

FIB - ECSDI 2022/23

Entrega final

Planificador de viajes



Carles Pedrals I Mansilla
Daniel Gallardo Peña
Genís Gutiérrez Bernal

1 Índice

1 Índice	1
2 Breve explicación del sistema desarrollado	4
3 System Specification	5
3.1 Analysis Overview Diagram	5
3.2 Actores externos	6
3.2.1 Usuario	6
3.2.2 Proveedor de transporte	6
3.2.3 Proveedor de hospedaje	6
3.2.4 Proveedor de actividades	7
3.2.5 Proveedor meteorológico	7
3.2.6 Entidad bancaria	7
3.3 Goal Overview Diagram	8
3.3.1 Preguntar satisfacción	8
3.3.1.1 Obtener viajes no valorados	8
3.3.1.2 Obtener satisfacción	8
3.3.1.3 Registrar satisfacción	8
3.3.2 Hacer recomendación	8
3.3.2.1 Obtener planes de viajes anteriores	8
3.3.2.2 Obtener registro de satisfacción usuario	9
3.3.2.3 Procesar recomendaciones	9
3.3.2.4 Obtener actividades similares	9
3.3.2.5 Decidir recomendaciones	9
3.3.2.6 Informar recomendaciones	9
3.3.3 Petición de viaje	9
3.3.3.1 Procesar petición	9
3.3.3.2 Planificar propuesta de viaje	9
3.3.3.2.1 Obtener hospedaje	10
3.3.3.2.2 Obtener transporte	10
3.3.3.2.3 Obtener actividades	10
3.3.3.2.4 Obtener satisfacciones registradas del usuario	10
3.3.3.2.5 Obtener viajes anteriores del usuario	10
3.3.3.2.6 Crear propuesta de viaje	10
3.3.3.3 Obtener confirmación	10
3.3.3.4 Gestión de pago	10
3.3.3.4.1 Obtener método de pago	11
3.3.3.4.2 Procesar pago	11
3.3.3.4.3 Registrar pago	11
3.3.3.4.4 Informar registro de pago	11
3.3.3.5 Registrar viaje	11
3.3.3.6 Informar resumen viaje usuario	11
3.3.4 Cambio de actividad planificada	11
3.3.4.1 Obtener predicción meteorológica	11

3.3.4.2 Obtener actividades canceladas	11
3.3.4.3 Obtener actividades a reemplazar	11
3.3.4.4 Obtener alternativa actividad	12
3.3.4.5 Registrar cambio de actividad	12
3.3.4.6 Informar cambio de actividad usuario	12
3.3.5 Actualizar estado viajes finalizados	12
3.3.5.1 Obtener viajes finalizados	12
3.3.5.2 Registrar viaje finalizado	12
3.4 Scenarios	14
3.4.1 Petición de viaje	15
3.4.1.1 Planificar propuesta de viaje	16
3.4.1.2 Gestión de pago	17
3.4.2 Cambio de actividad planificada	18
3.4.3 Preguntar satisfacción	19
3.4.4 Hacer recomendación	20
3.4.5 Actualizar estado viajes finalizados	21
3.5 System Roles	22
3.5.1 Contratador de viajes	23
3.5.2 Confirmador de viaje	24
3.5.3 Planificador de viajes	25
3.5.4 Preguntador de hospedaje	26
3.5.5 Preguntador de transporte	27
3.5.6 Preguntador de actividades	28
3.5.7 Preguntador de actividades canceladas	29
3.5.8 Registrador de viajes	32
3.5.9 Registrador de cambio de actividad	33
3.5.10 Gestor de pagos	34
3.5.11 Registrador de pagos	35
3.5.12 Recomendador	36
3.5.13 Informador de recomendaciones	37
3.5.14 Obtenedor de satisfacción	38
3.5.15 Registrador de satisfacción	39
3.5.16 Actualizador estado viajes	40
3.5.17 Comunicador viajes finalizados	41
4 Architectural Design	42
4.1 Data Coupling	42
4.1.1 Viajes confirmados	43
4.1.2 Viajes finalizados	43
4.1.3 Pagos	43
4.1.4 Satisfacciones	44
4.1.5 Actividades, Hospedajes y Transportes	44
4.2 Agent-Role Grouping	45
4.2.1 Ag. Contratador	46
4.2.2 Ag. Cobrador	46

