



ugr

Universidad
de Granada

FUNDAMENTOS DE INGENIERIA DEL SOFTWARE

Práctica 3.2: Modelo conceptual y contratos

CONTRATOS



Integrantes del grupo C2.3

Francisco Gómez Correa

Javier Martín Gómez

Jonathan Martín Valera

Jorge Soler Padial

Contrato 1.1: solicitarRegistro (Javier)

| | |
|------------------------|---|
| Nombre | <i>solicitarRegistro(direccionCorreo)</i> |
| Responsabilidad | Solicitar el registro de usuario para el sistema |
| Tipo | Tapuntas |
| Notas | Se necesita verificar la dirección de correo para dar de alta en el sistema |
| Excepciones | ----- |
| Salida | ----- |
| Precondiciones | Tener una dirección de correo creada |
| Postcondiciones | ----- |

Contrato 1.2: altaRegistro (Jorge)

| | |
|------------------------|---|
| Nombre | <i>altaRegistro(nombreUsuario, contraseña)</i> |
| Responsabilidad | Crear un nuevo usuario registrado en el sistema con el nombre del usuario y la contraseña proporcionada. |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si el nombre de usuario ya existe o no es correcto. -Si la contraseña no es válida. |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none"> Se creó un nuevo objeto usuario registrado con su correspondiente nombreUsuario , contraseña y direccionCorreo. |

Contrato 1.3: identificarse (Jorge)

| | |
|------------------------|--|
| Nombre | <i>identificarse(nombreUsuario, contraseña)</i> |
| Responsabilidad | Comprobar que el nombreUsuario y contraseña sean correctos para dar acceso al sistema al usuario |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si el nombre de usuario no existe. -Si la contraseña no es válida. |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | ----- |

Contrato 1.5: *obtenerDatosUsuario* (Jorge)

| | |
|------------------------|---|
| Nombre | <i>obtenerDatosUsuario(idUsuario)</i> |
| Responsabilidad | Devuelve los datos de un usuario identificado por su idUsuario. |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si el idUsuario noexiste. |
| Salida | -Datos del usuario idUsuario. |
| Precondiciones | ----- |
| Postcondiciones | ----- |

Contrato 1.7: *solicitarBaja* (Javier)

| | |
|------------------------|--|
| Nombre | <i>solicitarBaja(idUsuario)</i> |
| Responsabilidad | Solicitar la baja del usuario del sistema |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | ----- |
| Salida | ----- |
| Precondiciones | - Se puede eliminar el usuario del sistema |
| Postcondiciones | <ul style="list-style-type: none">• Se eliminó todos los objetos de la clase Usuario identificado por idUsuario• Se eliminó el enlace entre el objeto identificado por idUsuario de la clase Usuario con los objetos de la clase Alquiler• Se eliminó el enlace entre el objeto identificado por idUsuario de la clase Usuario con los objetos de la clase Viaje |

Contrato 2.1: definirPlanAlquiler (Jonathan)

| | |
|------------------------|--|
| Nombre | <i>definirPlanAlquiler(idPropietario,idVehiculo,fechaInicio,fechaFin,coste,ciudadRecogida)</i> |
| Responsabilidad | Crear un plan de viaje para poder ofertarlo |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -El idPropietario debe ser válido. -El idVehículo debe de ser válido. -FechaInicio debe de ser menor que fecha Fin(en segundos) -FechaInicio y FechaFin debe de ser mayor que la fecha actual(en segundos) -El coste no puede ser negativo -La ciudad de recogida debe de ser válida |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">• Se creó un objeto de la clase PlanAlquiler identificado por idPropietario,idVehiculo,fechaInicio,fechaFin inicializado:• {estado=visible,primerDiaAlquiler=fechaInicio,ultimoDiaAlquiler=fechaFin,costeAlquilerDia=coste,ciudadRecogida=ciudadRecogida}• Se creó un enlace entre Vehiculo y PlanAlquiler• Se creó un enlace entre UsuarioPropietario y PlanAlquiler |

Contrato 2.2: ObtenerPlanesAlquiler (Jonathan)

| | |
|------------------------|---|
| Nombre | <i>ObtenerPlanesAlquiler(idPropietario)</i> |
| Responsabilidad | Devuelve los planes de alquiler que tiene un usuario específico |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -El idPropietario debe de ser válido |
| Salida | misPlanesAlquiler={ primerDiaAlquiler,ultimoDiaAlquiler,costeAlquilerDia,ciudadRecogida } |
| Precondiciones | ----- |
| Postcondiciones | ----- |

