tmbs

Modelo de casos de prueba del Sprint 3

Versión 1.0

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 24/10/2013 | 1.0 | Creación del documento | Bruno Maiese |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contenido

[1 Objetivos 3](#_Toc370752073)

[2 Casos y Procedimientos de Pruebas Unitarias 3](#_Toc370752074)

[3 Procedimientos de Pruebas de Integración 3](#_Toc370752075)

[4 Casos y Procedimientos de Pruebas del Sistema 4](#_Toc370752076)

[4.1 Prototipo Toombs 2.0 4](#_Toc370752077)

[4.1.1 Requerimientos Funcionales 5](#_Toc370752078)

[4.1.2 Requerimientos no Funcionales 7](#_Toc370752079)

# Objetivos

En este documento se especificara como se realizara la verificación para el Prototipo Toombs 3.0, que es el prototipo que se obtendrá una vez finalizada la etapa de desarrollo del sprint 3.

# Casos y Procedimientos de Pruebas Unitarias

Se realizaran pruebas unitarias sobre todas las funcionalidades implementadas en el sprint, logrando una cobertura de código de por lo menos el 90 por ciento para los controladores, modelos, los adaptadores desarrollados para las herramientas JIRA y Redmine (JiraClient y RedmineClient), que como ya se especificó en el documento de la arquitectura del sistema son los encargados de obtener y persistir en las herramientas los distintos datos mediante REST y el ToolAdapter (encargado de procesar los datos provenientes de las herramientas de gestión).

A continuación se listan las historias sobre las que se realizaran los test unitarios:

1. Asignar horas a proyecto
2. Cargar horas sin proyecto asociado
3. Mapeo de estados no abiertos
4. Sincronización automática con JIRA (historia pendiente del sprint 2)
5. Sincronización automática con Redmine (historia pendiente del sprint 2)
6. Habilitar y deshabilitar categoría
7. Revisión de navegabilidad y botones de acción

Para realizar las pruebas de JiraClient, RedmineClient y ToolAdapter se utilizara el framework de Ruby unit test, y para testear los modelos y los controladores se utilizara minitest.

Las pruebas consisten en invocar a los diferentes métodos e ir verificando que el resultado de su invocación es consistente con lo esperado.

# Procedimientos de Pruebas de Integración

Al igual que en el sprint pasado, habrá tres grupos de desarrollo.

Por dificultades obtenidas en los sprint anteriores se decidió que cada grupo trabaje en una historia a la vez, en un branch destinado al desarrollo de esa historia exclusivamente. Al finalizar la historia el encargado de verificación realizara pruebas funcionales en un ambiente similar al que se realizara la demo. En caso de funcionar todo correctamente se mergea el branch de la historia con el branch master.

La conformación de los grupos es la siguiente:

* Grupo1
  + Horacio Pérez
  + Fabrizio Faggiani
  + German Mamberto

Historias: 1, 2.

* Grupo 2
  + Gabriel Pereyra
  + Miguel Da Silva
  + Gisel Cincunegui

Historias: 3,4, 5.

* Grupo 3
  + José Ignacio López
  + Martin Zanetti
  + Álvaro Acuña

Historias: 6,7.

Al integrar componentes o subsistemas se debe realizar lo siguiente:

Examinar y probar el sistema o subsistema obtenido según los casos de prueba.

