



## Casos de Uso

Basándose en el **enunciado de referencia de “GTD Task Manager”**, implementar los siguientes casos de uso empleando la tecnología **JSF/Primefaces**:



### Administrador: Autenticación

El administrador deberá acceder a la aplicación previa autenticación en un formulario donde deberá introducir su login y password. En caso de introducir correctamente sus credenciales (admin1/admin1) se accederá al menú de opciones principales de administrador.

**PR01:** Autenticar correctamente al administrador.

**PR02:** Fallo en la autenticación del administrador por introducir mal el login.

**PR03:** Fallo en la autenticación del administrador por introducir mal la password.

### Administrador: Iniciación de la Base de datos

Deberá incluirse una opción de menú del administrador que borre todos los datos de la base de datos menos el usuario administrador (admin1/admin1) e inserte los siguientes datos:

- 3 usuarios normales, con login/contraseña: *user1/user1*, *user2/user2* y *user3/user3*.
- 3 categorías por usuario: categoría1, categoría2 y categoría3.
- **Cada usuario** normal tendrá **30 tareas** registradas: 10 tareas previstas para los siguientes 6 días a hoy (sin categorizar), 10 tareas para el día que se ejecuta la tarea - (sin categorizar) y otras 10 retrasadas respecto al día que ejecute la prueba y pertenecientes a las categorías 1 (3 tareas), 2 (3 tareas) y 3 (4 tareas).

**PR04:** Probar que la base de datos contiene los datos insertados con conexión correcta a la base de datos.

### Administrador: Activar/Bloquear usuario

El administrador podrá ver una lista de los usuarios registrados en el sistema, que incluirá todos sus atributos excepto su contraseña. Seleccionando uno del listado se podrá cambiar su estado de ENABLED a DISABLED o viceversa. El administrador podrá ordenar los usuarios por login, email o status. Usar la ordenación de tablas de Primefaces.

**PR05:** Visualizar correctamente la lista de usuarios normales y verificar que son los que tienen que ser.

**PR06:** Cambiar el estado de un usuario de ENABLED a DISABLED y tratar de entrar con el usuario que se desactivado.

**PR07:** Cambiar el estado de un usuario de DISABLED a ENABLED. Y tratar de entrar con el usuario que se ha activado.

**PR08:** Ordenar por Login.



**PR09:** Ordenar por Email.

**PR10:** Ordenar por Status.

#### **Administrador: Eliminar un usuario del sistema**

Tras seleccionar un usuario de la misma forma que en el caso de uso anterior, el administrador podrá borrar completamente sus datos. El sistema deberá pedirle confirmación. Borrarlo implica eliminar del sistema todas sus tareas y categorías, y, finalmente, eliminar el registro del usuario.

**PR11:** Borrar una cuenta de usuario normal y datos relacionados.

#### **Anónimo: Crear cuenta de usuario**

Accediendo desde la pantalla principal de la aplicación, habrá una opción que permitirá registrar un nuevo usuario. En pantalla aparte, se le pedirá login, email y password dos veces. Tras la validación de estos campos se registrará al usuario con status ENABLED.

Las validaciones de datos consistirán en:

- El login no se puede repetir en el sistema.
- El email debe tener estructura sintáctica de dirección de correo.
- La password debe tener letras y números, y una longitud mínima de 8 caracteres.

**PR12:** Crear una cuenta de usuario normal con datos válidos.

**PR13:** Crear una cuenta de usuario normal con login repetido.

**PR14:** Crear una cuenta de usuario normal con Email incorrecto.

**PR15:** Crear una cuenta de usuario normal con Password incorrecta.

#### **Usuario: Ver lista de tareas**

Una vez iniciada sesión en el sistema, el usuario podrá ver las listas de tareas creadas pero no terminadas. Aparecerán en esa pantalla las opciones para acceder a las **pseudolistas** *inbox*, *hoy* y *semana*. Seleccionando una de estas listas podrá ver todas las tareas contenidas en ella.

La pseudolista **Inbox** relacionará todas las tareas que están sin categorizar ordenadas por fecha planeada. Si una tarea ya está retrasada, se mostrará la fecha en rojo. Esta lista deberá estar **paginada** por paginas de 8 en 8 tareas. Además se deberán incluir un filtrado por el nombre de la tarea. Se deberá usar la ordenación, paginación y filtrado de PrimeFaces.

**PR16:** Comprobar que en Inbox sólo aparecen listadas las tareas sin categoría y que son las que tienen que. Usar paginación navegando por las tres páginas.

**PR17:** Funcionamiento correcto de la ordenación por fecha planeada.

**PR18:** Funcionamiento correcto del filtrado.



La pseudolista **Hoy** mostrará todas las tareas cuya fecha planeada es hoy y todas las retrasadas. Estas últimas mostrarán la fecha prevista (ya pasada) en rojo. A su vez, las tareas se podrán ordenar por categoría de usuario y fecha planeada. Se deberá usar la ordenación de PrimeFaces.

