
Práctica 3 - Parte 2 FIS

Ingeniería de requisitos

Contratos



**UNIVERSIDAD
DE GRANADA**

Departamento de
Lenguajes y Sistemas
Informáticos



Raúl Del Pozo Moreno
Pablo Lombardero Ros
Nikita Stetskiy
Miguel Ángel Campos Cubillas

Índice

1. Pablo Lombardero Ros
 - 1.1. Control Agentes
 - 1.1.1. crearAgente
 - 1.1.2. modificarAgente
 - 1.1.3. eliminarAgente
 - 1.1.4. consultarAgente
 - 1.1.5. reflejarObservacion
 - 1.1.6. infoProduccionAgentes
 - 1.2. Clientes
 - 1.2.1. nuevoCliente
 - 1.2.2. modificarCliente
 - 1.2.3. eliminarCliente
 - 1.2.4. consultarCliente
 - 1.2.5. recibirPublicidad
 - 1.2.6. nuevoGrupo
 - 1.2.7. eliminarGrupo
2. Nikita Stetskiy
 - 2.1. Clientes
 - 2.1.1. modificarGrupo
 - 2.1.2. añadirClienteGrupo
 - 2.1.3. eliminarClienteGrupo
 - 2.1.4. consultarGrupo
 - 2.2. Pagos
 - 2.2.1. pagoPresencial
 - 2.2.2. pagoRemoto
 - 2.2.3. abono
 - 2.2.4. añadirOfertasEnvio
 - 2.2.5. ordenarEnvio
 - 2.3. Actividad Turística
 - 2.3.1. nuevaActividadTuristica
 - 2.3.2. modificarActividadTuristica
 - 2.3.3. añadirServicioActiviudadTuristica
 - 2.3.4. eliminarServicioActiviudadTuristica

3. Miguel Angel Campos Cubillas

3.1. Reservas

- 3.1.1. iniciarPaqueteTuristico(cliente)
- 3.1.2. iniciarPaqueteTuristico(grupo)
- 3.1.3. peticionActividadTuristica
- 3.1.4. peticionCircuitoTuristico
- 3.1.5. peticionServicioVuelo
- 3.1.6. añadirBilleteAvionActividadTuristica
- 3.1.7. añadirBilleteAvionCircuitoTuristico
- 3.1.8. añadirBilleteAvion
- 3.1.9. eliminarPeticionActividadTuristica
- 3.1.10. eliminarPeticionCircuitoTuristico
- 3.1.11. eliminarPeticionServicioVuelo
- 3.1.12. iniciarReserva
- 3.1.13. confirmarReserva
- 3.1.14. consultarPeticionBusqueda

4. Raúl Del Pozo Moreno

4.1. Actividad Turística

- 4.1.1. eliminarActividadTuristica
- 4.1.2. consultarActividadTuristica
- 4.1.3. nuevoHotel
- 4.1.4. nuevoVuelo
- 4.1.5. nuevoEspectaculo
- 4.1.6. eliminarServicioTuristico
- 4.1.7. modificarVuelo
- 4.1.8. modificarHotel
- 4.1.9. modificarEspectaculo
- 4.1.10. consultarServicioTuristico

4.2. Reserva

- 4.2.1. consultarReserva
- 4.2.2. cancelarPaqueteTuristico
- 4.2.3. cancelarReserva

1. Pablo Lombardero Ros

a. Control Agentes

i. crearAgente

Nombre	idAgente = crearAgente(nombre)
Responsabilidad	Crea un nuevo objeto de la clase Agente que se identificará por idAgente
Tipo	AgenciaViajes
Notas	
Excepciones	
Salida	idAgente = identificador del agente creado
Precondiciones	
Postcondiciones	Fue creado un objeto, agen, de la clase Agente debidamente inicializado

ii. modificarAgente

Nombre	modificarAgente(idAgente, nombre)
Responsabilidad	Modifica el objeto Agente identificado por idAgente
Tipo	AgenciaViajes
Notas	
Excepciones	Si el agente identificado por idAgente no existe
Salida	
Precondiciones	
Postcondiciones	Se modificó el nombre del objeto identificado por idAgente al valor contenido en el parámetro nombre

