Share my trip

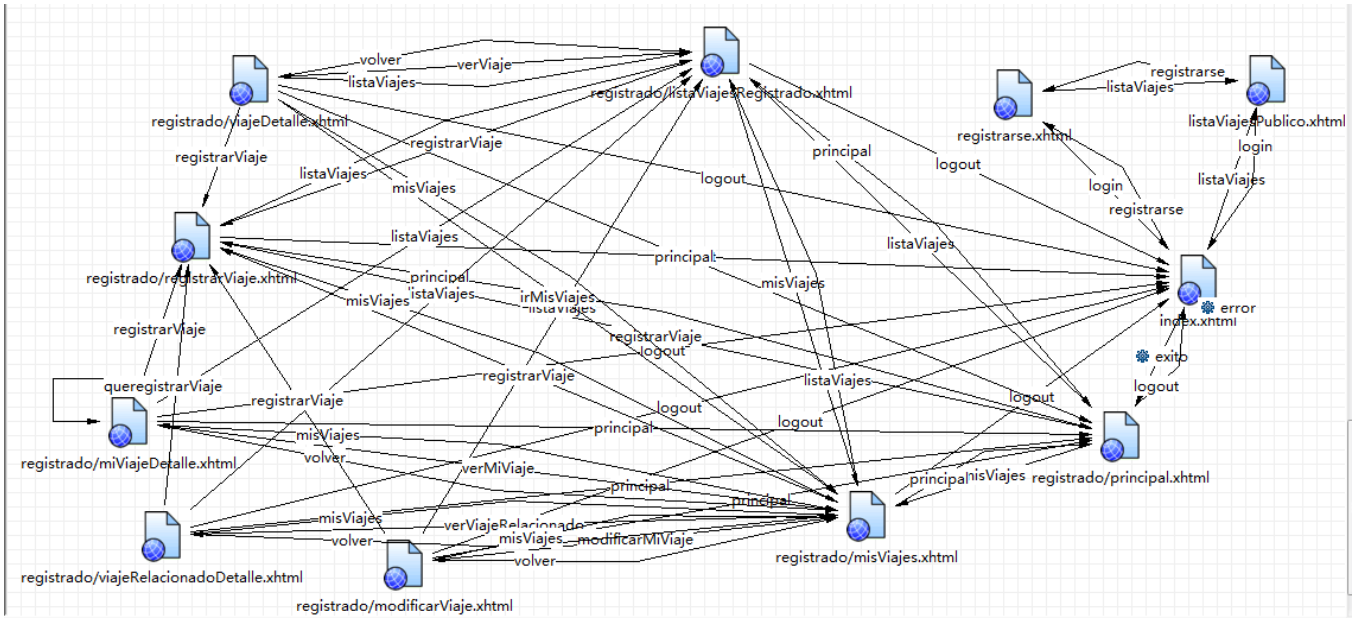
Documentación practica2 JSF

Changqu Ye

UO232770

Índice

Mapa de navegación



En esta práctica hay 2 rol, uno es usuario público y otro es usuario registrado. Un usuario público puede acceder solamente 3 vista que es registrarse.xhtml, index.xhtml y listaViajesPublico.xhtml. Mientras los usuarios registrados e identificados se pueden acceder a todas las vistas arribas mencionados. Las reglas de navegación están definidas en faces-config.xml.

Vistas

index.xhtml: Desde esta vista, un usuario registrado puede loquear a la aplicación si proporciona correctamente su identificador y contraseña.

registrarse.xhtml: Desde esta vista, cualquier usuario puede registrarse a la aplicación si proporcionan correctamente los datos que piden

listaViajesPublico.xhtml: Desde esta vista, un usuario puede ver una lista de viajes que están activos (plazo abierto, disponen plazas, etc.), la vista ofrece el mecanismo de ordenación, filtración y paginación.

registrado/principal.xhtml: Esta es la primera vista que se accede un usuario registrado cuando identifica correctamente en la aplicación.

registrado/listaViajesRegistrado.xhtml: Desde esta vista, un usuario puede ver una lista de viajes que están activos (plazo abierto, disponen plazas, etc.), la vista ofrece el mecanismo de ordenación, filtración y paginación. Si estás interesado en algún viaje puede pulsar ver para entrar a la vista registrado/viajeDetalle.xhtml para solicitar plaza en el viaje.

