PRÁCTICA 1 SDI

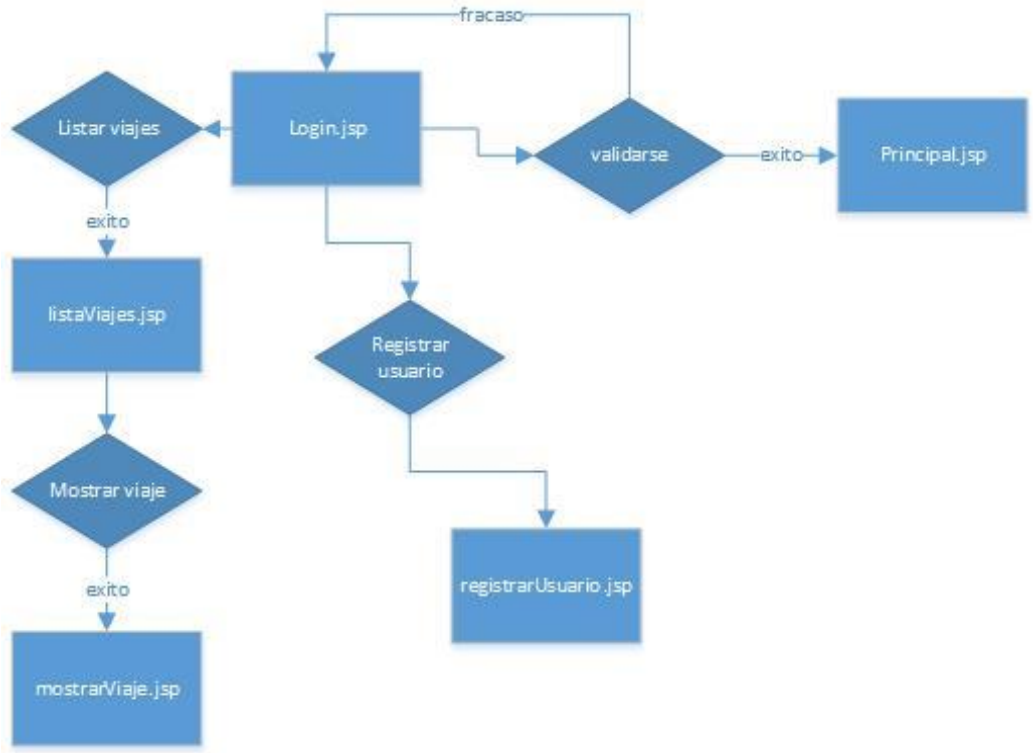
David Sariego Fernández

UO223935

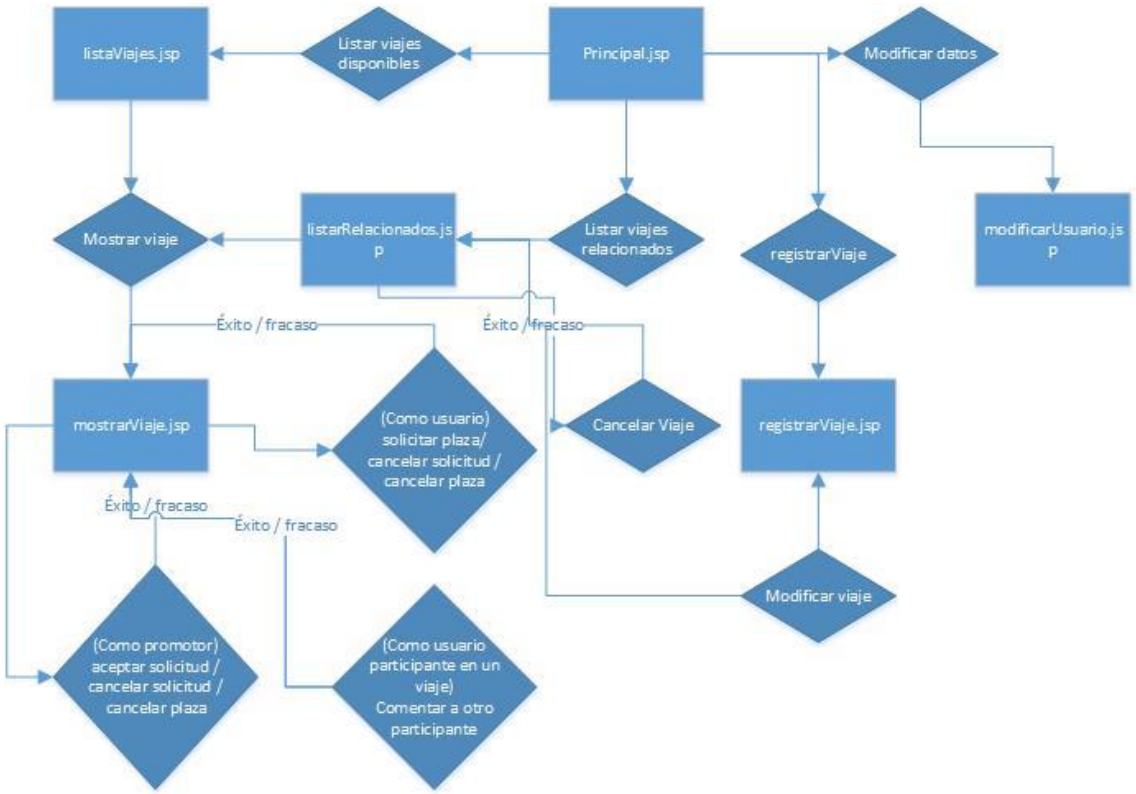
Contenido

Mapa de navegación

Como usuario publico



Como usuario registrado



Descripción de las acciones

Listener: ActualizarBD

Cada vez que se despliega la aplicación, se ejecutará este listener que comprobará que todos los datos en la base de datos sean consistentes.
-Comprueba que todos los viajes cuya fecha de cierre haya pasado no se encuentren en el estado 0 (OPEN)
-Comprueba, por si se realiza alguna modificación a mano o algo similar, que no haya datos inconsistentes del tipo de que un viaje que no tenga plazas disponibles siga abierto.
-Todas las solicitudes de los viajes abiertos pasan a ser seats con estado sin plaza.

AceptarSolicitudAction

Como Usuario PROMOTOR quieres aceptar a otro usuario que ha solicitado plaza en el viaje:
-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que el viaje siga activo y que el usuario en sesión es el promotor del viaje
-Recoge y borra el objeto Application (la solicitud)
-Comprueba que en el viaje queden plazas disponibles (teóricamente esto no haría falta pero lo he contemplado igual por si se modifica a mano la base de datos provocando estados inconsistentes de los datos)
-Crea un objeto Seat en estado aceptado con los datos del usuario y viaje y lo guarda en la base de datos.
-Decrementa en 1 el número de plazas disponibles en el viaje. Si con esto, las plazas disponibles llegan a cero, cerrará el viaje y pasará todas las peticiones pendientes a Seats en estado SIN_PLAZA

CancelarPlazaAction

Como usuario participante en el viaje, quieres cancelar tu plaza en un viaje todavía activo:
-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que el viaje esté activo
-Busca el objeto seat en la base de datos, cambia si estado a EXCLUDED y lo vuelve a guardar en la base de datos
-Incrementa en uno el número de plazas disponibles en el viaje
-Si el viaje estaba cerrado porque no quedaban plazas disponibles, pero aún no ha pasado la fecha de cierre, cambia su estado a abierto

CancelarSeatAction

Como promotor del viaje, quieres cancelar la plaza de un usuario participante que estaba aceptado:
-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que el viaje siga activo y que el usuario en sesión es el promotor del viaje
-Busca el seat en la base de datos, pone el estado a EXCLUDED y lo vuelve a guardar en la base de datos.
-Incrementa en uno el número de plazas disponibles en el viaje
-Si el viaje estaba cerrado porque no quedaban plazas disponibles, pero aún no ha pasado la fecha de cierre, cambia su estado a abierto

CancelarSolicitudAction

Como usuario interesado en un viaje, has realizado una petición para unirte a él, pero aún no has sido aceptado por el promotor, y quieres cancelar dicha solicitud:
-Comprueba que se le pasen todos los datos necesarios correctamente
-Busca el objeto Application en la base de datos y lo borra

CancelarSolicitudComoPromotorAction

Como usuario promotor de un viaje, otro usuario interesado ha realizado una solicitud para unirse a tu viaje:
-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que el usuario conectado sea el promotor del viaje
-Busca el objeto Application en la base de datos y lo borra. Si por algún problema que no debería darse, no existe la solicitud que se desea cancelar, dará un error.