iii. eliminarAgente

Nombre	eliminarAgente(idAgente)
Responsabilidad	Elimina el objeto Agente identificado por idAgente
Tipo	AgenciaViajes
Notas	
Excepciones	Si el agente identificado por idAgente no existe
Salida	
Precondiciones	
Postcondiciones	Fue destruido el objeto de la clase Agente identificado por idAgente

iv. consultarAgente

Nombre	infoAgente = consultarAgente(idAgente)
Responsabilidad	Muestra la información sobre el agente identificado por idAgente
Tipo	AgenciaViajes
Notas	
Excepciones	Si el agente identificado por idAgente no existe
Salida	infoAgente = nombre para el objeto de la clase Agente que se identifique por idAgente
Precondiciones	
Postcondiciones	

v. reflejarObservacion

Nombre	reflejarObservacion(idDirectivo, idAgente, descripcion, fecha)
Responsabilidad	Crea un nuevo objeto de la clase Observación inicializando su estado y asignándole el directivo que la realiza, así como el agente a la que va dirigida dicha observación, la observación en sí y la fecha en que ocurre
Tipo	AgenciaViajes
Notas	
Excepciones	- Si no existe el directivo identificado por idDirectivo - Si no existe el agente identificado por idAgente
Salida	
Precondiciones	
Postcondiciones	- Fue creado un objeto, obser, de la clase Observación debidamente inicializado - Fue creado un enlace entre obser y el objeto Directivo, identificado por idDirectivo - Fue creado un enlace entre obser y el objeto Agente, identificado por idAgente

vi. infoProduccionAgentes

Nombre	infoProduccionAgentes=produccionAgentes()
Responsabilidad	Muestra información sobre la producción de los agentes
Tipo	AgenciaViajes
Notas	
Excepciones	
Salida	infoProduccionAgentes = lista de {idAgente, produccion} para todos los objetos de la clase Agente
Precondiciones	
Postcondiciones	

b. Clientes

i. nuevoCliente

Nombre	idCliente=nuevoCliente(dni, nombre, dirección, teléfono, publicidad)
Responsabilidad	Crea un nuevo cliente inicializando su estado y asignándole un dni, un nombre, una dirección, un teléfono y una publicidad
Tipo	AgenciaViajes
Notas	
Excepciones	
Salida	idCliente = identificador del cliente creado
Precondiciones	
Postcondiciones	Fue creado un objeto, cliente, de la clase Cliente debidamente inicializado

ii. modificarCliente

Nombre	modificarCliente(idCliente, dni, nombre, dirección, teléfono, publicidad)
Responsabilidad	Modifica el objeto Cliente identificado por idCliente
Tipo	AgenciaViajes
Notas	
Excepciones	Si no existe el cliente identificado por idCliente
Salida	
Precondiciones	
Postcondiciones	<ul style="list-style-type: none">- Se modificó el dni del objeto identificado por idCliente al valor contenido en la variable dni- Se modificó el nombre del objeto identificado por idCliente al valor contenido en la variable nombre- Se modificó la dirección del objeto identificado por idCliente al valor contenido en la variable dirección- Se modificó el teléfono del objeto identificado por idCliente al valor contenido en la variable teléfono- Se modificó la publicidad del objeto identificado por idCliente al valor contenido en la variable publicidad

iii. eliminarCliente

Nombre	eliminarCliente(idCliente)
Responsabilidad	Elimina el objeto Cliente identificado por idCliente
Tipo	AgenciaViajes
Notas	
Excepciones	Si no existe el cliente identificado por idCliente
Salida	
Precondiciones	
Postcondiciones	Fue destruido el objeto de la clase Cliente identificado por idCliente

iv. consultarCliente

Nombre	infoCliente = consultarCliente(idCliente)
Responsabilidad	Muestra la información sobre el objeto Cliente identificado por idCliente
Tipo	AgenciaViajes
Notas	
Excepciones	Si el cliente identificado por idCliente no existe
Salida	infoCliente={dni, nombre, dirección, teléfono, publicidad} Para todos los objetos de la clase Cliente que cumplan con los criterios de consulta
Precondiciones	
Postcondiciones	

v. recibirPublicidad

Nombre	recibirPublicidad(idCliente, publicidad)
Responsabilidad	Cambiar la preferencia de publicidad del cliente
Tipo	AgenciaViajes
Notas	
Excepciones	Si no existe el cliente identificado por idCliente
Salida	
Precondiciones	
Postcondiciones	Se modificó la publicidad del objeto Cliente identificado por idCliente al valor contenido en la variable publicidad