Contrato 2.3: ofertarPlanAlquiler (Jonathan)

| | |
|------------------------|--|
| Nombre | <i>ofertarPlanAlquiler(idPropietario,idPlanAlquiler)</i> |
| Responsabilidad | Hacer visible a los usuarios el plan de alquiler |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -IdPropietario debe de ser válido -idPlanAlquiler debe de ser válido |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">Se modifica el estado del atributo visibilidad a valor true identificado por el idPlanAlquiler e idPropietario |

Contrato 2.4: buscarOfertasAlquiler (Jonathan)

| | |
|------------------------|--|
| Nombre | <i>buscarOfertasAlquiler(ciudadRecogida, fechaInicio, fechaFin)</i> |
| Responsabilidad | Devolver las ofertas de alquiler que cumplan nuestros requisitos |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -FechaInicio y FechaFin debe de ser mayor que la fecha actual(en segundos) -FechaInicio debe de ser menor que fecha Fin(en segundos) -La ciudad de recogida debe de ser válida |
| Salida | listaOfertas=Lista de {primerDia,ultimoDia,costeAlquilerAlDia,ciudadRecogida} para todos los objetos de la clase PlanAlquiler que cumplan los requisitos de búsqueda |
| Precondiciones | ----- |
| Postcondiciones | ----- |

Contrato 2.5: obtenerValoracionAlquiler (Jonathan)

| | |
|------------------------|---|
| Nombre | <i>obtenerValoracionAlquiler(idPropietario,idVehiculo)</i> |
| Responsabilidad | Consultar las valoraciones que se han hecho en los alquileres de ese vehículo |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -El idPropietario debe ser válido. -El idVehículo debe de ser válido. -IdVehiculo no tenga ninguna valoración: Se mostrará mensaje diciendo que no tiene valoraciones |
| Salida | valoracionesAlquiler=lista de{ fecha,comentario,puntuacion} para todos los objetos de la clase ValoraciónAlquiler cuyo valor coincida con idPropietario e idVehiculo |
| Precondiciones | ----- |
| Postcondiciones | ----- |

Contrato 2.6: solicitarAlquilerVehiculo (Jonathan)

| | |
|------------------------|--|
| Nombre | <i>solicitarAlquilerVehiculo(idUsuario,idPlanAlquiler, fechaInicio,fechaFin,formaPagoElegida)</i> |
| Responsabilidad | Realizar una petición por parte de un usuario a un plan de alquiler ofertado |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -El idUsuario debe ser válido. -El idPlanAquiler debe de ser válido. -FechaInicio y FechaFin debe de ser mayor que la fecha actual(en segundos) -formaPagoElegida debe de estar permitida |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">• Se ha creado un enlace entre Usuario y Alquiler• Se ha creado un objeto de la clase Alquiler identificado por idUsuario,idPlanAlquiler,fechaInicio,fechaFin inicializado: { estado=pendiente,fechaInicio=fechaInicio,fechaFin=fechaFin,formaPagoElegida=formaPagoElegida}• Se ha creado un enlace entre Alquiler y PlanAlquiler. |

Contrato 2.7: *obtenerSolicitudesAlquiler* (Jonathan)

| | |
|------------------------|---|
| Nombre | <i>obtenerSolicitudesAlquiler(idPlanAlquiler)</i> |
| Responsabilidad | Obtener una lista de las solicitudes que tiene un plan de alquiler |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -idPlanAlquiler debe de ser válido -En caso de no haber ninguna solicitud, mostrar mensaje a usuario |
| Salida | listaSolicitudes= Lista de{idUsuario,fechaInicio,fechaFin,formaPagoElegida} para todos los objetos de la clase Alquiler cuyo idPlanAlquiler coincida |
| Precondiciones | ----- |
| Postcondiciones | ----- |