registrado/viajeDetalle.xhtml: Desde esta vista un usuario registrado puede solicitar plaza al viaje que está interesado.

registrado/registrarViaje.xhtml: Desde esta vista un usuario registrado puede registrar un viaje nuevo rellenando todos los campos obligatorios de forma correcta.

registrado/misViajes.xhtml: Desde esta vista un usuario registrado puede ver una lista de viajes activos que figura él como promotor, puede modificarlo, cancelarlo y entrar para ver la detalle, también puede ver otro listado del viaje relacionado que había participado o

solicitado la plaza y entrar en el viaje para ver la detalle. La vista ofrece el mecanismo de ordenación, filtración y paginación para los listados de viajes.
registrado/miViajeDetalle.xhtml: Desde esta vista un usuario registrado que figura como promotor del viaje puede confirmar pasajero y excluir pasajero del viaje.
registrado/viajeRelacionadoDetalle.xhtml: Desde esta vista un usuario registrado que figura como participante o solicitante del viaje puede cancelar su solicitud del viaje.
registrado/modificarViaje.xhtml: Desde esta vista un usuario registrado que figura como promotor del viaje puede modificar su viaje si proporcionan datos válidos.

Descripción de acciones

Rol público

registrarse: Dirige a la vista al registrarse.xhtml.
listaViajes: Dirige la vista al listaViajePublico.xhtml y carga en esta vista una lista de viajes que están activos (plazo abierto, disponen plazas).
login: Dirige la vista al index.xhtml.
registrarse.registrarse: Para poder registrarse el usuario debe introducir un identificador usuario, nombre, apellidos, correo electrónico y contraseña por duplicado. En este caso, el identificador usuario debe ser único (que no esté ya utilizado), el correo electrónico debe contener @, y la contraseña debe coincidir con el repetir contraseña. En caso contrario se mostrará un mensaje indicando la naturaleza del problema o los mensajes de validación del campo que dio error.
login.verify: Un usuario registrado que quiere loquear a la aplicación debe proporcionar su usuario y contraseña correcto, en caso contrario, se muestra un error diciendo la naturaleza del problema. Una vez valido, la aplicación guarda una sesión correspondiente del usuario.

Rol registrado

principal: Dirige a la vista registrado/principal.xhtml.
listaViajes: Dirige a la vista registrado/listaViajesRegistrado.xhtml y carga en esta vista una lista de viajes que están activos (plazo abierto, disponen plazas).
verViaje: Desde la vista registrado/listaViajesRegistrado.xhtml dirige la vista al registrado/viajeDetalle.xhtml. Se carga en esta vista informaciones sobre el promotor, participantes y los solicitantes del viaje.
misViajes: Dirige a la vista registrado/misViajes.xhtml. Además, carga un listado de su viaje que no se ha realizado como promotor y un listado de viajes relacionados que no se ha realizado (puede ser participante o solicitante).
verMiViaje: Desde la vista registrado/misViajes.xhtml dirige la vista al registrado/miViajeDetalle.xhtml. Se carga en esta vista informaciones sobre el promotor, participantes y los solicitantes del viaje.
modificarMiViaje: Desde la vista registrado/misViajes.xhtml dirige la vista al registrado/modificarViaje.xhtml. Se carga en esta vista informaciones sobre el viaje que va a modificar.
verViajeRelacionado: Desde la vista registrado/misViajes.xhtml dirige la vista al

registrado/viajeRelacionadoDetalle.xhtml. Se carga en esta vista informaciones sobre el promotor, participantes y los solicitantes del dicho viaje.

registrarViaje: Dirige a la vista registrado/registrarViaje.xhtml.

logout: El usuario cierra su sesión invalidando la sesión obtenido del HttpServletRequest y dirige a la vista index.xhtml.

viajes.solicitarPlaza: Un usuario registrado puede solicitar una plaza al viaje que esté interesado a través del verDetalle de la vista registrado/listaViajesRegistrado.xhtml.