4.2.3 Ag. Recomendador Y Satisfacción	46
4.2.4 Ag. Gestor de Viajes	46
4.2.5 Agentes Proveedores (actividades, transporte, hospedaje, previsión meteorológica)	47
4.3 System Overview	48
4.4 Agent Acquaintance	49
5 Descripción ontología	50
6 Nivel avanzado	52
6.1 Agentes de información con caché	52
6.2 Gestión de pago:	52
6.3 Perfilado de usuarios y recomendaciones	52
6.4 Planificación para más de una ciudad	53
6.5 Servicio de tiempo meteorológico para modificar planes entregados	53
7 Detalles implementación	55
7.1 Actualización del tiempo solo para el mismo día, sin caché	55
7.2 Actividades lúdicas únicas aptas para climas adversos	55
7.3 Actualización del tiempo en intervalos de 24h	56
7.4 Cuando cambia estado viajes	56
8 Planificación y repartición del trabajo	56
9 Evaluación resultados	59

2 Breve explicación del sistema desarrollado

En este documento se detalla el proceso de diseño y construcción de un sistema capaz de planificar una visita turístico-cultural a una ciudad específica.

En este, como respuesta de salida a la entrada de unos filtros y parámetros detallados, obtendremos un viaje totalmente planificado que incluirá:

- El método de transporte de ida y vuelta a la destinación.
- Los lugares de hospedaje para cada día del viaje.
- La planificación de 3 actividades diarias para mañana, tarde y noche.

Desde esta herramienta también se debe poder realizar el pago del mismo, proporcionando así un único punto que centralice toda la gestión de la planificación, que de otra manera debería realizar el usuario, hacia el sistema.