vi. nuevoGrupo

Nombre	idGrupo=nuevoGrupo(descripción, listaidCliente)
Responsabilidad	Crea un nuevo objeto de la clase Grupo que consigne los clientes que lo forman y una descripción de sí mismo
Tipo	AgenciaViajes
Notas	
Excepciones	Si no existe alguno o varios de los clientes identificados por los idCliente de la lista listaidCliente
Salida	idGrupo = identificador del grupo creado
Precondiciones	
Postcondiciones	Fue creado un objeto, grup, de la clase Grupo, debidamente inicializado

vii. eliminarGrupo

Nombre	eliminarGrupo(idGrupo)
Responsabilidad	Elimina el objeto Grupo identificado por idGrupo
Tipo	AgenciaViajes
Notas	
Excepciones	Si no existe el grupo designado por idGrupo
Salida	
Precondiciones	
Postcondiciones	Fue destruido el objeto de la clase Grupo identificado por idGrupo

2. Nikita Stetskiy

a. Clientes

i. modificarGrupo

Nombre	modificarGrupo (idGrupo, descripción)
Responsabilidad	Modifica un grupo en base a su id del grupo y la descripción del dicho grupo
Tipo	AgenciaViajes
Notas	
Excepciones	Si el Grupo identificado por idGrupo no existe
Salida	
Precondiciones	
Postcondiciones	El Grupo en cuestión pasará a otro estado modificado

ii. añadirClienteGrupo

Nombre	añadirClienteGrupo (idGrupo, idCliente)
Responsabilidad	Añade un nuevo cliente inicializando su estado y asignándole un Grupo basándose en su id
Tipo	AgenciaViajes
Notas	
Excepciones	Si el Cliente identificado por idCliente no existe Si el Grupo identificado por idGrupo no existe
Salida	
Precondiciones	
Postcondiciones	Fue creado un enlace entre el el objeto de la clase Grupo identificado por los correspondientes idGrupo a través de idCliente

iii. eliminarClienteGrupo

Nombre	eliminarClienteGrupo (idGrupo, idCliente)
Responsabilidad	Elimina un cliente basándose en su estado definido a través de su idCliente y idGrupo
Tipo	AgenciaViajes
Notas	
Excepciones	Si el Cliente identificado por idCliente no existe Si el Grupo identificado por idGrupo no existe
Salida	
Precondiciones	
Postcondiciones	Se elimina el enlace entre el objeto de la clase Grupo identificado por los correspondientes idGrupo a través de idCliente

iv. consultarGrupo

Nombre	consultarGrupo (idGrupo)
Responsabilidad	Consulta los datos de un grupo basándose en su estado definido a través de su idGrupo
Tipo	AgenciaViajes
Notas	
Excepciones	Si el Grupo identificado por idGrupo no existe
Salida	infoGrupo = la consulta de datos trasladada a información
Precondiciones	
Postcondiciones	Se deja intactos los datos del objeto de la clase Grupo identificado por el correspondiente idGrupo

b. Pagos

i. pagoPresencial

Nombre	pagoPresencial (idCliente, idPaqueteTuristico, cantidad, fechaPago, TipoPago)
Responsabilidad	Realiza un pago presencial mediante el estado del id Cliente y asignándole el id del paquete turístico, la cantidad, la fecha del pago y su tipo
Tipo	AgenciaViajes
Notas	
Excepciones	Si el Cliente identificado por el idCliente no existe Si el Paquete Turistico identificado por el idPaqueteTuristico no existe Si el Tipo de Pago introducido es erroneo
Salida	
Precondiciones	
Postcondiciones	Se relaciona el pago Presencial mediante el id del cliente y el del paquete turístico. Los elementos de cantidad, la fecha del pago y su tipo.

ii. pagoRemoto

Nombre	pagoRemoto (idCliente, idPaqueteTuristico, cantidad, fechaPago, "TARJETA")
Responsabilidad	Realiza un pago remoto mediante el estado del id Cliente y asignándole el id del paquete turístico, la cantidad, la fecha del pago y la tarjeta del cliente
Tipo	AgenciaViajes
Notas	
Excepciones	Si el Cliente identificado por el idCliente no existe Si el Paquete Turistico identificado por el idPaqueteTuristico no existe Si la Tarjeta introducida es errónea
Salida	
Precondiciones	
Postcondiciones	Se relaciona el pago Remoto mediante el id del cliente y el del paquete turístico. Los elementos de cantidad, la fecha del pago y su tarjeta.