Contrato 2.8: *obtenerValoracionesDeUso* (Jonathan)

| | |
|------------------------|--|
| Nombre | <i>obtenerValoracionesDeUso(idUsuario)</i> |
| Responsabilidad | Obtener las valoraciones de uso de un usuario |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -idUsuario debe de ser válido -En caso de no haber ninguna valoración, mostrar mensaje a usuario |
| Salida | valoracionesDeUso=Lista de{ fecha,comentario,puntuación} para todos los objetos de la clase ValoraciónUsoAlquiler cuyo idUsuario coincida |
| Precondiciones | ----- |
| Postcondiciones | ----- |

Contrato 2.9: *acceptarSolicitudDeAlquiler* (Francisco)

| | |
|------------------------|---|
| Nombre | <i>acceptarSolicitudDeAlquiler(idUsuario,idPlanAlquiler*)</i> |
| Responsabilidad | Proceder a aceptar la solicitud de alquiler por parte del usuario (arrendador), para lo que se proporciona el idUsuario (arrendatario) y el idPlanAlquiler |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si fechaFin del anterior Alquiler es superior a la fechaInicio del Alquiler que se pretende aceptar su solicitud |
| Salida | ----- |
| Precondiciones | -El idUsuario existe y es válido -El idPlanAlquiler existe y es válido |
| Postcondiciones | <ul style="list-style-type: none">Se modificó el estado del objeto de la clase Alquiler, identificado por el idUsuario y el idPlanAlquiler, pasando su atributo estado a tener el valor de ACEPTADO |

*** idPlanAlquiler debería de ser idAlquiler**

Contrato 2.10: *rechazarSolicitudAlquiler* (Francisco)

| | |
|------------------------|--|
| Nombre | <i>rechazarSolicitudAlquiler(idUsuario,idPlanAlquiler*)</i> |
| Responsabilidad | Proceder a rechazar la solicitud de alquiler realizada por el usuario (arrendador), para lo que se proporciona el idUsuario (arrendatario) y el idPlanAlquiler |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si fechaFin del anterior Alquiler es superior a la fechaInicio del Alquiler que se pretende aceptar su solicitud |
| Salida | ----- |
| Precondiciones | -El idUsuario existe y es válido -El idPlanAlquiler existe y es válido |
| Postcondiciones | <ul style="list-style-type: none">Se modificó el estado del objeto de la clase Alquiler, identificado por el idUsuario y el idPlanAlquiler, pasando su atributo estado a tener el valor de RECHAZADO |

*** idPlanAlquiler debería de ser idAlquiler**

Contrato 2.11: *realizarPagoAlquiler* (Francisco)

| | |
|------------------------|---|
| Nombre | <i>realizarPagoAlquiler(idUsuario,idAlquiler,detallesPago)</i> |
| Responsabilidad | Realizar el pago del alquiler que ha solicitado por el usuario (arrendatario), siendo identificado por su idUsuario , idAlquiler y detallesPago |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si idAlquiler e idUsuario no concuerdan con el objeto Alquiler a pagar |
| Salida | ----- |
| Precondiciones | -El idUsuario existe y es válido -El idAlquiler existe y es válido |
| Postcondiciones | <ul style="list-style-type: none">• Se creó un nuevo objeto, pay, de la clase Pago con inicializándolo {fecha:FechaActual, importe:importe, detalles:detallesPago}• Se creó el enlace entre el objeto pay de la clase Pago y el objeto de la clase Alquiler, identificado por idAlquiler |

Contrato 2.12: *valorarAlquiler* (Francisco)

| | |
|------------------------|---|
| Nombre | <i>valorarAlquiler(idUsuario,idAlquiler,puntuación,comentario)</i> |
| Responsabilidad | Se procede a la valoración de un alquiler realizado por el usuario que es identificado por el idUsuario e idAlquiler, aportando una puntuación y/o comentario |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si idAlquiler e idUsuario no concuerdan con el objeto Alquiler a valorar |
| Salida | ----- |
| Precondiciones | -El idUsuario existe y es válido -El idAlquiler existe y es válido |
| Postcondiciones | <ul style="list-style-type: none">• Se creó un nuevo objeto, val, de la clase ValoraciónAlquiler inicializándolo {fecha:FechaActual, puntuacion:puntuacion, comentario:comentario}• Se creó el enlace entre el objeto val de la clase valoraciónAlquiler y el objeto de la clase Alquiler, identificado por idAlquiler |

