

INGENIERÍA DE REQUISITOS

Modelo conceptual y contratos

Alicia Rodríguez Gómez
Francisco Javier Morales Piqueras
Rubén Morales Pérez
Samia Mikou

17 de mayo de 2017

Índice

1. Introducción	1
2. Diagramas realizados por Alicia Rodríguez Gómez	1
3. Diagramas realizados por Francisco Javier Morales Piqueras	7
4. Diagramas realizados por Rubén Morales Pérez	12
5. Diagramas realizados por Samia Mikou	17

1. Introducción

En este documento se adjuntan algunos de los contratos de las operaciones del sistema *Tapuntas*, basándonos en el modelo conceptual adjunto a la práctica.

2. Diagramas realizados por Alicia Rodríguez Gómez

Nombre	eliminarSolicitudesAlquilerRechazadasCaducadas()
Responsabilidad	Elimina la solicitudes de alquiler que están caducadas o rechazadas
Tipo	Tapuntas
Notas	
Excepciones	
Salida	
Precondiciones	<ul style="list-style-type: none"> - Existan solicitudes de alquiler rechazadas pendientes de ser eliminadas - Existan solicitudes de alquiler caducadas pendientes de se eliminadas
Poscondiciones	<ul style="list-style-type: none"> - Las solicitudes de alquiler rechazadas han sido eliminadas - Las solicitudes de alquiler caducadas han sido eliminadas

Nombre	eliminarPlanesAlquileresNoUsadosCaducados()
Responsabilidad	Elimina los planes de alquiler que no han sido usados y se han caducado.
Tipo	Tapuntas
Notas	
Excepciones	
Salida	
Precondiciones	<ul style="list-style-type: none"> - Existan planes de alquiler que no han sido usados y que han caducado en el tiempo.
Poscondiciones	<ul style="list-style-type: none"> - Los planes de alquiler no usados y que han caducado en el tiempo han sido eliminados

Nombre	nuevoPlanViaje(idViaje,idVehiculo,plazas,dia,origen,destino,comentario)
Responsabilidad	Planificar un viaje nuevo por un conductor, en el cual se informa del vehículo, fecha y lugares donde se inicia el viaje y donde se finaliza.
Tipo	Tapuntas
Notas	
Excepciones	<ul style="list-style-type: none"> - Si el Viaje identificado por idViaje no existe - Si el Vehículo identificado por idVehiculo no existe - Si la fecha identificada por día no está dentro de una rango apropiado - Si $!(0 < plazas < 4)$ - Si Parada identificada por origen y por destino no existen
Salida	idPlanViaje
Precondiciones	- El vehículo identificado por idVehiculo no tiene ningún viaje previsto para la fecha identificado por día
Poscondiciones	<ul style="list-style-type: none"> - Fue creado un objeto, idPlanViaje, de la clase PlanViaje - Fue creado un enlace entre idPlanViaje y el objeto de la la clase Vehículo identificado por idVehiculo - Fue creado un enlace entre idPlanViaje y el objeto de la clase Viaje identificado por idViaje - Fue creado un enlace entre idViaje y un objeto de la clase Parada

Nombre	modificarInfoParada(idParada, hora, coste)
Responsabilidad	Modificar información sobre una parada de un viaje
Tipo	Tapuntas
Notas	Este método se aplica para todas las paradas de las que está compuesta un viaje
Excepciones	<ul style="list-style-type: none"> - Si la Parada identificado por idParada no existe - Si la hora identificada por hora no está dentro de un rango apropiado - Si $!(coste > 0)$
Salida	
Precondiciones	- Que estén todas las paradas definidas
Poscondiciones	- Los objetos identificados por idParara, de la clase Parada, establecen un nuevo valor de hora y coste

Nombre	ofertarPlanViaje(idPlanViaje)
Responsabilidad	Anuncio, de un viaje planificado, al resto de usuarios
Tipo	Tapuntas
Notas	
Excepciones	- Si el PlanViaje identificado por idPlanViaje no existe
Salida	
Precondiciones	- Que el Viaje asociado a PlanViaje está definido
Poscondiciones	- Fue creado un enlace entre idPlanViaje y un objeto de la clase Viaje