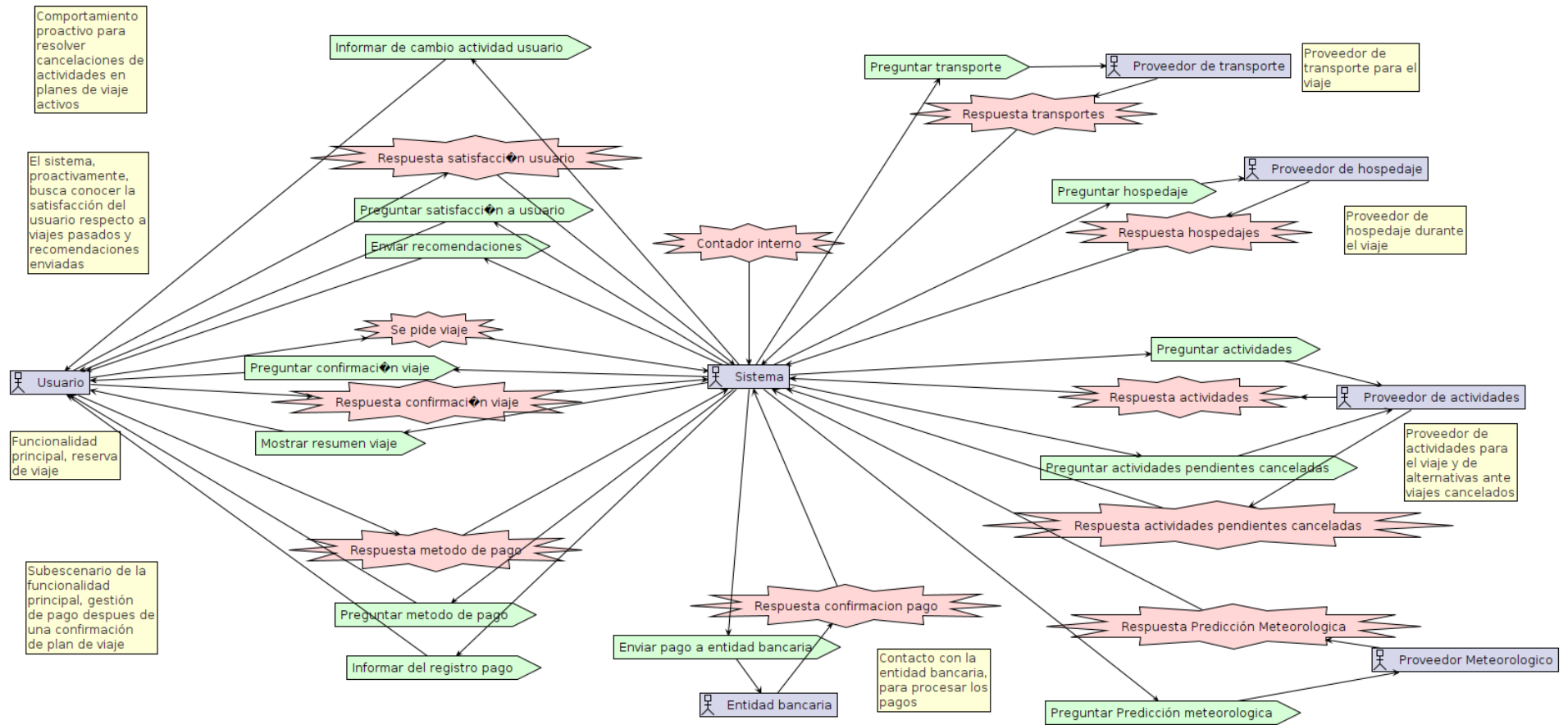
Este debe ser capaz de reaccionar de manera dinámica ante los posibles imprevistos que puedan surgir de la planificación de los viajes. Estos contratiempos pueden ser las cancelaciones de último momento de algunas actividades por parte de los organizadores o que el tiempo atmosférico impida realizarlas. En ambos casos deberán ser solucionadas modificando aquellas actividades afectadas por nuevas alternativas que puedan ser ajustadas a la planificación inicial.

Por último, el sistema también permitirá de manera dinámica obtener la valoración de los viajes que ha planificado anteriormente a sus usuarios, y ofrecer a estos nuevas recomendaciones basadas en sus gustos y preferencias.

Esta herramienta se desarrollará utilizando la metodología PROMETHEUS, que permitirá la confección de un sistema basado en agentes altamente desacoplados y dinámicos.

3 System Specification

3.1 Analysis Overview Diagram



3.2 Actores externos

Nuestro sistema interactúa con diversos actores externos para con tal de llevar a cabo su función. De todos ellos cabe destacar la diferencia entre el actores Usuario (actores de salida), que en última instancia es toda persona o sistema que utiliza nuestras funcionalidades, y el resto de actores de los cuales consumimos algún tipo de recurso o nos proveen con datos (actores de entrada).

3.2.1 Usuario

Único actor que pide datos de manera activa a nuestro sistema, donde después de especificar y enviar un seguido de preferencias a nuestro sistema, este recibe una propuesta de viaje personalizado y planificado teniendo en cuenta tanto viajes anteriores como valoraciones anteriores si existen. Si acepta la propuesta de viaje, este debe introducir los datos bancarios necesarios para el procesamiento del pago íntegro del viaje.

Fuera del comportamiento activo del actor usuario respecto al sistema, este también recibe de manera ocasional preguntas para la confirmación de conformidad respecto a una nueva actividad planificada que ha tenido que ser substituida por otra que ha sido cancelada o por el contrario la búsqueda de una nueva. Así, también puede recibir encuestas respondiendo por la valoración de las actividades y los hospedajes que ha recibido y completado del sistema.

3.2.2 Proveedor de transporte

Actor que provee con diferentes opciones de transporte a la destinación al especificar el día y la fecha tanto de ida como de vuelta a esta.