Contrato 2.13 *valorarUsoAlquiler* (Francisco)

| | |
|------------------------|--|
| Nombre | <i>valorarUsoAlquiler(idUsuario,idPropietario,puntuación,comentario)</i> |
| Responsabilidad | Se procede a la valoración del uso de un alquiler realizado por el usuario arrendador que es identificado por el idPropietario, que valorará el uso hecho por parte del usuario arrendatario identificado mediante su idUsuario, aportando una puntuación y un comentario |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | ----- |
| Salida | ----- |
| Precondiciones | -El idUsuario existe y es válido -El idPropietario existe y es válido |
| Postcondiciones | <ul style="list-style-type: none"> • Se creó un nuevo objeto, valuso, de la clase valoraciónUsoAlquiler inicializándolo {fecha:FechaActual, puntuacion:puntuacion, comentario:comentario} • Se creó el enlace entre el objeto valuso de la clase valoraciónUsoAlquiler y el objeto de la clase Usuario, identificado por idUsuario |

Contrato 2.14: *cancelarSolicitudAlquiler* (Francisco)

| | |
|------------------------|---|
| Nombre | <i>cancelarSolicitudAlquiler(idUsuario,idAlquiler)</i> |
| Responsabilidad | Se procede a la cancelación de una solicitud por parte del usuario (arrendatario) identificado por su idUsuario, de un alquiler identificado por su idAlquiler, quedando éste eliminado |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | ----- |
| Salida | ----- |
| Precondiciones | -El idAlquiler existe y es válido -El idUsuario existe y es válido |
| Postcondiciones | <ul style="list-style-type: none"> • Se eliminó el enlace entre el objeto de la clase Alquiler y el objeto de la clase PlanAlquiler relacionado con esa solicitud de alquiler. • Se eliminó el enlace entre el objeto de la clase Alquiler y el objeto de la clase ValoraciónAlquiler relacionado de ese alquiler. • Se eliminó el enlace entre el objeto de la clase Alquiler y el objeto de la clase Pago relacionado de ese alquiler. • Se eliminó el enlace entre el objeto de la clase Alquiler y el objeto de la clase Usuario relacionado de ese alquiler. • Se eliminó el objeto de la clase Alquiler. |

Contrato 2.15: cancelarPlanAlquiler (Francisco)

| | |
|------------------------|---|
| Nombre | <i>cancelarPlanAlquiler(idPlanAlquiler)</i> |
| Responsabilidad | Procede a cancelar un plan de alquiler previamente creado y que se identifica con su idPlanAlquiler |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Comprobar que no existen objetos de la clase Alquiler relacionados con dicho plan de alquiler, cuyo estado sea ACEPTADO |
| Salida | ----- |
| Precondiciones | -El idPlanAlquiler existe y es válido |
| Postcondiciones | <ul style="list-style-type: none"> • Se modificó el objeto, plan, de la clase PlanAlquiler identificado por idPlanAlquiler, cambiando el atributo visible a false • (Excepción) Se modificó el estado de todas las solicitudes de la clase Alquiler, pasando de ACEPTADO a RECHAZADO, notificando a su vez a los correspondientes usuarios del cambio de estado en sus solicitudes. |

Contrato 2.16: EliminarSolicitudesAlquilerRechazadasCaducadas (Francisco)

| | |
|------------------------|--|
| Nombre | <i>EliminarSolicitudesAlquilerRechazadasCaducadas()</i> |
| Responsabilidad | Se procede a la eliminación por parte del sistema de las solicitudes que han sido rechazadas y han caducado |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | ----- |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <p>Para cada objeto de la clase Alquiler cuyo estado:RECHAZADO y cuya fechaFin es posterior a la fecha actual {</p> <ul style="list-style-type: none"> • Se eliminó el enlace entre el objeto de la clase Alquiler y el objeto de la clase PlanAlquiler relacionado con esa solicitud de alquiler. • Se eliminó el enlace entre el objeto de la clase Alquiler y el objeto de la clase ValoraciónAlquiler relacionado de ese alquiler. • Se eliminó el enlace entre el objeto de la clase Alquiler y el objeto de la clase Pago relacionado de ese alquiler. • Se eliminó el objeto de la clase Alquiler. <p>}</p> |