Nombre	solicitarViaje(idPlanViaje, idUsuario, origenViajero, destinoViajero, formaPago)
Responsabilidad	Solicitar un viaje, previamente publicado por un usuario, donde se elige el lugar de origen y destino y la forma de pago del viaje.
Tipo	Tapuntas
Notas	
Excepciones	<ul style="list-style-type: none"> - Si el PlanViaje identificado por idPlanViaje no existe - Si el Usuario identificado por idUsuario no existe - Si el TipoTransaccion identificada por formaPago no existe
Salida	
Precondiciones	<ul style="list-style-type: none"> - El PlanViaje identificado por idPlanViaje está registrado en la base de datos - La Parada identificada por origenViajero está dentro del Viaje identificado por idPlanViaje - La Parada identificada por destinoViajero está dentro del Viaje identificado por idPlanViaje
Poscondiciones	<ul style="list-style-type: none"> - Se ha creado un objeto de tipo Viaje - Se ha creado un enlace entre el objeto identificado por idPlanViaje de tipo PlanViaje y el objeto de tipo Viaje - Se ha creado un enlace entre el objeto identificado por origenViajero de tipo Parada y el objeto de tipo Viaje. - Se ha creado un enlace entre el objeto identificado por destinoViajero de tipo Parada y el objeto de tipo Viaje - Se ha creado un enlace entre el objeto identificado por idUsuario de tipo Usuario y el objeto de tipo Viaje

Nombre	aceptarAcompañante(idPlanViaje, idUsuario)
Responsabilidad	Aceptación de la solicitud de un usuario a un viaje ofertado
Tipo	Tapuntas
Notas	
Excepciones	<ul style="list-style-type: none"> - Si el PlanViaje identificado por idPlanViaje no existe - Si el Usuario identificado por idUsuario no existe
Salida	
Precondiciones	El Usuario identificado por idUsuario ha organizado un PlanViaje identificado por idPlanViaje
Poscondiciones	<ul style="list-style-type: none"> - Se crea un objeto de tipo UsuarioPropietario - Se crea un objeto de tipo Viaje - Se crea un enlace entre el Usuario identificado por idUsuario y el objeto creado de tipo UsuarioPropietario - Se crea un enlace entre el Usuario identificado por idUsuario y el objeto creado de tipo Viaje - Se crea un enlace entre el PlanViaje identificado por idPlanViaje y el objeto creado de tipo Viaje - Se crea un enlace entre el PlanViaje identificado por idPlanViaje y el objeto creado de tipo UsuarioPropietario

Nombre	valorarAcompañante(idUsuarioValora, idUsuarioValorado, puntuación, comentario)
Responsabilidad	Puntuación a un usuario que ha sido acompañante durante un viaje realizado.
Tipo	Tapuntas
Notas	
Excepciones	<ul style="list-style-type: none"> - Si el Usuario identificado por idUsuarioValorado no existe - Si el Usuario identificado por idUsuarioValora no existe - Si $!(0 \leq \text{puntuacion} \leq 5)$
Salida	
Precondiciones	El Usuario identificado por idUsuarioValora ha realizado un viaje con el Usuario identificado por idUsuarioValorado

Poscondiciones	<ul style="list-style-type: none"> - Se crea un objeto de tipo ValoraciónAcompañante - Se crea un objeto de tipo ValoraciónUsuario - Se crea un objeto de tipo ValoraciónUsuario identificada por realizadas - Se crea un objeto de tipo ValoraciónUsuario identificada por recibidas - Se crea un enlace entre el objeto de tipo ValoraciónAcompañante y el objeto de tipo ValoraciónUsuario - Se crea un enlace entre el objeto de tipo ValoraciónUsuario identificada por realizadas y el objeto de tipo Usuario identificado por idUsuarioValora - Se crea un enlace entre el objeto de tipo ValoraciónUsuario identificada por recibidas y el objeto de tipo Usuario identificado por idUsuarioValorado
-----------------------	---

3. Diagramas realizados por Francisco Javier Morales Piqueras

Nombre	aceptarSolicitudAlquiler(idUsuario, idPlanAlquiler)
Responsabilidad	Dar como aceptada la solicitud del usuario identificado por idUsuario en el alquiler identificado por idPlanAlquiler .
Tipo	Tapuntas
Notas	
Excepciones	-Si el usuario identificado por idUsuario no existe -Si el plan de alquiler identificado por idPlanAlquiler no existe
Salida	
Precondiciones	-El usuario identificado por idUsuario ha solicitado previamente el plan de alquiler identificado por idPlanAlquiler
Poscondiciones	-El estado del viaje del usuario identificado por idPlanAlquiler pasa a ACEPTADO

