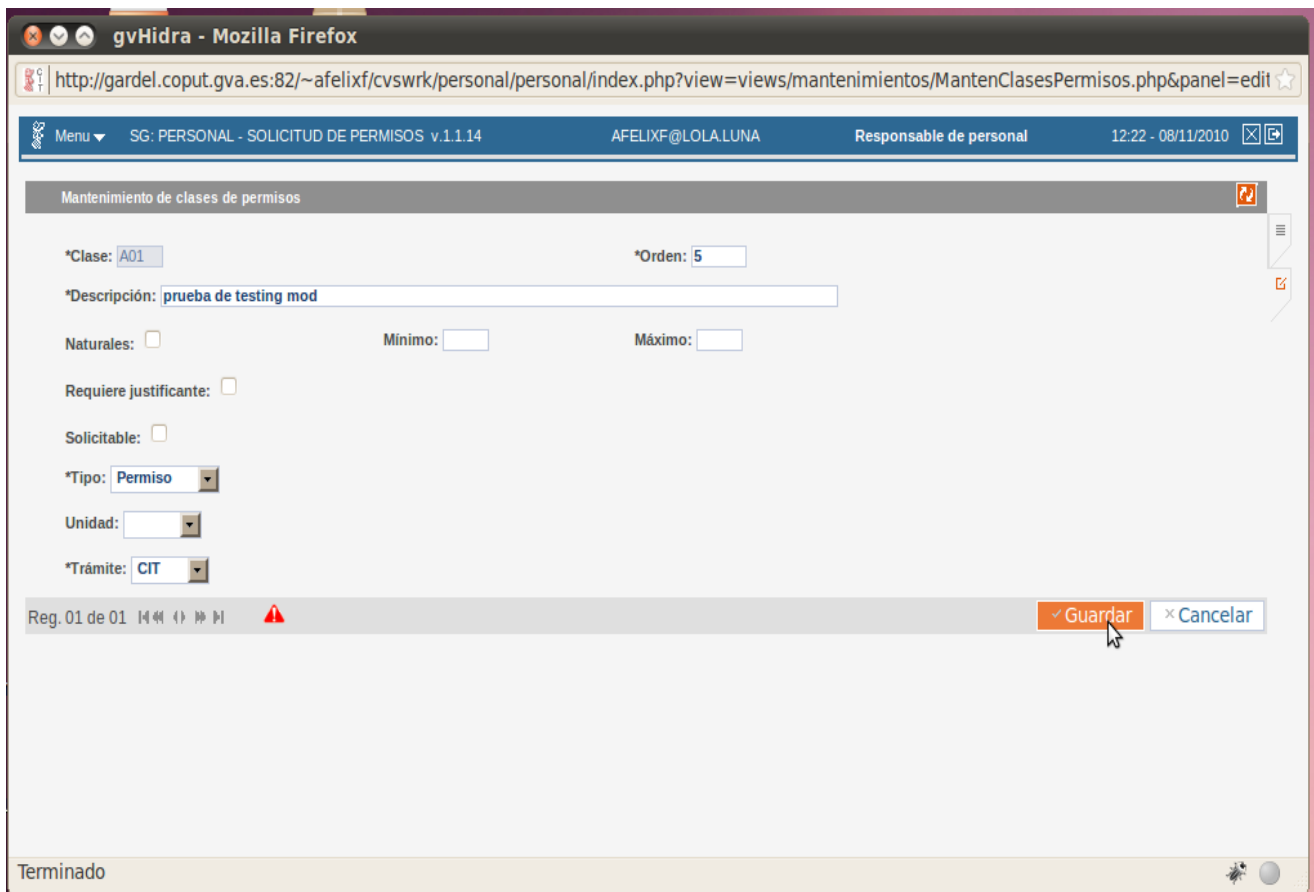
(Nº reg. 37) Pag. 01 de 03 14/11/10 02:03

Terminado

una vez dentro del panel de los datos, no modificamos nada, y le damos al botón cancelar. Y cuando se vuelve a la ventana anterior, verificamos (con asserts como antes) que la clase y la descripción siguen siendo como antes (que es lo esperado):

```
$this->assertEquals("prueba de testing", $this->getValue("cam__dclase__Tabla1_7"));
```

ahora modificamos los datos del registro, poniendo en orden el valor 5 por ejemplo, y en descripción escribimos "prueba de testing mod":



código generado:

```
$this->click("check_Tabla1_7");
$this->click("img_btllModificar_lis");
$this->click("cam__dclase__FichaEdicion_0");
$this->type("cam__dclase__FichaEdicion_0", "prueba de testing mod");
$this->type("cam__orden__FichaEdicion_0", "5");
$this->click("bnedi_guardar");
```

para hacer la comprobación verificamos con assertEquals que la cadena nueva de descripción esta presente :

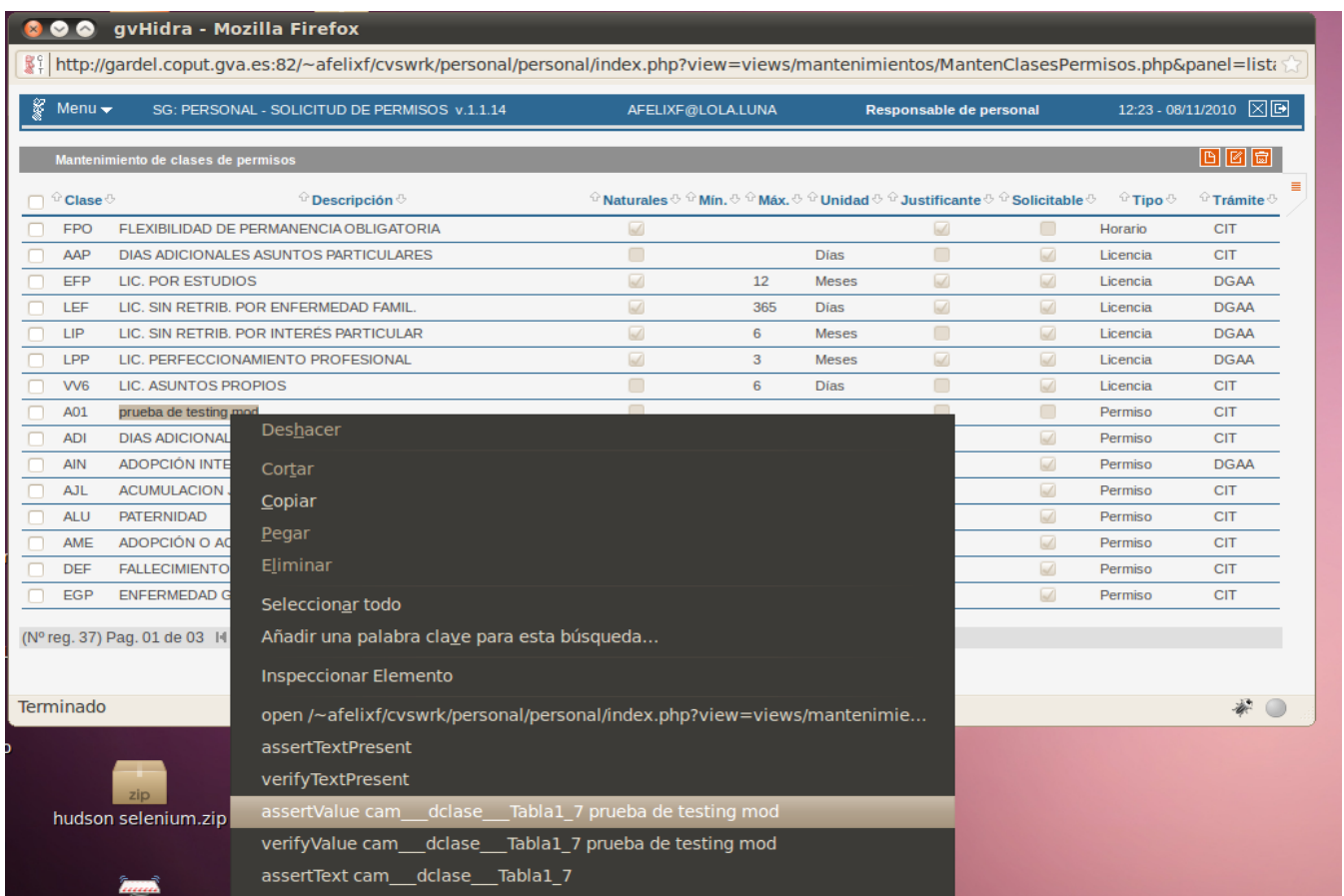
```
$this->assertEquals("prueba de testing mod", $this->getValue("cam__dclase__Tabla1_7"));
```

Paso nº4 (modificar datos sin grabar):

- Se modifican los datos
- Se pulsa en el botón “Cancelar”



y cuando volvemos a la ventana de la tabla de permisos verificamos que la descripción que se despliega sigue siendo la antigua:



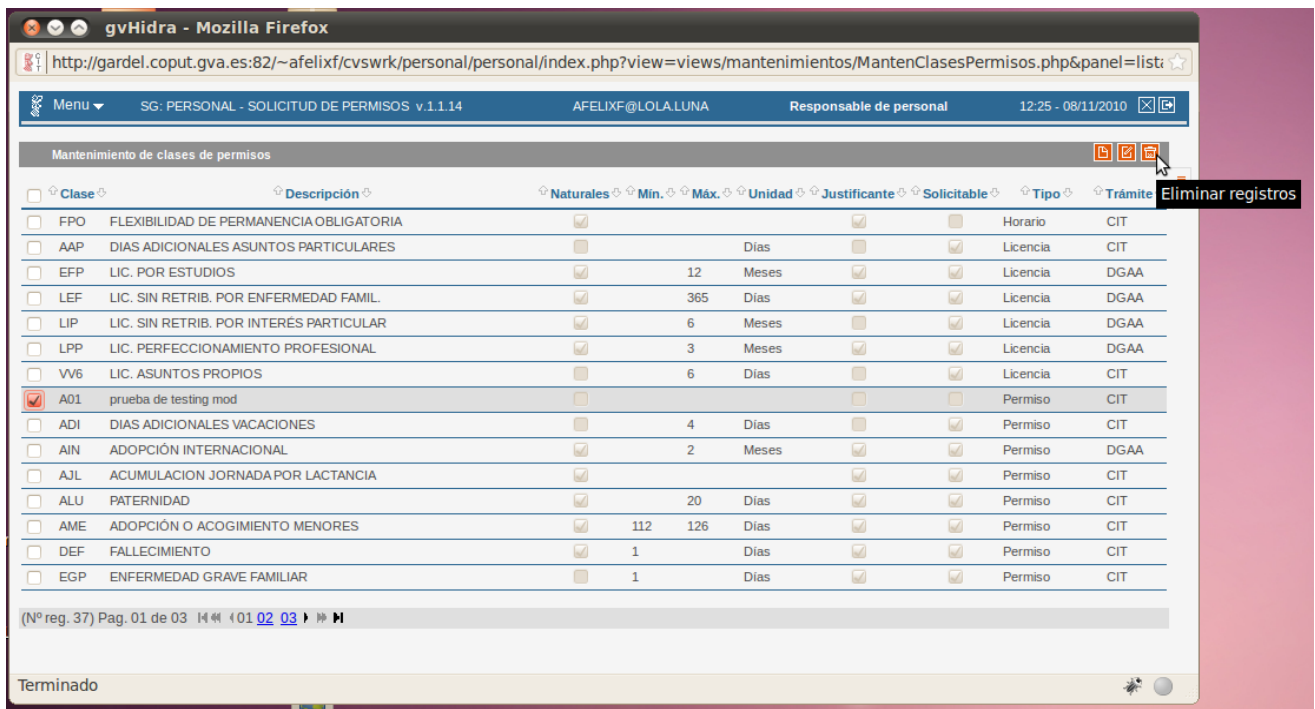
código generado:

```
$this->click("check_Tabla1_7");
$this->click("img_btllModificar_lis");
$this->type("cam__dclase__FichaEdicion_0", "prueba de testing mod sin grabar");
$this->click("pag_FichaEdicion_0");
$this->click("bnedi_cancelar");
```

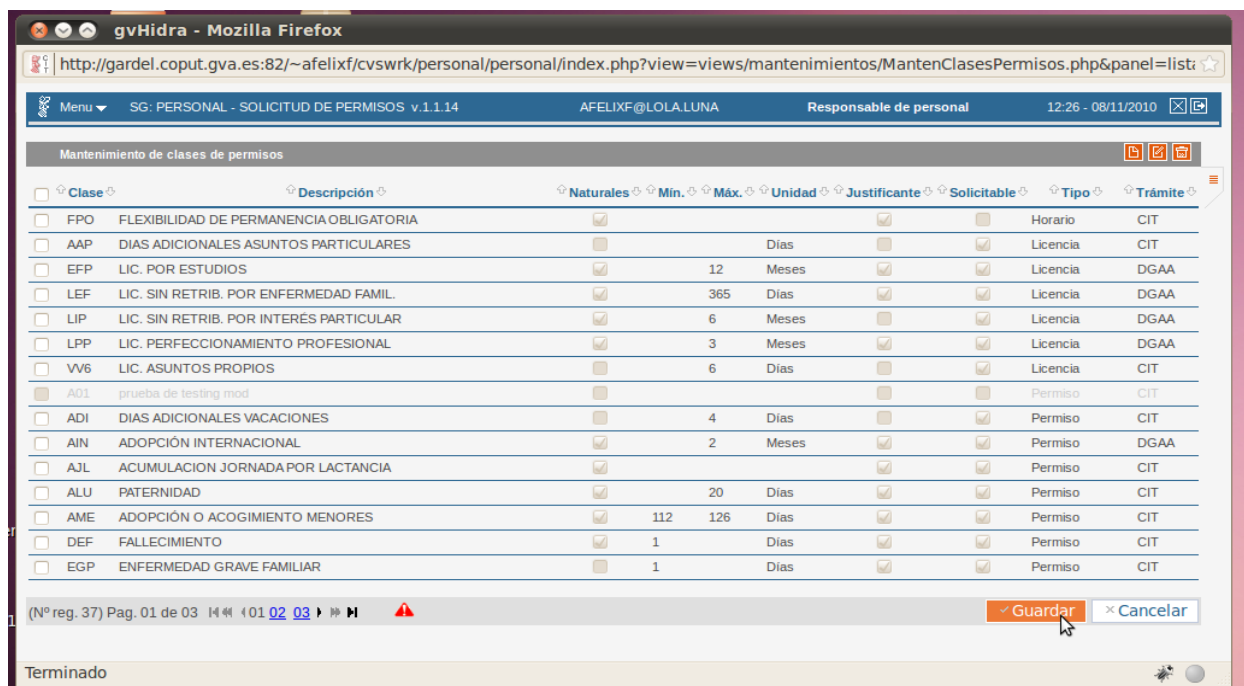
```
$this->assertEquals("prueba de testing mod", $this->getValue("cam__dclase__Tabla1_7"));
```

Paso nº5 (comprobar el borrado):

- Seleccionar una clase de permiso marcando el checkbox de la izquierda
- Pulsar en el botón de borrar registros de la esquina superior derecha



- Pulsar en el botón de “Guardar”



código generado :

```
$this->click("check_Tabla1_7");
$this->click("img_btllEliminar_lis");
$this->click("bnlis_guardar");
```

para comprobar que el borrado se hizo correctamente, podemos añadir manualmente, **una vez guardado el archivo del test** (ya que, escribir directamente dentro de la ventana de Selenium Ide puede generar problemas de ordenación de código como mencionamos anteriormente), podemos añadir dos llamadas a

“assertNotEquals” para comprobar que ya no existe el texto ni la descripción correspondientes al registro “A01” que borramos:

```
$this->assertNotEquals("prueba de testing", $this->getValue("cam__dclase__Tabla1_7"));
$this->assertNotEquals("A01", $this->getValue("cam__cclase__Tabla1_7"));
```

Cuando terminamos de utilizar la aplicación, volvemos a focalizar la ventana de Selenium Ide, y vemos que se ha generado el código del test que, por defecto esta en html, pero podemos cambiarlo a cualquiera de los lenguajes de programación que soporta Selenium Ide, y lo hacemos desde el menú opciones → Format → seleccionamos el lenguaje (php en nuestro caso).

Finalmente guardamos el test en un archivo, y le damos un nombre explicativo (por ejemplo el nombre del caso de uso en cuestión).

El código de test generado en la sección anterior lo mas probable es que falle si lo ejecutamos tal como esta, ya que faltan algunas modificación importantes que debemos considerar y que están descritas en el Apéndice de este proyecto.

Después de leer y aplicar las modificaciones descritas en el Apéndice(están en color verde), el contenido del fichero del test sería finalmente:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';
class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        //maquina donde esta corriendo el selenium-server.jar
        //puede ser local host o una IP .
        $this->setHost("localhost");
        /* Importante : el numero de puerto debe coincidir con el
        numero que se habrá utilizado al arrancar selenium-server (java -jar apache-server.jar -port 4445) en el nodo que se
        vaya a utilizar para lanzar el navegador cuando se inicie el desde el servidor apache .
        Si en el arranque de selenium-server.jar no se especifica un numero distinto , el puerto sería por defecto el 4444
        */
        $this->setPort(4445);

        $this->setBrowser("**chrome");
```

```
$this->setBrowserUrl("http://intranet.coput.gva.es:2005/inicio/");
$this->setSleep(5);
}
public function testMyTestCase()
{
    $this->open("/servicis-generals/aplicacions-informatiques/acces-personalitzat-a-aplicacions/");
    //$this->selectFrame("");
    $this->selectFrame("contenido");
    $this->type("as_usuario", <usuario>);
    $this->type("as_clave", <password>);
    $this->click("Aceptar");
    $this->waitForPageToLoad("30000");
    $this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
    $this->selectFrame("relative=up");
    $this->selectFrame("oculto");
    //$this->waitForPopUp("personal__W_PERDES", "30000");
    $this->openWindow("", "personal__W_PERDES");
    $this->selectWindow("name=personal__W_PERDES");
    $this->click("btnAceptar");
    $this->click("link=Mantenimientos");
    $this->click("link=Clases de permisos");
    $this->click("img_btllInsertar_lis");
    $this->type("ins__cclase__FichaEdicion_0", "A01");
    $this->type("ins__orden__FichaEdicion_0", "2");
    $this->type("ins__dclase__FichaEdicion_0", "prueba de testing");
    $this->focus("bnedi_guardar");
    $this->click("bnedi_guardar");
    $this->assertEquals("A01", $this->getValue("cam__cclase__Tabla1_0"));
    $this->assertEquals("prueba de testing", $this->getValue("cam__dclase__Tabla1_0"));
    $this->click("//img[@alt='X']");
    $this->waitForPageToLoad("30000");
```



```
$this->click("link=Mantenimientos");  
$this->click("link=Clases de permisos");  
$this->click("img_bttilInsertar_lis");  
$this->type("ins__cclase__FichaEdicion_0", "A02");  
$this->type("ins__dclase__FichaEdicion_0", "test_alta_permiso_mal");  
$this->type("ins__orden__FichaEdicion_0", "");  
$this->click("ins__dclase__FichaEdicion_0");  
$this->focus("bnedi_guardar");  
$this->click("bnedi_guardar");  
$this->assertTrue($this->isTextPresent("IGEP-901"));  
$this->assertTrue($this->isTextPresent("Faltan campos por rellenar"));  
$this->click("btnAceptar");  
$this->focus("bnedi_cancelar");  
$this->click("bnedi_cancelar");  
$this->click("check_Tabla1_7");  
$this->click("img_bttilModificar_lis");  
$this->focus("bnedi_cancelar");  
$this->click("bnedi_cancelar");  
$this->assertEquals("A01", $this->getValue("cam__cclase__Tabla1_7"));  
$this->assertEquals("prueba de testing", $this->getValue("cam__dclase__Tabla1_7"));  
$this->click("check_Tabla1_7");  
$this->click("img_bttilModificar_lis");  
$this->click("cam__dclase__FichaEdicion_0");  
$this->type("cam__dclase__FichaEdicion_0", "prueba de testing mod");  
$this->type("cam__orden__FichaEdicion_0", "5");  
$this->focus("bnedi_guardar");  
$this->click("bnedi_guardar");  
$this->assertEquals("prueba de testing mod", $this->getValue("cam__dclase__Tabla1_7"));  
$this->click("check_Tabla1_7");  
$this->click("img_bttilModificar_lis");  
$this->type("cam__dclase__FichaEdicion_0", "prueba de testing mod sin grabar");
```