viajes.cancelarSolicitud: A través del misViajes, el usuario puede ver los viajes que ha solicitado plaza o ya es participantes de él, de allí, el usuario que está en sesión tiene opción de cancelar su solicitud. Si el usuario cancela su solicitud pendiente, se elimina de la TApplication, si el usuario cancela su plaza confirmado, también se elimina del Tseat y la plaza libre del viaje se aumenta 1 si dicho usuario es del estado ACCEPTED. Si por dicha cancelación el viaje estaba CLOSED por plaza libre, pues el viaje pasa al estado OPENED.

viajes.cancelarViajes: Un usuario registrado en la sección del misViaje puede ver su viaje que no se ha realizado y él como promotor, en este caso, puede seleccionar varios viajes a cancelar, los viajes seleccionados pasan al estado CNCELLED y no se va mostrar en lista de viajes, pero las participantes de este viaje pueden ver en misViajes que este viaje esta cancelado.

viajes.confirmarPasajero: Un usuario registrado en su sección misViajes puede ver su viaje que no ha realizado y él como el promotor, si pulsa ver, entra ver la detalle que están las participantes y los solicitantes del viaje. En este caso el usuario tiene una opción que es ConfirmarPasajero, es decir confirmar a los pasajeros que mandaron solicitud y les convierten en participantes del viaje, como consecuencia se disminuye 1 plaza libre a cada pasajero confirmado y si plaza libre pasa al 0, el estado del viaje pasa al CLOSED.

viajes.excluirPasajero: Un usuario registrado en su sección misViajes puede ver su viaje que no ha realizado y él como el promotor, si pulsa ver, entra ver la detalle que están las participantes y los solicitantes del viaje. En este caso el usuario tiene una opción que es ExcluirPasajero, es decir excluir a los pasajeros que están como participantes del viaje, su estado pasa al EXCLUDED como consecuencia se aumenta 1 plaza libre a cada pasajero excluido, si viaje está en CLOSED, la exclusión del pasajero hace que el viaje pasa al estado OPENED.

registrarViaje.registrarViaje: Un usuario registrado puede registrar su viaje propio, en este caso el usuario debe introducir las informaciones de salida, las informaciones de llegada, fecha límite, plaza máxima, plaza libre, coste a repartir, etc. Si todo es correcto, se registra el viaje, en caso contrario, mostrando un error indicando la naturaleza de error o los mensajes de validación del campo que dio error.

modificarViaje.modificarViaje: Un usuario registrado en la sección del misViaje puede ver su viaje que no se ha realizado y él como promotor, en este caso, el usuario tiene opción de modificar dicho viaje, debe introducir datos correctos, en caso contrario se muestra un error indicando que la acción no tuvo éxito o los mensajes de validación del campo que dio error.

Otros

setting.setSpanish: Cambia el idioma de aplicación al español.

setting.setEnglish: Cambia el idioma de aplicación al inglés.

setting.setChinese: Cambia el idioma de aplicación al chino.

Seguridad

La técnica de autorización que ha utilizado es un filtro http, se ha creado una clase que se llama LoginFilter, implemente la interfaz javax.servlet.Filter e implementar su método principal doChain(). Hacen control de todas las páginas situados en url /registrado/*, si previamente no hay un usuario guardado en sesión no dejará acceder la aplicación a estas vistas controlado. A parte de eso se ha utilizado la librería Logging para registrar el funcionamiento interno de la aplicación.

Arquitectura

La aplicación está diseñada con el patrón arquitectónico N-capas.
En la capa de presentación, está diseñado por patrón MVC empleando los recursos de JSF, Modelo que corresponde al Bean, vista que corresponde a los archivos .xhtml, controlador que corresponde con faces-config.xml. Se ha incorporado la librería de Primefaces en esta capa y se ha hecho internacionalización en tres idiomas de todos los textos posibles. En los formularios se ha utilizado los métodos de validación automática implícita y explícita. También emplea plantillas en la vista para reducir la duplicidad de código JSF.
En la capa de negocio se ha seguido un diseño basado en interfaces de servicios empleando los patrones Fachada y Factoría.
En la capa de persistencia se emplea el patrón DAO para el diseño del acceso de la base de datos, así como el uso de consultas SQL empleando jdbc. La base de datos se ha rellenado con suficientes datos simulando una situación real y se ha utilizado control de concurrencia optimista (uso de commit/rollback) para mantener la consistencia de la base de datos.