Nombre	rechazarSolicitudAlquiler(idUsuario, idPlanAlquiler)
Responsabilidad	Dar como rechazada la solicitud del usuario identificado por idUsuario en el alquiler identificado por idPlanAlquiler .
Tipo	Tapuntas
Notas	
Excepciones	-Si el usuario identificado por idUsuario no existe -Si el plan de alquiler identificado por idPlanAlquiler no existe
Salida	
Precondiciones	-El usuario identificado por idUsuario ha solicitado previamente el plan de alquiler identificado por idPlanAlquiler
Poscondiciones	-El estado del viaje del usuario identificado por idPlanAlquiler pasa a RECHAZADO

Nombre	realizarPagoAlquiler(idUsuario, idAlquiler, detallesPago)
Responsabilidad	Que el usuario identificado por idUsuario pague el alquiler identificado por idAlquiler según los detalles establecidos en detallesPago .
Tipo	Tapuntas
Notas	
Excepciones	-Si el usuario identificado por idUsuario no existe -Si el alquiler identificado por idAlquiler no existe -Si la información de detallesPago no es correcta
Salida	
Precondiciones	-El usuario identificado por idUsuario había contratado previamente el alquiler identificado por idAlquiler -La formaPagoElegida del alquiler identificado por idAlquiler no es EFFECTIVO
Poscondiciones	-El usuario propietario del alquiler identificado por idAlquiler ha recibido el dinero

Nombre	valorarAlquiler(idUsuario, idAlquiler, puntuación, comentario)
Responsabilidad	Que el usuario identificado por idUsuario valore el alquiler identificado por idAlquiler con la puntuación puntuación y el comentario comentario .
Tipo	Tapuntas
Notas	
Excepciones	-Si el usuario identificado por idUsuario no existe -Si el alquiler identificado por idAlquiler no existe -Si la puntuación no es válida
Salida	
Precondiciones	-El usuario identificado por idUsuario había contratado previamente el alquiler identificado por idAlquiler
Poscondiciones	-Se ha creado un objeto del tipo ValoraciónAlquiler -Se crea un enlace entre el objeto de tipo ValoraciónAlquiler y el alquiler identificado por idAlquiler

Nombre	valorarUsoAlquiler(idUsuario, idPropietario, puntuación, comentario)
Responsabilidad	Que el propietario identificado como idPropietario valore al usuario identificado por idUsuario con la puntuación puntuación y el comentario comentario .
Tipo	Tapuntas
Notas	
Excepciones	-Si el usuario identificado por idUsuario no existe -Si el propietario identificado por idPropietario no existe
Salida	
Precondiciones	-El usuario identificado por idUsuario ha contratado algún alquiler cuyo propietario es idPropietario
Poscondiciones	-Se crea un objeto de tipo ValoraciónUsoAlquiler -Se crea un enlace entre el objeto de tipo ValoraciónUsoAlquiler y el usuario identificado por idPropietario con el rol de realizada -Se crea un enlace entre el objeto de tipo ValoraciónUsoAlquiler y el usuario identificado por idUsuario con el rol de recibida

Nombre	cancelarSolicitudAlquiler(idUsuario, idAlquiler)
Responsabilidad	Que un usuario identificado por idUsuario cancele una solicitud suya de un alquiler identificado por idAlquiler .
Tipo	Tapuntas
Notas	
Excepciones	-Si el usuario identificado por idUsuario no existe -Si el alquiler identificado por idAlquiler no existe
Salida	
Precondiciones	-El usuario identificado por idUsuario ha solicitado previamente el alquiler identificado por idAlquiler
Poscondiciones	-La solicitud ha sido cancelada

Nombre	cancelarPlanAlquiler(idPlanAlquiler)
Responsabilidad	Cancelar un plan de alquiler identificado por idPlanAlquiler
Tipo	Tapuntas
Notas	
Excepciones	-Si el plan de alquiler identificado por idPlanAlquiler no existe
Salida	
Precondiciones	
Poscondiciones	-El objeto de tipo PlanAlquiler identificado por idPlanAlquiler es eliminado

4. Diagramas realizados por Rubén Morales Pérez

Nombre	altaRegistro(nombreUsuario, contraseña)
Responsabilidad	Crear una nueva cuenta de usuario en el sistema
Tipo	Tapuntas
Notas	
Excepciones	<ul style="list-style-type: none"> - El nombreUsuario ya existe en el sistema - La contraseña es demasiado corta, mínimo 8 caracteres.
Salida	
Precondiciones	
Poscondiciones	Existe un nuevo usuario cuyo nombre de usuario y contraseña coinciden con los parámetros de la función.