Contrato 2.17: *EliminarPlanesAlquilerNoUsadosCaducados* (Francisco)

| | |
|------------------------|--|
| Nombre | <i>EliminarPlanesAlquilerNoUsadosCaducados()</i> |
| Responsabilidad | Se procede a la eliminación por parte del sistema de los planes de alquiler que no han sido usados y han caducado |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | ----- |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | Para cada objeto de la clase PlanAlquiler que no se haya usado y cuya fecha del ultimoDiaAlquiler es posterior a la fecha actual { <ul style="list-style-type: none">• Se eliminó el enlace entre el objeto de la clase PlanAlquiler y el objeto de la clase Alquiler relacionado con ese plan de alquiler.• Se eliminó el enlace entre el objeto de la clase PlanAlquiler y el objeto de la clase Vehículo asignado a dicho plan de alquiler.• Se eliminó el enlace entre el objeto de la clase PlanAlquiler y el objeto de la clase UsuarioPropietario relacionado con ese plan de alquiler• Se eliminó el objeto de la clase PlanAlquiler. } |

Contrato 3.1: *introducirPerfil* (Javier)

| | |
|------------------------|---|
| Nombre | <i>introducirPerfil</i> (<i>nombre,telefono,añoNacimiento,breveDescPersonal,foto,direccionPostal</i>) |
| Responsabilidad | Introducir los datos del perfil al registrarse en el sistema |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | - Uno o más datos obligatorios no han sido introducidos - El añoNacimiento está en un rango no existente |
| Salida | idUsuario={nombredeUsuario} |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">• Se creó un objeto de la clase Restaurante con nombre, teléfono, añoNacimiento, breveDescPersonal, foro y direccionPostal. |

Contrato 3.2: modificarPerfil (Javier)

| | |
|------------------------|--|
| Nombre | <i>modificarPerfil (idUserio, nombre,teléfono,añoNacimiento,breveDescPersonal,foto,direccionPostal)</i> |
| Responsabilidad | Permite al usuario identificado como idUsuario modificar los datos del perfil |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | - El añoNacimiento está en un rango no existente |
| Salida | ----- |
| Precondiciones | -El usuario identificado por idUsuario se ha conectado en el sistema correctamente |
| Postcondiciones | <ul style="list-style-type: none">• Se modificaron los atributos del objeto de la clase Usuario identificado por idUsuario |

Contrato 3.3: consultarPerfil (Javier)

| | |
|------------------------|--|
| Nombre | <i>consultarPerfil (idUserio)</i> |
| Responsabilidad | Permite consultar los datos del perfil del usuario identificado como idUsuario |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | ----- |
| Salida | infoPerfil={ nombre,teléfono,añoNacimiento,breveDescPersonal,foto,direccion Postal } |
| Precondiciones | -El usuario identificado por idUsuario se ha conectado en el sistema correctamente |
| Postcondiciones | ----- |

Contrato 3.4: modificarVisibilidadPerfil (Javier)

| | |
|------------------------|---|
| Nombre | <i>modificarVisibilidadPerfil (idUserario, nuevaVisibilidad)</i> |
| Responsabilidad | Permite al usuario identificado como idUsuario cambiar la visibilidad de su perfil |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | ----- |
| Salida | ----- |
| Precondiciones | -El usuario identificado por idUsuario se ha conectado en el sistema correctamente |
| Postcondiciones | <ul style="list-style-type: none">• -Se modificó el atributo nuevaVisibilidad del objeto de la clase Usuario identificado por idUsuario |

Contrato 3.5: establecerFormaPago (Javier)

| | |
|------------------------|--|
| Nombre | <i>establecerFormaPago(idUsuario,listaTipoTransaccion)</i> |
| Responsabilidad | Permite al usuario identificado por idUsuario establecer una forma de pago para pagar un servicio, eligiendo dicha forma de pago en una lista identificada como listaTipoTransaccion |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | - Si algunas de las formas de pago identificadas por alguno de los elementos de listaTipoTransaccion no exista. |
| Salida | ----- |
| Precondiciones | El usuario identificado por idUsuario se ha conectado en el sistema correctamente |
| Postcondiciones | <ul style="list-style-type: none">• Se modificó el atributo preferenciaPago de la clase Usuario |