```

$this->click("pag_FichaEdicion_0");
$this->focus("bnedi_cancelar");
$this->click("bnedi_cancelar");
$this->assertEquals("prueba de testing mod", $this->getValue("cam___dclase___Tabla1_7"));
$this->click("check_Tabla1_7");
$this->click("img_btflEliminar_lis");
$this->focus("bnlis_guardar");
$this->click("bnlis_guardar");
$this->assertNotEquals("prueba de testing", $this->getValue("cam___dclase___Tabla1_7"));
$this->assertNotEquals("A01", $this->getValue("cam___cclase___Tabla1_7"));
}
}
?>

```

Verificaciones a nivel de reglas de negocio:

El test construido en la sección anterior realiza las comprobaciones solamente a nivel de la interfaz gráfica, basándose en verificaciones de cadenas de texto, filas en las tablas, existencia de determinados elementos en la ventana, etc... Y aunque esto puede ser suficiente en nuestro caso concreto, pero en muchas ocasiones nos puede interesar para mas seguridad, y también puede ser necesaria, una comprobación basada en las reglas de negocio accediendo a la propia base de datos para verificar que los datos están realmente consistentes.

Esto se puede conseguir añadiendo después de los métodos setUp y testMyTestCase las definiciones de nuestras propias funciones que hacen la comprobación a nivel de negocio, y luego invocarlas donde sea necesario dentro del código del test .

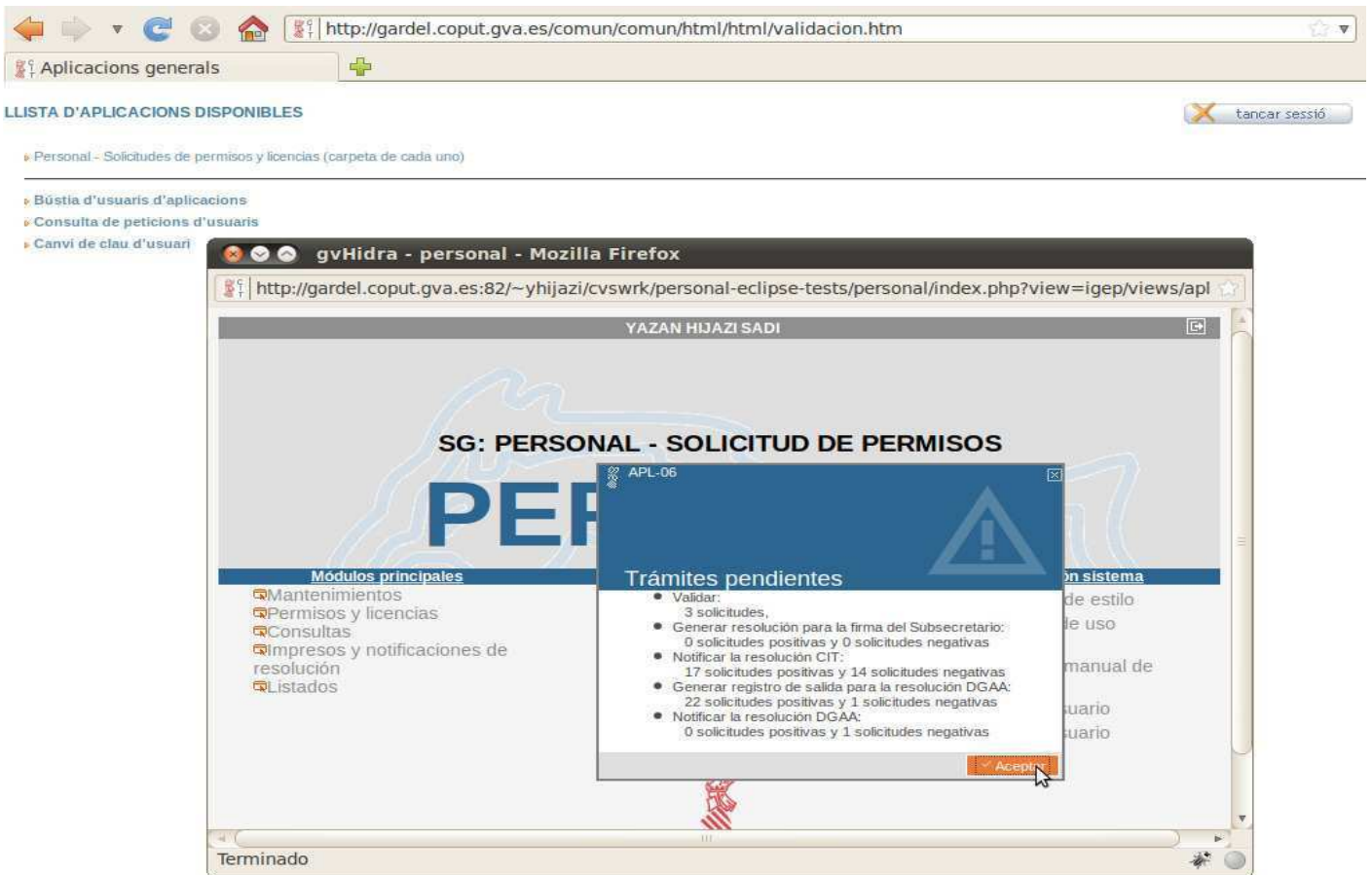
por ejemplo después de una inserción, podríamos insertar una llamado a una función que implementamos y que hace mediante una select la comprobación de que la tupla ha sido realmente insertada.

O también dentro del código de la función setUp (que es lo primero que se ejecuta del test), podríamos invocar a una función que realiza la conexión con la base de datos.

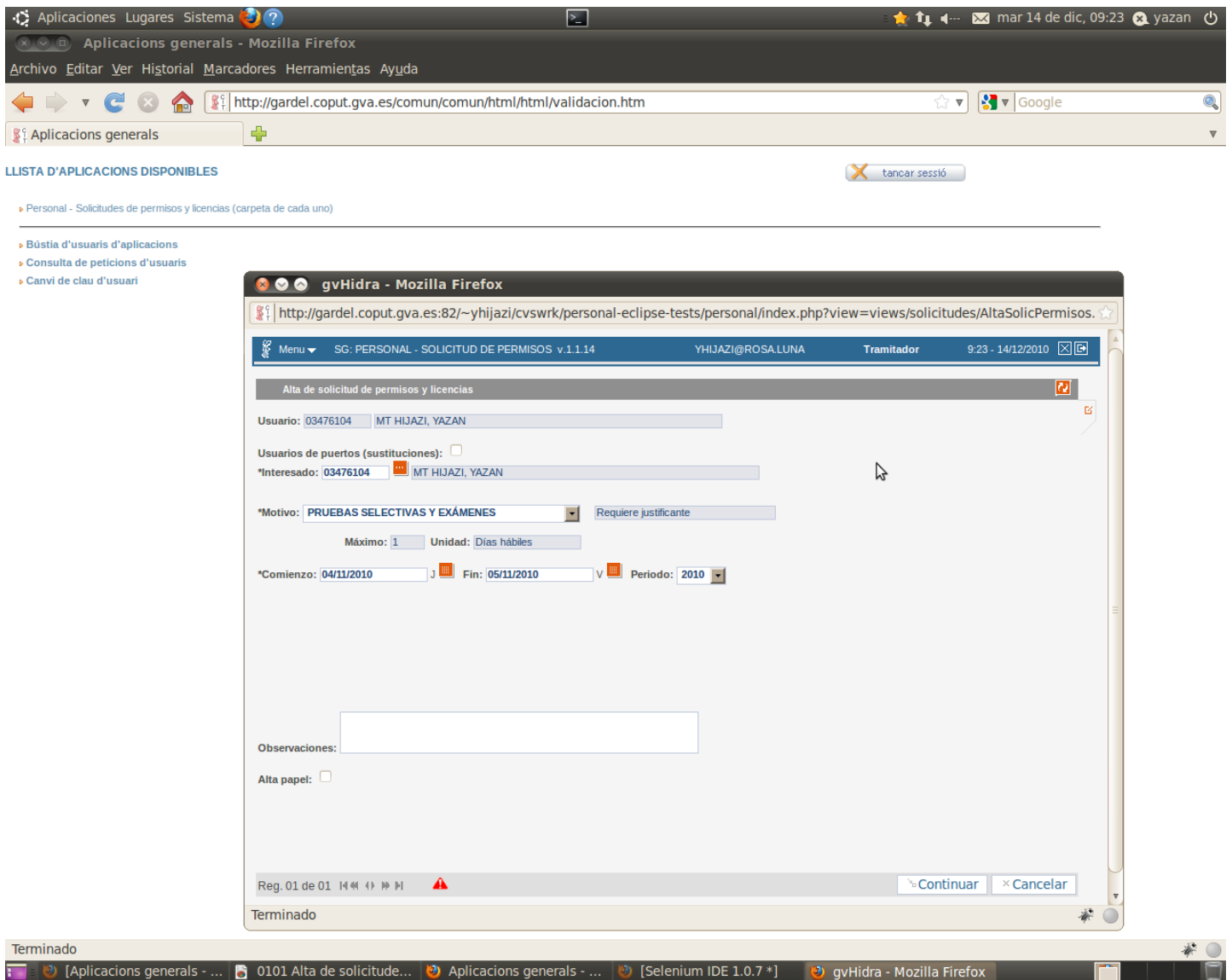
5.2.2. Tests para “Solicitud de permisos y licencias”

A lo largo de este apartado vamos a seguir ordenadamente los pasos descritos en la tabla de casos de uso del apartado 4.2.2 y utilizando exactamente los mismos datos especificados, e igual como en el ejemplo anterior, las modificación las distinguimos con color verde.

Primero arrancamos la aplicación pasando por la pagina de validación, e iniciamos el plugin de Selenium Ide a partir de esta pagina.

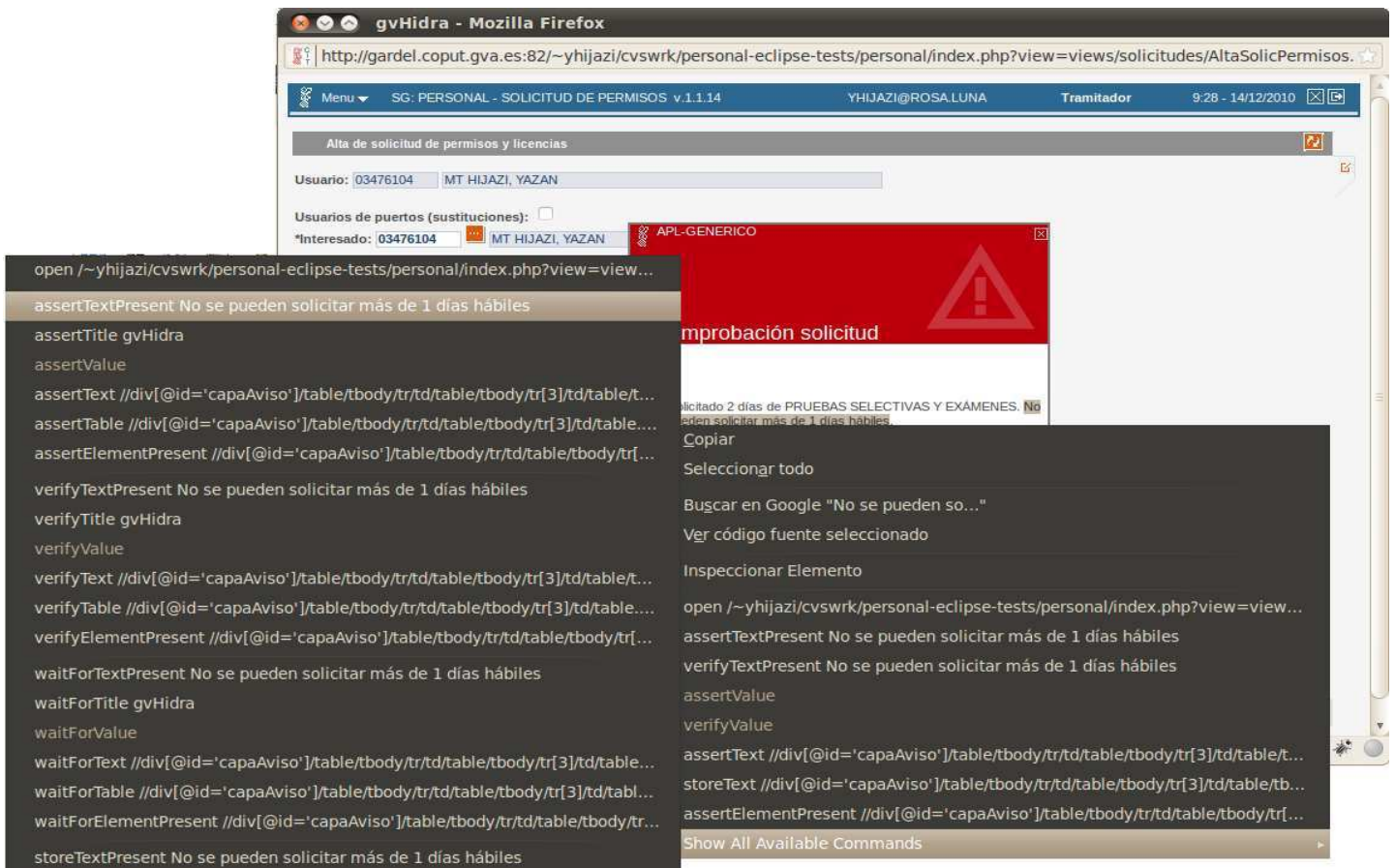


A1
 Se seleccionan dos días para pruebas selectivas y exámenes como nos indican en la tabla:



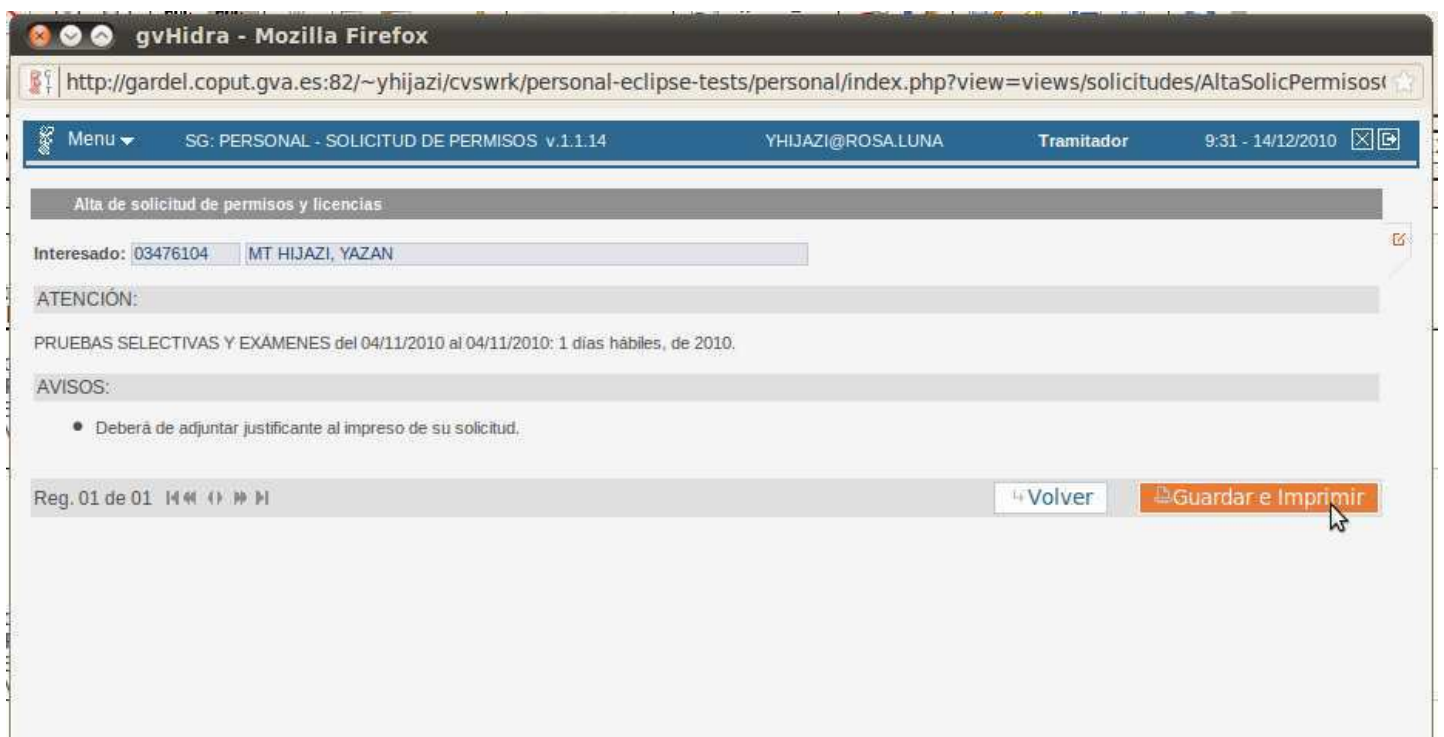
y al darle al botón continuar salta el error como se esperaba en este caso, por lo tanto verificamos su presencia mediante una llamada (botón derecho y seleccionar la opción `assert text present`) al método “`assertTrue`”:

```
$this->assertTrue($this->isTextPresent("No se pueden solicitar más de 1 días hábiles"));
```

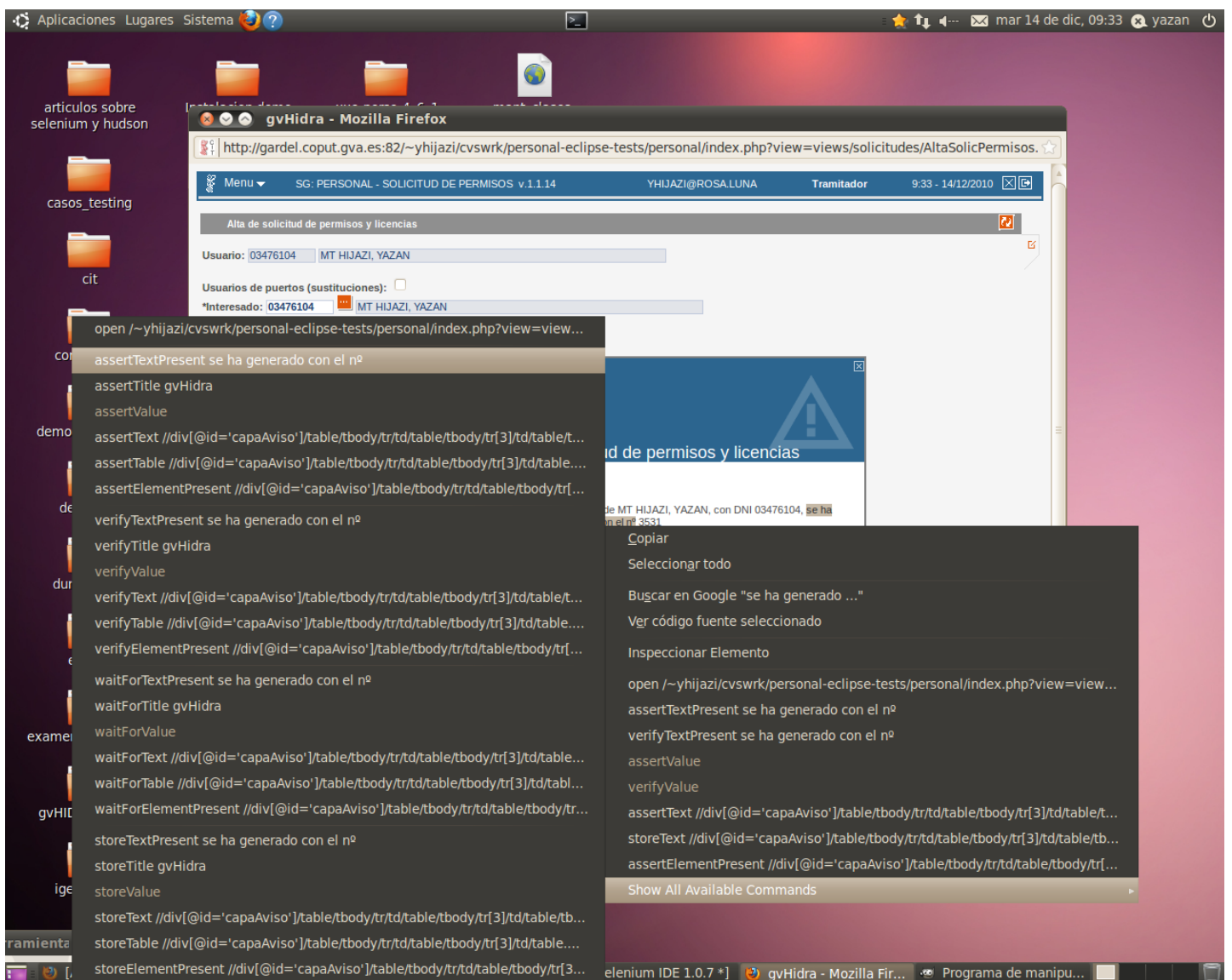
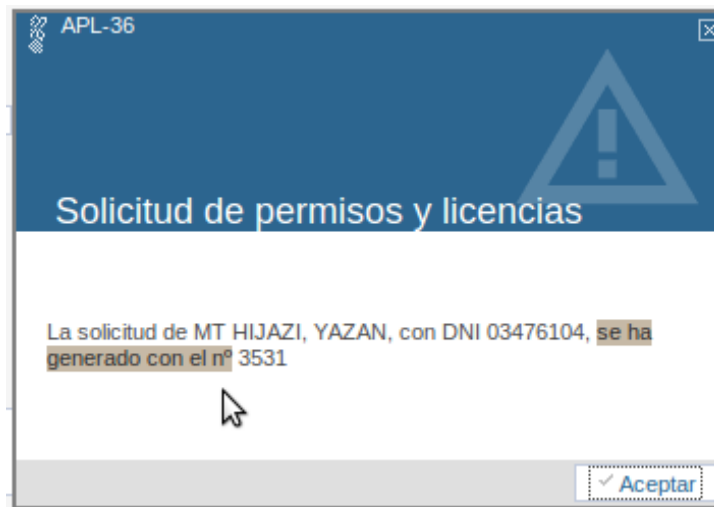



A2

Se ajusta el numero de días a 1, por lo cual la solicitud se realiza con éxito y se graba al darle al botón guardar e imprimir:

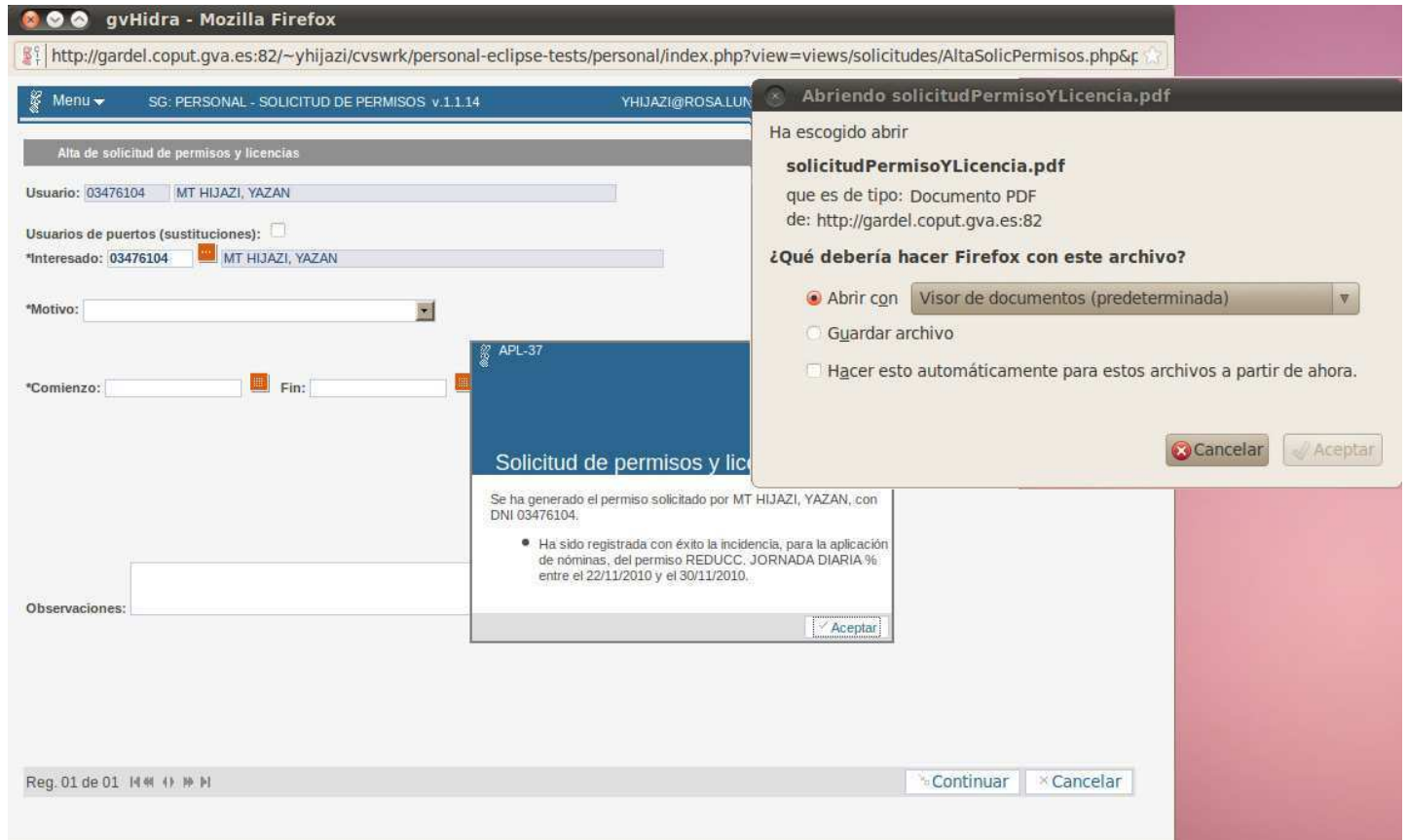


verificamos entonces que el mensaje que confirma la inserción del nuevo registro se muestra correctamente.



Y como se explica en el Apéndice, en vez de cerrar o darle al botón cancelar del dialogo que nos salta “Abrir o Guardar archivo pdf” (ya que esta acción no se añade al código generado), simplemente se le vuelve a dar el foco a la ventana de la aplicación con la llamada

`$this->windowFocus(...)`, dejando el cierre de la misma para el final cuando Selenium termine el proceso de firefox al terminarse la ejecución del test.



El código generado para las partes A1 y A2 es el siguiente:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';
class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setPort(4445);
        $this->setHost("172.19.80.73");
        $this->setBrowser("**chrome");
        $this->setBrowserUrl("http://gardel.coput.gva.es/comun/comun/html/html/validacion.htm");
        $this->setSleep(7);
    }
}
```

```

}
public function testMyTestCase()
{
$this->open("/comun/comun/html/html/validacion.htm");
$this->selectFrame("contenido");
$this->type("as_usuario", "<usuario>");
$this->type("as_clave", "<password>");
$this->click("Aceptar");
$this->waitForPageToLoad("30000");
$this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
$this->selectFrame("relative=up");
$this->selectFrame("oculto");
//$this->waitForPopUp("personal__W_PERDES", "30000");
$this->openWindow("", "personal__W_PERDES");
$this->selectWindow("name=personal__W_PERDES");
$this->click("btnAceptar");
$this->click("link=Permisos y licencias");
$this->click("link=Alta");
$this->select("cam__motivo__FichaEdicion_0", "label=PRUEBAS SELECTIVAS Y EXÁMENES");
$this->click("//select[@id='cam__motivo__FichaEdicion_0']/option[9]");
$this->click("cam__fechalni__FichaEdicion_0");
$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");
//$this->click("cal_cam__fechalni__FichaEdicion_0");
$this->type("cam__fechalni__FichaEdicion_0", "04/11/2010");
//$this->click("//div[2]/table/tbody/tr[1]/td[5]");
$this->click("cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "");
$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");
//$this->click("cal_cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "05/11/2010");
//$this->click("//div[3]/table/tbody/tr[1]/td[6]");