iii. abono

Nombre	abono (idCliente, idPaqueteTuristico, tipoAbono)
Responsabilidad	Realiza un abono, que puede ser metálico o electrónico, mediante el estado del id Cliente y asignándole el id del paquete turístico
Tipo	AgenciaViajes
Notas	
Excepciones	Si el Cliente identificado por el idCliente no existe Si el Paquete Turistico identificado por el idPaqueteTuristico no existe Si el tipo Abono introducido no existe
Salida	cantidad = la cantidad abonada
Precondiciones	
Postcondiciones	Se relaciona el abono mediante el id del cliente y el del paquete turístico. Los elementos del tipo del abono que pueden llegar a ser en metálico o electrónico

iv. añadirOfertasEnvio

Nombre	añadirOfertasEnvio (idListaDistribucion, idOferta)
Responsabilidad	Se añade una nueva oferta de envío inicializando su estado en base a la lista de distribución y la oferta correspondida.
Tipo	AgenciaViajes
Notas	
Excepciones	Si la lista de distribución identificada por el idListaDistribucion no existe
Salida	
Precondiciones	
Postcondiciones	Se relaciona la oferta creada con la clase de distribución, creando un enlace con el id de la lista y oferta

v. ordenarEnvio

Nombre	ordenarEnvio (idListaDistribucion)
Responsabilidad	Se ordena una oferta de envío inicializando en base a la lista de distribución correspondida.
Tipo	AgenciaViajes
Notas	
Excepciones	Si la lista de distribución identificada por el idListaDistribucion no existe
Salida	informacionEnvio = la información dada a la ordenación de la lista de distribución
Precondiciones	
Postcondiciones	Se relaciona la oferta ordenada con la clase de distribución, creando un enlace con el id de la lista y oferta

c. Actividad Turística

i. nuevaActividadTuristica

Nombre	nuevaActividadTuristica (descripcion, fechaAlta, fechaFinalizacion, listaIdServicioTuristico)
Responsabilidad	Se crea una nueva actividad turística basándose en la lista de Servicio Turístico, gracias a la descripción y las fechas de alta y finalización.
Tipo	AgenciaViajes
Notas	
Excepciones	Si la lista de Servicio Turistico identificada por el listaIdServicioTuristico no existe
Salida	idActividadTuristica = la actividad turística creada
Precondiciones	
Postcondiciones	Se creará el objeto basado en la descripcion, fecha alta y fecha finalización utilizada. Se creará una enlace con la lista id de Servicio Turistico

ii. modificarActividadTuristica

Nombre	modificarActividadTuristica (idActividadTuristica, descripcion, fechaAlta, fechaFinalizacion)
Responsabilidad	Se modifica una actividad turística basándose en la Actividad Turística, se podrá editar a la descripción, las fechas de alta y finalización.
Tipo	AgenciaViajes
Notas	
Excepciones	Si Actividad Turística identificada por el idActividadTuristica no existe
Salida	
Precondiciones	
Postcondiciones	Se modifica el estado de la Actividad Turística, utilizando en el enlace en el id de esta. Así se podrá editar su descripción, fechaAlta y fechaFinalizacion

iii. añadirServicioActividadTursitica

Nombre	añadirServicioActividadTursitica (idActividadTuristica, idServicioTuristico)
Responsabilidad	Se añadirá un servicio basándose en la actividad turística determinada
Tipo	AgenciaViajes
Notas	
Excepciones	Si Actividad Turística identificada por el idActividadTuristica no existe Si Servicio Turistico identificada por el idServicioTuristico no existe
Salida	
Precondiciones	
Postcondiciones	Se añadirá un nuevo servicio, creando así un objeto que enlaza el id de la Actividad Turística y el id del Servicio Turistico

iv. eliminarServicioActividadTuristica

Nombre	eliminarServicioActividadTuristica (idActividadTuristica, idServicioTuristico)
Responsabilidad	Se eliminara el Servicio Turistico gracias a la actividad relacionada
Tipo	AgenciaViajes
Notas	
Excepciones	Si Actividad Turística identificada por el idActividadTuristica no existe Si Servicio Turistico identificada por el idServicioTuristico no existe
Salida	
Precondiciones	
Postcondiciones	Se eliminará un servicio, destruyendo así un objeto que enlazaba el id de la Actividad Turística y el id del Servicio Turistico