Nombre	identificarse(nombreUsuario, contraseña)
Responsabilidad	Acceso de un usuario al sistema
Tipo	Tapuntas
Notas	
Excepciones	- La contraseña no se corresponde con el nombreUsuario especificado en los parámetros.
Salida	
Precondiciones	- nombreUsuario existe dentro del sistema
Poscondiciones	

Nombre	obtenerDatosUsuario(idUsuario)
Responsabilidad	Obtener toda la información respectiva al usuario identificado por idUsuario dd, las asociaciones, las variables heredadas de la clase UsuarioRegistrado y, en caso de pertenecer a la clase UsuarioPropietario , también los datos referentes a dicha clase
Tipo	Tapuntas
Notas	
Excepciones	- El parámetro idUsuario no se corresponde con ningún usuario del sistema
Salida	Variable datosUsuario
Precondiciones	
Poscondiciones	<u>La variable datosUsuario está correctamente inicializada</u>

Nombre	<u>introducirPerfil(nombre, teléfono, añoNacimiento, breveDescPersonal, foto, direccionPostal)</u>
Responsabilidad	Una vez registrado el usuario con esta función almacena la información principal de su perfil.
Tipo	Tapuntas
Notas	
Excepciones	
Salida	<u>El identificador del usuario que llama a la función, idUsuario</u>
Precondiciones	<ul style="list-style-type: none"> - El teléfono es válido, con su respectivo prefijo - El añoNacimiento es posterior a 1900 - La breveDescPersonal y la foto cumple las condiciones de uso del sistema - direcciónPostal válida
Poscondiciones	El perfil es modificado con los valores pasados como argumento.

Nombre	consultarPerfil(idUsuario)
Responsabilidad	Devolver la información de un usuario que sea susceptible de ser mostrada al resto de usuarios, la contraseña o la preferencia de pago no se devolverían.
Tipo	Tapuntas
Notas	
Excepciones	- El parámetro idUsuario no se corresponde con ningún usuario del sistema
Salida	infoPerfil
Precondiciones	El usuario ha insertado en su cuenta información susceptible de ser mostrada
Poscondiciones	

Nombre	establecerFormaPago(idUsuario, listaTipoTransaccion)
Responsabilidad	Definir el orden de preferencia en el TipoTransaccion de un UsuarioPropietario
Tipo	Tapuntas
Notas	
Excepciones	- Variable idUsuario no identifica a ningún usuario del sistema
Salida	
Precondiciones	- listaTipoTransaccion tiene hasta cuatro valores distintos dentro de los especificados en TipoTransaccion
Poscondiciones	Se ha actualizado la variable preferenciaCobro del UsuarioRegistrado con los valores especificados en listaTipoTransaccion

Nombre	añadirVehículo(idPropietario, matrícula, marca, modelo, color, numeroPlazas, categoría, confort)
Responsabilidad	Crear un vehículo y asociarlo con el propietario identificado por idPropietario
Tipo	Tapuntas
Notas	
Excepciones	<ul style="list-style-type: none"> - El parámetro idPropietario no identifica a ningún usuarios - La matrícula es válida
Salida	
Precondiciones	Tanto la marca , el modelo , color , numeroPlazas , categoría y confort son válidos
Poscondiciones	Tras crear el nuevoVehículo se ha creado un enlace entre el usuario identificado por idPropietario y dicho vehículo

5. Diagramas realizados por Samia Mikou

Nombre	definirPlanAlquiler(idPropietario, idVehiculo, fechaInicio, fechaFin, coste, ciudadRecogida)
Responsabilidad	Definir un plan de alquiler propuesto por un propietario, especificando el vehículo, la fecha de inicio , el coste y la ciudad de recogida.
Tipo	Tapuntas
Notas	
Excepciones	-Si el propietario identificado por idPropietario no existe. -Si el vehículo identificado por idVehiculo no existe. -Si la fechaInicio no está entre el rango apropiado. -Si la fechaFin no está en el rango apropiado. -Si la ciudad identificada por ciudadDeRecogida no existe.
Salida	
Precondiciones	-Que el propietario este registrado.
Poscondiciones	-Fue creado un objeto PlanAlquiler, plan , debidamente inicializado. -Fue creado un enlace entre Vehículo (identificado por idVehiculo) y plan . -Fue creado un enlace entre UsuarioPropietario (identificado por idPropietario) y plan .