```

```

$this->focus("bnsaltoConfirmar_saltaredi");

$this->click("bnsaltoConfirmar_saltaredi");

$this->assertTrue($this->isTextPresent("No se pueden solicitar más de 1 días hábiles"));

$this->click("btnAceptar");

$this->focus("bnedi_cancelar");

$this->click("bnedi_cancelar");

$this->click("link=Permisos y licencias");

$this->click("link=Alta");

$this->select("cam__motivo__FichaEdicion_0", "label=PRUEBAS SELECTIVAS Y EXÁMENES");

$this->click("cam__fechalni__FichaEdicion_0");

$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");

//$this->click("cal_cam__fechalni__FichaEdicion_0");

$this->type("cam__fechalni__FichaEdicion_0", "04/11/2010");

//$this->click("//div[2]/table/tbody/tr[1]/td[5]");

$this->click("cam__fechaFin__FichaEdicion_0");

$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");

//$this->click("cal_cam__fechaFin__FichaEdicion_0");

//$this->click("//div[3]/table/thead/tr[2]/td[2]/div");

$this->type("cam__fechaFin__FichaEdicion_0", "04/11/2010");

//$this->click("//div[3]/table/tbody/tr[1]/td[5]");

$this->focus("bnsaltoConfirmar_saltaredi");

$this->click("bnsaltoConfirmar_saltaredi");

$this->focus("bnedi_particular");

$this->click("bnedi_particular");

//$this->waitForPopUp("ventana", "30000");

$this->windowFocus();

$this->assertTrue($this->isTextPresent("se ha generado con el nº"));

$this->click("btnAceptar");

$this->focus("bnedi_cancelar");

$this->click("bnedi_cancelar");

}

```

```
}  
?>
```

A3

Alta de solicitud de vacaciones superando el máximo permitido:

gvHidra - Mozilla Firefox
http://gardel.coput.gva.es:82/~yhijazi/cvswrk/personal-eclipse-tests/personal/index.php?view=views/solicitudes/AltaSolicPermisos.php&panel=edit

Menu SG: PERSONAL - SOLICITUD DE PERMISOS v.1.1.14 YHIJAZI@ROSA.LUNA Tramitador 9:48 - 14/12/2010

Alta de solicitud de permisos y licencias

Usuario: 03476104 MT HIJAZI, YAZAN

Usuarios de puertos (sustituciones):

*Interesado: 03476104 MT HIJAZI, YAZAN

*Motivo: VACACIONES No requiere justificante

Máximo: 22 Unidad: Días hábiles

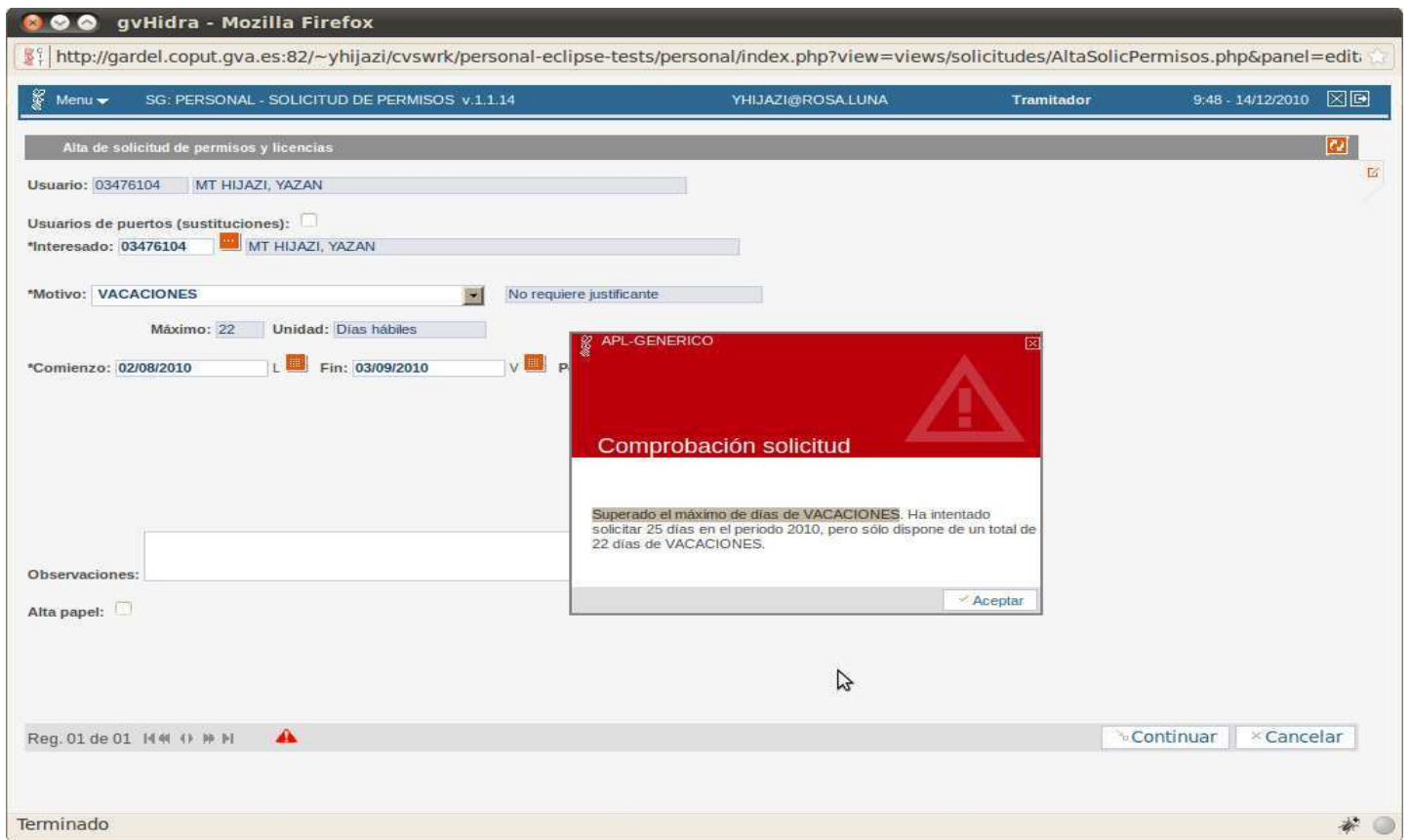
*Comienzo: 02/08/2010 L Fin: 03/09/2010 V Periodo: 2010

Observaciones:

Alta papel:

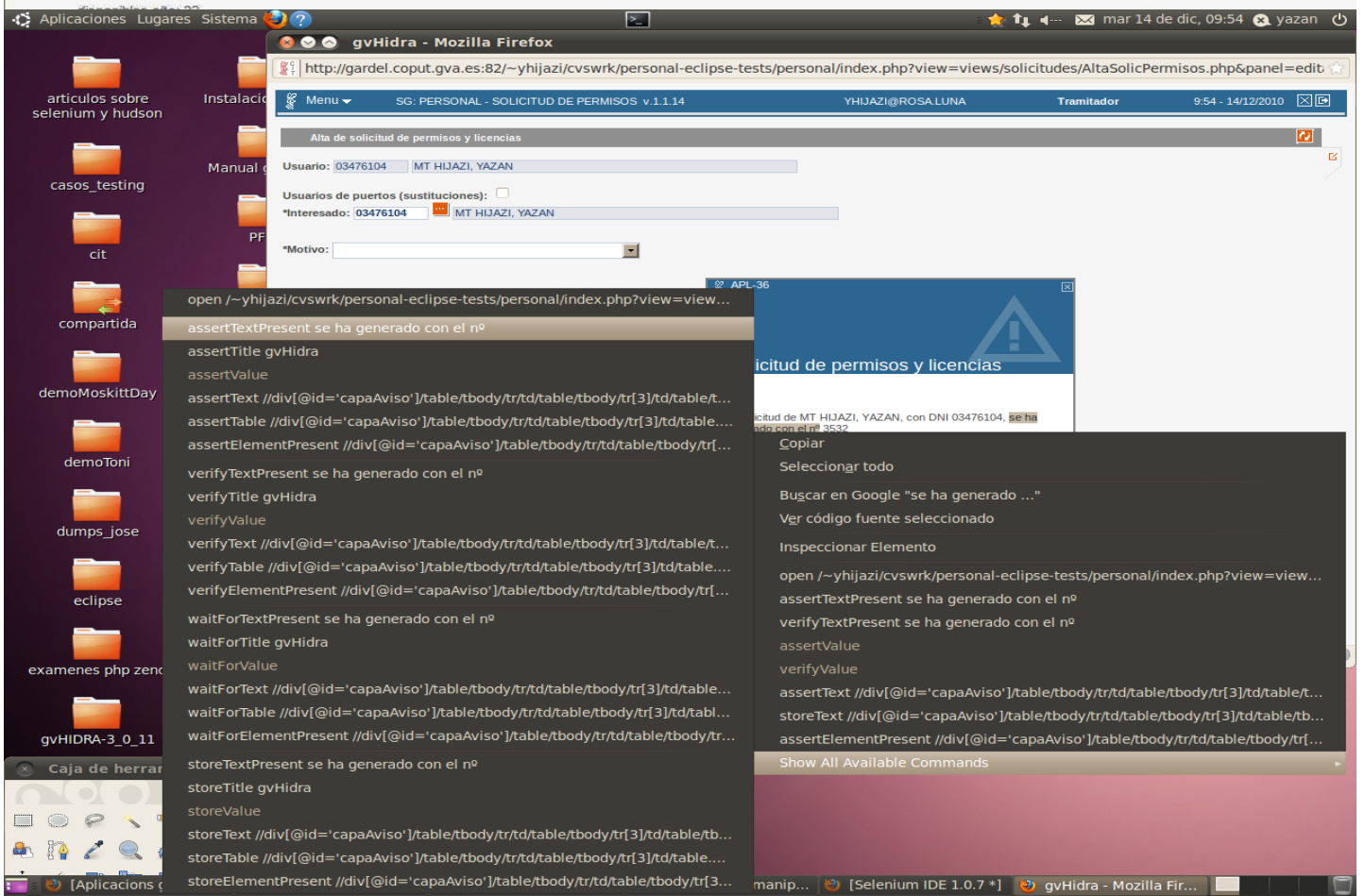
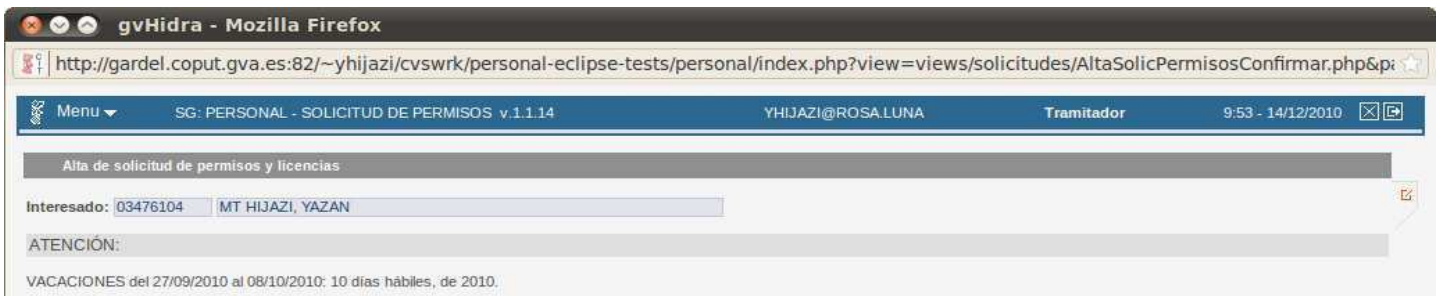
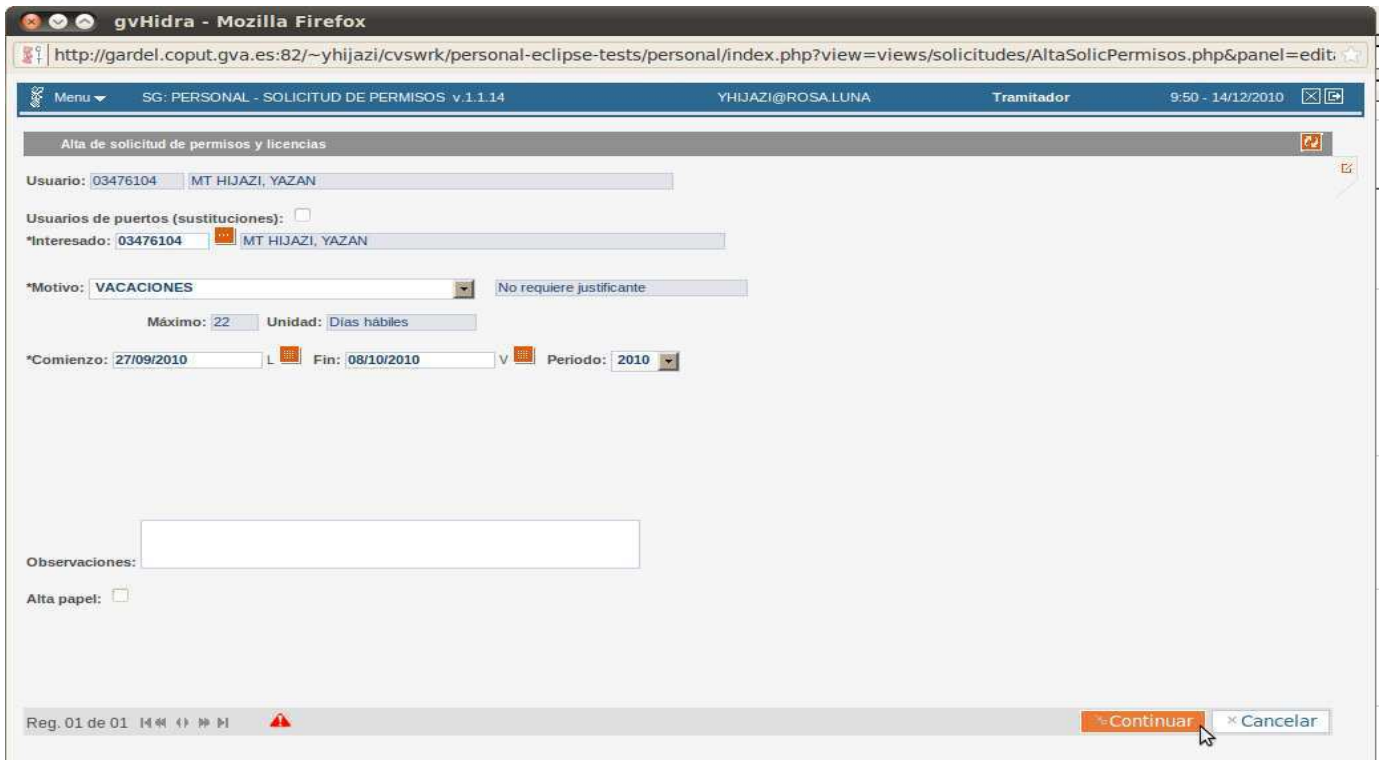
Reg. 01 de 01 Continuar Cancelar

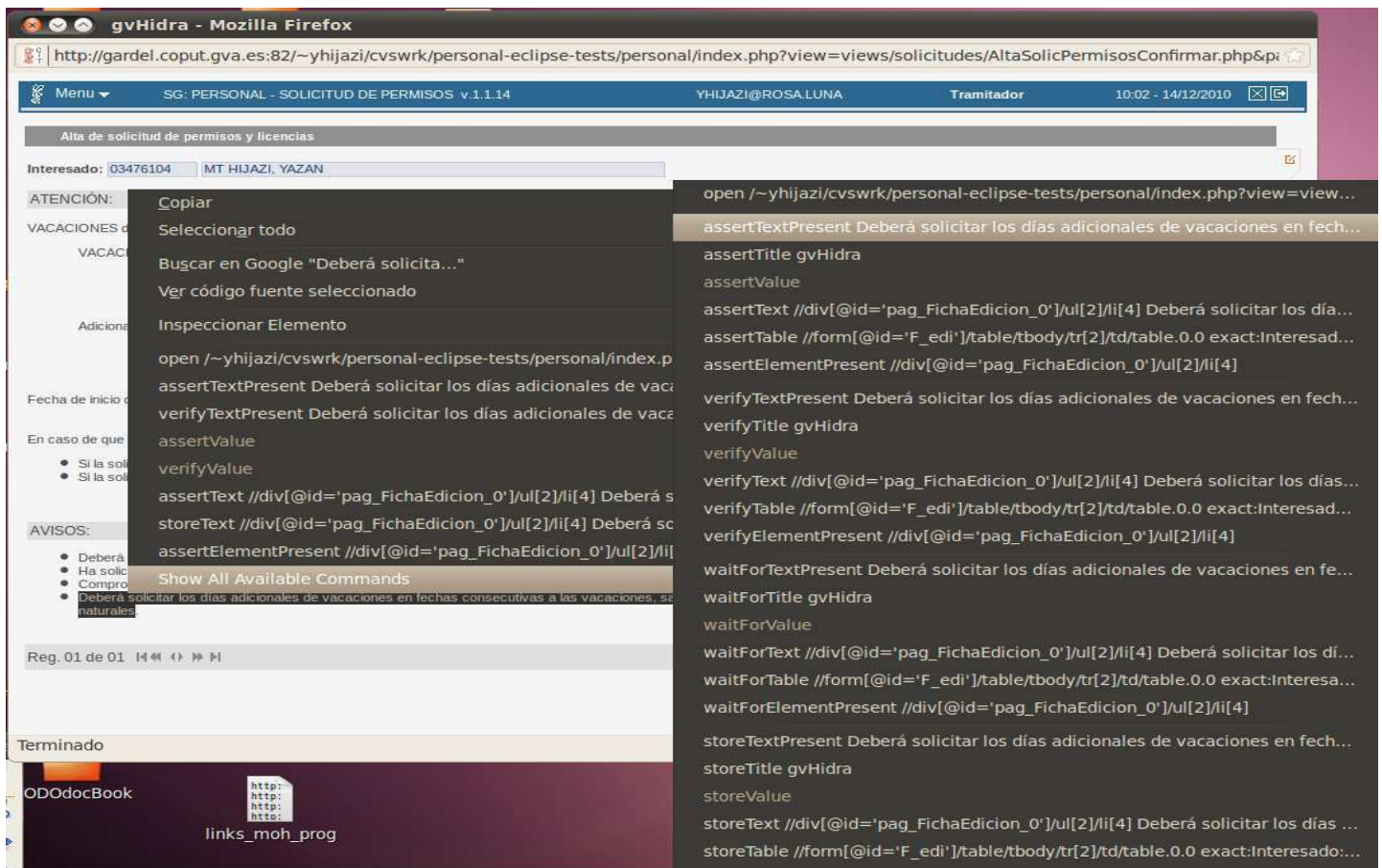
verificamos la presencia del texto del error como en el caso anterior:



A4, A4.1, A4.2:

Se dan de alta las solicitudes de vacaciones indicadas en la tabla, verificando como siempre el mensaje de confirmación:





y el código para los tests A3, A4, A4.1 y A4.2 sera el siguiente:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';

class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setPort(4445);
        $this->setHost("172.19.80.73");
        $this->setBrowser("**chrome");
        $this->setBrowserUrl("http://gardel.coput.gva.es/comun/comun/html/html/validacion.htm");
        $this->setSleep(7);
    }
}
```

```

public function testMyTestCase()
{
    $this->open("/comun/comun/html/html/validacion.htm");

    $this->selectFrame("contenido");

    $this->type("as_usuario", "<usuario>");
    $this->type("as_clave", "<password>");

    $this->click("Aceptar");

    $this->waitForPageToLoad("30000");

    $this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");

    $this->selectFrame("relative=up");
    $this->selectFrame("oculto");

    //$this->waitForPopUp("personal__W_PERDES", "30000");
    $this->openWindow("", "personal__W_PERDES");

    $this->selectWindow("name=personal__W_PERDES");

    $this->click("btnAceptar");

    $this->click("link=Permisos y licencias");

    $this->click("link=Alta");

    $this->select("cam__motivo__FichaEdicion_0", "label=VACACIONES");
    $this->click("//select[@id='cam__motivo__FichaEdicion_0']/option[5]");
    $this->click("cam__fechalni__FichaEdicion_0");

    $this->doubleClick("cal_cam__fechalni__FichaEdicion_0");

    //$this->click("cal_cam__fechalni__FichaEdicion_0");

    $this->type("cam__fechalni__FichaEdicion_0", "02/08/2010");

    //$this->click("//tbody/tr[2]/td[2]");

    $this->click("cam__fechaFin__FichaEdicion_0");

    $this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");

    //$this->click("cal_cam__fechaFin__FichaEdicion_0");

    //$this->click("//div[3]/table/thead/tr[2]/td[4]/div");

    $this->type("cam__fechaFin__FichaEdicion_0", "03/09/2010");

    //$this->click("//div[3]/table/tbody/tr[1]/td[6]");

```

```

$this->focus("bnsaltoConfirmar_saltaredi");

$this->click("bnsaltoConfirmar_saltaredi");

$this->click("btnAceptar");

$this->click("cam__fechalni__FichaEdicion_0");

$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");

//$this->click("cal_cam__fechalni__FichaEdicion_0");

$this->type("cam__fechalni__FichaEdicion_0", "27/09/2010");

//$this->click("//div[5]/table/tbody/tr[5]/td[2]");

$this->click("cam__fechaFin__FichaEdicion_0");

$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");

//$this->click("cal_cam__fechaFin__FichaEdicion_0");

$this->type("cam__fechaFin__FichaEdicion_0", "08/10/2010");

//$this->click("//div[7]/table/tbody/tr[2]/td[6]");

$this->focus("bnsaltoConfirmar_saltaredi");

$this->click("bnsaltoConfirmar_saltaredi");

$this->focus("bnedi_particular");

$this->click("bnedi_particular");

$this->assertTrue($this->isTextPresent("se ha generado con el nº"));