3. Miguel Angel Campos Cubillas

a. Actividad Turística

i. iniciarPaqueteTurístico(cliente)

Nombre	idPaquete = iniciarPaqueteTurístico (idCliente)
Responsabilidad	Crear un nuevo paquete turístico identificado por idCliente
Tipo	AgenciaViajes
Notas	
Excepciones	· Si el cliente identificado por idCliente no existe
Salida	idPaquete = identificador del paquete creado
Precondiciones	
Postcondiciones	· Fue creado un objeto de la clase PaqueteTuristico debidamente inicializado · Fue creado un enlace entre cliente y paquete turístico, identificados por idCliente e idPaquete respectivamente

ii. iniciarPaqueteTurístico(grupo)

Nombre	idPaquete = iniciarPaqueteTurístico (idGrupo)
Responsabilidad	Crear un nuevo paquete turístico identificado por idGrupo
Tipo	AgenciaViajes
Notas	
Excepciones	· Si el grupo identificado por idGrupo no existe
Salida	idPaquete = identificador de paquete creado
Precondiciones	
Postcondiciones	· Fue creado un objeto de la clase paqueteTuristico debidamente inicializado · Fue creado un enlace entre grupo y paquete turístico, identificados por idGrupo e idPaquete respectivamente

iii. `peticionActividadTuristica`

Nombre	<code>peticionActividadTuristica</code> (<code>idPaquete</code> , <code>listaidActividadTuristica</code>)
Responsabilidad	Crear una nueva lista de actividades turísticas identificada por <code>listaidActividadTuristica</code>
Tipo	<code>AgenciaViajes</code>
Notas	
Excepciones	<ul style="list-style-type: none"> · Si el paquete identificado por <code>idPaquete</code> existe · Si las actividades turísticas identificadas por <code>listaidActividadTuristica</code> existen
Salida	
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Se creó una lista de objetos de <code>ActividadTuristica</code> identificados por <code>listaidActividadTuristica</code> · Se creó un enlace entre un objeto de la clase <code>PaqueteTuristico</code> identificado por <code>idPaquete</code> y cada uno de los objetos de la clase <code>ActividadTuristica</code> identificados por <code>listaidActividadTuristica</code>

iv. `peticionCircuitoTuristico`

Nombre	<code>peticionCircuitoTuristico</code> (<code>idPaquete</code> , <code>listaidCircuitoTuristico</code>)
Responsabilidad	Crear una nueva lista de circuitos turísticos identificados por <code>listaidCircuitoTuristico</code>
Tipo	<code>AgenciaViajes</code>
Notas	
Excepciones	<ul style="list-style-type: none"> · Si el paquete identificado por <code>idPaquete</code> existe · Si los circuitos turísticos identificados por <code>listaidCircuitoTuristico</code> existen
Salida	
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Se creó una lista de objetos de <code>CircuitoTuristico</code> identificados por <code>listaidCircuitoTuristico</code> · Se creó un enlace entre un objeto de la clase <code>PaqueteTuristico</code> identificado por <code>idPaquete</code> y cada uno de los objetos de la clase <code>CircuitoTuristico</code> identificados por <code>listaidActividadTuristica</code>

v. peticionServicioVuelo

Nombre	peticionServicioVuelo (idPaquete, fechaAlta, fechaFinalizacion, precioEstimado, origen, destino)
Responsabilidad	Crear una petición de vuelo identificada por idServicioTuristico
Tipo	AgenciaViajes
Notas	
Excepciones	
Salida	idServicioTurístico = identificador de servicio turístico creado
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Fue creado un objeto de la clase Servicio turístico debidamente inicializado · Fue creado un enlace entre servicioTuristico y paquete, identificado por idServicioTuristico