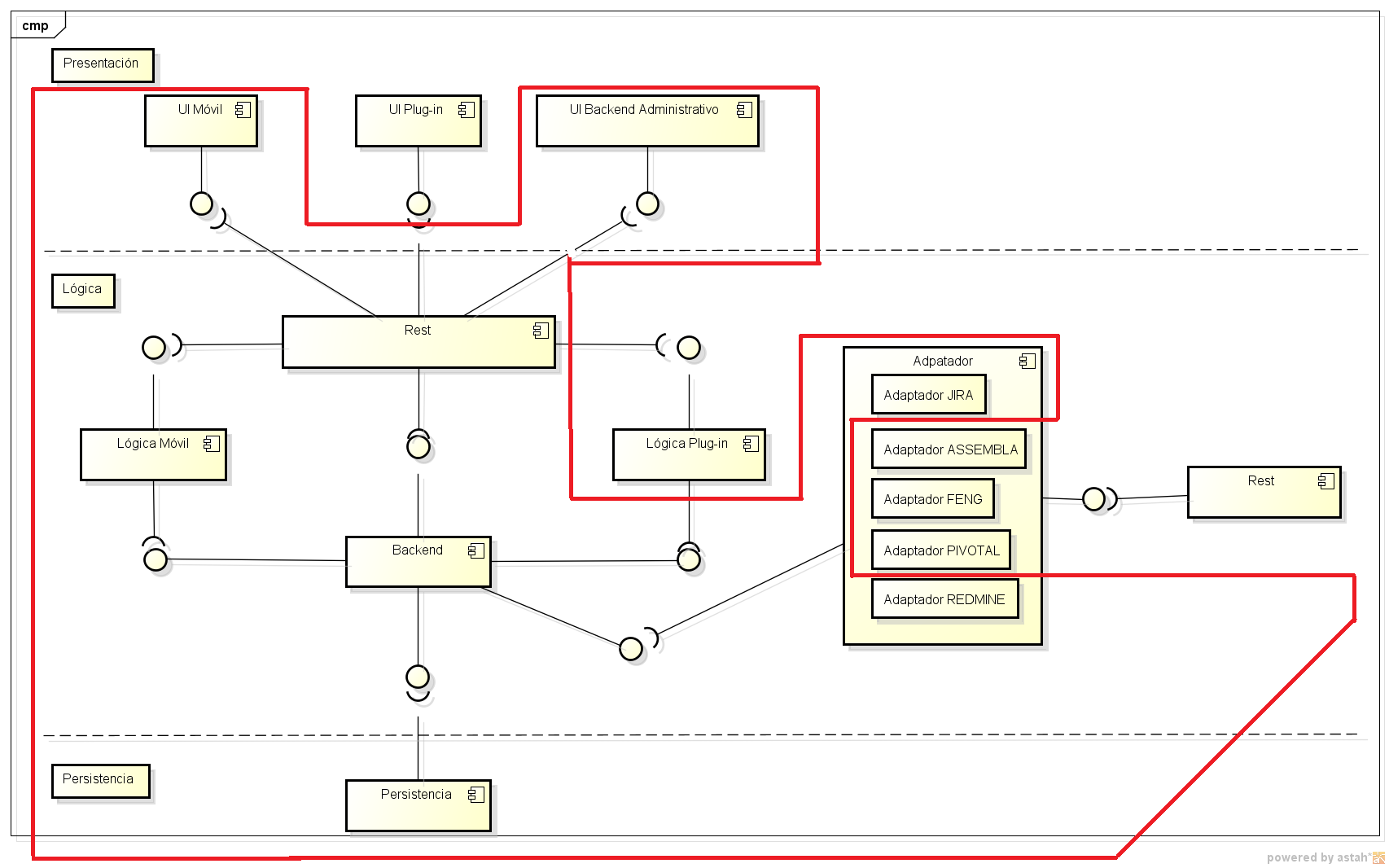
Si el sistema o subsistema contiene objetos con error se debe determinar si dichos errores son graves o no.

* En caso que sean graves se considera que la Integración no es correcta y se debe deshacer, terminando aquí el procedimiento de Integración.
* Si no hay errores graves la Integración se considera finalizada.

# Casos y Procedimientos de Pruebas del Sistema

## Prototipo Toombs 2.0

Subsistemas que lo integran:



### 

### Requerimientos Funcionales

Se describirán los requerimientos funcionales correspondientes a las historias del sprint 3. Es decir aquellos requerimientos que se pueden probar haciendo uso de las interfaces del sistema. En nuestro caso debido a las historias que se verificaran, haremos uso del front-end para la aplicación móvil y del front-end para el backend administrativo.

#### Testing planificado

##### Asignar horas a proyecto

Se desea ingresar horas asociadas a un proyecto, pero no a una tarea.

Flujo principal:

1. Ingresar a la aplicación.
2. Pulsar en el botón de opciones.
3. Pulsar en New Entry.
4. Ingresar los datos de la entry, seleccionando un proyecto e ingresando las horas con un formato valido.
5. Pulsar en el tick para confirmar la entry.
6. Fin CU.