$this->click("btnAceptar");

$this->click("cam__motivo__FichaEdicion_0");

$this->select("cam__motivo__FichaEdicion_0", "label=VACACIONES");

$this->click("cam__fechalni__FichaEdicion_0");

$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");

//$this->click("cal_cam__fechalni__FichaEdicion_0");

$this->type("cam__fechalni__FichaEdicion_0", "18/10/2010");

//$this->click("//tr[4]/td[2]");

$this->click("cam__fechaFin__FichaEdicion_0");

$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");

//$this->click("cal_cam__fechaFin__FichaEdicion_0");

$this->type("cam__fechaFin__FichaEdicion_0", "29/10/2010");

//$this->click("//div[3]/table/tbody/tr[5]/td[6]");

```

```

$this->focus("bnsaltoConfirmar_saltaredi");

$this->click("bnsaltoConfirmar_saltaredi");

$this->focus("bnedi_particular");

$this->click("bnedi_particular");

$this->assertTrue($this->isTextPresent("se ha generado con el nº"));

$this->click("btnAceptar");

$this->click("cam__motivo__FichaEdicion_0");

$this->select("cam__motivo__FichaEdicion_0", "label=VACACIONES");

$this->click("cam__fechaIni__FichaEdicion_0");

$this->doubleClick("cal_cam__fechaIni__FichaEdicion_0");

//$this->click("cal_cam__fechaIni__FichaEdicion_0");

$this->type("cam__fechaIni__FichaEdicion_0", "17/11/2010");

//$this->click("//tbody/tr[3]/td[4]");

$this->click("cam__fechaFin__FichaEdicion_0");

$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");

//$this->click("cal_cam__fechaFin__FichaEdicion_0");

$this->type("cam__fechaFin__FichaEdicion_0", "18/11/2010");

//$this->click("//div[3]/table/tbody/tr[3]/td[5]");

$this->focus("bnsaltoConfirmar_saltaredi");

$this->click("bnsaltoConfirmar_saltaredi");

$this->assertTrue($this->isTextPresent("Deberá solicitar a continuación los días adicionales de vacaciones para que el total solicitado no sea inferior a 7 días naturales consecutivos"));

$this->assertTrue($this->isTextPresent("Deberá solicitar los días adicionales de vacaciones en fechas consecutivas a las vacaciones, salvo que su solicitud, de días adicionales de vacaciones, tenga una duración de al menos 7 días naturales"));

$this->focus("bnedi_particular");

$this->click("bnedi_particular");

$this->assertTrue($this->isTextPresent("se ha generado con el nº"));

$this->click("btnAceptar");

}

}

?>

```

A6

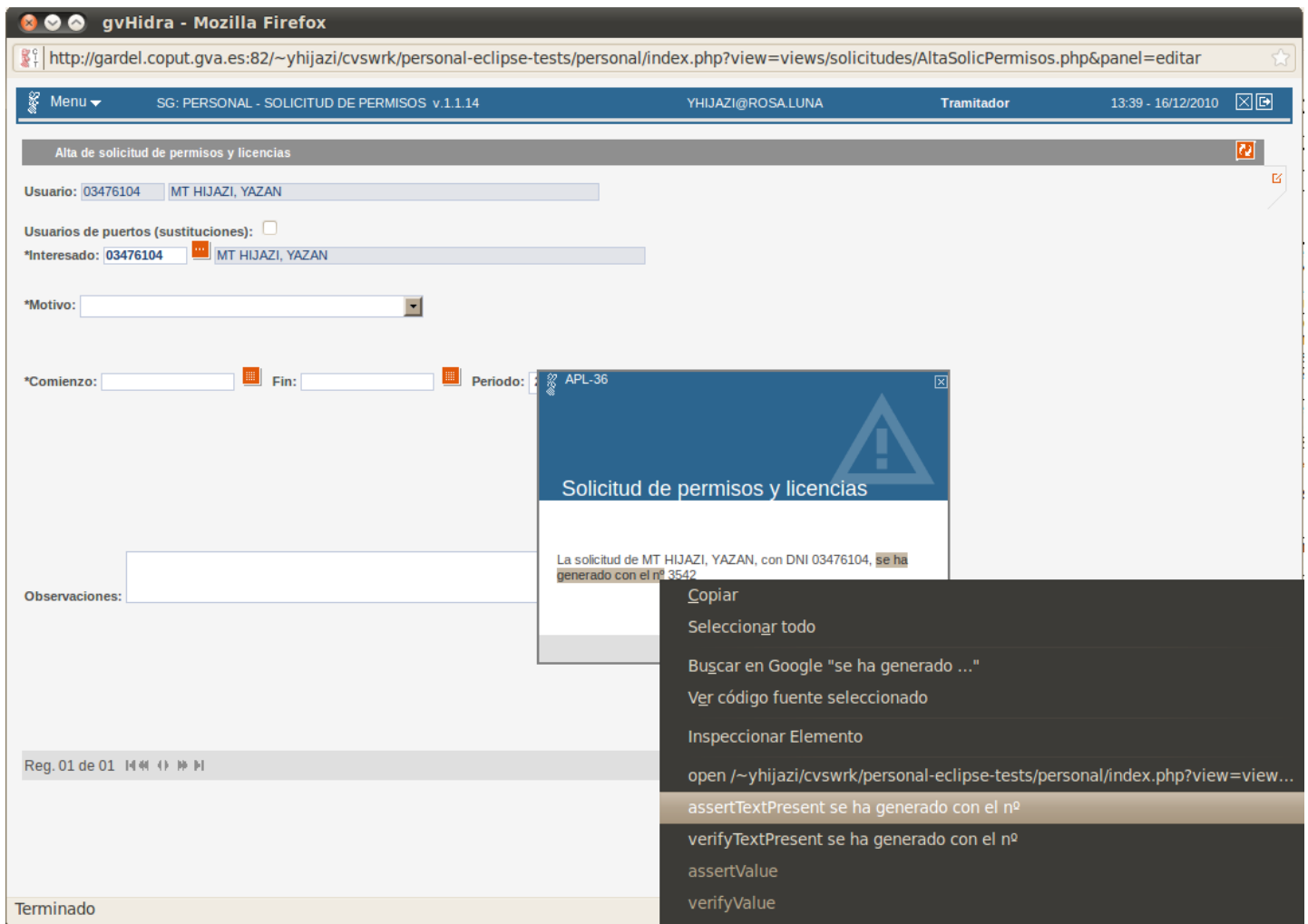
Se solicitan dos días adicionales de vacaciones:

The screenshot shows a web browser window titled 'gvHidra - Mozilla Firefox' with the URL 'http://gardel.coput.gva.es:82/~yhijazi/cvswrk/personal-eclipse-tests/personal/index.php?view=views/solicitudes/AltaSolicPermisos.php&panel=editar'. The page header includes 'Menu', 'SG: PERSONAL - SOLICITUD DE PERMISOS v.1.1.14', 'YHIJAZI@ROSALUNA', 'Tramitador', and '13:37 - 16/12/2010'. The main content area is titled 'Alta de solicitud de permisos y licencias' and contains the following form elements:

- Usuario:** 03476104 MT HIJAZI, YAZAN
- Usuarios de puertos (sustituciones):**
- *Interesado:** 03476104 MT HIJAZI, YAZAN
- *Motivo:** DIAS ADICIONALES VACACIONES (dropdown menu) No requiere justificante
- Máximo:** 2 **Unidad:** Dias hábiles
- *Comienzo:** 02/11/2010 **Fin:** 03/11/2010 **Periodo:** 2010
- Observaciones:** [Empty text box]
- Alta papel:**

At the bottom of the page, there is a navigation bar with 'Reg. 01 de 01' and a red warning icon. On the right side of this bar, there are two buttons: 'Continuar' (highlighted in orange) and 'Cancelar'.

y se verifica que la solicitud se ha generado correctamente:



Como se indica en la tabla del caso de uso, para esta funcionalidad en concreto (y para la A.6) se requiere también una verificación del estado de la base de datos (mediante una select).

Para poder realizar esta verificación implementamos dos métodos que incluiremos en el propio archivo del test. El primer método “connectToApplication()” realiza la conexión con la base de datos, invocando internamente a los métodos ya implementados en el framework gvHIDRA para tal fin:

```
public function connectToApplication() {
include_once('igep/include_class.php');
include_once('igep/include_all.php');
$clasePrincipal = new AppMainWindow();
//error global
global $g_error;
$g_error = new IgepError();
$conf = ConfigFramework::getConfig();
}
```

este método lo invocamos una única vez al principio del test (por ejemplo dentro del método setUp), y luego para cada consulta o verificación del estado de la base de datos invocamos al método correspondiente cuando se necesite (en nuestro caso haríamos una select justo después de la inserción).

El segundo método “checkInsert()” ejecuta la consulta verificando que el resultado obtenido es el que se espera realmente:

```
private function checkInsert() {
$conf = ConfigFramework::getConfig();
$dsn = $conf->getDSN('g_dsn');
$con = new IgepConexion($dsn);
$res = $con->consultar('SELECT diasadi_act FROM TPER_PERSONAS WHERE
nregpgv = \'03476104W0\');
if($res[0]['diasadi_act']==2)
return true;
return false;
}
```

en nuestro caso estamos verificando (instrucción if), que los días adicionales de vacaciones que tiene asignados la persona en cuestión realmente son igual a 2.

A6.1

Solicitamos mas días de los permitidos para provocar el error, y verificamos igual que en los casos anteriores la presencia de la ventana y de la cadena del texto del error:

The screenshot shows a web browser window with the URL `http://gardel.coput.gva.es:82/~yhijazi/cvswrk/personal-eclipse-tests/personal/index.php?view=views/solicitudes/AltaSolicPermisos.php&panel=editar`. The page title is "Alta de solicitud de permisos y licencias". The form contains the following fields:

- Usuario: 03476104 MT HIJAZI, YAZAN
- Usuarios de puertos (sustituciones):
- *Interesado: 03476104 MT HIJAZI, YAZAN
- *Motivo: DIAS ADICIONALES VACACIONES (No requiere justificante)
- Máximo: 2 Unidad: Días hábiles
- *Comienzo: 05/11/2010 Fin: 05/11/2010 Periodo: APL-GENERICO

A red modal dialog box titled "Comprobación solicitud" is overlaid on the form. The message inside reads: "Ha solicitado 3 DIAS ADICIONALES VACACIONES, y no puede solicitar más de 2 días hábiles. No se pueden solicitar permisos de DIAS ADICIONALES VACACIONES de menos de 7 días naturales." There is an "Aceptar" button at the bottom right of the dialog.

Y el código para las funcionalidades A6 y A6.1 sera:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';
class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setPort(4445);
        $this->setHost("172.19.80.73");
        $this->setBrowser("**chrome");
        $this->setBrowserUrl("http://gardel.coput.gva.es/comun/comun/html/html/validacion.htm");
        $this->setSleep(7);
        $this->connectToApplication();
    }
    public function testMyTestCase()
    {
        $this->open("/comun/comun/html/html/validacion.htm");
        $this->selectFrame("contenido");
        $this->type("as_usuario", "<usuario>");
        $this->type("as_clave", "<password>");
        $this->click("Aceptar");
        $this->waitForPageToLoad("30000");
        $this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
        $this->selectFrame("relative=up");
        $this->selectFrame("oculto");
        //$this->waitForPopUp("personal__W_PERDES", "30000");
        $this->openWindow("", "personal__W_PERDES");
        $this->selectWindow("name=personal__W_PERDES");
        $this->click("btnAceptar");
        $this->click("link=Permisos y licencias");
        $this->click("link=Alta");
    }
}
```

```

$this->select("cam__motivo__FichaEdicion_0", "label=DIAS ADICIONALES VACACIONES");
$this->click("cam__fechaIni__FichaEdicion_0");
$this->doubleClick("cal_cam__fechaIni__FichaEdicion_0");
//$this->click("cal_cam__fechaIni__FichaEdicion_0");
$this->type("cam__fechaIni__FichaEdicion_0", "02/11/2010");
$this->click("cam__fechaFin__FichaEdicion_0");
$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");
//$this->click("cal_cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "03/11/2010");
$this->focus("bnsaltoConfirmar_saltaredi");
$this->click("bnsaltoConfirmar_saltaredi");
$this->focus("bnedi_particular");
$this->click("bnedi_particular");
$this->assertTrue($this->isTextPresent("se ha generado con el nº"));
$this->click("btnAceptar");
//Comprobación integridad de la BBDD.
$this->assertTrue($this->checkInsert());
$this->select("cam__motivo__FichaEdicion_0", "label=DIAS ADICIONALES VACACIONES");
$this->click("cam__fechaIni__FichaEdicion_0");
$this->doubleClick("cal_cam__fechaIni__FichaEdicion_0");
//$this->click("cal_cam__fechaIni__FichaEdicion_0");
//$this->click("//tr[2]/td[2]/div");
$this->type("cam__fechaIni__FichaEdicion_0", "05/11/2010");
$this->click("cam__fechaFin__FichaEdicion_0");
$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");
//$this->click("cal_cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "05/11/2010");
$this->focus("bnsaltoConfirmar_saltaredi");
$this->click("bnsaltoConfirmar_saltaredi");
$this->assertTrue($this->isTextPresent("Ha solicitado 3 DIAS ADICIONALES VACACIONES, y no puede solicitar
más de 2 días hábiles"));

```

```

$this->assertTrue($this->isTextPresent("No se pueden solicitar permisos de DIAS ADICIONALES VACACIONES de
menos de 7 días naturales"));
$this->click("btnAceptar");
//Comprobación integridad de la BBDD.
$this->assertTrue($this->checkInsert());
}
private function checkInsert() {
    $conf = ConfigFramework::getConfig();
    $dsn = $conf->getDSN('g_dsn');
    $con = new IgepConexion($dsn);
    $res = $con->consultar('SELECT diasadi_act FROM TPER_PERSONAS WHERE nregpgv = \'03476104W0\');

    if($res[0]['diasadi_act']==2)
        return true;

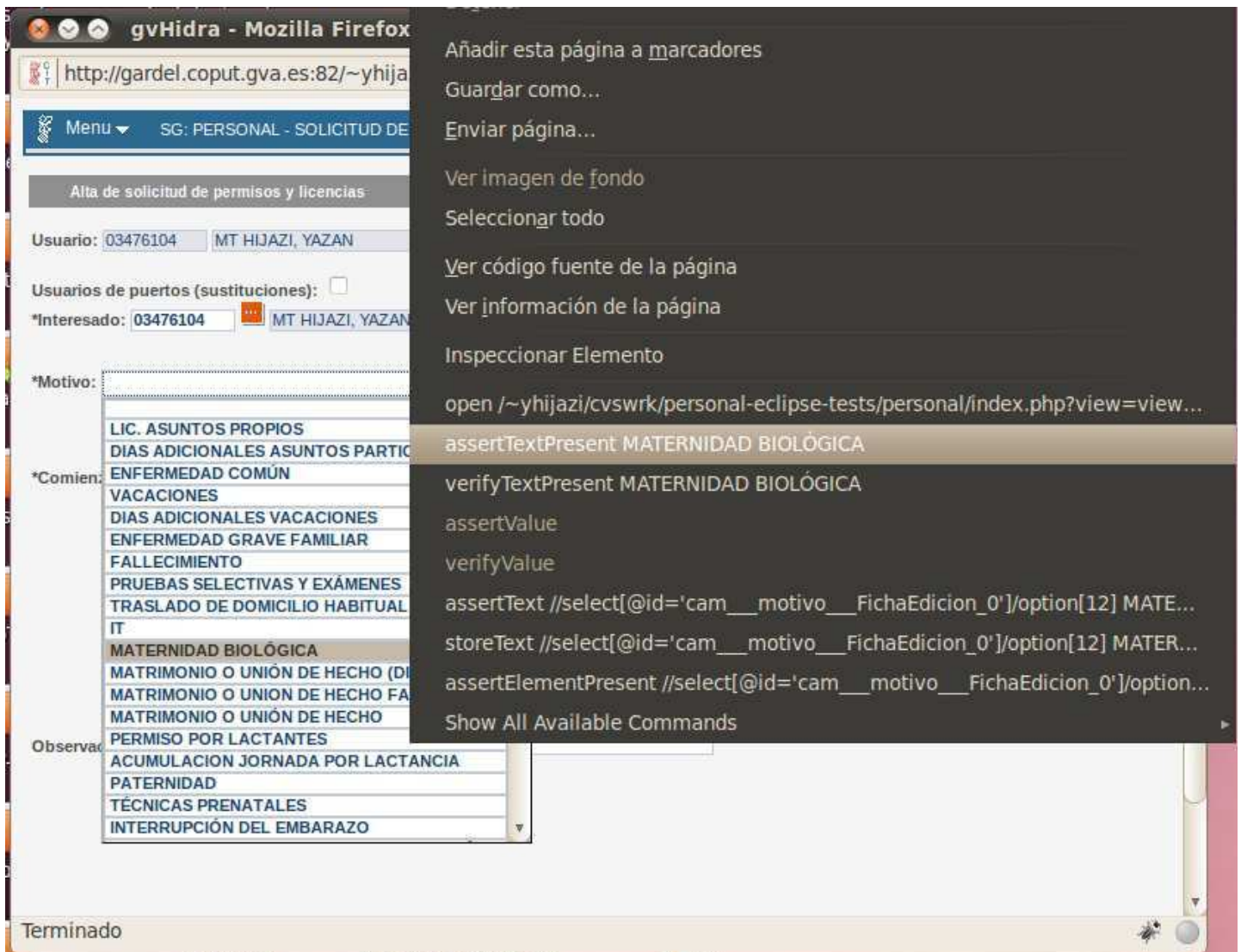
    return false;
}

public function connectToApplication() {
    include_once('igep/include_class.php');
    include_once('igep/include_all.php');
    $clasePrincipal = new AppMainWindow();

    //error global
    global $g_error;
    $g_error = new IgepError();
    $conf = ConfigFramework::getConfig();
}
}
?>

```

Para esta funcionalidad, simplemente comprobamos la presencia de las opciones mencionadas en la lista desplegable para el perfil de usuario “Tramitador” como se nos indica en la tabla:



Código para A7:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';
class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setPort(4445);
        $this->setHost("172.19.80.73");
        $this->setBrowser("**chrome");
        $this->setBrowserUrl("http://gardel.coput.gva.es/comun/comun/html/html/validacion.htm");
        $this->setSleep(7);
    }
}
```

```
}

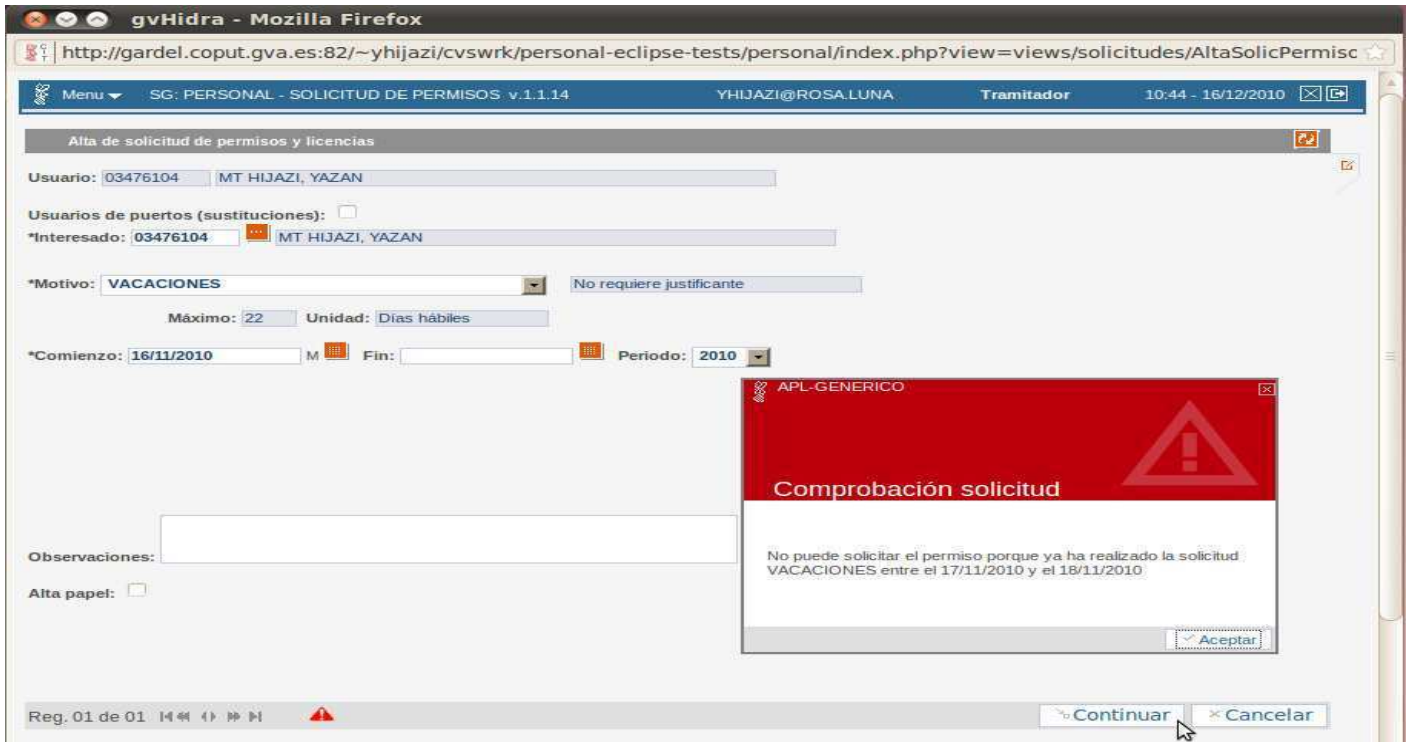
public function testMyTestCase()
{
    $this->open("/comun/comun/html/html/validacion.htm");
    $this->selectFrame("contenido");
    $this->type("as_usuario", "<usuario>");
    $this->type("as_clave", "<password>");
    $this->click("Aceptar");
    $this->waitForPageToLoad("30000");
    $this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
    $this->selectFrame("relative=up");
    $this->selectFrame("oculto");
    //$this->waitForPopUp("personal__W_PERDES", "30000");
    $this->openWindow("", "personal__W_PERDES");
    $this->selectWindow("name=personal__W_PERDES");
    $this->click("btnAceptar");
    $this->click("link=Permisos y licencias");
    $this->click("link=Alta");
    $this->assertTrue($this->isTextPresent("MATERNIDAD BIOLÓGICA"));
    $this->assertTrue($this->isTextPresent("FLEXIBILIDAD DE PERMANENCIA OBLIGATORIA"));
    $this->assertTrue($this->isTextPresent("IT"));
    $this->assertTrue($this->isTextPresent("REDUCC. HORARIO DE 9 A 14"));
    $this->assertTrue($this->isTextPresent("REDUCC. ENFERMEDAD MUY GRAVE"));
    $this->assertTrue($this->isTextPresent("REDUCC. 1H. DIARIA POR ENFERMEDAD"));
    $this->assertTrue($this->isTextPresent("REDUCC. DIARIA GUARDA LEGAL"));
    $this->assertTrue($this->isTextPresent("REDUCC. 1H. DIARIA POR MINUSVALÍA"));
    $this->assertTrue($this->isTextPresent("REDUCC. JORNADA DIARIA %"));
    $this->assertTrue($this->isTextPresent("REDUCC. POR VIOLENCIA SOBRE MUJER"));
    $this->focus("bnedi_cancelar");
    $this->click("bnedi_cancelar");
}
```