3.2.3 Proveedor de hospedaje

Aplicación externa que provee opciones de hospedaje como respuesta a una petición que especifica una franja de días, una franja de precios y algunos tags con preferencias y restricciones.

3.2.4 Proveedor de actividades

Actor encargado de proporcionar actividades según unos parámetros preestablecidos. Así, se puede escoger el tipo de actividad (lúdico, recreativa, festiva), la fecha, el momento (mañana, tarde, noche) y otro tipo de preferencias o restricciones a través de tags.

Este actor también proporciona información sobre las actividades que en algún momento ha planificado y han sido canceladas por cualquier motivo.

3.2.5 Proveedor meteorológico

Sistema externo que nos provee de predicciones de datos meteorológicos fiables para una franja temporal y espacial concreta.

3.2.6 Entidad bancaria

Actor encargado de manejar el contacto con la entidad bancaria. A través de este actor se realizan y confirman los pagos de los viajes creados por el sistema.

3.3 Goal Overview Diagram

Nuestro sistema está compuesto por siete grandes objetivos, donde cada uno de ellos formará un escenario. Dos de ellos, *Planificar Propuesta de Viaje* y *Gestión de pago* son subobjetivos de *Petición de Viaje*. A continuación se mencionan los objetivos que lo componen:

3.3.1 Preguntar satisfacción

El sistema cada cierto tiempo revisa los viajes que han finalizado recientemente para preguntar a los clientes la satisfacción del plan propuesto.

Este objetivo está formado por los siguientes subobjetivos:

3.3.1.1 Obtener viajes no valorados

Se obtiene toda la información del conjunto de viajes que han acabado recientemente y no han sido valorados.

3.3.1.2 Obtener satisfacción

Del conjunto de viajes obtenidos anteriormente, se pregunta al cliente la valoración del transporte, el hospedaje y las actividades realizadas.

3.3.1.3 Registrar satisfacción

Se registran las valoraciones emitidas por los clientes en la base de datos del sistema.

3.3.2 Hacer recomendación

El sistema cada cierto tiempo envía sugerencias de viajes a los clientes basándose en sus gustos y en planes de viaje que han realizado anteriormente.

Este objetivo está formado por los siguientes subobjetivos:

3.3.2.1 Obtener planes de viajes anteriores

Se obtiene toda la información de los viajes ya realizados anteriormente.

3.3.2.2 Obtener registro de satisfacción usuario

Se obtienen las valoraciones de satisfacción de los viajes realizados anteriormente.

3.3.2.3 Procesar recomendaciones

El sistema procesa la información de los dos objetivos anteriores y obtiene los tipos de actividades que más se adaptan a los gustos de los usuarios.

3.3.2.4 Obtener actividades similares

Se obtienen las actividades que se van a realizar en el futuro que más se adaptan a los patrones obtenidos en el objetivo anterior.

3.3.2.5 Decidir recomendaciones

Con la información de la información obtenida se decide qué recomendaciones se van a sugerir a los usuarios.

3.3.2.6 Informar recomendaciones

Se informa a los usuarios de los planes generados que pueden ser de su interés.

3.3.3 Petición de viaje

El sistema recibe una petición de planificación de viaje con ciertos parámetros de entrada y genera un plan de viaje.

Este objetivo está formado por los siguientes subobjetivos:

3.3.3.1 Procesar petición

El sistema analiza la petición y extrae los parámetros clave para poder generar el plan de viaje.

3.3.3.2 Planificar propuesta de viaje

El sistema genera una propuesta de viaje que satisfaga la petición del usuario.

Este objetivo a su vez está compuesto por los siguientes subobjetivos:

3.3.3.2.1 Obtener hospedaje

Se obtienen los hospedajes disponibles para el viaje acorde con los parámetros de la petición del usuario.

3.3.3.2.2 Obtener transporte

Se obtienen los transportes disponibles para el viaje acorde con los parámetros de la petición del usuario.

3.3.3.2.3 Obtener actividades

Se obtienen las actividades disponibles para el viaje acorde con los parámetros de la petición del usuario.