Contrato 3.6: establecerFormaCobro (Javier)

| | |
|------------------------|--|
| Nombre | <i>establecerFormaCobro(idPropietario,listaTipoTransaccion)</i> |
| Responsabilidad | Permite al propietario identificado por idPropietario establecer una forma de cobro para recibir pagos por un servicio, eligiendo dicha forma de cobro en una lista identificada como listaTipoTransaccion |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | - Si algunas de las formas de cobro identificadas por alguno de los elementos de listaTipoTransaccion no exista. |
| Salida | ----- |
| Precondiciones | -El propietario identificado por idPropietario se ha conectado en el sistema correctamente |
| Postcondiciones | <ul style="list-style-type: none">• Se modificó el atributo prefenciaCobro de la clase UsuarioPropietario• Se creó un enlace entre UsuarioPropietario y CuentaBancaria |

Contrato 3.7: añadirVehículo (Javier)

| | |
|------------------------|---|
| Nombre | <i>añadirVehículo(idPropietario,matricula,marca,modelo,color,numeroPlazas,categoria,confor)</i> |
| Responsabilidad | El propietario identificado como idPropietario pueda añadir un vehículo al sistema que estará identificado con la matrícula |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | - El vehículo identificado por la matrícula no existe |
| Salida | ----- |
| Precondiciones | -El propietario identificado por idPropietario se ha conectado en el sistema correctamente |
| Postcondiciones | <ul style="list-style-type: none">• Se creó un objeto de la clase Vehículo identificado por matrícula• Se creó un enlace entre UsuarioPropietario y Vehículo |

Contrato 3.8: eliminarVehículo (Javier)

| | |
|------------------------|---|
| Nombre | <i>eliminarVehículo(idPropietario,matrícula)</i> |
| Responsabilidad | El propietario identificado como idPropietario pueda eliminar un vehículo del sistema que está identificado por la matrícula |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | - El vehículo identificado por la matrícula no existe |
| Salida | ----- |
| Precondiciones | -El propietario identificado por idPropietario se ha conectado en el sistema correctamente |
| Postcondiciones | <ul style="list-style-type: none">• Se eliminó un enlace entre UsuarioPropietario y Vehículo• Se eliminó un objeto de la clase Vehículo identificado por matrícula |

Contrato 4.1: buscarPlanesViaje (Jorge)

| | |
|------------------------|---|
| Nombre | <i>buscarPlanesViaje(origen, destino, fecha)</i> |
| Responsabilidad | El sistema consulta los planes de viaje y devuelve los que coinciden con el origen, destino y fecha proporcionados. |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si el origen no existe. -Si el destino no existe. -Si la fecha no es correcta. |
| Salida | Colección de los planes de viaje que coinciden con la solicitud identificados por idPlanViaje. |
| Precondiciones | ----- |
| Postcondiciones | ----- |

Contrato 4.6: *nuevoPlanViaje* (Jorge)

| | |
|------------------------|--|
| Nombre | <i>nuevoPlanViaje(idUsuario,idVehiculo,plazas,dia,origen,destino,comentario)</i> |
| Responsabilidad | Crear un nuevo plan de viaje identificado por el idPlanViaje, creado por el usuario idUsuario con el vehiculo idVehiculo, con la información correspondiente. |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si el vehículo no está disponible. -Si la información no es correcta. |
| Salida | IdPlanViaje. |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">• Se creó un nuevo objeto Plan de viaje , PlanViaje inicializado con la información pasada por parámetros.• Se creó un enlace entre el Plan de viaje y el usuario identificado por idUsuario. |

Contrato 4.9: *modificarInfoParada* (Jorge)

| | |
|------------------------|--|
| Nombre | <i>modificarInfoParada(idParada,hora,coste)</i> |
| Responsabilidad | Modificar la información de la parada identificada por idParada con la hora y coste proporcionados. |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si la parada no existe. -Si la información no es correcta. |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">• Se modifica la hora y coste del objeto parada identificada por idParada. |

Contrato 4.11: ofertarPlanViaje (Jorge)

| | |
|------------------------|--|
| Nombre | <i>ofertarPlanViaje(idPlanViaje)</i> |
| Responsabilidad | Modificar el objeto plan de viaje identificado por idPlanViaje para que sea visible. |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | Si el idPlanViaje no existe |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">• Se modificó el objeto PlanViaje , identificado por idPlanViaje para que sea visible.• Se creó un enlace entre el Plan de viaje y el usuario identificado por idUsuario. |