```

}
}
?>
    
```

A8

Introduciendo la fecha de inicio como se nos indica en la tabla y dejando la fecha de fin vacía, verificamos que se nos salta el mensaje de error:



A9

gvHidra - Mozilla Firefox

http://gardel.coput.gva.es:82/~yhijazi/cvswrk/personal-eclipse-tests/personal/index.php?view=views/solicitudes/AltaSolicPermisos.php&panel=editar

Menu SG: PERSONAL - SOLICITUD DE PERMISOS v.1.1.14 YHIJAZI@ROSA.LUNA Tramitador 10:39 - 16/12/2010

Alta de solicitud de permisos y licencias

Usuario: 03476104 MT HIJAZI, YAZAN

Usuarios de puertos (sustituciones):

*Interesado: 03476104 MT HIJAZI, YAZAN

*Motivo: LIC. PERFECCIONAMIENTO PROFESIONAL Requiere justificante

Máximo: 3 Unidad: Meses naturales

*Comienzo: 12/11/2010 Fin: 17/11/2010 Periodo: 2010

Observaciones:

Alta papel:

Reg. 01 de 01

Continuar Cancelar

Código para A8 y A9:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';
class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setPort(4445);
        $this->setHost("172.19.80.73");
        $this->setBrowser("*chrome");
        $this->setBrowserUrl("http://gardel.coput.gva.es/comun/comun/html/html/validacion.htm");
        $this->setSleep(7);
    }
    public function testMyTestCase()
    {
```

```

$this->open("/comun/comun/html/html/validacion.htm");
$this->selectFrame("contenido");
$this->type("as_usuario", "<usuario>");
$this->type("as_clave", "<password>");
$this->click("Aceptar");
$this->waitForPageToLoad("30000");
$this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
$this->selectFrame("relative=up");
$this->selectFrame("oculto");
//$this->waitForPopUp("personal__W_PERDES", "30000");
$this->openWindow("", "personal__W_PERDES");
$this->selectWindow("name=personal__W_PERDES");
$this->click("btnAceptar");
$this->click("link=Permisos y licencias");
$this->click("link=Alta");
$this->select("cam__motivo__FichaEdicion_0", "label=VACACIONES");
$this->click("//select[@id='cam__motivo__FichaEdicion_0']/option[5]");
$this->click("cam__fechalni__FichaEdicion_0");
$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");
//$this->click("cal_cam__fechalni__FichaEdicion_0");
$this->type("cam__fechalni__FichaEdicion_0", "22/11/2010");
//$this->click("//div[7]/table/tbody/tr[4]/td[2]");
$this->click("cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "");
$this->focus("bnsaltoConfirmar_saltaredi");
$this->click("bnsaltoConfirmar_saltaredi");
$this->assertTrue($this->isTextPresent("Dado el motivo de la solicitud indicado, la fecha de finalización es obligatoria"));
$this->click("btnAceptar");
$this->click("cam__fechalni__FichaEdicion_0");
$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");

```

```

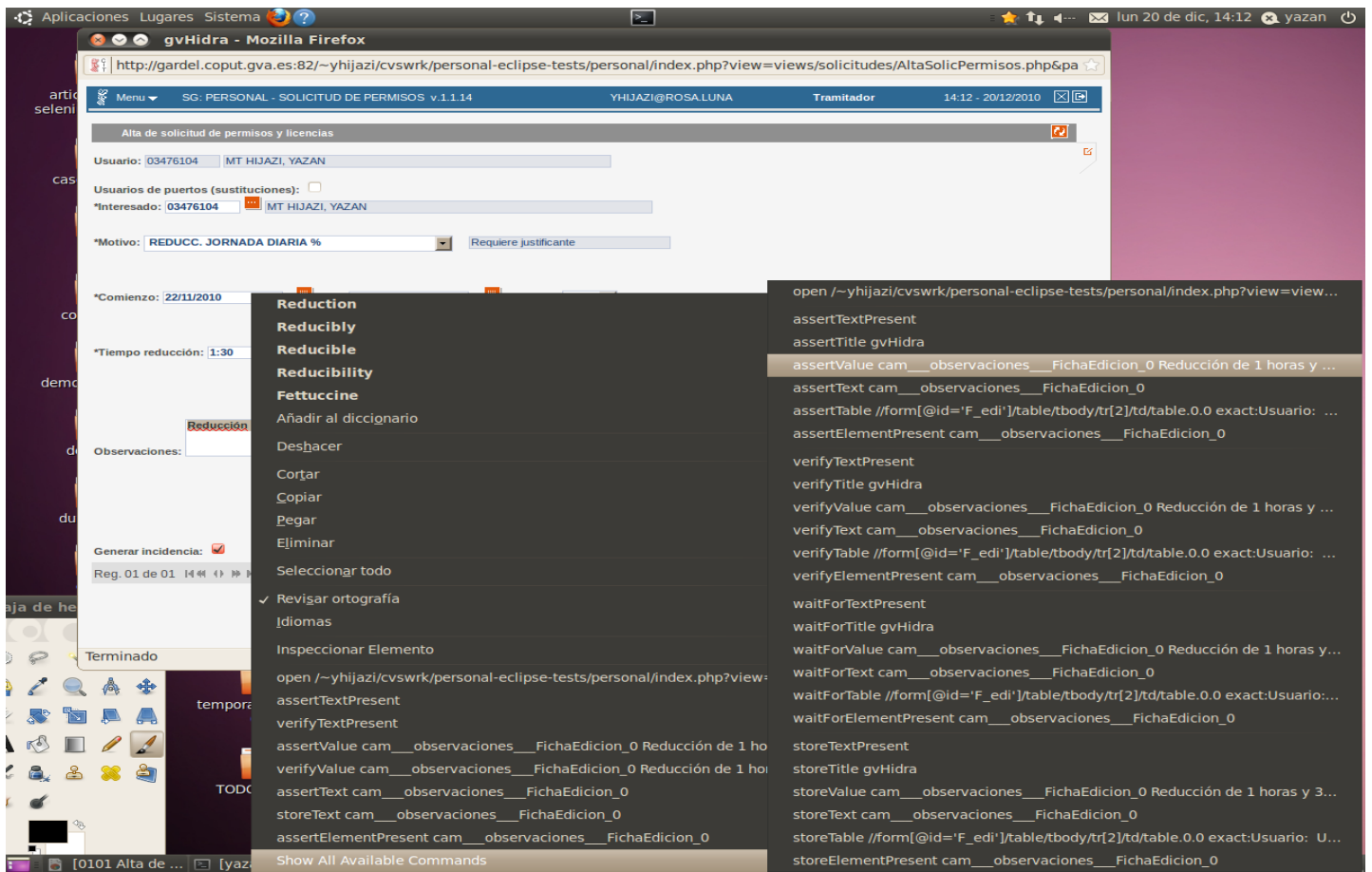
//$this->click("cal_cam__fechalni__FichaEdicion_0");
$this->type("cam__fechalni__FichaEdicion_0", "16/11/2010");
//$this->click("//div[11]/table/tbody/tr[3]/td[3]");
$this->click("cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "");
$this->focus("bnsaltoConfirmar_saltaredi");
$this->click("bnsaltoConfirmar_saltaredi");
$this->assertTrue($this->isTextPresent("No puede solicitar el permiso porque ya ha realizado la solicitud
VACACIONES entre el"));
$this->click("btnAceptar");

$this->select("cam__motivo__FichaEdicion_0", "label=LIC. PERFECCIONAMIENTO PROFESIONAL");
$this->click("cam__fechalni__FichaEdicion_0");
$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");
//$this->click("cal_cam__fechalni__FichaEdicion_0");
$this->type("cam__fechalni__FichaEdicion_0", "12/11/2010");
//$this->click("//tr[2]/td[6]");
$this->click("cam__fechaFin__FichaEdicion_0");
$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");
//$this->click("cal_cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "17/11/2010");
//$this->click("//div[3]/table/tbody/tr[3]/td[4]");
$this->focus("bnsaltoConfirmar_saltaredi");
$this->click("bnsaltoConfirmar_saltaredi");
$this->assertTrue($this->isTextPresent("No puede solicitar el permiso porque ya ha realizado la solicitud
VACACIONES entre el"));
$this->click("btnAceptar");
}
}
?>

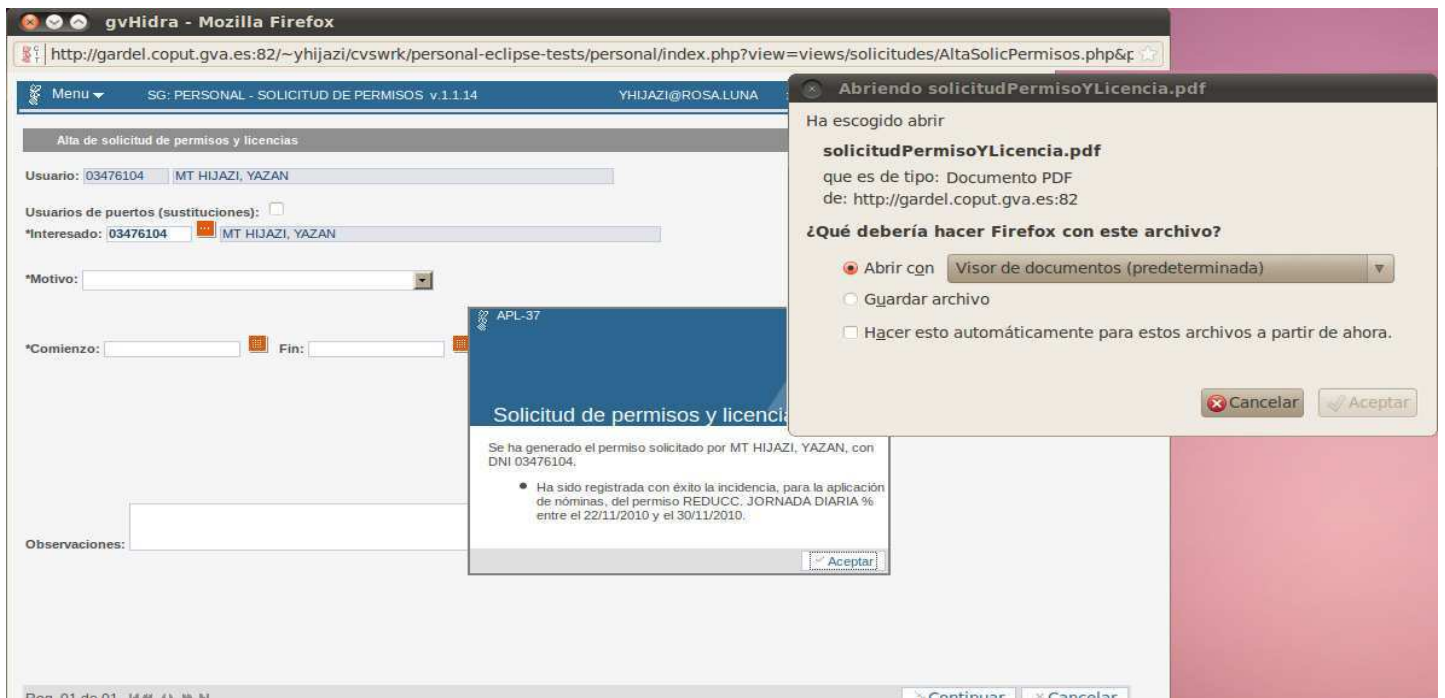
```

A10

Una vez introducidos las fechas indicadas comprobamos que se genero realmente el texto “Reduccion de 1 hora y 30 minutos” en el campo de texto de observaciones:

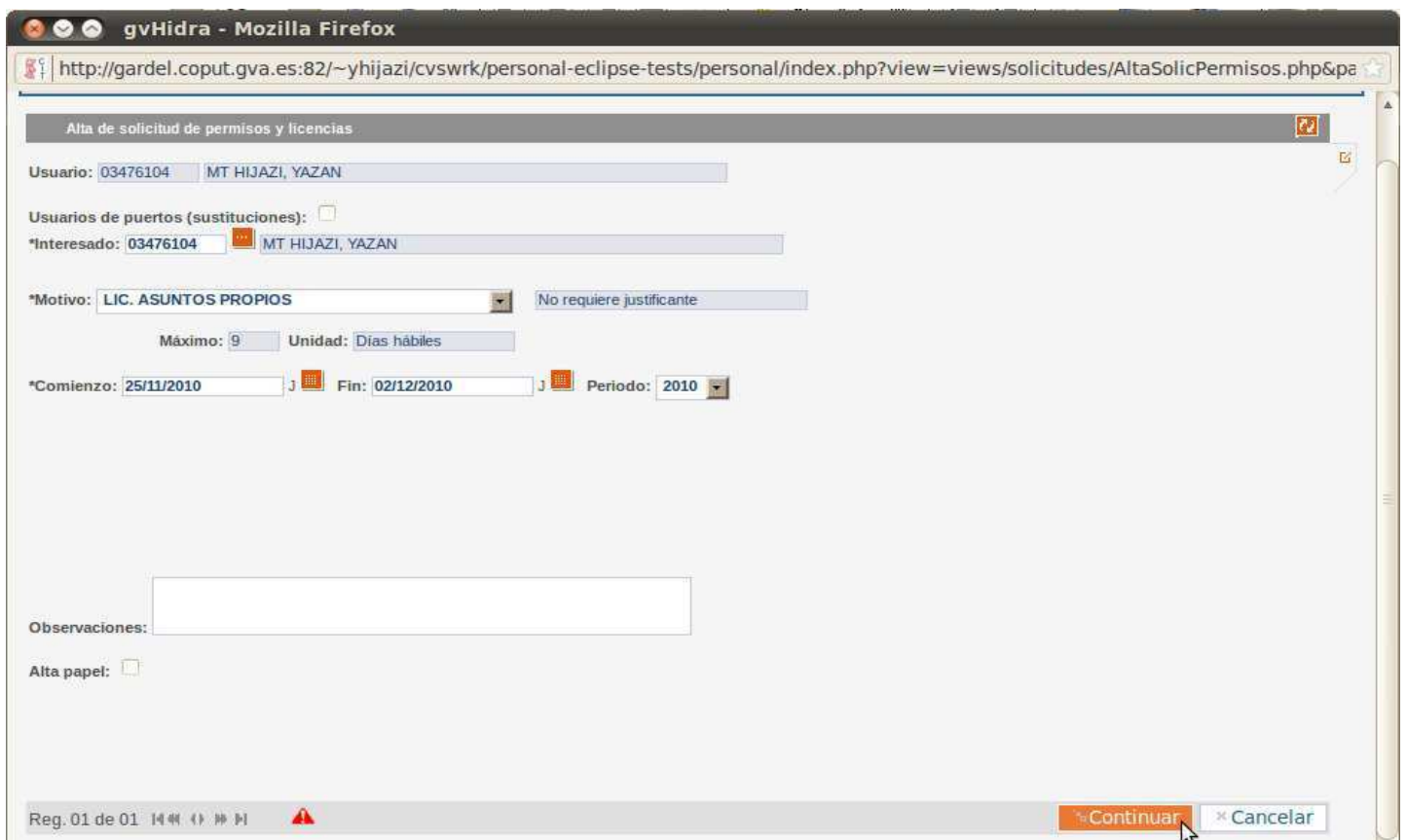


Y finalmente comprobamos el mensaje de confirmación:



A11

Se da de alta a la “solicitud de Lic. Asuntos propios” para las fechas indicadas:



Y se comprueba de nuevo el texto del mensaje de confirmación

Código para A10 y A11:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';
class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setPort(4445);
        $this->setHost("172.19.80.73");
        $this->setBrowser("*chrome");
        $this->setBrowserUrl("http://gardel.coput.gva.es/comun/comun/html/html/validacion.htm");
        $this->setSleep(7);
    }
    public function testMyTestCase()
    {
        $this->open("/comun/comun/html/html/validacion.htm");
        $this->selectFrame("contenido");
        $this->type("as_usuario", "<usuario>");
        $this->type("as_clave", "<password>");
        $this->click("Aceptar");
        $this->waitForPageToLoad("30000");
        $this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
        $this->selectFrame("relative=up");
        $this->selectFrame("oculto");
        //$this->waitForPopUp("personal__W_PERDES", "30000");
        $this->openWindow("", "personal__W_PERDES");
        $this->selectWindow("name=personal__W_PERDES");
        $this->click("btnAceptar");
        $this->click("link=Permisos y licencias");
        $this->click("link=Alta");
        $this->select("cam__motivo__FichaEdicion_0", "label=REDUCC. JORNADA DIARIA %");
        $this->click("cam__fechalni__FichaEdicion_0");
    }
}
```

```

$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");
//$this->click("cal_cam__fechalni__FichaEdicion_0");
$this->type("cam__fechalni__FichaEdicion_0", "22/11/2010");
$this->click("cam__fechaFin__FichaEdicion_0");
$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");
//$this->click("cal_cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "30/11/2010");
//$this->click("//div[21]/table/tbody/tr[5]/td[3]");
$this->click("cam__horas__FichaEdicion_0");
$this->type("cam__horas__FichaEdicion_0", "1:30");
//$this->click("cam__observaciones__FichaEdicion_0");
$this->focus("bnsaltoConfirmar_saltaredi");
$this->assertEquals("Reducción de 1 horas y 30 minutos.", $this-
>getValue("cam__observaciones__FichaEdicion_0"));
$this->click("bnsaltoConfirmar_saltaredi");
$this->focus("bnedi_particular");
$this->click("bnedi_particular");
//$this->waitForPopUp("ventana", "30000");
$this->windowFocus();
$this->assertTrue($this->isTextPresent("Se ha generado el permiso solicitado por"));
$this->assertTrue($this->isTextPresent("Ha sido registrada con éxito la incidencia, para la aplicación de nóminas, del
permiso REDUCC. JORNADA DIARIA % entre el"));
$this->click("btnAceptar");
$this->select("cam__motivo__FichaEdicion_0", "label=LIC. ASUNTOS PROPIOS");
$this->click("//select[@id='cam__motivo__FichaEdicion_0']/option[2]");
$this->click("cam__fechalni__FichaEdicion_0");
$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");
//$this->click("cal_cam__fechalni__FichaEdicion_0");
$this->type("cam__fechalni__FichaEdicion_0", "25/11/2010");
//$this->click("//tr[4]/td[5]");
$this->click("cam__fechaFin__FichaEdicion_0");

```

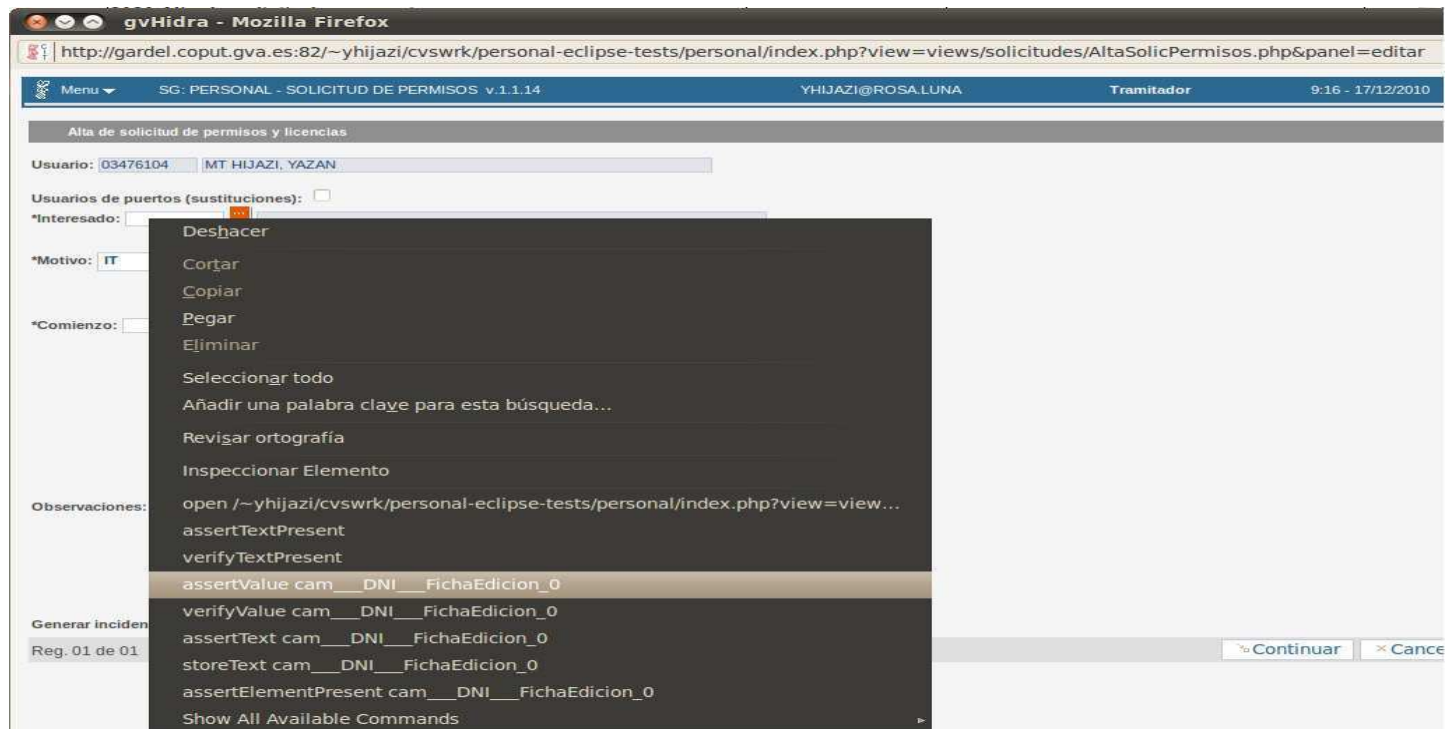
```

$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");
//$this->click("cal_cam__fechaFin__FichaEdicion_0");
//$this->click("//div[3]/table/thead/tr[2]/td[4]/div");
$this->type("cam__fechaFin__FichaEdicion_0", "02/12/2010");
//$this->click("//div[3]/table/tbody/tr[1]/td[5]");
$this->focus("bnsaltoConfirmar_saltaredi");
$this->click("bnsaltoConfirmar_saltaredi");
$this->focus("bnedi_particular");
$this->click("bnedi_particular");
$this->assertTrue($this->isTextPresent("se ha generado con el nº"));
$this->click("btnAceptar");
}
}
?>