vi. añadirBilleteAvionActividadTuristica

Nombre	añadirBilleteAvionActividadTuristica (idPaquete, idActividadTuristica, numeroVuelo, compañía, diaVuelo, asiento, tarifa, clase, precio)
Responsabilidad	Añade un billete de avión con toda su información a una actividad turística perteneciente a un paquete
Tipo	AgenciaViajes
Notas	
Excepciones	<ul style="list-style-type: none"> · Si no existe un paquete identificado por idPaquete · Si no existe una Actividad Turística identificada por idActividadTuristica
Salida	
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Fue creado un objeto, BilleteAvion debidamente inicializado · Fue creado un enlace entre BilleteAvion y ActividadTuristica, identificado por idActividadTuristica

vii. añadirBilleteAvionCircuitoTuristico

Nombre	añadirBilleteAvionCircuitoTuristico (idPaquete, idCircuitoTuristico, numeroVuelo, compañía, diaVuelo, asiento, tarifa, clase, precio)
Responsabilidad	Añade un billete de avión con toda su información a un circuito turístico perteneciente a un paquete
Tipo	AgenciaViajes
Notas	
Excepciones	<ul style="list-style-type: none"> · Si no existe el PaqueteTuristico identificado por idPaquete · Si no existe el CircuitoTuristico identificado por idCircuitoTuristico
Salida	
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Fue creado un objeto, BilleteAvion debidamente inicializado · Fue creado un enlace entre BilleteAvion y CircuitoTuristico, identificado por idCircuitoTuristico

viii. añadirBilleteAvion

Nombre	añadirBilleteAvion (idPaquete, numeroVuelo, compañía, diaVuelo, asiento, tarifa, clase, precio)
Responsabilidad	Añade un billete de avión con toda su información a un paquete turístico
Tipo	AgenciaViajes
Notas	
Excepciones	<ul style="list-style-type: none"> · Si no existe el paquete turistico identificado por idPaquete
Salida	
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Fue creado el objeto BilleteAvion debidamente inicializado · Fue creado un enlace entre BilleteAvion y PaqueteTuristico identificado por idPaquete

ix. eliminarPeticionActividadTuristica

Nombre	eliminarPeticionActividadTuristica (idPaquete, idActividadTuristica)
Responsabilidad	Eliminar una petición de Actividad Turística
Tipo	AgenciaViajes
Notas	
Excepciones	<ul style="list-style-type: none"> · Si no existe un paquete turistico identificado por idPaquete · Si no existe una actividad turística identificada por idActividadTuristica
Salida	
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Fue eliminado el objeto de la clase ActividadTuristica identificado por idActividadTuristica · Se eliminó el enlace entre el objeto de la clase ActividadTuristica identificado por idActividadTuristica y el objeto de la clase PaqueteTuristico identificado por idPaquete

x. eliminarPeticionCircuitoTuristico

Nombre	EliminarPeticionCircuitoTuristico (idPaquete, idCircuitoTuristico)
Responsabilidad	Eliminar una petición de Circuito Turistico
Tipo	AgenciaViajes
Notas	
Excepciones	<ul style="list-style-type: none"> · Si no existe un paquete turístico identificado por idPaquete · Si no existe un circuito turístico identificado por idCircuitoTuristico
Salida	
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Fue eliminado el objeto de la clase CircuitoTuristico identificado por idCircuitoTuristico · Se eliminó el enlace entre el objeto de la clase CircuitoTuristico identificado por idCircuitoTuristico y el objeto de la clase PaqueteTuristico identificado por idPaquete

xi. eliminarPeticionServicioVuelo

Nombre	eliminarPeticionServicioVuelo (idPaquete, idServicioTuristico)
Responsabilidad	Eliminar una petición de Servicio Turistico
Tipo	AgenciaViajes
Notas	
Excepciones	<ul style="list-style-type: none"> · Si no existe un paquete turistico identificado por idPaquete · Si no existe un servicio turistico identificado por idServicioTuristico
Salida	
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Fue eliminado el objeto de la clase ServicioTuristico identificado por idCircuitoTuristico · Se eliminó el enlace entre el objeto de la clase ServicioTuristico identificado por idServicioTuristico y el objeto de la clase PaqueteTuristico identificado por idPaquete