**PR19:** Funcionamiento correcto de la ordenación por categoría.

**PR20:** Funcionamiento correcto de la ordenación por fecha planeada.

**PR21:** Comprobar que las tareas que no están en rojo son las de hoy y además las que deben ser.

**PR22:** Comprobar que las tareas retrasadas están en rojo y son las que deben ser.

La pseudolista **Semana**, mostrará todas las tareas, nombre y categoría, desde hoy (incluyendo las retrasadas) hasta dentro de 7 días (incluido hoy), ordenadas por día o nombre. Las retrasadas mostrarán en rojo el nombre de la categoría.

**PR23:** Comprobar que las tareas de hoy y futuras no están en rojo y que son las que deben ser.

**PR24:** Funcionamiento correcto de la ordenación por día.

**PR25:** Funcionamiento correcto de la ordenación por nombre.

Todas estas listas mostrarán por defecto las tareas creadas y sin terminar. En el caso de *inbox* se podrán ver también las tareas terminadas. Para ello aparecerá en pantalla una opción que deshabilitará este filtro. En esta situación, todas las tareas finalizadas aparecerán al final del listado definiendo el nombre de la tarea en color verde.

**PR26:** Confirmar una tarea, inhabilitar el filtro de tareas terminadas, ir a la pagina donde está la tarea terminada y comprobar que se muestra en verde.

#### **Usuario: Añadir tarea**

Desde una opción de menú independiente de las listas, el usuario podrá registrar una nueva tarea. Basta con registrar el nombre de la tarea. El resto de campos tomarán valores automáticos o por defecto: la fecha de creación será la del sistema, las fechas planeada se rellenará mediante un calendario de PrimeFaces y la fecha finalizada estará vacía, la tarea no tendrá comentario y sin categoría. Tras la creación se redireccionará el control automáticamente a la siguiente lista: Inbox si se ha creado sin categoría, Hoy si se ha creado con categoría y fecha planificada hoy y a Semana si se ha creado con fecha planificada posterior a Hoy y con categoría.

**PR27:** Crear una tarea sin categoría y comprobar que se muestra en la lista **Inbox**.

**PR28:** Crear una tarea con categoría categoria1 y fecha planeada **Hoy** y comprobar que se muestra en la lista Hoy.

**PR29:** Crear una tarea con categoría categoria1 y fecha planeada posterior a Hoy y comprobar que se muestra en la lista **Semana**.



### **Usuario: Editar tarea**

Seleccionando cualquier tarea en cualquier listado, el usuario podrá editar el nombre, el comentario y la fecha planeada.

También se podrá modificar la categoría a la que pertenece. Al cambiarla de categoría las pseudolistas se deberán refrescar.

**PR30:** Editar el nombre, y categoría de una tarea sin categoría (se le cambia a categoría1) de la lista Inbox y comprobar que las tres pseudolista se refrescan correctamente.

**PR31:** Editar el nombre, y categoría (se cambia a sin categoría) de una tarea de la lista Hoy y comprobar que las tres pseudolistas se refrescan correctamente.

### **Usuario: Marcar tarea como finalizada**

Seleccionando cualquier tarea del listado el usuario podrá marca una tarea como terminada. Eso hará que la tarea desaparezca de los listados en los que pueda estar saliendo.

**PR32:** Marcar una tarea como finalizada. Comprobar que desaparece de las tres pseudolistas.

### **Administrador / usuario: cerrar sesión**

Cualquier usuario con la sesión iniciada podrá solicitar la finalización de la sesión.

**PR33:** Salir de sesión desde cuenta de administrador.

**PR34:** Salir de sesión desde cuenta de usuario normal.

### **Internacionalización**

La aplicación deberá estar internacionalizada a dos idiomas al menos.

**PR35:** Cambio del idioma por defecto a un segundo idioma. (Probar algunas vistas)

**PR36:** Cambio del idioma por defecto a un segundo idioma y vuelta al idioma por defecto. (Probar algunas vistas)

### **Seguridad**

Se deberá emplear la técnica de autenticación/autorización más adecuada a este contexto. Se recomienda el uso de un URL inicial diferente para cada rol.

**PR37:** Intento de acceso a un URL privado de administrador con un usuario autenticado como usuario normal.

**PR38:** Intento de acceso a un URL privado de usuario normal con un usuario no autenticado.



## Pruebas automatizadas

Se deberá suministrar un proyecto Java JUnit con las pruebas unitarias solicitadas empleando el framework Selenium (Se suministrará en el campus virtual un proyecto plantilla de ejemplo que deberán entregarse con los mismos archivos pero modificando el nombre del proyecto).

Todas las pruebas deben incluir el click en las opciones de menú correspondientes, etc, etc, igual que lo haría un usuario real.