3.3.3.2.4 Obtener satisfacciones registradas del usuario

Se obtienen las valoraciones emitidas de viajes anteriores del usuario.

3.3.3.2.5 Obtener viajes anteriores del usuario

Se obtienen los viajes anteriores realizados por el usuario.

3.3.3.2.6 Crear propuesta de viaje

A partir de las opciones disponibles de transporte, actividades y hospedaje el sistema determina un plan de viaje teniendo en cuenta los viajes anteriores realizados y las valoraciones emitidas.

3.3.3.3 Obtener confirmación

Se requiere la confirmación por parte del usuario para confirmar el plan propuesto, si se rechaza este tendrá que realizar otra petición.

3.3.3.4 Gestión de pago

El sistema hace toda la gestión necesaria para cobrar el plan confirmado al cliente

Este objetivo a su vez está compuesto por los siguientes subobjetivos:

3.3.3.4.1 Obtener método de pago

Se obtiene el método de pago con el que el usuario desea pagar el viaje.

3.3.3.4.2 Procesar pago

Se procesa el pago del viaje contratado por el cliente.

3.3.3.4.3 Registrar pago

Se registra el pago en la base de datos del sistema.

3.3.3.4.4 Informar registro de pago

Se informa al cliente del resultado del proceso de pago.

3.3.3.5 Registrar viaje

Se registra que el viaje ha sido confirmado y pagado.

3.3.3.6 Informar resumen viaje usuario

Se informa al usuario que ha contratado un viaje con éxito.

3.3.4 Cambio de actividad planificada

El sistema puede cambiar las actividades planificadas que ha sido canceladas por nuevas actividades teniendo en cuenta el tipo de actividad que está siendo sustituido y las condiciones meteorológicas de la fecha.

3.3.4.1 Obtener predicción meteorológica

Se obtiene la predicción meteorológica de los próximos días.

3.3.4.2 Obtener actividades canceladas

Se obtienen las actividades que han sido canceladas.

3.3.4.3 Obtener actividades a reemplazar

Se obtiene todo el conjunto de actividades que se han cancelado o que por condiciones meteorológicas no se van a poder realizar.

3.3.4.4 Obtener alternativa actividad

Se obtienen alternativas para el conjunto de actividades que ya no se van a poder realizar obtenidas en el apartado anterior.

3.3.4.5 Registrar cambio de actividad

Se registran los cambios de actividad en la base de datos del sistema de los planes que lo requieran.

3.3.4.6 Informar cambio de actividad usuario

Se informa a los usuarios que hayan visto cambiado su plan de los cambios de actividades realizados.