Flujos alternativos:

4 A- El usuario ingresa una hora con formato inválido.

4 A.1- Mensaje de error

4 A.2- Fin CU.

5 A- El usuario presiona back

5 A.1 El usuario es redirigido a la página principal

5 A.2 Fin CU.

Escenarios:

|  |  |  |
| --- | --- | --- |
| Nombre de escenario | Flujo Inicial | Flujo alternativo |
| E1 Escenario 1 | Flujo principal |  |
| E2 Escenario 2 | Flujo principal | 4 A |
| E3 Escenario 3 | Flujo principal | 5 A |

Planilla de Condiciones:

|  |  |  |
| --- | --- | --- |
| Escenario – Condición | Salida Esperada | Casos |
| 1.E1- | -Mensaje: Successful entry -Re direccionamiento a la pantalla principal. | 1 |
| 2.E2- | Mensaje de error: Incorrect format. | 2 |
| 3E2- | - | 3 |

##### Cargar horas sin proyecto asociado

El usuario desea ingresar horas, pero que las mismas no estén asociadas a ningún proyecto, por lo tanto tampoco a una tarea.

Flujo principal:

1. Ingresar a la aplicación.
2. Pulsar en el botón de opciones.
3. Pulsar en New Entry.
4. Ingresar los datos de la entry, dejando el combo box de proyecto vacío e ingresando las horas con un formato valido.
5. Pulsar en el tick para confirmar la entry.
6. Fin CU.

Flujos alternativos:

4 A- El usuario ingresa una hora con formato inválido.

4 A.1- Mensaje de error

4 A.2- Fin CU.

5 A- El usuario presiona back

5 A.1 El usuario es redirigido a la página principal

5 A.2 Fin CU.

Escenarios:

|  |  |  |
| --- | --- | --- |
| Nombre de escenario | Flujo Inicial | Flujo alternativo |
| E1 Escenario 1 | Flujo principal |  |
| E2 Escenario 2 | Flujo principal | 4 A |
| E3 Escenario 3 | Flujo principal | 5 A |

Planilla de Condiciones:

|  |  |  |
| --- | --- | --- |
| Escenario – Condición | Salida Esperada | Casos |
| 1.E1- | -Mensaje: Successful entry -Re direccionamiento a la pantalla principal. | 1 |
| 2.E2- | Mensaje de error: Incorrect format. | 2 |
| 3E2- | - | 3 |

##### Habilitar y deshabilitar categoría

Se desea que el usuario administrador pueda habilitar y deshabilitar las diferentes categorías de tareas existentes en el sistema.

Flujo principal:

1. Ingresar a la aplicación y loguearse como administrador.
2. Pulsar el icono correspondiente a las categorías
3. Pulsar en Disable para deshabilitar una categoría
4. Pulsar en Enable para habilitar una categoría
5. Ingresar una nueva entry verificando las categorías disponibles
6. Fin CU.

Escenarios:

|  |  |  |
| --- | --- | --- |
| Nombre de escenario | Flujo Inicial | Flujo alternativo |
| E1 Escenario 1 | Flujo principal |  |

Planilla de Condiciones:

|  |  |  |
| --- | --- | --- |
| Escenario – Condición | Salida Esperada | Casos |
| 1.E1- | Categoría habilitada/deshabilitada | 1 |

#### Testing Exploratorio

Para probar la sincronización automática y la revisión de la navegabilidad y botones de acción, se utilizara la estrategia de testing exploratorio basado en sesiones.

El mapeo de estados queda fuera de la verificación, por no haberse podido implementar la historia.

Las misiones serán:

1) Ingresar a la herramienta JIRA y realizar modificaciones en las tareas para verificar que luego se sincronicen dichos cambios en Toombs.

2) Ingresar a la herramienta Redmine y realizar modificaciones en las tareas para verificar que luego se sincronicen dichos cambios en Toombs.

3) Ingresar horas de forma manual y mediante el cronometro, verificando que las mismas impacten en la instancia de JIRA o Redmine que corresponda.

### Requerimientos no Funcionales

Se verificará la adaptabilidad de las pantallas en distintos dispositivos móviles y la buena experiencia de usuario (que la aplicación sea intuitiva y fácil de usar).