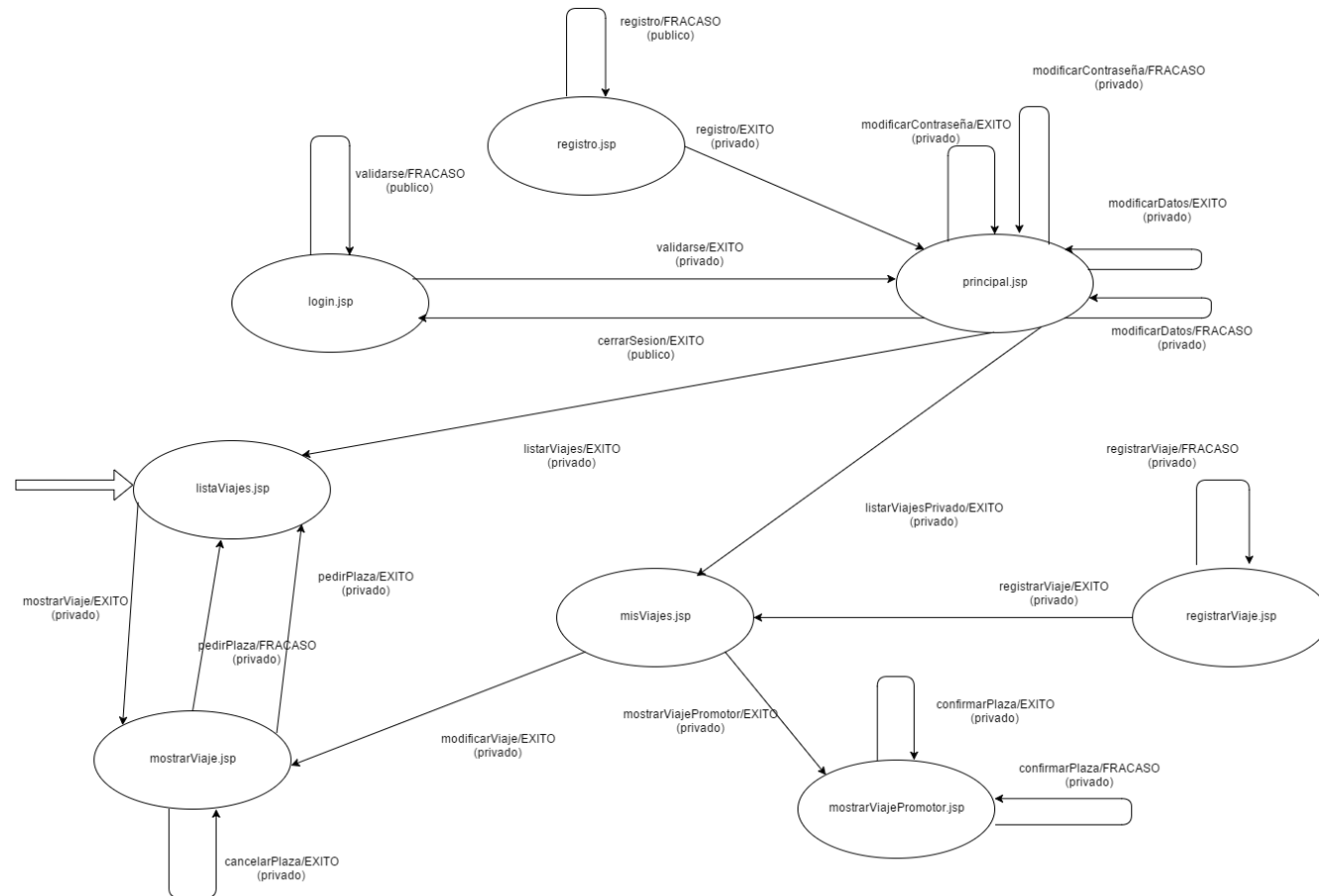


Mapa de navegación:



*Desde todas las páginas JSP se puede navegar sin acción asociada como:

REGISTRADO: listaViajes.jsp, misViajes.jsp, principal.jsp

PUBLICO: listaViajes.jsp, login.jsp, registro.jsp

Descripción de las acciones:

CancelarPlaza

Borrar las entradas en las tablas applications y seats para el usuario y el viaje que se le pide. El id del usuario lo obtiene de la sesión, mientras que el id del viaje de la petición.

Si la plaza ya había sido confirmada por el promotor se suma 1 a las plazas disponibles.

CerrarSesion

Utiliza el método invalidate del objeto sesión para borrar todo el contenido y hacer que el usuario pase del rol "REGISTRADO" a "PUBLICO".

ConfirmarPlaza

Elimina la entrada de application correspondiente a un usuario y un viaje, y lo introduce en la tabla seats. En caso de que el parámetro action sea "aceptar", el asiento tendría el estado "ACEPTADO", si el parámetro es "cancelar", este sería "EXCLUIDO"

ListarViajes

Si no recibe el en la petición el parámetro "criterio" devuelve un listado de todos los viajes en la base de datos con plazas libres, estado abierto y fecha de límite de admisión de petición posterior al día actual.