Descripciones de parte opcionales

Opcional1: Filtrado1 Filtrado2

En la vista listaViajePublico.xhtml, listaViajeRegistrado.xhtml y misviajes.xhtml el usuario puede filtrar el listado por los campos combinados Origen y Destino para localizar rápidamente los viajes de su interés. Dicho filtrado se ha implementado utilizando componente de Primefaces Filter que es una extensión de JSF.

Opcional2: Ordenación1 Ordenación2

En la vista listaViajePublico.xhtml, listaViajeRegistrado.xhtml y misviajes.xhtml el usuario puede seleccionar el tipo de ordenación para el listado por los campos origen, destino, fecha cierre, fecha salida, fecha llegada, estado, coste estimado y promotor para localizar rápidamente los viajes de su interés. Dicha ordenación se ha implementado utilizando componente de Primefaces Sort que es una extensión de JSF.

Opcional3: Paginación1 Paginación2

En la vista listaViajePublico.xhtml, listaViajeRegistrado.xhtml y misviajes.xhtml el usuario puede seleccionar el número de filas visualizables en este caso se ha establecido para

listaViajePublico.xhtml y listaViajeRegistrado.xhtml unos 10 filas, para misviajes.xhtml unos 5filas. Se puede elegir que cada página cargue entre 5 10 o 15 filas, el resto sobrante se meten en las siguientes páginas. Dicha paginación se ha implementado utilizando componente de Primefaces Paginator que es una extensión de JSF.

Opcional5: i18n

Se ha desarrollado internalización (i+18chars+n) en tres idiomas de todos los idiomas posibles, estos tres idiomas son Chino, Español e Inglés. Por lo tanto hay 3 ficheros de .properties en el directorio src del proyecto, que son messages_cn.properties, messages_en.properties y messages_es.properties. El uso de estos tres ficheros es controlado por msg que está declarado en faces-config.xml.

Opcional8: Autocompletado en Alta de viaje

En la vista registrarViaje.xhtml, el usuario puede completar los campos como ciudad y provincia de origen y destino utilizando mecanismo de autocompletado de Primefaces, de tal manera que si un usuario se sitúa en el campo y teclea una letra se debe saltar por abajo elementos de autocompletados para elegir.

Opcional9: Mantenimiento programado de la base de datos

En este caso, dado que un viaje pasado fecha de cierre debemos cambiar su estado al CLOSED y un viaje que ha pasado su fecha de llegada debemos cambiar su estado al DONE. Además, dado que un viaje una vez supera su fecha de cierre puede que haya usuario en estado PENDIENTE cuando debería ser SIN_PLAZA, aunque la relación del pasajero con el viaje podemos verlo cambiado cuando miramos detalle del viaje relacionado si ocurre lo anterior. He creado en la package listener una clase que se llama viajeMantenimiento.java que se declaran como listener en web.xml, emplea la clase Timer de java con una frcuencia de ejecución de cada 5s para actualizar base de datos teniendo cuenta lo anterior.

Pruebas unitarias

1.[RegVal]

Registro de usuario con datos válidos. En este caso se ha rellenado la prueba con datos válidos, un identificador que no hay en base de datos, contraseña coincide con repetir contraseña y restos de campos rellenados cumpliendo las validaciones del campo. Pero si vuelves ejecutar dicha prueba te dirá que el identificador de usuario ya está utilizado y no te deja hacer registro.

2.[RegInval]

Registro de usuario con datos inválidos (contraseñas diferentes). En este caso se ha llenado la prueba con campos válidos excepto la contraseña y repetir contraseña no se coinciden, la aplicación muestra un mensaje de error indicando que las contraseñas no coinciden.

2.[RegInval2]

Registro de usuario con datos inválidos (identificador ya existe). En este caso se ha llenado la prueba con todos los campos válidos excepto un identificador de usuario que ya existe en base de datos, la aplicación muestra un mensaje de error indicando que el identificador de

usuario ya está utilizado.

2.[RegInval3]

Registro de usuario con datos inválidos (falta de datos). En este caso se ha llenado la prueba solamente con identificador de usuario y nombre, el resto lo dejamos en vacío, la aplicación muestra todos los errores de validación que hay en los campos no rellenados.