xii. iniciarReserva

Nombre	idReserva = iniciarReserva (idCliente, idPaquete, idAgente, fecha)
Responsabilidad	Crear una nueva reserva
Tipo	AgenciaViajes
Notas	
Excepciones	<ul style="list-style-type: none"> · Si no existe un cliente identificado por idCliente · Si no existe un paquete turístico identificado por idPaquete · Si no existe un agente identificado por idAgente
Salida	idReserva = identificador de reserva creado
Precondiciones	
Postcondiciones	<ul style="list-style-type: none"> · Fue creado un objeto de la clase Reserva debidamente inicializado · Fue creado un enlace entre un objeto de la clase paquete turístico identificado por idPaquete y un objeto de la clase Reserva identificado por idReserva · Fue creado un enlace entre un objeto de la clase Cliente identificado por idCliente y un objeto de la clase Reserva identificado por idReserva · Fue creado un enlace entre un objeto de la clase Agente identificado por idAgente y un objeto de la clase Reserva identificado por idReserva

xiii. confirmarReserva

Nombre	confirmarReserva (idReserva, fechaContrato)
Responsabilidad	Confirmar una reserva
Tipo	AgenciaViajes
Notas	
Excepciones	· Si no existe una reserva identificada por idReserva
Salida	
Precondiciones	
Postcondiciones	· Se modificó el atributo de estado de la reserva identificado por idReserva a activo

xiv. consultarPeticionBusqueda

Nombre	infoPaqueteTuristico = consultarPeticionBusqueda (idPaquete)
Responsabilidad	Consultar información sobre un paquete turístico
Tipo	AgenciaViajes
Notas	
Excepciones	· Si no existe un paquete turístico identificado por idPaquete
Salida	infoPaqueteTuristico = información sobre un paquete turistico
Precondiciones	
Postcondiciones	· infoPaqueteTuristico = {codigo, fechaContrato} para el objeto de la clase PaqueteTuristico identificado por idPaquete

4. Raúl Del Pozo Moreno

a. Actividad Turística

i. eliminarActividadTuristica

Nombre	eliminarActividadTuristica(idActividadTuristica)
Responsabilidad	Eliminar el objeto ActividadTuristica identificado por idActividadTuristica
Tipo	AgenciaViajes
Notas	-
Excepciones	Si la actividad turística identificada por idActividadTuristica no existe
Salida	-
Precondiciones	-
Postcondiciones	Se destruyó el objeto de la clase ActividadTuristica identificado por idActividadTuristica

ii. consultarActividadTuristica

Nombre	infoActividadTuristica = consultarActividadTuristica(idActividadTuristica)
Responsabilidad	Obtener información de la actividad turística identificada por idActividadTuristica
Tipo	AgenciaViajes
Notas	-
Excepciones	Si la actividad turística identificada por idActividadTuristica no existe
Salida	infoActividadTuristica
Precondiciones	-
Postcondiciones	-

iii. nuevoHotel

Nombre	idHotel = nuevoHotel(descripción, fechaAlta, fechaFinalizacion, precioEstimado, nombre, categoria, direccion, telefono, zona)
Responsabilidad	Crear un nuevo objeto Hotel identificado por idHotel a partir de sus datos
Tipo	AgenciaViajes
Notas	-
Excepciones	Si todos o algunos de los atributos identificados por descripcion, fechaAlta, fechaFinalizacion, precioEstimado, nombre, categoría, dirección, teléfono o zona no existe
Salida	idHotel
Precondiciones	-
Postcondiciones	Fue creado un objeto, idHotel de la clase Hotel debidamente inicializado

iv. nuevoVuelo

Nombre	idVuelo = nuevoVuelo(descripción, fechaAlta, fechaFinalizacion, precioEstimado, origen, destino)
Responsabilidad	Crear un nuevo objeto Vuelo identificado por idVuelo a partir de sus datos
Tipo	AgenciaViajes
Notas	-
Excepciones	Si todos o algunos de los atributos identificados por descripcion, fechaAlta, fechaFinalizacion, precioEstimado, origen o destino no existe
Salida	idVuelo
Precondiciones	-
Postcondiciones	Fue creado un objeto, idVuelo de la clase Vuelo debidamente inicializado

v. nuevoEspectaculo

Nombre	idEspectaculo = nuevoEspectaculo(descripción, fechaAlta, fechaFinalizacion, precioEstimado)
Responsabilidad	Crear un nuevo objeto Espectaculo identificado por idEspectaculo a partir de sus datos
Tipo	AgenciaViajes
Notas	-
Excepciones	Si todos o algunos de los atributos identificados por descripcion, fechaAlta, fechaFinalizacion o precioEstimado no existe
Salida	idEspectaculo
Precondiciones	-
Postcondiciones	Fue creado un nuevo objeto identificado por idEspectaculo debidamente inicializado