## El protocolo de prueba

Para probar cada proyecto el profesor realizará los siguientes pasos:

- Descomprimirá el zip suministrado por el alumno.
- Lanzará la base de datos del proyecto suministrado por el alumno (sdi2-n/data/startup.bat).
- Copiará el archivo war a la carpeta deployments del servidor wildflyext (sólo el archivo war).
- Importará el proyecto sdi2-test en eclipse.
- Ejecutará el proyecto sdi2-test.

## Otras consideraciones importantes

Funcionalidades obligatorias:

- Ajaxificación de aquellas pantallas/formularios donde sea oportuno.
- También es **obligatorio** utilizar la librería de *logging* para registrar el funcionamiento interno de la aplicación.

Funcionalidades opcionales:

- Aunque se proporciona una capa de persistencia totalmente funcional, es posible mejorarla: utilizar JPA. Esta posibilidad es **opcional** (OPCIONAL 1, hasta 1 punto).

## Arquitectura de la aplicación

La aplicación deberá estar diseñada siguiendo un patrón arquitectónico de N-Capas hibridado con el patrón MVC en la línea de lo visto en la sesión de la laboratorio sobre Diseño Arquitectónico.

## Capa de Presentación

Deberán emplearse los siguientes recursos JSF como mínimo:

- Managed Beans: Emplear face-config.xml.
- Reglas de navegación. Emplear face-config.xml
- Internacionalización (i18n) en dos idiomas de todos los textos posibles.



- Uso del método de validación adecuado en formularios.
- Empleo coherente de plantillas en orden a reducir la duplicidad de código JSF.
- Uso de etiquetas Primefaces siempre que sea posible.

## Entrega

- La carpeta exportada sin comprimir del **proyecto Eclipse** donde se ha desarrollo de entrega (Export/General/File System). El nombre de dicho proyecto será sdi2-n, con el nombre asignado en la lista publicada al efecto en el campus virtual. Por ejemplo, sdi2-10.
- Un archivo **sdi2-10.war** que se despliegue correctamente en el servidor Wildfly **externo** y que se corresponda con la exportación del proyecto Eclipse anterior. (El puerto de escucha del servidor externo es **8180**)
- La carpeta exportada sin comprimir de un **segundo proyecto Eclipse** (Export/General/File System), con nombre **sdi2-n-pruebas**, con los casos de pruebas funcionales. Las pruebas se ejecutarán contra la **aplicación desplegada en el servidor externo**. El proyecto plantilla del que se partirá para este segundo proyecto es Proyecto-Selenium-Junit1617 (Seminario4) que se deberá renombrar y la suite de pruebas que se deberá modificar es el archivo PlantillaSDI2\_Tests1617.java que NO se deberá renombrar.
- Un documento **PDF** (con el mismo nombre que el proyecto) que contenga:
  - Una captura de pantalla de la pestaña JUnit después de pasar las pruebas donde se vea el nombre del proyecto de pruebas y el listado de pruebas superadas correctamente (Puede que no se vean todas).
  - El Mapa de navegación.
  - Una descripción de los aspectos técnicos decididos en las funcionalidades desarrolladas.
- Una hoja Excel **sdi2-n.xlsx** con las estadísticas de pruebas y funcionalidades superadas. (Se adjunta plantilla en el CV).
- Todo lo anterior **dentro de un fichero ZIP** (con el mismo nombre que el proyecto) deberá ser subido a una tarea del campus virtual que se creará al efecto. La subirá sólo el miembro del grupo cuyo UO se haya usado para el nombre del proyecto. Estructura del fichero ZIP:

sdi2-n.ZIP

```
└ sdi2-n           // proyecto Eclipse JSF/Primefaces
└ sdi2-n-pruebas  // proyecto Eclipse JUnit/Selenium
└ sdi2-n.war
└ sdi2-n.pdf
└ sdi2-n.xlsx     // Hoja Excel suministrada para
```



## Fecha máxima de entrega

Días del **26 al 30 de Marzo a las 20:00**, coincidiendo con el día **ANTERIOR** a que se imparte el grupo de laboratorio al que pertenece el alumno del grupo que **tiene clase en primer lugar esa semana**.

## Evaluación

La implementación de todas las pruebas funcionales relacionadas en este enunciado dará el 80% de la nota total. El 20% de la nota total se corresponde con la validación del código y el informe. Se podrá obtener hasta un 10% extra mediante las funcionales opcionales.

En resumen:

- 8 puntos procederán de las pruebas funcionales.
- 2 puntos de código e informe.
- 1 punto de las funcionalidades opcionales.
- Total: 11 puntos máximo.

Se **valorarán**, además, los siguientes aspectos técnicos:

Se penalizará:

- La escasa claridad del código.
- Que el código Java NO se ajuste a las *Java Code Conventions*.
- Que la compilación del código genere “*warnings*”.
- Que se presenten problemas durante el despliegue.