CancelarViajeAction

-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que el usuario conectado sea el promotor del viaje
-Comprueba que el viaje no haya pasado ya
-Cambia el estado del viaje a CANCELLED
-Para todas las plazas (objetos Seat) pone su estado en EXCLUDED y lo guarda
-Para todas las solicitudes (objetos Application) las borra

CerrarSesionAction

-Comprueba si hay sesión abierta, si la hay la cierra, en caso contrario da error

ComentarSeatAction

-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que el usuario que realiza el comentario no sea el mismo que sobre quién se realiza
-Comprueba que el usuario conectado sea el promotor del viaje o haya participado en él
-Comprueba que ese usuario no haya realizado ya una valoración sobre ese otro usuario en ese viaje
-Guarda el objeto Rating

ExitoAction

Para navegaciones de una página a otra, donde no necesitamos realizar ningún tipo de comprobación, usaremos este action que devuelve siempre éxito para poder navegar.

FiltrarAction

Para la página listaViajes.jsp, realiza la acción de filtrar por la cadena introducida:
-Comprueba si se ha introducido algún dato, si no se introdujo, devolverá todos los viajes de forma normal como si no hubiese filtro (para quitar un filtro que hubieses metido antes)
-Llama a la base de datos pasándole la cadena por la que filtrar y mete la nueva lista en la request

FiltrarRelacionadosAction

Lo mismo que el FiltrarAction, pero para la página listarRelacionados.jsp

ListarviajesAction

Lista todos los viajes disponibles, si hay un usuario conectado, listará todos los disponibles en los que no esté participando ya.
-Recoge en un parámetro "orderBy" si se desea ordenar de alguna forma los viajes.
-En función de esto, y de si hay usuario conectado o no, llamará a un método diferente para devolver la lista de viajes apropiada

ListarViajesRelacionadosAction

Lista todos los viajes en los que un usuario participa o promociona
-Recoge en un parámetro "orderBy" si se desea ordenar de alguna forma los viajes.
-En función de esto, y de si hay usuario conectado o no, llamará a un método diferente para devolver la lista de viajes apropiada

ModificarDatosAction

-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que se haya introducido bien la contraseña (en caso contrario no hará nada)
-Modifica los datos
-Si además, los campos passwordNueva y passwordNueva2 tienen algo introducido, comprueba que sean iguales, en caso afirmativo, cambia la contraseña

ModificarViajeAction

Es un action solo para navegar al jsp de registrarViaje.jsp, con la única peculiaridad de que introduce en la request el viaje que se desea modificar para que la jsp muestre los datos.

MostrarViajeAction

Cuando desde alguno de los jsp que listan viajes, se desea ver los detalles de algún viaje, este action cargará todos los datos necesarios para el viaje.
-Comprueba que se le pasen todos los datos necesarios correctamente
-Busca el viaje en la base de datos y lo mete en la request
-Si hay usuario conectado:
-si no es el promotor:
*Comprueba si tiene plaza en el viaje, si la tiene también la mete en la request, si no la tiene comprueba si tiene una solicitud pendiente y hace lo mismo
-Si el usuario es promotor:
*Carga una lista de todas las solicitudes pendientes para ese viaje y las mete en la request
-A continuación, para ambos, mete todos los asientos vinculados al viaje y sus ratings

RegistrarUsuarioAction

Como usuario público, deseo registrarme
-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que no exista ya un usuario con ese login en la base de datos
-Comprueba que las dos contraseñas que se han introducido coincidan
-Guarda el usuario y lo mete en la sesión

RegistrarViajeAction

Como usuario registrado, deseo registrar un nuevo viaje
-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que no haya inconsistencias en los datos introducidos
*El número de plazas sea mayor que 0
*El coste estimado sea mayor de 0
*Las 3 fechas introducidas sean válidas (primero fecha de cierre, luego fecha de salida y la más lejana fecha de llegada)
*Creo el viaje, introduzco todos sus datos y lo guardo en la base de datos.