vi. eliminarServicioTuristico

Nombre	eliminarServicioTuristico(idServicioTuristico)
Responsabilidad	Eliminar un objeto ServicioTuristico identificado por idServicioTuristico
Tipo	AgenciaViajes
Notas	-
Excepciones	Si el objeto identificado por idServicioTuristico no existe
Salida	-
Precondiciones	-
Postcondiciones	Se destruyó un objeto de la clase ServicioTuristico identificado por idServicioTuristico

vii. modificarVuelo

Nombre	modificarVuelo(idVuelo, descripción, fechaAlta, fechaFinalizacion, precioEstimado, origen, destino)
Responsabilidad	Modificar la información del objeto Vuelo identificado por idVuelo
Tipo	AgenciaViajes
Notas	-
Excepciones	Si todos o algunos de los atributos identificados por idVuelo, descripcion, fechaAlta, fechaFinalizacion, precioEstimado, origen o destino no existe
Salida	-
Precondiciones	-
Postcondiciones	Se modificó correctamente el objeto Vuelo identificado por idVuelo

viii. modificarHotel

Nombre	modificarHotel(idHotel, descripción, fechaAlta, fechaFinalizacion, precioEstimado, nombre, categoria, direccion, telefono, zona)
Responsabilidad	Modificar la información del objeto Hotel identificado por idHotel
Tipo	AgenciaViajes
Notas	-
Excepciones	Si todos o algunos de los atributos identificados por idHotel, descripcion, fechaAlta, fechaFinalizacion, precioEstimado, nombre, categoría, dirección, teléfono o zona no existe
Salida	-
Precondiciones	-
Postcondiciones	Se modificó correctamente el objeto Hotel identificado por idHotel

ix. modificarEspectaculo

Nombre	modificarEspectaculo(idEspectaculo, descripción, fechaAlta, fechaFinalizacion, precioEstimado)
Responsabilidad	Modificar la información del objeto Espectaculo identificado por idEspectaculo
Tipo	AgenciaViajes
Notas	-
Excepciones	Si todos o algunos de los atributos identificados por descripcion, fechaAlta, fechaFinalizacion, precioEstimado, nombre, categoría, dirección, teléfono o zona no existe
Salida	-
Precondiciones	-
Postcondiciones	Se modificó correctamente el objeto Espectaculo identificado por idEspectaculo

x. consultarServicioTuristico

Nombre	infoServicioTuristico = consultarServicioTuristico(idServicioTuristico)
Responsabilidad	Obtener información del objeto ServicioTuristico identificado por idServicioTuristico
Tipo	AgenciaViajes
Notas	-
Excepciones	Si el servicio turístico identificado por idServicioTuristico no existe
Salida	infoServicioTuristico
Precondiciones	-
Postcondiciones	-

b. Reserva

i. consultarReserva

Nombre	infoReserva = consultarReserva(idReserva)
Responsabilidad	Obtener información de la reserva identificada por idReserva
Tipo	AgenciaViajes
Notas	-
Excepciones	Si la reserva identificada por idReserva no existe
Salida	infoReserva
Precondiciones	-
Postcondiciones	-

ii. cancelarPaqueteTuristico

Nombre	cancelarPaqueteTuristico(idPaquete)
Responsabilidad	Eliminar un objeto PaqueteTuristico identificado por idPaquete
Tipo	AgenciaViajes
Notas	-
Excepciones	Si el paquete turístico identificado por idPaquete no existe
Salida	-
Precondiciones	-
Postcondiciones	Se destruyó el enlace entre el objeto PaqueteTuristico identificado por idPaquete y el cliente

iii. cancelarReserva

Nombre	cancelarReserva(idReserva)
Responsabilidad	Eliminar un objeto Reserva identificado por idReserva
Tipo	AgenciaViajes
Notas	-
Excepciones	Si la reserva identificada por idReserva no existe
Salida	-
Precondiciones	-
Postcondiciones	Se destruyó el enlace entre el objeto Reserva identificado por idReserva y el cliente