Nombre	obtenerPlanesAlquiler(idPropietario)
Responsabilidad	Obtener los planes de alquiler del propietario identificado por idPropietario .
Tipo	Tapuntas
Notas	
Excepciones	-Si el propietario identificado por idPropietario no existe
Salida	misPlanesAlquiler.
Precondiciones	-Que el propietario identificado por idPropietario este registrado.
Poscondiciones	-Fue creado un enlace entre PlanAlquiler y UsuarioPropietario (identificado por idPropietario).

Nombre	ofertarPlanAlquiler(idPropietario, idPlanAlquiler)
Responsabilidad	Ofertar un plan de alquiler identificado por idPlanAlquiler .
Tipo	Apuntes
Notas	
Excepciones	-Si el propietario identificado por idPropietario no existe. -Si el plan Alquiler identificado por idPlanAlquiler no existe.
Salida	
Precondiciones	-Que el propietario este registrado. -Que el plan de alquiler identificado por idPlanAlquiler este definido.
Poscondiciones	-Se ha actualizado la variable visible de PlanAlquiler a true.

Nombre	buscarOfertasAlquiler(ciudadRecogida, fechaInicio, FechaFin)
Responsabilidad	Buscar ofertas de alquiler especificando la ciudad de recogida, la fecha de inicio y la fecha de fin.
Tipo	Tapuntas
Notas	
Excepciones	-Si la ciudad recogida no existe. -Si la fecha de inicio no está dentro del rango apropiado. -Si la fecha de fin no está dentro del rango apropiado.
Salida	listaOfertas
Precondiciones	
Poscondiciones	-Se buscan las ofertas de alquiler válidas.

Nombre	obtenerValoracionAlquiler(idPropietario,idVehiculo)
Responsabilidad	Obtener la valoración de alquiler de vehículo ofrecido por un usuario propietario.
Tipo	Tapuntas
Notas	
Excepciones	-Si el propietario identificado por idPropietario no existe. -Si el vehículo identificado por idVehiculo no existe.
Salida	valoracionesAlquiler.
Precondiciones	-Que el propietario identificado por idPropietario este registrado.
Poscondiciones	-Se ha creado un enlace entre UsuarioPropietario(identificado por idPropietario) y Vehículo (identificado por idVehiculo).

Nombre	solicitarAlquilerVehiculo(idUsuario,idPlanAlquiler,fechaInicio,fechaFin,formaPagoElegida)
Responsabilidad	Solicitar el alquiler de un vehículo especificando el plan de alquiler, la fecha de inicio, fecha de fin y la forma de pago.
Tipo	Tapuntas.
Notas	
Excepciones	Si el usuario identificado por idUsuario no existe. Si el plan de alquiler identificado por idPlanAlquiler no existe. Si la fechaInicio no está dentro de un rango apropiado. Si la fechaFin no está dentro de un rango apropiado.
Salida	
Precondiciones	-El usuario identificado por idUsuario este registrado. -El plan de alquiler identificado por idPlanAlquiler está registrado en la base de datos.
Poscondiciones	-Se ha creado un enlace entre PlanAlquiler (identificado por idPlanAlquiler) y Usuario (identificado por idUsuario).

Nombre	obtenerSolicitudesAlquiler(idPlanAlquiler)
Responsabilidad	Mostrar las solicitudes del plan de alquiler identificado por idPlanAlquiler .
Tipo	Tapuntas.
Notas	
Excepciones	-Si el plan de alquiler identificado por idPlanAlquiler no existe.
Salida	listaSolicitudes.
Precondiciones	-El plan de alquiler identificado por idPlanAlquiler este registrado en la base de datos.
Poscondiciones	

Nombre	obtenerValoracionesDeUso(idUsuario)
Responsabilidad	Obtener la valoración que tiene el usuario identificado por idUsuario por el uso que ha hecho de un recurso propuesto.
Tipo	Tapuntas
Notas	
Excepciones	-Si el usuario identificado por idUsuario no existe.
Salida	valoracionesDeUso
Precondiciones	-El usuario identificado por idUsuario este registrado.
Poscondiciones	-Se ha creado un enlace entre ValoraciónUsuario y Usuario (identificado por idUsuario)