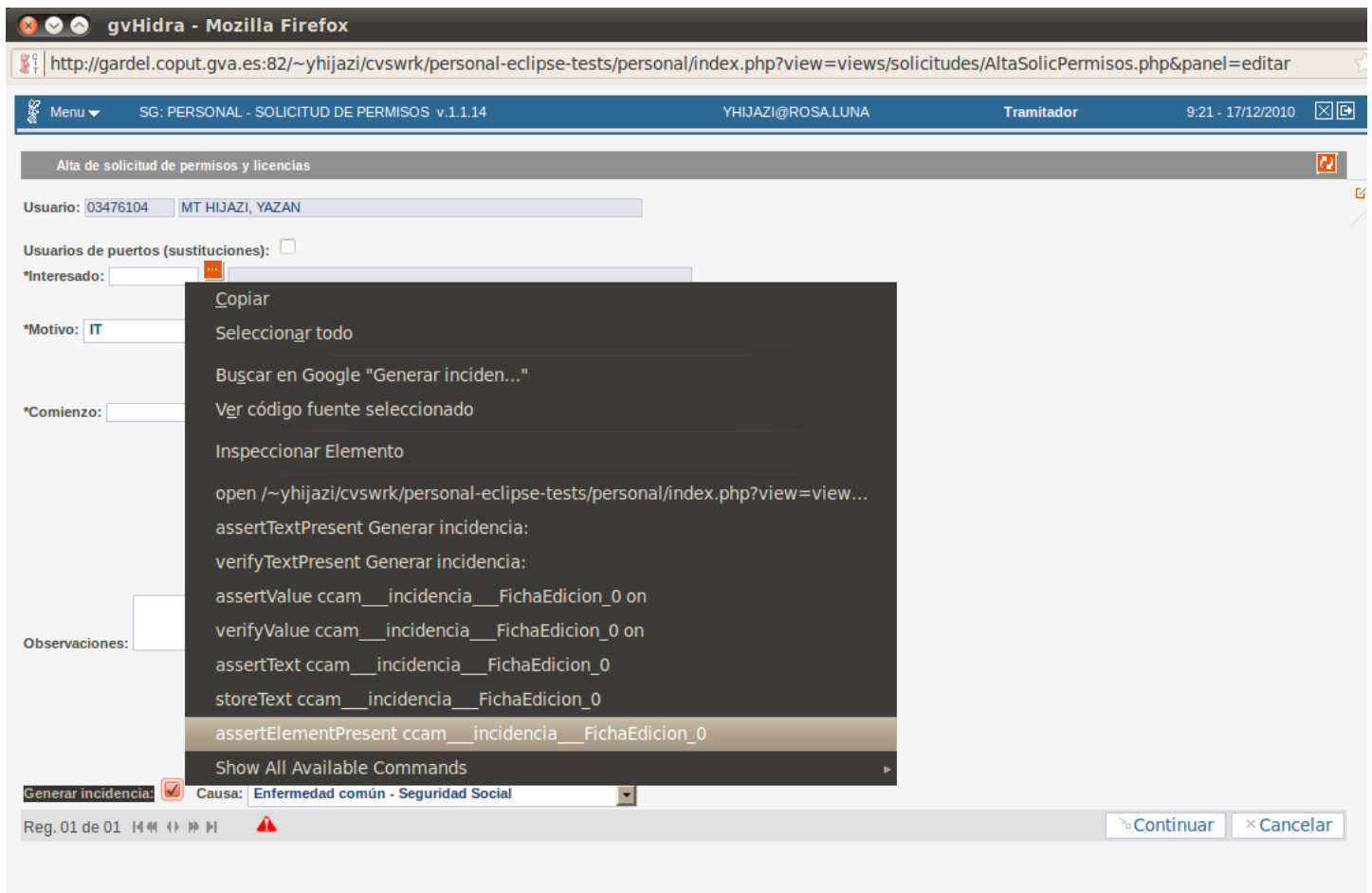
```

A12

Al seleccionar el motivo IT comprobamos que el campo “Interesado” esta vacio como nos indican en la tabla del caso de uso:



y también verificamos, con la llamada al método “assertElementPresent”, la existencia del checkbox “Generar Incidencia” y de la lista “causa”:



A13

Se da de alta, y se realiza la comprobación de la misma forma que en los casos anteriores.

Código para A12 y A13:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';
class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setPort(4445);
        $this->setHost("172.19.80.73");
        $this->setBrowser("*chrome");
        $this->setBrowserUrl("http://gardel.coput.gva.es/comun/comun/html/html/validacion.htm");
        $this->setSleep(7);
    }
}
```

```
public function testMyTestCase()
{
    $this->open("/comun/comun/html/html/validacion.htm");
    $this->selectFrame("contenido");
    $this->type("as_usuario", "<usuario>");
    $this->type("as_clave", "<password>");
    $this->click("Aceptar");
    $this->waitForPageToLoad("30000");
    $this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
    $this->selectFrame("relative=up");
    $this->selectFrame("oculto");
    //$this->waitForPopUp("personal__W_PERDES", "30000");
    $this->openWindow("", "personal__W_PERDES");
    $this->selectWindow("name=personal__W_PERDES");
    $this->click("btnAceptar");
    $this->click("link=Permisos y licencias");
    $this->click("link=Alta");
    $this->click("cam__motivo__FichaEdicion_0");
    $this->select("cam__motivo__FichaEdicion_0", "label=IT");
    $this->assertEquals("", $this->getValue("cam__DNI__FichaEdicion_0"));
    $this->assertTrue($this->isElementPresent("ccam__incidencia__FichaEdicion_0"));
    $this->assertTrue($this->isElementPresent("cam__causa__FichaEdicion_0"));
    $this->assertEquals("", $this->getValue("cam__DNI__FichaEdicion_0"));
    $this->assertTrue($this->isTextPresent("Generar incidencia:"));
    $this->assertTrue($this->isElementPresent("ccam__incidencia__FichaEdicion_0"));
    $this->assertTrue($this->isElementPresent("cam__causa__FichaEdicion_0"));
    $this->assertTrue($this->isTextPresent("Enfermedad común - Seguridad Social"));
    $this->assertTrue($this->isTextPresent("Enfermedad común - Muface"));
    $this->assertTrue($this->isTextPresent("Accidente trabajo/Enf. profesional - Seguridad Social"));
    $this->assertTrue($this->isTextPresent("Accidente trabajo/Enf. profesional - Muface"));
    $this->assertTrue($this->isTextPresent("Accidente no laboral - Seguridad Social"));
}
```

```

$this->assertTrue($this->isTextPresent("Accidente no laboral - Muface"));
$this->focus("bnedi_cancelar");

$this->click("bnedi_cancelar");

$this->click("link=Permisos y licencias");

$this->click("link=Alta");

$this->select("cam__motivo__FichaEdicion_0", "label=REDUCC. 1H. DIARIA POR MINUSVALÍA");

$this->assertTrue($this->isElementPresent("cam__tipo__FichaEdicion_0"));

$this->assertTrue($this->isTextPresent("Padres"));

$this->assertTrue($this->isTextPresent("Hijos"));

$this->assertTrue($this->isTextPresent("Propios"));

$this->assertTrue($this->isTextPresent("Otros"));

$this->focus("bnedi_cancelar");

$this->click("bnedi_cancelar");

}
}
?>

```

A15

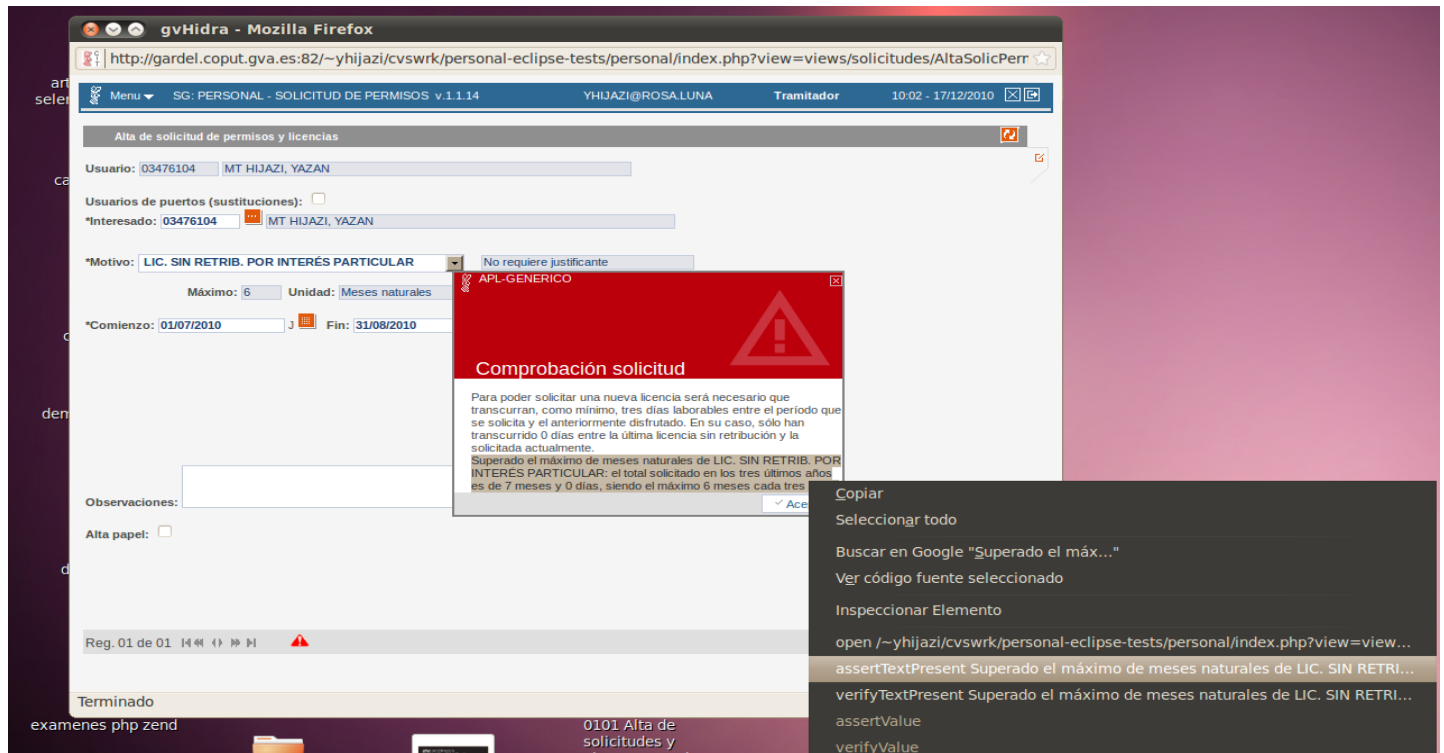
The screenshot shows a web browser window titled 'gvHidra - Mozilla Firefox' with the URL 'http://gardel.coput.gva.es:82/~yhijazi/cvswrk/personal-eclipse-tests/personal/index.php?view=views/solicitudes/AltaSolicPermisos.php&panel=editar'. The page header includes 'Menu', 'SG: PERSONAL - SOLICITUD DE PERMISOS v.1.1.14', 'YHIJAZI@ROSA.LUNA', 'Tramitador', and '9:47 - 17/12/2010'. The main content area is titled 'Alta de solicitud de permisos y licencias' and contains the following fields:

- Usuario: 03476104 MT HIJAZI, YAZAN
- Usuarios de puertos (sustituciones):
- *Interesado: 03476104 MT HIJAZI, YAZAN
- *Motivo: LIC. SIN RETRIB. POR INTERÉS PARTICULAR (No requiere justificante)
- *Comienzo: 01/09/2009 M Fin: 31/10/2009 S Periodo: 2009
- Observaciones:
- Alta papel:

At the bottom right, there are buttons for 'Continuar' and 'Cancelar'. The footer shows 'Reg. 01 de 01' and navigation icons.

A16 y A17

se realizan de igual manera que A15 pero cambiando las fechas, y verificando el error que salta en el caso de A17:



Código para A15, A16 y A17:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';
class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setPort(4445);
        $this->setHost("172.19.80.73");
        $this->setBrowser("**chrome");
        $this->setBrowserUrl("http://gardel.coput.gva.es/comun/comun/html/html/validacion.htm");
        $this->setSleep(7);
    }
    public function testMyTestCase()
    {
```



```

$this->open("/comun/comun/html/html/validacion.htm");
$this->selectFrame("contenido");
$this->type("as_usuario", "<usuario>");
$this->type("as_clave", "<password>");
$this->click("Aceptar");
$this->waitForPageToLoad("30000");
$this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
$this->selectFrame("relative=up");
$this->selectFrame("oculto");
//$this->waitForPopUp("personal__W_PERDES", "30000");
$this->openWindow("", "personal__W_PERDES");
$this->selectWindow("name=personal__W_PERDES");
$this->click("btnAceptar");
$this->click("link=Permisos y licencias");
$this->click("link=Alta");
$this->select("cam__motivo__FichaEdicion_0", "label=LIC. SIN RETRIB. POR INTERÉS PARTICULAR");
$this->click("cam__fechalni__FichaEdicion_0");
$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");
//$this->click("cal_cam__fechalni__FichaEdicion_0");
$this->type("cam__fechalni__FichaEdicion_0", "01/09/2009");
//$this->click("//div[2]/table/tbody/tr[1]/td[3]");
$this->click("cam__fechaFin__FichaEdicion_0");
$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");
//$this->click("cal_cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "31/10/2009");
//$this->click("//div[3]/table/tbody/tr[5]/td[7]");
$this->select("cam__periodo__FichaEdicion_0", "label=2009");
$this->click("//select[@id='cam__periodo__FichaEdicion_0']/option[2]");
$this->focus("bnsaltoConfirmar_saltaredi");
$this->click("bnsaltoConfirmar_saltaredi");
$this->focus("bnedi_particular");

```

```

$this->click("bnedi_particular");

//$this->waitForPopUp("ventana", "30000");
$this->windowFocus();

$this->assertTrue($this->isTextPresent("se ha generado con el nº"));

$this->click("btnAceptar");

$this->select("cam__motivo__FichaEdicion_0", "label=LIC. SIN RETRIB. POR INTERÉS PARTICULAR");

$this->click("//select[@id='cam__motivo__FichaEdicion_0']/option[25]");

$this->click("cam__fechalni__FichaEdicion_0");

$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");

//$this->click("cal_cam__fechalni__FichaEdicion_0");

//$this->click("//tr[2]/td[2]/div");

$this->type("cam__fechalni__FichaEdicion_0", "01/04/2010");

//$this->click("//div[2]/table/tbody/tr[1]/td[5]");

$this->click("cam__fechaFin__FichaEdicion_0");

$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");

//$this->click("cal_cam__fechaFin__FichaEdicion_0");

$this->type("cam__fechaFin__FichaEdicion_0", "30/06/2010");

//$this->click("//div[3]/table/tbody/tr[5]/td[4]");

$this->focus("bnsaltoConfirmar_saltaredi");

$this->click("bnsaltoConfirmar_saltaredi");

$this->focus("bnedi_particular");

$this->click("bnedi_particular");

//$this->waitForPopUp("ventana", "30000");

$this->windowFocus();

$this->click("btnAceptar");

$this->select("cam__motivo__FichaEdicion_0", "label=LIC. SIN RETRIB. POR INTERÉS PARTICULAR");

$this->click("cam__fechalni__FichaEdicion_0");

$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");

//$this->click("cal_cam__fechalni__FichaEdicion_0");

$this->type("cam__fechalni__FichaEdicion_0", "01/07/2010");

//$this->click("//div[2]/table/tbody/tr[1]/td[5]");

```

```

$this->click("cam___fechaFin___FichaEdicion_0");

$this->doubleClick("cal_cam___fechaFin___FichaEdicion_0");

//$this->click("cal_cam___fechaFin___FichaEdicion_0");

//$this->click("//div[3]/table/thead/tr[2]/td[4]/div");

$this->type("cam___fechaFin___FichaEdicion_0", "31/08/2010");

//$this->click("//div[3]/table/tbody/tr[6]/td[3]");

$this->focus("bnsaltoConfirmar_saltaredi");

$this->click("bnsaltoConfirmar_saltaredi");

$this->assertTrue($this->isTextPresent("Superado el máximo de meses naturales de LIC. SIN RETRIB. POR
INTERÉS PARTICULAR: el total solicitado en los tres últimos años es de 7 meses y 0 días, siendo el máximo 6 meses
cada tres años.));

$this->click("btnAceptar");

$this->focus("bnedi_cancelar");

$this->click("bnedi_cancelar");

}

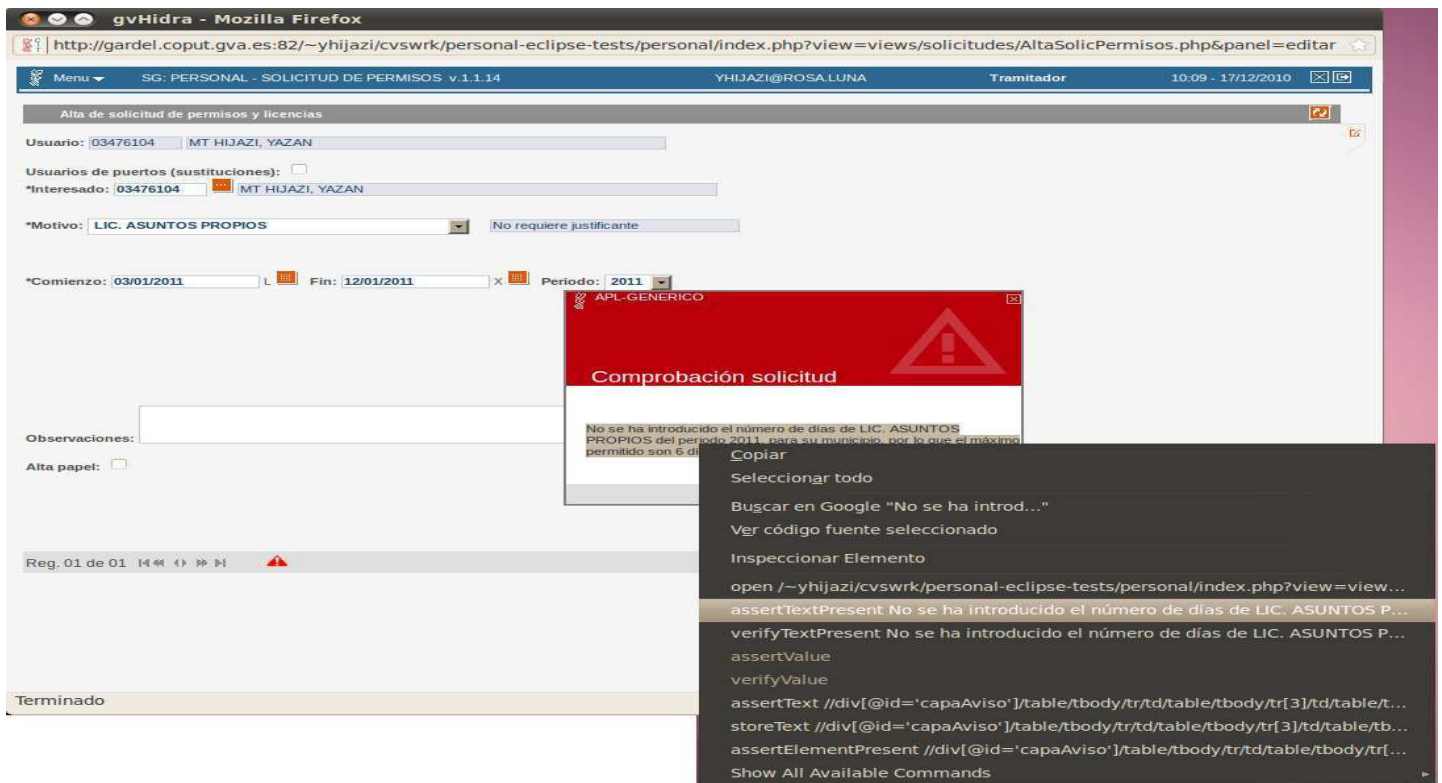
}

?>

```

A18

Para esta funcionalidad, además de la comprobación de la presencia del texto de error, se requiere una verificación de la base de datos insertando la sentencia de consulta select apropiada, y utilizando los mismos métodos descritos en los casos A6 y A6.1:



Código para A18:

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';

class Example extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setPort(4445);
        $this->setHost("172.19.80.73");
        $this->setBrowser("*chrome");
        $this->setBrowserUrl("http://gardel.coput.gva.es/comun/comun/html/html/validacion.htm");
        $this->setSleep(7);
        $this->connectToApplication();
    }

    public function testMyTestCase()
    {
        $this->open("/comun/comun/html/html/validacion.htm");
        $this->selectFrame("contenido");
    }
}
```

```

$this->type("as_usuario","<usuario>");
$this->type("as_clave", "<password>");
$this->click("Aceptar");
$this->waitForPageToLoad("30000");
$this->click("link=Personal - Solicitudes de permisos y licencias (carpeta de cada uno)");
$this->selectFrame("relative=up");
$this->selectFrame("oculto");
//$this->waitForPopUp("personal__W_PERDES", "30000");
$this->openWindow("", "personal__W_PERDES");
$this->selectWindow("name=personal__W_PERDES");
$this->click("btnAceptar");
$this->click("link=Permisos y licencias");
$this->click("link=Alta");
$this->select("cam__motivo__FichaEdicion_0", "label=LIC. ASUNTOS PROPIOS");
$this->click("cam__fechalni__FichaEdicion_0");
$this->doubleClick("cal_cam__fechalni__FichaEdicion_0");
//$this->click("cal_cam__fechalni__FichaEdicion_0");
//$this->click("//td[5]/div");
$this->type("cam__fechalni__FichaEdicion_0", "03/01/2011");
//$this->click("//tbody/tr[2]/td[2]");
$this->click("cam__fechaFin__FichaEdicion_0");
$this->doubleClick("cal_cam__fechaFin__FichaEdicion_0");
//$this->click("cal_cam__fechaFin__FichaEdicion_0");
$this->type("cam__fechaFin__FichaEdicion_0", "12/01/2011");
//$this->click("//div[3]/table/tbody/tr[3]/td[4]");
$this->select("cam__periodo__FichaEdicion_0", "label=2011");
$this->click("//select[@id='cam__periodo__FichaEdicion_0']/option[3]");
$this->focus("bnsaltoConfirmar_saltaredi");
$this->click("bnsaltoConfirmar_saltaredi");
$this->assertTrue($this->isTextPresent("No se ha introducido el número de días de LIC. ASUNTOS PROPIOS del periodo 2011, para su municipio, por lo que el máximo permitido son 6 días."));

```

```

$this->click("btnAceptar");

//Comprobación integridad de la BBDD.

$this->assertTrue($this->checkInsert());
}

private function checkInsert() {
    $conf = ConfigFramework::getConfig();
    $dsn = $conf->getDSN('g_dsn');
    $con = new IgepConexion($dsn);
    $str_select = <<<querySelect
select CASE
WHEN max(maxdias) IS NULL THEN 0
ELSE max(maxdias)
END as diasMaximo
from TPER_PERMISOSMAXDIASLOCAL
where cclase= 'VV6' and
periodo = 2011 and
(cmun = '014' or ('03' = '46' and cmun='250') or ('03' = '12' and cmun='040')
or ('03' = '03' and cmun='014')) and Cpro = '03';
querySelect;
    $res = $con->consultar("$str_select");
    if($res[0]['diasMaximo']==0)
        return true;
    return false;
}

public function connectToApplication() {
    include_once('igep/include_class.php');
    include_once('igep/include_all.php');
    $clasePrincipal = new AppMainWindow();

    //error global
    global $g_error;
    $g_error = new IgepError();
}

```

```
$conf = ConfigFramework::getConfig();  
}  
}  
?>
```

6. Pruebas y Resultados

6.1. Lanzamiento de los tests

Una vez generados los archivos de prueba y modificados siguiendo los patrones explicados en el apéndice “Patrones para la adaptación de los tests”, se guardan esos archivos en el directorio tests de la aplicación. Por ejemplo en nuestro caso la ruta sería: “<dir>/personal/tests”, donde <dir> suele ser la ruta hacia el directorio público de la instalación de apache que tenemos en el servidor, por ejemplo: “/usr/local/httpd-2.2.8/htdocs/”.

A continuación, se pasa a crear las tareas en hudson que determinan los pasos de prueba que se van a ejecutar cuando esas tareas sean lanzadas. Además debemos crear y/o especificar los nodos (maquinas locales o remotas), que van a soportar esa ejecución.