SolicitarPlazaAction

Como usuario registrado estoy interesado en un viaje y deseo solicitar una plaza para participar en el viaje
-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que el viaje siga activo
-Comprueba que el usuario no sea promotor del viaje
-Comprueba que el viaje tenga plazas disponibles
-Comprueba que no se haya realizado ya una solicitud
-Por como tengo hecha la interfaz, si el usuario tiene una plaza como excluido o sin plaza en el viaje, al realizar la nueva solicitud, se borrará esta entrada en la tabla Seats (para ser reemplazada por la nueva cuando el usuario sea aceptado o cancelado)
-Crea y guarda la solicitud (objeto Application)

ValidarseAction

Cuando te logueas en la aplicación con una cuenta ya existente
-Comprueba que se le pasen todos los datos necesarios correctamente
-Comprueba que no haya un usuario ya logueado en la sesión
-Comprueba que el usuario y contraseña ha sido bien introducidos
-Guarda el objeto user en sesión

Partes opcionales implementadas

OPCIONAL A: El usuario podrá filtrar el listado anterior para localizar rápidamente los viajes de su interés (hasta 0,5 puntos)
OPCIONAL B: El usuario podrá seleccionar el tipo de ordenación para los viajes: origen, destino e id (hasta 0,5 puntos)
OPCIONAL C: El usuario podrá filtrar el listado anterior para localizar rápidamente los viajes de su interés (hasta 0,5 puntos)
OPCIONAL D: El usuario podrá seleccionar el tipo de ordenación para los viajes: origen, destino e id (hasta 0,5 puntos)
OPCIONAL E: Una vez ha pasado la fecha del viaje, los usuarios participantes y el promotor podrán comentar y otorgar puntuaciones al resto de compañeros del viaje (ningún usuario podrá comentarse o puntuarse a sí mismo) [..] (hasta 1 punto)
¿OPCIONAL H?:No usé etiquetas de acción ya que no las requerí en ningún momento (únicamente usé if y for), usé EL, aunque no usé ni una sola vez scriptlets, que creo que es lo que “solicita” mayormente esta opcional, por eso lo pongo aquí

Opccional sin letra: Realicé pruebas junit extra para probar toda la funcionalidad, aunque la prueba para probar el "opcional e" la he dejado comentada, ya que por algún motivo que no he conseguido comprender, no encuentra el campo para introducir la valoración.

Información extra

-En la carpeta raíz del proyecto tengo un fichero "data.zip" con la base de datos en su estado inicial, para que, descomprimiendo y sobrescribiendo, vuelvas la base de datos a su estado inicial
-Las pruebas de introducir un nuevo viaje y cancelar un viaje solo funcionarán las primeras veces que se ejecutan si no se vuelve la base de datos a su estado inicial.
-El estado inicial de la base de datos consta de los usuarios y viajes solicitados en el enunciado, pero para poder ver la funcionalidad completa de la aplicación:
-Cada usuario promociona dos viajes, uno abierto y otro que ya pasó (para poder introducir ratings)
-El usuario "user2" participa en todos los viajes
-El usuario "user1" participa en los dos viajes de "user2"
-El usuario "user3" participa en el viaje id 31 (propiedad de "user2")
-El usuario "user3" solicita plaza en el viaje 30 (propiedad de "user1")

De esta forma, únicamente con loguearte con "user1"
-Puedes ver, si vas a viajes disponibles, que únicamente muestra el viaje id 42 (propiedad de "user3") ya que los viajes de user2 ya participa en ellos, el otro viaje de "user3" ya pasó y por lo tanto no debería salir, y los propios no deberían salir aquí tampoco.
Entrando en este viaje podrás solicitar plaza (y cancelar la solicitud una vez solicitada)
-En viajes relacionados podrás ver los 4 viajes con los que está relacionados:
*Sus dos viajes que promociona, el que ya pasó no podrá borrarlo ni modificarlo, pero el viaje que no pasó todavía sí tendrá estas opciones disponibles (29 y 30 respectivamente)
*Si entra en el viaje 30, podrá cancelar plaza del user2 que ya está sentado, o aceptar/cancelar la plaza del user3 que solicitó plaza.
*Si entra en el viaje 29 podrá realizar un comentario sobre un usuario participante en el viaje como promotor del viaje
*Si entra en el viaje 31 podrá realizar un comentario sobre otros usuarios participantes en el viaje (pero no sobre sí mismo)
*Si entra en el viaje 39 podrá cancelar su plaza ya aceptada en el viaje

Por otro lado, realicé un listener, aunque no es solicitado en el enunciado, que comprueba que los datos son consistentes siempre que se despliega la aplicación (Descrito en el apartado "Descripción de las acciones")