Contrato 4.13: solicitarViaje (Jorge)

| | |
|------------------------|---|
| Nombre | <i>solicitarViaje(idPlanViaje, idUsuario, origen Viajero, destino Viajero, Pago)</i> |
| Responsabilidad | Crear un nuevo Viaje. |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si el origen no existe. -Si el destino no existe. -Si el pago no es correcto. -Si el idPlanViaje no existe. |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">• Se creó un nuevo objeto Viaje con estado pendiente, asociado con plan de viaje por idPlanViaje, asociado a un usuario por idUsuario, con la parada de origen identificada por origenViajero y la parada de destino por destinoViajero, y el pago asociado por Pago. |

Contrato 4.18: *acceptarAcompañante (Jorge)*

| | |
|------------------------|--|
| Nombre | <i>acceptarAcompañante(idViaje,idUsuario).</i> |
| Responsabilidad | Aceptar la solicitud de viaje |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si el idViaje no existe -Si el usuario no existe. |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">Se cambia el estado del viaje identificado por idViaje a aceptado. |

Contrato 4.19: *rechazarAcompañante (Jorge)*

| | |
|------------------------|---|
| Nombre | <i>rechazarAcompañante(idViaje,idUsuario).</i> |
| Responsabilidad | Rechazar la solicitud de viaje. |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -Si el idViaje no existe -Si el usuario no existe. |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">Se cambia el estado del viaje identificado por idViaje a rechazado. |

Contrato 4.20: *cancelarSolicitudPlan Viaje(Jonathan)*

| | |
|------------------------|--|
| Nombre | <i>cancelarSolicitudPlan Viaje(idPlan Viaje,idUsuario)</i> |
| Responsabilidad | Se procede a la cancelación de una solicitud por parte del usuario (arrendatario) identificado por su idUsuario, de un plan de viaje identificado por su idPlanViaje, quedando éste eliminado |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | - El idPlanViaje existe y es válido - El idUsuario existe y es válido - Comprobar que no existen objetos de la clase Alquiler relacionados con dicho plan de alquiler, cuyo estado sea ACEPTADO |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">Se modificó el objeto de la clase Plan Viaje identificado poridPlanViaje, cambiando el atributo visible a false(Excepción) Se modificó el estado de todas las solicitudes de la clase Alquiler, pasando de ACEPTADO a RECHAZADO, notificando a su vez a los correspondientes usuarios del cambio de estado en sus solicitudes. |

Contrato 4.24: *valorarViaje* (Francisco)

| | |
|------------------------|---|
| Nombre | <i>valorarViaje(idPlanViaje*,idUsuario,puntuación,comentario)</i> |
| Responsabilidad | Se procede a la valoración de un viaje realizado por el usuario que es identificado por el idPlanViaje e idUsuario, aportando una puntuación y un comentario |
| Tipo | Tapuntas |
| Notas | |
| Excepciones | -Si idPlanViaje e idUsuario no concuerdan con el objeto Viaje a valorar |
| Salida | |
| Precondiciones | |
| Postcondiciones | <ul style="list-style-type: none">• Se creó un nuevo objeto, val, de la clase ValoraciónViaje inicializándolo {fecha:FechaActual, puntuacion:puntuacion, comentario:comentario}• Se creó el enlace entre el objeto val de la clase ValoraciónViaje y el objeto de la clase Viaje, identificado por idPlanViaje |

* idPlanViaje debería de ser idViaje

Contrato 4.25: *valorarAcompañante* (Jonathan)

| | |
|------------------------|--|
| Nombre | <i>valorarAcompañante(idUsuarioValora,idUsuarioValorado,puntuación,comentario)</i> |
| Responsabilidad | Valorar al acompañante de viaje |
| Tipo | Tapuntas |
| Notas | ----- |
| Excepciones | -idUsuarioValora debe de ser válido -idUsuarioValorado debe de ser válido -Puntuación debe estar comprendida entre 1 y 10 |
| Salida | ----- |
| Precondiciones | ----- |
| Postcondiciones | <ul style="list-style-type: none">• Se ha creado un objeto de la clase valoracionAcompañante inicializado con valores:• { fecha=dateActual,comentario=comentario,puntuación=puntuación }• Se ha creado un enlace entre la clase valoraciónUsuario y la clase Usuario |