Para preparar el entorno descrito, seguimos los pasos del anexo de este proyecto “Instalación y manejo de PHPUnit, Selenium y Hudson”, creando tantas tareas como archivos de tests habíamos generado, aunque este número puede ser distinto dependiendo del grado de paralelismo que deseamos obtener y que depende a su vez también de la relación y la dependencia entre los tests y del orden correcto en el que estos deben ser ejecutados.

En nuestro ejemplo hemos optado por incluir las pruebas muy relacionadas entre ellas en un mismo archivo.

El resto de las dependencias se controla configurando adecuadamente el orden y la periodicidad del lanzamiento de cada tarea, y eso se consigue fácilmente siguiendo el orden especificado en la tabla del caso de uso que nos suministran, y colocando las tareas temporalmente una detrás de otra para garantizar que el estado de la base de datos antes de ejecutar cada prueba vaya a ser el adecuado.

Incluimos aquí las capturas para las tres primeras tareas (archivos **A1-A2.php**, **A3-A4.php** y **A6-A6-1.php**), el resto se configuran de forma similar apoyándonos en el manual de hudson del anexo de este proyecto:

The screenshot shows the Hudson web interface in a Mozilla Firefox browser. The main content area displays a table of jobs with the following data:

S	W	Tarea ↓	Último éxito	Último fallo	Última duración	Console
●		personal-A1-A2	N/D	N/D	N/D	N/A
●		personal-A3-A4	N/D	N/D	N/D	N/A
●		personal-A6	N/D	N/D	N/D	N/A

On the left side, under 'Trabajos en la cola', it shows 'No hay tareas pendientes'. Under 'Estado de los nodos', it shows 'Master 0/2' and 'nodo-gardel fuera de linea'.

Dependiendo de la maquina donde residirán los tests y el binario de PHPUnit, tendríamos que crear o no (siguiendo el manual del anexo) un nodo en hudson que represente esta maquina. En nuestro ejemplo lo llamamos “nodo-gardel” y es el que se ve en la parte izquierda inferior de la imagen. Aunque también podemos mantener toda la funcionalidad en el “localhost” si la aplicación reside en nuestra maquina local, y no tendríamos necesidad de crear nodos extra.

Las líneas de comando que ejecutaran esas tareas son casi las mismas (solamente se diferencian por el nombre del fichero de prueba).

Para el primer test (fichero A1-A2.php) las líneas de comandos serian:

```
cd /home/yazan/mnt/gardel/public_html/cvswrk/personal-eclipse-tests/personal/  
phpunit Example tests/A1-A2.php
```

La primera línea es necesaria para colocarnos en el directorio de la aplicación, y en la segunda colocamos en la invocación de phpunit el nombre de la clase principal del fichero .php que contiene el tests (por defecto Selenium la llama "Example").

A continuación consultamos el manual de Selenium y Hudson para establecer la conexión entre el nodo principal del servidor de hudson y el nodo donde se ejecuta phpunit (no hace falta hacer esto si todo esta en localhost). También arrancamos la herramienta Selenium-Server dentro de la maquina es donde se va a abrir el navegador y reproducir el test.

```
java -jar selenium/selenium-remote-control-1.0.3/selenium-server-1.0.3/selenium-server.jar -port 4445
```

En nuestro ejemplo esta en 172.19.80.73 y el puerto es el 4445, estos dos valores son los que se utilizan dentro del código del método setUp del test:

```
$this->setPort(4445);  
$this->setHost("172.19.80.73");
```

En este momento estamos preparados para lanzar las tareas.

6.2. Los Resultados

Una vez concluida correctamente la ejecución de las tareas que lanzan los tests en el orden correcto, y que se deduce a partir de la tabla del caso de uso, tendríamos la pagina principal de Hudson con el siguiente aspecto (solo incluimos tres pruebas como ejemplo, el resto se realiza igual):

Panel de control [Hudson] - Mozilla Firefox
 Archivo Editar Ver Historial Marcadores Herramientas Ayuda
 http://localhost:8080/

Hudson

ACTIVAR AUTO REFRESCO

[Crear nueva Tarea](#)
[Administrar Hudson](#)
[Actividad](#)
[Historia de ejecuciones](#)

Trabajos en la cola
 No hay tareas pendientes

Estado de los nodos
 Master 0/2
 Idle

S	W	Tarea ↓	Último éxito	Último fallo	Última duración	Console
		personal-A1-A2	2 Min 36 Seg (#1)	N/D	0,68 Seg	
		personal-A3-A4	23 Seg (#1)	N/D	0,14 Seg	
		personal-A6	17 Seg (#1)	N/D	0,13 Seg	

Icono: [S](#) [M](#) [L](#)

Suscribirse a RSS de: [todos los trabajos](#) [sólo los fallidos](#) [los más recientes](#)

como podemos apreciar en la imagen, todas las tareas se han ejecutado correctamente y sin errores. En caso de que se produzca un error (por ejemplo ejecutando equivocadamente una segunda vez la primera prueba), tenemos la posibilidad de revisar el historial de ejecuciones, y las respuestas producidos por la consola, y que nos indican en que momento se ha producido el error ademas de la cause.

[Insertar descripción](#)

S	W	Tarea ↓	Último éxito	Último fallo	Última duración	Console
		personal-A1-A2	8 Min 21 Seg (#1)	11 Seg (#2)	0,68 Seg	
		personal-A3-A4	6 Min 9 Seg (#1)	N/D	0,14 Seg	

[Volver al Proyecto](#)
[Estado](#)
[Cambios](#)
[Build Now](#)
[Salida de consola](#)
[Configure](#)
[Ejecucion previa](#)

Salida de consola

```
Lanzada por el usuario anonymous
Ejecutando en el nodo principal
[workspace] $ /bin/sh -xe /tmp/hudson4317983613326482153.sh
+ cd /home/yazan/mnt/gardel/public_html/cvswrk/personal-eclipse-tests/personal/
cd: 1: can't cd to /home/yazan/mnt/gardel/public_html/cvswrk/personal-eclipse-tests/personal/
Sending e-mails to: hijazi_yaz@gva.es
[DEBUG] Skipping watched dependency update for build: personal-A1-A2 #5 due to result: FAILURE
Finished: FAILURE
```

[Mostrar en texto plano](#)

Además, vemos que nos ha llegado un correo electrónico con los detalles del problema, ya que en este ejemplo se había configurado la tarea para que se comuniquen los errores por correo electrónico a los responsables de construir y/o corregir los tests:

join us for MySQL webcasts in August Oracle MySQL 11/06/11 17:10
La ejecución en Hudson ha fallado: personal-A1-A2 #5 hijazi_yaz@gva.es 13:39

from You ☆ reply forward archive junk delete
subject La ejecución en Hudson ha fallado: personal-A1-A2 #5 13:39
to You ☆ other actions ▾

Echa un vistazo a <http://localhost:8080/job/personal-A1-A2/5/>

Lanzada por el usuario anonymous
Ejecutando en el nodo principal
[workspace] \$ /bin/sh -xe /tmp/hudson4317983613326482153.sh
+ cd /home/yazan/mnt/gardel/public_html/cvswrk/personal-eclipse-tests/personal/
cd: 1: can't cd to /home/yazan/mnt/gardel/public_html/cvswrk/personal-eclipse-tests/personal/

7. Conclusiones

A partir del estudio de los principales elementos teóricos sobre Integración Continua, se sentaron las bases para la elaboración de la propuesta y el estudio de algunas herramientas que permiten crear entornos de Integración Continua para diferentes lenguajes de programación. Este estudio proporciona datos valiosos para aquellos proyectos que deseen implantarlo.

7.1. Aportaciones realizadas

Aplicando el proceso de Integración Continua siguiendo los conceptos y las practicas propuestas de este proyecto hemos conseguido lo siguiente:

- 1) descubrir la mayor parte de los errores del sistema en el mismo día en el que se integran los módulos intervinientes.
- 2) reducir drásticamente los tiempos para determinar el problema y resolverlo.
- 3) introducir el uso de las herramientas Selenium y PHPUnit que conjuntamente nos han sido de gran ayuda para poder generar y ejecutar los tests de manera rápida y sencilla.
- 4) Aplicar el proceso de integración continua al framework gvHIDRA que, a su vez, es un entorno de desarrollo rápido de las aplicaciones, consiguiendo en la totalidad, aplicaciones PHP completas y probadas en menos tiempo que en el desarrollo convencional.

7.2. Problemas Encontrados y Mantenimiento Futuro

El problema básico con el que nos hemos encontrado a la hora de utilizar Selenium con gvHIDRA ha sido detectar las acciones generadas por Selenium y que teníamos que modificar manualmente y/o controlar su momento de ejecución adecuado (colocando esperas cuando sea necesario), para que la ejecución de las pruebas se realice correctamente.

Para resolver esto, se hicieron muchas pruebas de los diferentes componentes del framework por separado para determinar en cuales de ellos se suele manifestar ese problema, y cuales son las modificaciones que habría que realizar para solucionarlo, elaborando finalmente un Apéndice en el que se enumeran todas esas situaciones para que el usuario vaya aplicando las que sean necesarias en un momento dado.

Con respecto al mantenimiento del servidor de Hudson, la principal función de la persona responsable de ello sera crear y configurar correctamente los nodos y las tareas para ejecutar los ficheros de prueba y en el orden y la periodicidad correcta que se le indica.

Determinar ese orden sera responsabilidad de la persona que construye el archivo de prueba basándose en la tabla del caso de uso y prestando especial atención a las pruebas que tienen que ejecutarse secuencialmente y diferenciarlas de las que podemos lanzar en paralelo.

Normalmente esa tarea no suele ser difícil para la persona que desarrolla o participa en el desarrollo de la aplicación, y por ese motivo, también recomendamos en este proyecto(aunque no es preciso), que sea esa persona que construya sus propios tests, debido a que suele tener un conocimiento bastante amplio del código, Y de esta manera conseguiríamos generar los tests en un tiempo bastante menor, especialmente si configuramos las tareas en hudson que ejecutan esas pruebas para que el resultado de la ejecución se pueda

comunicar (como hemos visto anteriormente) por correo electrónico a ese desarrollador para que se encargue de resolver los errores de su propio código

8. Bibliografía

- [1]. Fowler, M. (2006, Mayo 1). Martin Fowler. Retrieved Septiembre 17, 2009, from <http://martinfowler.com/articles/continuousIntegration.html>
- [2]. Bartlett, D. (2009). 97 Things Every Software Architect Should Know.
- [3]. web de gvHIDRA: <http://www.gvpontis.gva.es/cast/gvhidra-herramienta/>
- [4]. PHPUnit: <http://pear.php.net/package/PHPUnit>
- [5]. Selenium: <http://seleniumhq.org/>
- [6]. Hudson: <http://hudson-ci.org/>
- [7]. Jenkins: <http://jenkins-ci.org/>

9. Apéndice: Patrones para la adaptación de los tests

Las modificaciones que se detallan en esta sección (excepto la primera), pueden no ser de importancia cuando se trata de una navegación normal y simple (por ejemplo una búsqueda en google, o rellenar un formulario en alguna pagina web, etc...).

Pero se ha detectado durante la elaboración de este proyecto y después de varias pruebas, que esas modificaciones son imprescindibles para el caso concreto de cuando se trabaje con una aplicación realizada utilizando gvHIDRA, ya que existen varias funcionalidades del framework (ventanas de selección, menús, algunos botones, etc...) que para ser efectuadas correctamente, utilizan internamente unos scripts de JavaScript y/o necesitan esperas entre un paso del código y el siguiente para poder cargar las paginas adecuadamente.

Para ello, de los puntos que se detallan a continuación, aplicamos en nuestro test las que tocan para el caso de uso concreto que estamos realizando.

- En el método setUp que esta al principio del código modificamos la ruta que esta en la llamada `$this->setBrowserUrl(“”)`, y ponemos la ruta de la pagina a partir de la cual habíamos iniciado la generación del test (el arranque del Selenium Ide).
- opcionalmente podemos cambiar en el método setBrowser el tipo de navegador utilizado, que por defecto es el firefox (`*chrome`).
- opcionalmente podemos especificar el puerto en el que esta corriendo el servidor de selenium, añadiendo al principio del método setUp la linea `$this->setPort(<puerto>);` por defecto el puerto es el 4444.
- opcionalmente podemos especificar la IP de la maquina donde se abrirá el navegador y se ejecutaran los pasos del test `$this->setHost(<IP >);` por defecto la maquina es el localhost.
- Para que los menús se desplieguen bien, y las paginas y/o ventanas de selección (que pueden conllevar ejecución de código en javascript) se carguen y funcionen correctamente, habrá que introducir una espera entre los pasos como habíamos mencionado antes, y lo hacemos añadiendo al final del método setUp la linea:
`$this->setSleep(<segundos>);`
- dependiendo de la rapidez o la lentitud de la comunicación con el Servidor de Selenium (por ejemplo cuando se utilizan maquinas virtuales), el numero de segundos se puede ajustar según las necesidades, pero normalmente un valor de 5 a 8 segundos suele ser suficiente.
- En todo caso, si el tiempo requerido para ejecutar el test no nos importa mucho (por ejemplo si se ejecuta por la noche y cuando no haya mucha carga en la red), seria aconsejable aumentar el numero de segundos de espera para estar mas seguros.
- normalmente cuando se quiere abrir una ventana de selección lo hacemos con un doble clic, pero selenium añade al test en vez de ello dos llamadas `$this->click(“nombre-ventana-selección”)`, lo que produce que la ventana no se abiera. Entonces tendríamos que reemplazarlas por una única llamada
`$this->doubleClick(“nombre-ventana-selección”)`.
- A veces cuando se intenta abrir desde el código del test una ventana nueva (popup), selenium genera la instrucción `selectWindow(“nombre_ventana”)`, pero esta instrucción falla y selenium no consigue encontrar la ventana, por lo tanto se sobrepasa el timeout de la función `waitForPopUp` y el test termina fallando.
- Para solucionar esto, insertamos antes de la instrucción `selectWindow`, una llamada
`$this->openWindow(<url>,<nombre_ventana>);` y le pasamos en el argumento `<url>` una cadena vacía, consiguiendo con esto que al invocar posteriormente al método `selectWindow(“nombre_ventana”)`, no se produzca el error anterior y selenium encuentra la ventana.

- algunos botones (especialmente las que hacen que ejecute un código en javascript), al hacer clic sobre ellos a través del test no se producen los resultados esperados, por ejemplo al intentar guardar un nuevo registro en la base de datos presionando el botón guardar, se comprobó que el registro no se había grabado. La solución a esto es añadir una línea de código para focalizar el botón antes de la llamada

`$this->click("nombre-botón")` que genera selenium.

La línea insertada sería: `$this->focus("nombre-botón");`

- normalmente cuando se presiona con el ratón sobre un panel o un frame, selenium genera la llamada

`$this->selectFrame("nombre-frame")`, pero algunas veces añade antes la línea: `this->selectFrame("");` que hace que el test fallara. Habría que borrar o comentar esa línea.

- en la llamada `$this->assertTrue($this->isTextPresent("texto"))`; que genera selenium cuando queremos verificar la existencia de algún texto en la página, a veces selenium no introduce el texto seleccionado como argumento de `isTextPresent`, y lo deja con la cadena vacía. Entonces tendríamos que añadirlo manualmente.

- en algunas aplicaciones, la acción de guardar un registro conlleva dar la opción al usuario para imprimir un informe (botones guardar e imprimir), lo que produce el despliegue de la típica ventana de diálogo para seleccionar la ubicación del archivo, y se coloca sobre la ventana de la aplicación pudiendo producir resultados inesperados. Para ignorar este diálogo invocamos a la llamada : `$this->windowFocus()`; para devolverle el foco a la pantalla original de la aplicación, y el test pueda continuar ejecutándose sin fallos.

- El diálogo de guardar e imprimir se cierra automáticamente al finalizarse la ejecución del test ya que selenium termina el proceso de firefox al finalizar la ejecución del test.

10. Anexo. Instalación de PHPUnit, Selenium y Hudson

En este manual explicamos los pasos mas importantes en la preparación del entorno de trabajo necesario para llevar a cabo la ejecución de los test unitarios, y que consiste básicamente en la Instalación de tres módulos: **PHPUnit**, **Selenium** y **Hudson**.

Daremos también algunos ejemplo del uso de estas herramientas para reforzar la comprensión de las mismas.

10.1. PHPUnit

Es una utilidad de php que lanza hacia el servidor de Selenium (que explicaremos en el siguiente apartado), una por una los pasos (pruebas unitarias, básicamente instrucciones de php) contenidos en un fichero de test. Se instala a través de PEAR (una utilidad que viene ya incluida con la instalación de php a partir de la versión 5.2+) .

Entramos a la carpeta donde tenemos el binario de pear (en el caso de este manual, la tenemos en /usr/local/php-5.2.17/bin/).

Y se lanzan las siguientes ordenes :

```
sudo ./pear channel-discover pear.phpunit.de
sudo ./pear update-channels
sudo ./pear install phpunit/PHPUnit
```

Buscamos ahora el binario de phpunit, y puede estar en el directorio PHPUnit (que normalmente esta en /ruta/de/php/bin/PEAR/PHPUnit), o puede estar en /ruta/de/php/bin/bin.

Una vez localizado el binario “phpunit” creamos un enlace simbólico a el en la carpeta /usr/sbin/ para poder invocarlo como un comando desde cualquier ubicación.

Por ultimo, para asegurarnos de que phpunit funciona correctamente ejecutamos:

```
phpunit --version
```

y debería darnos una salida parecida a :

```
PHPUnit 3.4.14 by Sebastian Bergmann .
```

10.2. Selenium

Selenium IDE versión 1.0.7 :

Plugin para Firefox que permite grabar una navegación realizada de forma manual y generar, en diferentes lenguajes (java, php, html, perl, etc...), el código necesario para reproducirla posteriormente.

<http://release.seleniumhq.org/selenium-ide/1.0.7/selenium-ide-1.0.7.xpi>

Para la generación de los ficheros de pruebas unitarias, y una vez instalado el Selenium Ide, lo arrancamos desde “menú herramientas > Selenium Ide” antes de comenzar la navegación.

Selenium Remote Control 1.0.3 :

Esta solución está pensada para permitir la ejecución de tests de Selenium bajo un sistema de integración continua.

Los tests se integran dentro del código fuente de una plataforma tecnológica concreta como Java o PHP. El código fuente de estos tests puede ser obtenido mediante Selenium IDE.

Los tests se ejecutan en una única máquina que puede ser la misma máquina que los genera una máquina distinta. Para ello, la máquina que ejecuta los tests tiene un control remoto que permite la comunicación entre el proceso que lanza los tests(el PHPUnit), y el que los ejecuta.

Es posible utilizar uno o varios navegadores para reproducir las pruebas en cuestión.

<http://selenium.googlecode.com/files/selenium-remote-control-1.0.3.zip>

Dentro del directorio de selenium remote control reside el servidor de selenium, es simplemente un fichero .jar de java “selenium-server.jar”, y que comunica con la herramienta PHPUnit que vimos en el apartado anterior.

10.3. Hudson

Para instalar Hudson entramos en la pagina web <http://hudson-ci.org/> , desde la cual podemos descargar el archivo hudson.war que contendrá la ultima versión , o bien , como se hizo para este documento , elegir nuestra distribución de linux , y ejecutar las lineas de comando que se nos indican.

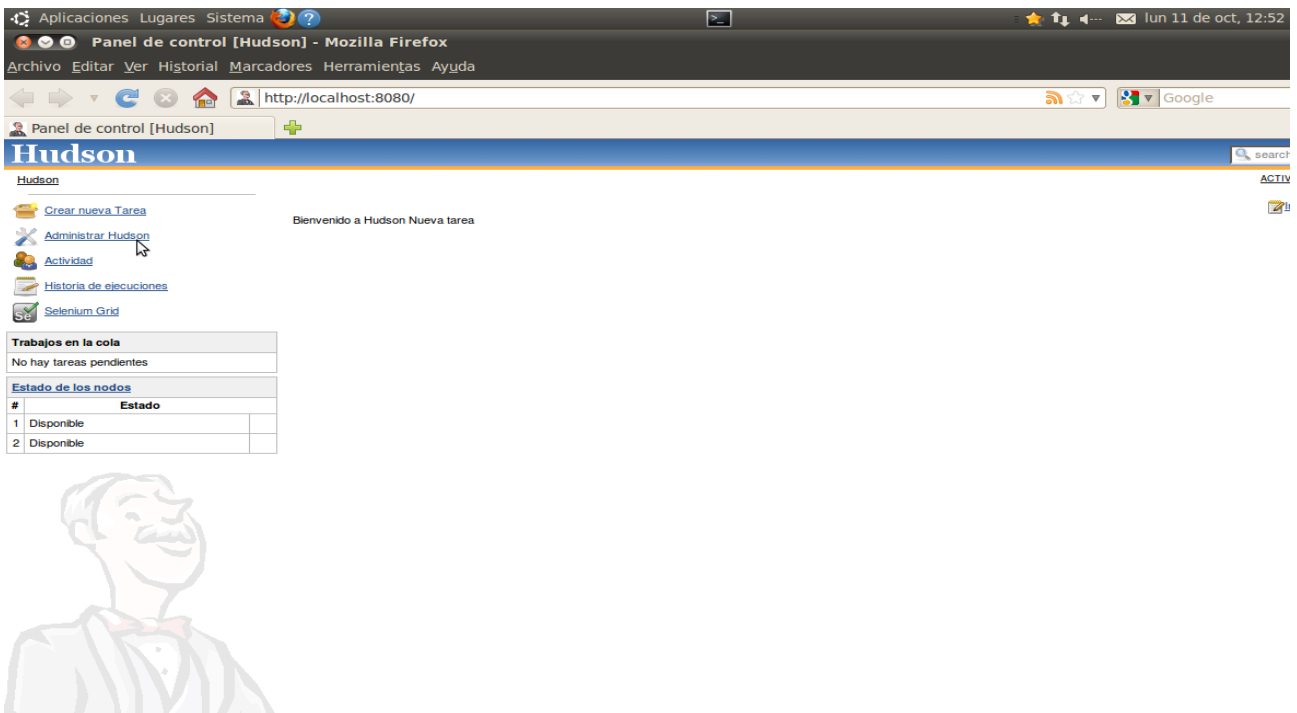
Por ejemplo, para ubuntu, se instala Hudson a través de apt-get lanzando los siguientes comandos :

```
wget -o /tmp/key http://hudson-ci.org/debian/hudson-ci.org.key
sudo apt-key add /tmp/key
wget -o /tmp/hudson.deb http://hudson-ci.org/latest/debian/hudson.deb
sudo dpkg --install /tmp/hudson.deb
```

Una vez completados los pasos, aceptando la descarga e instalación de los paquetes desde la consola, podremos iniciar/parar hudson con las ordenes: `sudo /etc/init.d/hudson start/stop`

Para trabajar con hudson , abrimos mediante el navegador la pagina <http://localhost:8080>

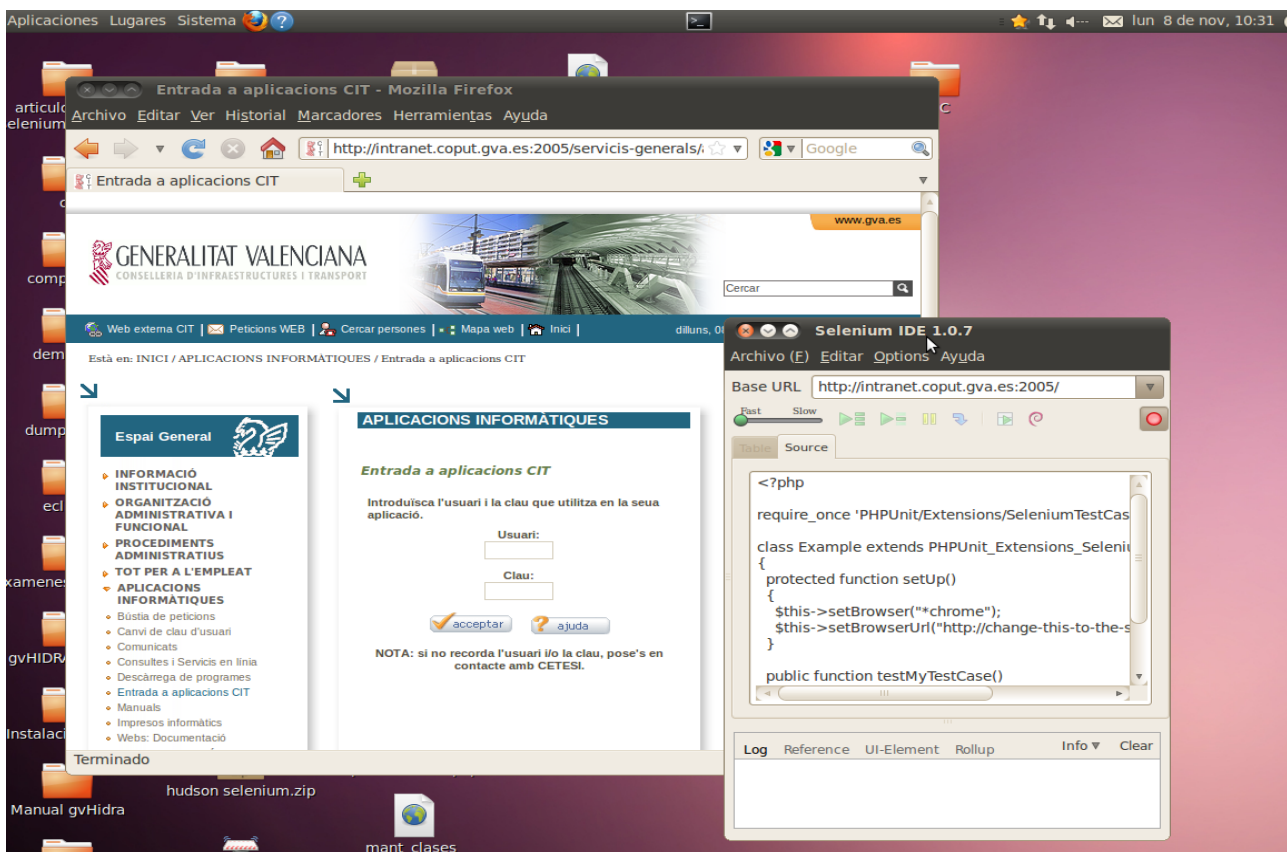
donde localhost se podría reemplazar por la IP de la maquina donde reside la instalación de hudson.