3.3.5 Actualizar estado viajes finalizados

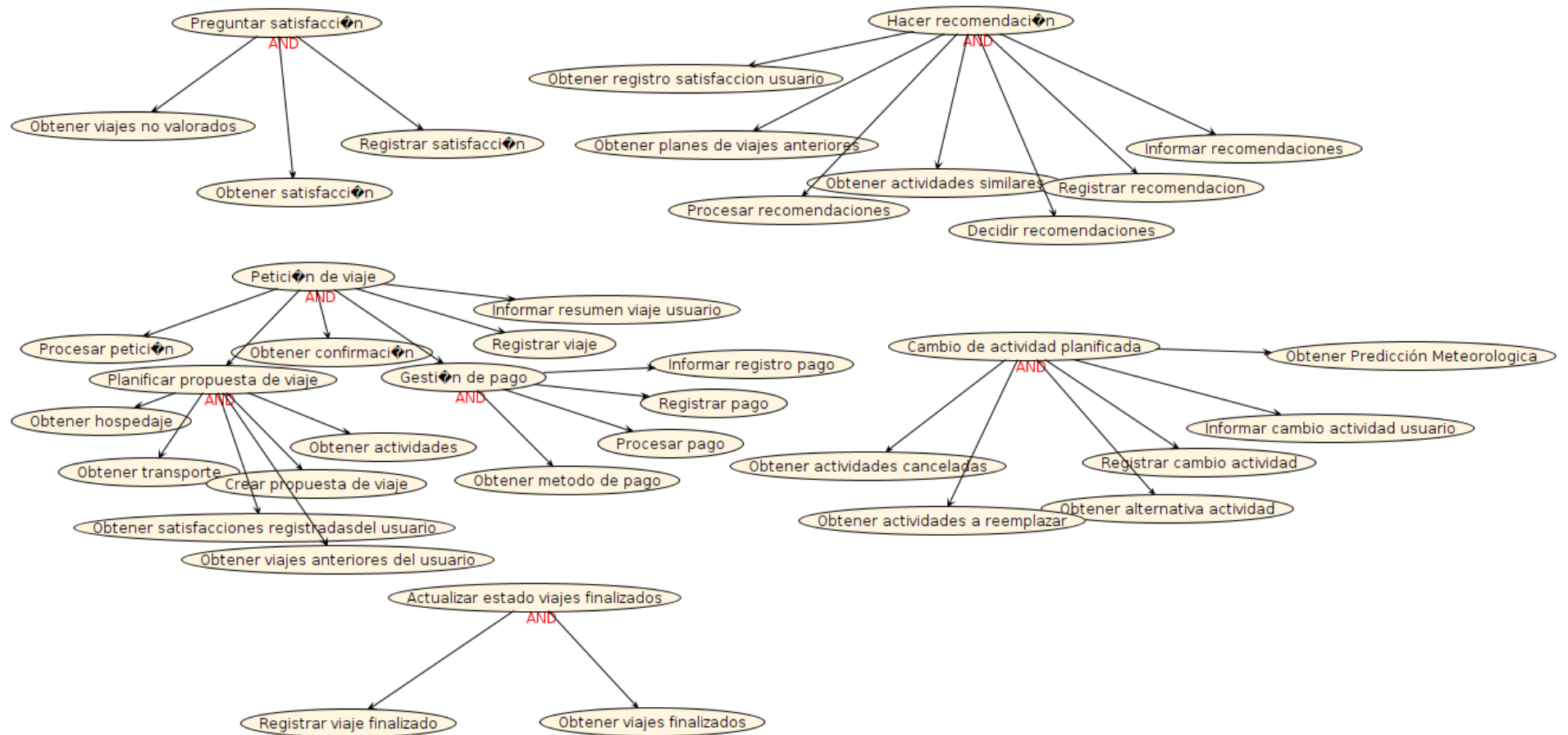
Cada día el sistema actualiza el estado de los viajes que han finalizado el día anterior en su base de datos.

3.3.5.1 Obtener viajes finalizados

Se obtiene el conjunto de los viajes que han finalizado el día anterior.