3. [IdVal]

Identificación de Usuario registrado con datos válidos. En este caso se rellena el formulario de login con los datos correctos, por lo tanto, el usuario se loquea correctamente en la aplicación.

4. [IdInval]

Identificación de usuario registrado con datos inválidos (password inválido). En este caso el usuario introduce mal su contraseña, la aplicación muestra un error indicando que el password o usuario está incorrecto.

4. [IdInval2]

Identificación de usuario registrado con datos inválidos (falta datos). En este caso el usuario introduce solo ha introducido su identificador, pero no ha introducido su contraseña, la aplicación muestra un error de validación del campo no rellenado

5. [AccInval]

Intento de acceso con URL desde un usuario no público (no identificado). Intento de acceso a vistas de acceso privado. En este caso con el usuario no identificado, prueba acceder a alguna vista pública, la aplicación lo deja hacer sin problema, por otro lado también intenta acceder las vistas no públicas, como tiene mecanismo de autorización Filter implementado, todos los intentos se redirigen a la página al index.xhtml.

6. [RegViajeVal]

Registro de un viaje nuevo con datos válidos. En este caso un usuario registrado se loquea en la aplicación y dirige la vista registrarViaje.xhtml, rellenando todos los datos que hagan falta en el registro de forma valido, al pulsar botón registrar viaje, la aplicación muestra un mensaje indicando que el registro tuvo éxito.

7. [RegViajeInVal]

Registro de un viaje nuevo con datos inválidos (error relación plaza). En este caso un usuario registrado se loquea en la aplicación y dirige la vista registrarViaje.xhtml, rellenando todos los datos que hagan falta en el registro de forma valido, pero se ha puesto que la plaza libre sea un número mayor que plaza máxima, al pulsar botón registrar viaje, la aplicación muestra un mensaje de error indicando que el registro tuvo error en relación de plazas.

7. [RegViajeInVal2Autocompletado]

Registro de un viaje nuevo con datos inválidos (falta datos). En este caso un usuario registrado se loquea en la aplicación y dirige la vista registrarViaje.xhtml, rellenamos los campos de salida ciudad, llegada ciudad, salida provincia y llegada provincia utilizando el mecanismo de autocompletado de Primefaces, pero no rellenamos nada de otros campos obligatorios, al pulsar botón registrar viaje, la aplicación muestran los mensajes de validación de los campos no rellenados.

8. [EditViajeVal]

Edición de viaje existente con datos válidos. En este caso un usuario registrado se loquea en la aplicación y dirige la vista misViajes.xhtml, selecciona un viaje a modificar, modificamos el

campo comentario con nuevo comentario válido, al pulsar botón de modificar viaje, la aplicación muestra un mensaje indicando que la modificación tuvo éxito.

9. [EditViajeInVal]

Edición de viaje existente con datos inválidos (error relación plaza). En este caso un usuario registrado se loquea en la aplicación y dirige la vista misViajes.xhtml, selecciona un viaje a modificar, modificamos el campo plaza libre a un número mayor que plaza máxima, al pulsar botón de modificar viaje, la aplicación muestra un mensaje de error indicando que la modificación produjo un error al intentar modificar plaza libre.

10. [CancelViajeVal]

Cancelación de un viaje existente por un promotor. En este caso un usuario registrado se loquea en la aplicación y dirige la vista misViajes.xhtml, selecciona un viaje a cancelar marcando el checkbox delante del viaje, al pulsar botón de cancelar viaje, vemos que el estado del viaje con checkbox marcado se cambia al CANCELLED.

11. [CancelMulViajeVal]

Cancelación de múltiples viajes existentes por un promotor. n este caso un usuario registrado se loquea en la aplicación y dirige la vista misViajes.xhtml, selecciona varios viajes a cancelar marcandoles checkbox delante del viaje, al pulsar botón de cancelar viaje, vemos que el estado los viajes con checkbox marcado se cambia al CANCELLED.