En caso de recibir el parámetro "criterio" lo utiliza para solo mostrar los viajes en los que la ciudad de destino o de partida contenga total o parcialmente el contenido del "criterio"

ListarViajesPrivados

Recopila todos los viajes en los que está involucrado el usuario como promotor, admitido, pendiente, excluido o sin plaza. Una vez recopilados los viajes devuelve una lista de objetos MiViajeConEstado, que no es más que un contenedor que guarda el viaje (objeto Trip) y la relación del usuario con él.

El objeto MiViajeConEstado también tiene un método auxiliar isCancelable que ayuda al jsp que los lista a saber si debe mostrar el botón de cancelar el viaje.

ModificarContraseña

Se solicita la contraseña antigua al usuario, y se pide la nueva por duplicado. Si coincide, se actualiza la contraseña.

ModificarDatos

Se pide al usuario que introduzca los campos de los campos que desea cambiar, y estos se actualizan siempre y cuando no sean una cadena en blanco

MostrarViaje

Devuelve el objeto trip que se le pide pasándole como parámetro el id del viaje. También recopila la información, valoración y comentarios del promotor y los participantes aceptados y la devuelve en un objeto RaitingsContainer que tiene tres métodos: getAverageRatingValue, getComments y isEnElViaje, que ayudan a la vista a obtener la valoración media, los comentarios de cada usuario y saber si el usuario está ya aceptado en viaje o es el promotor para no mostrarle el botón de solicitar plaza.

MostrarViajePromotor

Si el usuario es promotor de un viaje, mostrará la información de un viaje pasando el id como parámetro. Mostrará todos los datos de las personas que ya son participantes de dicho viaje, y las personas que solicitan viaje. Desde aquí se podrá llamar a la acción ConfirmarPlaza pasándole un atributo "action" para confirmar o denegar al usuario. No puede confirmar ni denegar si la fecha actual es posterior a la fecha límite del viaje.

PedirPlaza

Registra una petición del usuario que ha iniciado sesión en la tabla applications del viaje en el que se encuentra actualmente, para ello recibe como parámetro el id del viaje.

En caso de que el usuario ya haya solicitado una plaza, se le mostrara un mensaje de error indicando que debe esperar a que el promotor le acepte.

Registrarse

Primero comprueba que los datos introducidos por el usuario en el formulario son correctos, ninguno en blanco y que las contraseñas coinciden, y si es así inserta el usuario y sus datos a la base de datos.

En caso de que ya exista un usuario con el mismo login o que alguno de los campos del formulario no esté correctamente rellenado.

RegistrarViaje

FALTA EL RESTO DE LA EXPLICACIÓN

En caso de que el usuario ya haya registrado un viaje para en la misma fecha, se le mostrara un mensaje de error indicando que ya que tiene un viaje para esa fecha.

Validarse

Comprueba que el login y contraseña que ha introducido el usuario se corresponde con una entrada en la base de datos y en ese caso le devuelve un objeto User que se guardará en la sesión y que permitirá identificar al usuario como registrado.

En caso de ocurrir un error en la validación se mostrará un mensaje de error al usuario.

Partes opcionales:

Opcional A. El usuario podrá filtrar el listado anterior para localizar rápidamente los viajes de su interés.

En listarViajes.jsp incluimos un cuadro de texto al inicio del listado que permite buscar por ciudad de origen o destino, todo ello funciona con el form que hace una petición get a la clase ListarViajesAccion que sí detecta que la petición incluye un parámetro criterio lo utiliza otra consulta SQL mediante el TripDao que lo utiliza para filtrar por el nombre de la ciudad tanto origen como destino.

Opcional C. El usuario podrá filtrar el listado anterior para localizar rápidamente los viajes de su interés.

En misviajes.jsp incluimos un cuadro de texto para buscar por ciudad de origen o destino, todo ello funciona en la lógica, ya que se hace busque de los viajes en los que se participa, y en los viajes en los que el usuario es promotor. Tras seleccionar los viajes que cumplen esa condición, se añaden a la lista y se muestran al usuario.

Opcional G. Uso de layout web.

Para el diseño de la interfaz de SharaMyTrip hemos utilizado el framework Bootstrap, que aparte de contener todo tipo de elementos personalizados como: tablas, barras de navegación... y que nos permite hacer un mejor diseño sin necesidad de dedicar tiempo a la construcción de un CSS propio incluye su propio sistema de grid que es el que hemos usado en nuestra implementación.

Opcional H. Uso de etiquetas de acción.

A lo largo de todos los .JSPs de la aplicación hemos usado etiquetas de la librería “core” para ocultar/mostrar información si el usuario ha iniciado sesión en la aplicación, para mostrar todo el contenido de una lista y para redirigir si el usuario trataba de entrar a una zona que requiere que haya iniciado sesión o un preprocesamiento y él ha ido directamente al JSP.

También de la librería “fmt” para dar formato a las fechas de los viajes.

Otra información:

Entregamos un script de iniciación, “script.sql”, con los datos que se pide, por defecto la base de datos solo viene con lo necesario para que pasen los test que se piden.

Para poder ejecutar los test por segunda vez adjuntamos un script, “preparar_db_test.sql”, que contiene las sentencias a ejecutar para que la base de datos esté lista para los test y no de errores porque los datos ya existan.