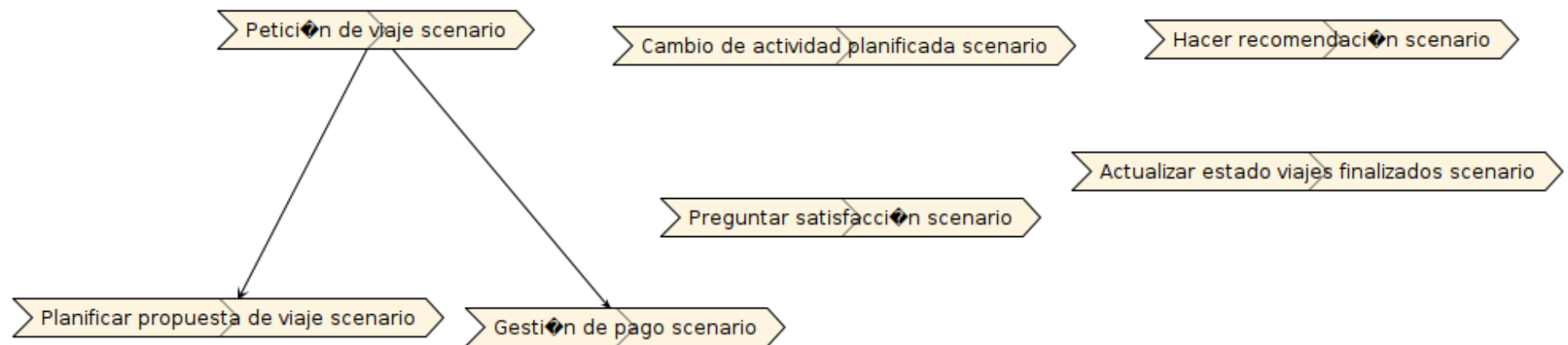
3.3.5.2 Registrar viaje finalizado

Se actualiza el estado de los viajes obtenidos en el objetivo anterior y se marcan como viajes finalizados.



3.4 Escenarios

Los escenarios están divididos según las diferentes funcionalidades que ofrece el sistema, como se puede ver en el diagrama.



Podemos ver como *Planificar propuesta de viaje* y *Gestión de pago*, son subescenarios de *Petición de viaje*, ya que son actividades lo suficientemente grandes y relevantes de forma independiente como para necesitar un escenario propio. Sin embargo, teniendo en cuenta el alcance de nuestro proyecto siempre serán llevadas a cabo como subactividad del primer escenario.

3.4.1 Petición de viaje

Scenario Petición de viaje scenario

Name	Petición de viaje scenario						
Description	Funcionalidad principal, reserva de viaje						
Priority	Not Specified						
Actors							
Initiated by	System						
Trigger							
Steps	#	Type	Name	Role	Description	Data used	Data produced
	1	Percept	Se pide viaje	Contratador de viajes	El Usuario realiza una petición de viaje		
	2	Goal	Procesar petición	Contratador de viajes	El sistema parsea la petición y extrae los parámetros clave para poder procesar la petición		
	3	Scenario	Planificar propuesta de viaje scenario		El sistema debe generar una propuesta de viaje para responder a la petición del usuario		
	4	Goal	Obtener confirmación	Confirmador de Viaje	Se requiere obtener la confirmación del plan de viaje por parte del usuario, si la rechaza se espera otra petición		
	5	Action	Preguntar confirmación viaje	Contratador de viajes	El sistema realiza la acción de preguntar al usuario si quiere reservar el plan de viaje que el anterior ha realizado		
	6	Percept	Respuesta confirmación viaje	Confirmador de Viaje	Se recibe la respuesta a la propuesta de viaje por parte del usuario		
	7	Scenario	Gestión de pago scenario		El sistema realiza la gestión del pago del viaje contratado.		
	8	Goal	Registrar viaje	Registrador de viajes	Una vez el pago ha sido confirmado, se registra el viaje (como viaje confirmado) y se elimina de viajes pendientes de confirmacion		Viajes confirmados , Viajes pendientes de confirmacion
	9	Goal	Informar resumen viaje usuario	Confirmador de Viaje	El objetivo del sistema es informar al Usuario del éxito del pago y de la información sobre el viaje que acaba de contratar		
	10	Action	Mostrar resumen viaje	Confirmador de Viaje	Se realiza la acción de mostrar el resumen al usuario del viaje que ha contratado		
Variation							