12. [Ins1ViajeAceptVal]

Inscribir en un viaje un solo usuario y ser admitido por el promotor. En este caso un usuario registrado se loquea en la aplicación y dirige a la vista listaViajeRegistrado.xhtml, selecciona un viaje tiene interés, pulsa botón ver para ver el detalle del viaje y luego pulsa solicitar plaza. A continuación, este usuario cierra su sesión, el promotor del viaje seleccionado se loquea a la aplicación, dirige a la vista misViajes.xhtml, y entra ver detalle a ese viaje, confirma el usuario anterior que ha solicitado la plaza, la aplicación muestra un mensaje indicando que el usuario ha sido confirmado correctamente al viaje. Esta prueba no pasa al segundo ya que modifica base de datos.

13. [Ins2ViajeAceptVal]

Inscribir en un viaje dos usuarios y ser admitidos los dos por el promotor. En este caso un usuario registrado se loquea en la aplicación y dirige a la vista listaViajeRegistrado.xhtml, selecciona un viaje tiene interés, pulsa botón ver para ver el detalle del viaje y luego pulsa solicitar plaza. A continuación, este usuario cierra su sesión, otro usuario registrado se loquea en la aplicación y dirige a la vista listaViajeRegistrado.xhtml, selecciona el mismo viaje que el usuario anterior, pulsa botón ver para ver el detalle del viaje, luego pulsa solicitar plaza y cierra la sesión. Por el último, el promotor del viaje seleccionado se loquea a la aplicación, dirige a la vista misViajes.xhtml, y entra ver detalle a ese viaje, confirma los usuarios anteriores que han solicitado la plaza, la aplicación muestra un mensaje indicando que el usuario ha sido confirmado correctamente al viaje. Esta prueba no pasa al segundo ya que modifica base de datos.

14. [Ins3ViajeAceptInval] (Esta prueba depende de la prueba 13, debe hacer prueba 13 antes para poder hacer esta prueba correctamente)

Inscribir en un viaje (2 plazas máximo) dos usuarios y ser admitidos los dos y que un tercero intente inscribirse en ese mismo viaje, pero ya no pueda por falta de plazas. En este caso un usuario registrado se loquea en la aplicación y dirige a la vista listaViajeRegistrado.xhtml,

selecciona un viaje tiene interés, pulsa botón ver para ver el detalle del viaje y luego pulsa solicitar plaza. A continuación, este usuario cierra su sesión, otro usuario registrado se loquea en la aplicación y dirige a la vista listaViajeRegistrado.xhtml, selecciona el mismo viaje que el usuario anterior, pulsa botón ver para ver el detalle del viaje, luego pulsa solicitar plaza y cierra la sesión. A continuación, el tercer usuario registrado se loquea en la aplicación y dirige a la vista listaViajeRegistrado.xhtml, selecciona el mismo viaje que el usuario anterior, pulsa botón ver para ver el detalle del viaje, luego pulsa solicitar plaza y cierra la sesión. Por el último, el promotor del viaje seleccionado se loquea a la aplicación, dirige a la vista misViajes.xhtml, y entra ver detalle a ese viaje, confirma los dos usuarios anteriores que han solicitado la plaza, la aplicación muestra un mensaje indicando que el usuario ha sido confirmado correctamente al viaje, cuando el usuario promotor intenta confirmar el tercer usuario como participantes del viaje, la aplicación muestra un error indicando que no se puede confirmar pasajero ya que no queda plaza para el viaje. Esta prueba no pasa al segundo ya que modifica base de datos.

15. [CancelNoPromotorVal]

Un usuario no promotor Cancela plaza. En este caso, un usuario registrado se loquea en la aplicación y dirigea la vista misViajes.xhtml, selecciona un viaje relacionado, pulsa ver, entra en detalle de este viaje y pulsa botón cancelar solicitud, la aplicación redirige a la vista misViajes.xhtml y podemos ver que este viaje cancelado ya se ha quitado de la lista del viaje relacionado. Esta prueba no pasa al segundo ya que modifica base de datos.