Página generada el: 11-oct-2010 12:52:06

10.4. Ejemplo de uso:

Como habíamos comentado anteriormente, cuando queremos generar un test cualquiera (desde una navegación sencilla por Internet, hasta ejecutar varias funcionalidades de una aplicación en php), arrancamos el plugin Selenium ID desde el menú de herramientas de firefox, y a continuación comenzamos nuestro uso ordinario de la aplicación:



Nota importante :

mientras navegamos por la aplicación procurar no pinchar con el ratón sobre ningún sitio del código dentro de la pestaña “source” en la ventanita de Selenium Ide, para que los pasos siguientes de navegación que haremos, se sigan insertando correctamente al final del mismo y no se intercalen por el medio, lo cual hará que el test fallara.

Minimizamos la ventana de Selenium Ide mientras utilizamos la aplicación.

El código generado en la pestaña source esta por defecto en html, y para cambiarlo al lenguaje de programación que nos interesa (por ejemplo PHP), cuando terminamos la navegación, en la ventana de Selenium Ide seleccionamos el menú Opciones>Formatos>PHPUnit , y comprobamos que el código se reescribe en ese lenguaje En este momento guardamos el test(menú Archivo>Guardar Como) en la ubicación que queremos.

Creación de los nodos y las tareas en el servidor de Hudson:

Para añadir un nodo nuevo (que representaría una maquina física o virtual), primero abrimos la pagina principal de Hudson (escribir en el navegador <http://<ip o localhost>.8080>), y cuando este listo pinchar en la izquierda sobre “administrar Hudson”, y luego en el panel de la derecha en “administrar nodos”:

Hudson

[Hudson](#)

- [Crear nueva Tarea](#)
- [Administrar Hudson](#)
- [Actividad](#)
- [Historia de ejecuciones](#)
- [Selenium Grid](#)

Trabajos en la cola

No hay tareas pendientes

Estado de los nodos

#	Estado
1	Disponible
2	Disponible

Administrar Hudson

⚠ Hay una nueva versión de Hudson disponible (1.379). [descarga](#)

- [Configurar el Sistema](#)
Configurar variables globales y rutas.
- [Actualizar configuración desde el disco duro.](#)
Descartar todos los datos cargados en memoria y a disco.
- [Administrar Plugins](#)
Añadir, borrar, desactivar y activar plugins que extienden Hudson.
- [Información del sistema](#)
Muestra información del entorno que puedan ayudar a solucionar problemas.
- [Registro del Sistema](#)
El log del sistema captura la salida de la clase javahelper.
- [Estadísticas de Carga](#)
Comprobar la utilización de los recursos y comprobar el estado de los nodos.
- [Interfaz de línea de comandos \(CLI\) de Hudson](#)
Accede y administra Hudson desde la consola, o desde un script.
- [Consola de scripts](#)
Ejecutar script para la administración, diagnóstico y depuración.
- [Administrar Nodos](#)
Añadir, borrar, desactivar y monitorizar los nodos sobre los que se ejecutan las tareas.

- En la izquierda de la pagina que se abre le damos a la entrada “nuevo nodo” y ponemos un nombre (preferentemente descriptivo del sistema operativo y/o del test, etc...), seleccionamos la opción “secundario pasivo”, y aceptamos.
- En la configuración del nodo, escribimos 1 en “numero de ejecutores” .
- Descripción y etiquetas son opcionales.

- En “directorio raíz remoto” ponemos “/” .
- En la lista “usar” seleccionamos: “dejar este nodo para ejecutar solamente tareas vinculadas a él”.
- Finalmente en “método de ejecución” seleccionamos “ejecutar el agente en el nodo secundario utilizando JNLP”, y guardamos.

Para ver los nodos creados volvemos a “administrar hudson> administrar nodos”, y observamos como el nodo ha sido creado (con una aspa roja encima, ya que aun el nodo aun no esta conectado a Hudson) .

The screenshot shows the Hudson web interface in a Mozilla Firefox browser. The main content area displays a table of nodes and a configuration form for a new node named 'nodo_servidor_apache'.

S	Nombre	Tiempo de respuesta	Espacio de intercambio libre	Espacio de disco libre	Espacio temporal libre	Arquitectura	Diferencia entre los relojes
	principal	12ms	868MB	39GB	39GB	Linux (386)	Sincronizados
	nodo_servidor_apache	N/A	N/A	N/A	N/A		N/A

The configuration form for 'nodo_servidor_apache' includes the following fields:

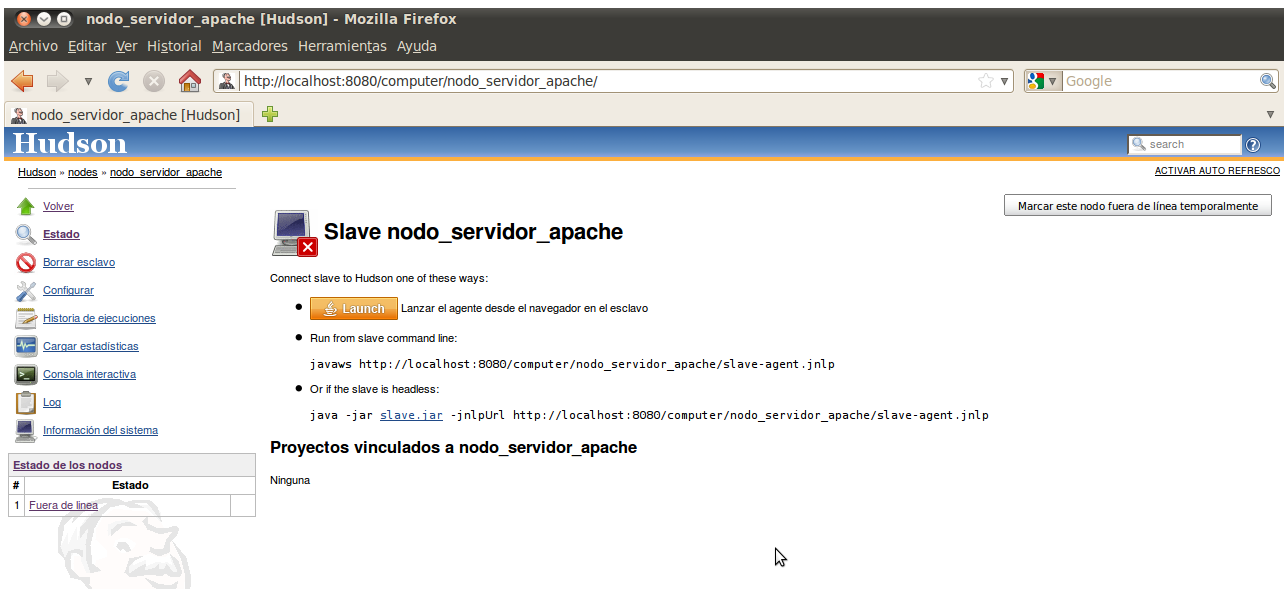
- Nombre: nodo_servidor_apache
- Descripción: (empty)
- Número de ejecutores: 1
- Directorio raíz remoto: /
- Etiquetas: (empty)
- Usar: Dejar este nodo para ejecutar solamente tareas vinculadas a él
- Método de ejecución: Ejecutar el agente en el nodo secundario utilizando JNLP
- Disponibilidad: Mantener este nodo en línea el máximo de tiempo posible

At the bottom of the form, there are checkboxes for 'Propiedades del nodo':

- Variables de entorno
- Localización de herramientas

A 'Guardar' button is located at the bottom of the configuration form.

Si pinchamos sobre el nombre del nodo, Hudson nos muestra tres maneras para establecer la conexión , la tercera (que es la que nos interesa ya que se trata de automatizar el proceso) se realiza a través de la línea de comandos en el nodo que ejecuta el test :



Nota: primero descargamos el archivo slave.jar en alguna ubicación el nodo esclavo, por ejemplo en algún directorio dedicado a la conexión con Hudson que creamos previamente (esto solamente se hace la primera vez que conectamos el nodo con el servidor de hudson).

luego ejecutamos la orden de conexión (hacerlo como root utilizando sudo, y también no olvidemos de reemplazar la cadena “localhost” por la IP del servidor de hudson) :

If the slave is headless:

```
sudo java -jar slave.jar -jnlpUrl <ip-servidor-hudson>:8080/computer/nodo_servidor_apache/slave-agent.jnlp
```

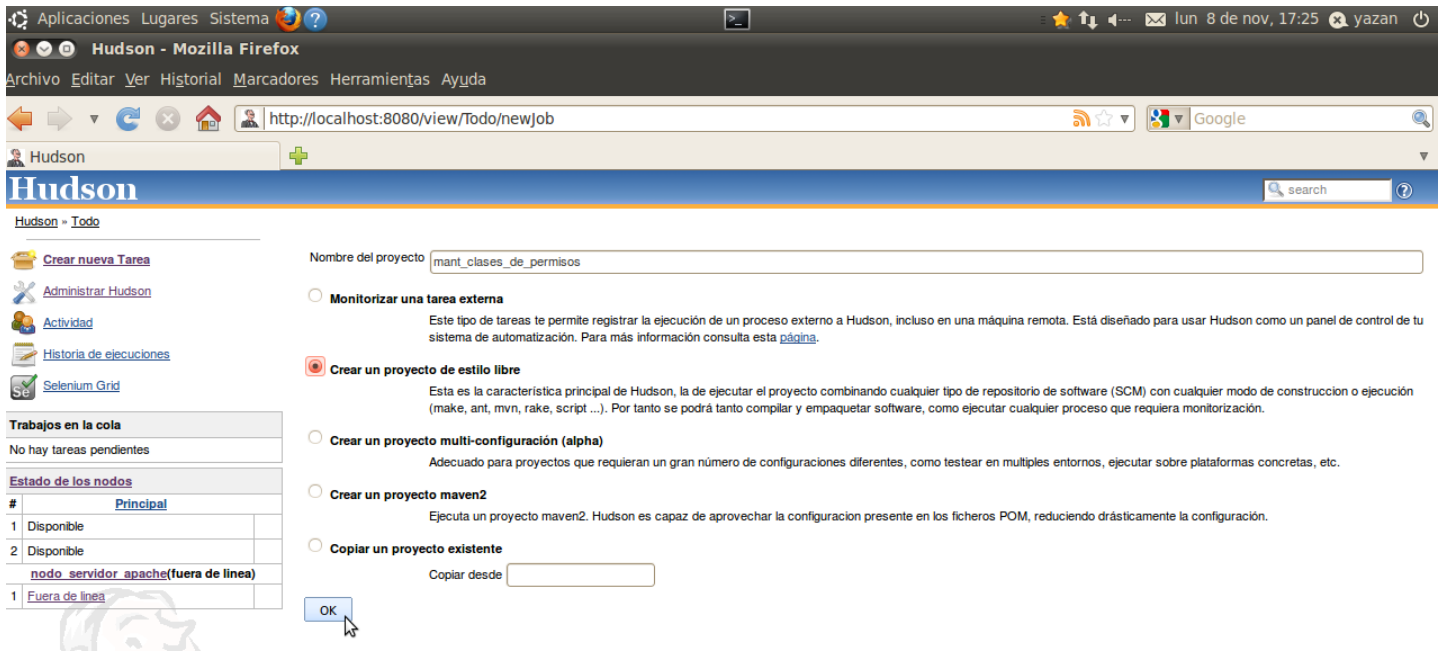
La salida sería algo parecido a:

```
13-oct-2010 10:47:10 hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Hudson agent is running in headless mode.
13-oct-2010 10:47:10 hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://<ip-hudson>:8080/]
13-oct-2010 10:47:10 hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to <ip-hudson>:57663
13-oct-2010 10:47:10 hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
13-oct-2010 10:47:10 hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

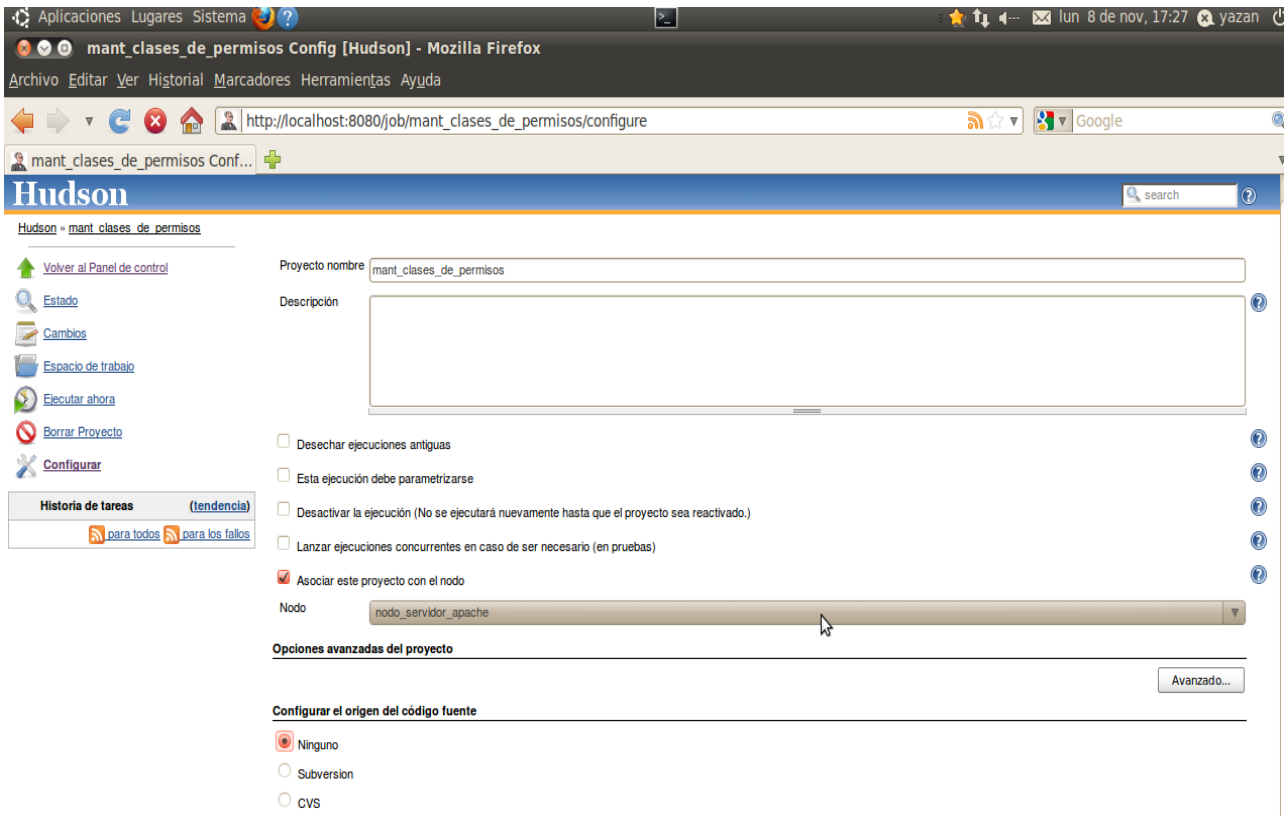
Creación de una tarea:

Desde la ventana principal de Hudson y en la parte izquierda pinchamos sobre “crear nueva tarea”.

le damos un nombre al proyecto, y seleccionamos la opción “crear un proyecto de estilo libre” y luego ok.



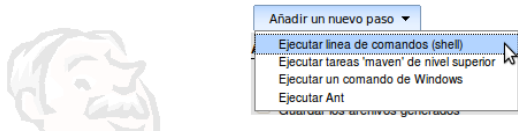
Una vez dentro de la configuración del proyecto, marcamos la opción de “asociar este proyecto con el nodo”, y de la lista que nos aparece elegimos el nodo que habíamos creado al principio.



En la sección *Disparadores de Ejecución* se puede seleccionar la opción *Ejecutar periódicamente* y en el campo *Programador* ingresar por ejemplo la expresión `0 1 * * *`, y que significa: todos los días a los 0 minutos y 1 horas .

Podemos usar el botón de ayuda (?) para comprender mejor esta sintaxis.

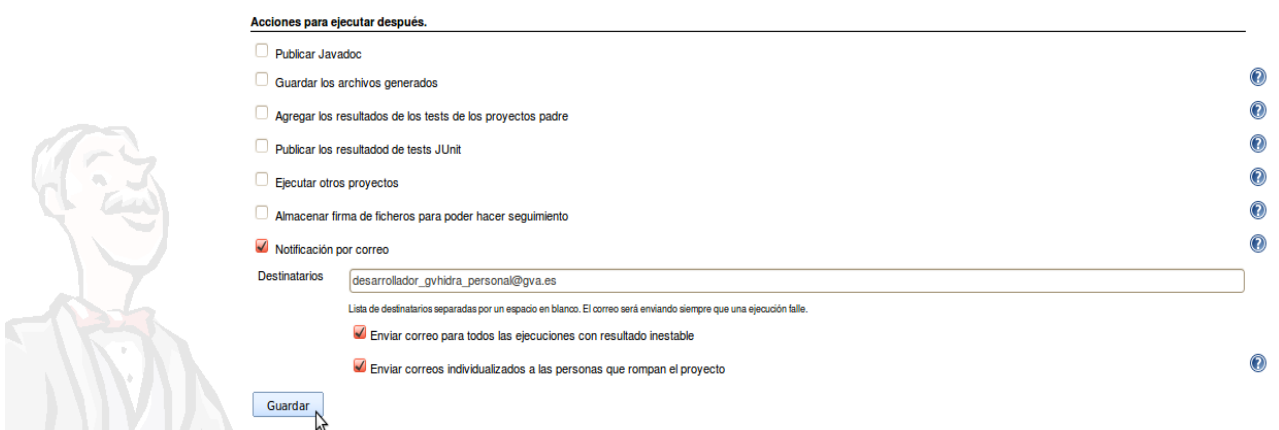
A continuación indicamos los comandos de shell que se vayan a ejecutar en esta tarea:



escribimos en el cuadro de texto los comandos a ejecutar, por ejemplo:

```
cd /usr/local/httpd-2.2.8/htdocs/personal
phpunit Example tests/mant_clases_permisos.php
```

Finalmente podemos marcar, si nos interesa, la opción de mandar un correo electrónico en el caso de que la ejecución falle por algún motivo (con una previa configuración del servidor smtp que vayamos a utilizar):



En el caso de que la ejecución falle, en el panel de la izquierda podemos pinchar sobre el nombre del nodo en el que se ejecuto la tarea, y luego le damos en la parte izquierda a “historial de ejecuciones”, donde podremos averiguar las causas del fallo para cada ejecución fallida pinchando sobre el símbolo de consola negro situado a la derecha de cada ejecución.

Historia de tareas ejecutadas en nodo_servidor_apache

Ejecución	Fecha	Estado
test_mant_clases_permisos :#51	4 días 0 hor	fallido desde la ejecución #48
test_mant_clases_permisos :#50	4 días 0 hor	fallido desde la ejecución #48
test_mant_clases_permiso :#1	4 días 0 hor	estable
test_mant_clases_permisos :#49	4 días 1 hor	fallido desde esta ejecución
test_mant_clases_permisos :#48	4 días 1 hor	cancelado
test_mant_clases_permisos :#47	4 días 1 hor	volvió a normal
test_mant_clases_permisos :#46	4 días 1 hor	fallido desde la ejecución #40
test_mant_clases_permisos :#45	4 días 1 hor	fallido desde la ejecución #40
test_mant_clases_permisos :#44	4 días 1 hor	fallido desde la ejecución #40
test_mant_clases_permisos :#43	4 días 1 hor	fallido desde la ejecución #40
test_mant_clases_permisos :#42	4 días 2 hor	fallido desde la ejecución #40
test_mant_clases_permisos :#41	4 días 2 hor	fallido desde esta ejecución

La causas pueden ser por ejemplo un fallo en la aplicación que pretendemos testear, fallo por permisos de acceso, un archivo no encontrado, que la maquina virtual java o que phpunit no este instalado, etc...

Para arrancar los servidores de selenium (selenium-server.jar), podríamos hacerlo creando también otra tarea(que seria la primera tarea lanzada) y que ejecute la siguiente orden(donde “-port puerto” es opcional, y por defecto es el 4444, pero seria conveniente cambiarlo):

```
cd /ruta/carpeta/selenium-server/
java -jar selenium-server.jar [-port <puerto>]
```

Nota importante: si se especifica un numero de puerto distinto al 4444, recordar que habrá que añadir en el método “setUp” del test una llamada al método setPort especificando el mismo numero de puerto como argumento , por ejemplo: \$this->setPort(4445);

Con todo lo anterior ya tenemos todo el entorno preparado y ya podemos empezar a lanzar las tareas (manualmente o configurando el/los instante/s y la periodicidad con la que estas serán lanzadas).