16. [Rech1ViajeVal] (Esta prueba depende de la prueba 13 y 14 debe hacer prueba 13 y 14 antes para poder hacer esta prueba correctamente)

Inscribir en un viaje un usuario que será admitido y después rechazarlo por el promotor. En este caso un usuario registrado se loquea en la aplicación y dirige a la vista listaViajeRegistrado.xhtml, selecciona un viaje tiene interés, pulsa botón ver para ver el detalle del viaje y luego pulsa solicitar plaza. A continuación, este usuario cierra su sesión, el promotor del viaje seleccionado se loquea a la aplicación, dirige a la vista misViajes.xhtml, y entra ver detalle a ese viaje, confirma el usuario anterior que ha solicitado la plaza, la aplicación muestra un mensaje indicando que el usuario ha sido confirmado correctamente al viaje, ahora pulsa el botón de excluir pasajero, la aplicación muestra un mensaje indicando que el usuario ha sido excluido correctamente al viaje. Esta prueba no pasa al segundo ya que modifica base de datos.

17. [i18N1]

Cambio del idioma por defecto a un segundo idioma. (Probar algunas vistas) En este caso, cambiamos el idioma de visualización, se loquea un usuario y vamos a las diferentes vistas de la aplicación, podemos ver los cambios de idiomas producidos en estas vistas que hemos ido.

18. [i18N2]

Cambio del idioma por defecto a un segundo idioma y vuelta al idioma por defecto. (Probar algunas vistas) En este caso, cambiamos el idioma de visualización, se loquea un usuario y vamos a las diferentes vistas de la aplicación, podemos ver los cambios de idiomas producidos en estas vistas que hemos ido. Volvemos a cambiar idioma original, seguimos cambiando vistas y al final cierra la sesión.

19. [OpFiltrado]

Prueba para el filtrado opcional. En este caso accedemos a la vista listaViajePublico.xhtml, en

el componente Primefaces de filtrado, mete una letra en filtrado de ciudad y otra letra en filtrado de provincia, al final vemos que la lista se filtra por estos campos que han introducido letra.

20. [OpOrden]

Prueba para la ordenación opcional. En este caso, un usuario se loquea en la aplicación y accedemos a la vista listaViajeRegistrado.xhtml, pulsa en las diferentes componentes de Primefaces de ordenado, vemos que la lista se ordena por estos componentes de ordenado pulsado.

21. [OpPag]

Prueba para la paginación opcional. En este caso, un usuario se loquea en la aplicación y accedemos a la vista listaViajeRegistrado.xhtml, pulsa en el componente de Primefaces de paginación de avanzar a la siguiente página de listado, vemos que la lista se salta a la siguiente página de los elementos de la lista.

22. [OpMante] (Esta prueba depende de la prueba 13 y 14 debe hacer prueba 13 y 14 antes para poder hacer esta prueba correctamente)

Prueba del mantenimiento programado opcional. En este caso el usuario user6 se loquea en la aplicación, dirige a la vista misViajes.xhtml, pulsa ver un viaje relacionado con estado CLOSED, en el detalle de viaje relacionado vemos que la relación del usuario con este viaje es SIN_PLAZA.

Otros

Se han creado varios métodos en la clase de prueba que consideramos que las acciones en el método pueden ser común para algunas pruebas para evitar duplicidad de los códigos.

Otras informaciones

Pruebas unitarias

Algunas pruebas pasan de primera vez sin error, pero no del segundo ya que el base de datos esta modificado por primera prueba. Existe pruebas que depende la ejecución de la prueba anterior, de manera que si no había ejecutado el anterior no pasa la prueba en este. Durante la prueba es posible que casque algo, ya que alguna vez por el problema del navegador algún elemento esperado por la prueba puede no aparecer, pero he probado muchas veces que debería pasar todas las pruebas, si encuentra alguna esto problema si ejecuta otra vez debería pasar la prueba. En peor de los casos como había modificado cosas de base de datos, se recomienda recargue la base de datos original y empezar la prueba de nuevo todos.

Importar el proyecto sdi2-180.

Al importar el proyecto sdi2-180 es posible que pierde la referencia a la librería **Web App Libraries** y sale un motón de error en el proyecto para solucionar el problema hay que hacer lo siguiente: **1**.Right click al proyecto sdi-180-> Properties **2**.seleccionar Project Facets -> Check Dyanmic Web Module and Java **3**.Apply Setting.

Ahora tienes que ver la librería **Web App Libraries** con los jar por dentro y desaparecer todos los errores que ha surgido del importar proyecto.

Despliegue del archivo sdi2-180.war

El archivo sdi2-180.war se debe depositar en la carpeta S:\wildfly\standalone\deployments para que se despligue correctamente en el servidor.