3.4.1.1 Planificar propuesta de viaje

Scenario Planificar propuesta de viaje escenario

Name	Planificar propuesta de viaje escenario						
Description	Se obtiene un viaje planificado acorde a los parámetros de entrada						
Priority	Not Specified						
Actors							
Initiated by	System						
Trigger							
Steps	#	Type	Name	Role	Description	Data used	Data produced
	1	Goal	Obtener hospedaje	Preguntador hospedaje, Planificador de viajes	Se requiere obtener los posibles hospedajes que entran dentro los parametros del usuario		
	2	Goal	Obtener transporte	Preguntador transporte, Planificador de viajes	Se requiere obtener los posibles transportes para desplazarse que entran dentro los parametros del usuario		
	3	Goal	Obtener actividades	Planificador de viajes, Preguntador actividades	Se requiere obtener los posibles actividades a realizar durante la estada, que entran dentro los parametros del usuario		
	4	Action	Preguntar hospedaje	Preguntador hospedaje	Se preguntan los hospedajes que cumplen los criterios a la entidad externa		
	5	Action	Preguntar transporte	Preguntador transporte	Se preguntan los transportes que cumplen los criterios a la entidad externa		
	6	Action	Preguntar actividades	Preguntador actividades	Se preguntan las actividades a realizar que cumplen los criterios a la entidad externa		
	7	Percept	Respuesta hospedajes	Preguntador hospedaje	Se reciben los hospedajes que cumplen los criterios por parte de la entidad externa		
	8	Percept	Respuesta transportes	Preguntador transporte	Se reciben los transportes que cumplen los criterios por parte de la entidad externa		
	9	Percept	Respuesta actividades	Preguntador actividades	Se reciben las actividades a realizar que cumplen los criterios por parte de la entidad externa		
	10	Goal	Obtener satisfacciones registradas del usuario	Planificador de viajes	Se obtienen las satisfacciones anteriores del usuario para poder realizar una propuesta a medida	Satisfacciones	
	11	Goal	Obtener viajes anteriores del usuario	Planificador de viajes	Se obtienen los viajes realizados anteriormente por el usuario	Viajes finalizados	
	12	Goal	Crear propuesta de viaje	Planificador de viajes	Se crea la propuesta de viaje para ser enviada al usuario		Viajes pendientes de confirmacion
Variation							

3.4.1.2 Gestión de pago

Scenario Gestión de pago escenario

Name	Gestión de pago escenario					
Description	Se realiza el cobro de un viaje contratado					
Priority	Not Specified					
Actors						
Initiated by	System					
Trigger	Un cliente ha contratado un viaje					
Steps	#	Type	Name	Role	Description	Data used
	1	Goal	Obtener metodo de pago	Gestor de pagos	Obtener metodo de pago del viaje contratado por el cliente	
	2	Action	Preguntar metodo de pago	Gestor de pagos	Se pregunta al cliente por los datos para realizar el pago	
	3	Percept	Respuesta metodo de pago	Gestor de pagos	Respuesta de los datos de pago del cliente	
	4	Goal	Procesar pago	Gestor de pagos	Procesar pago del cliente	
	5	Action	Enviar pago a entidad bancaria	Gestor de pagos	Se envian los datos de pago a la entidad bancaria para realizar el pago	
	6	Percept	Respuesta confirmacion pago	Gestor de pagos	Se recibe la respuesta del pago hecho por el cliente	
	7	Goal	Registrar pago	Registrador de pagos	Se registra el pago realizado por el cliente	Pagos
	8	Goal	Informar registro pago	Gestor de pagos	Se informa al cliente del pago que ha realizado	
	9	Action	Informar del registro pago	Gestor de pagos	Se muestra un resumen al cliente del pago realizado	
Variation						

3.4.2 Cambio de actividad planificada

Scenario Cambio de actividad planificada escenario

Name	Cambio de actividad planificada escenario						
Description							
Priority	Not Specified						
Actors							
Initiated by	System						
Trigger							
Steps	#	Type	Name	Role	Description	Data used	Data produced
	1	Percept	Contador interno	Obtenedor actividades canceladas	Se activa el trigger de un contador interno del sistema		
	2	Goal	Obtener Predicción Meteorologica	Preguntador previsión meteorologica, Obtenedor actividades canceladas	Se obtiene la prevision meteorologica de los proximos dias		
	3	Action	Preguntar Predicción meteorologica	Preguntador previsión meteorologica	Se pregunta la prevision meteorologica de los proximos dias		
	4	Percept	Respuesta Predicción Meteorologica	Preguntador previsión meteorologica	Se obtiene la respuesta de la prevision meteorologica de los proximos dias		
	5	Goal	Obtener actividades canceladas	Obtenedor actividades canceladas	Se obtienen las actividades que han sido canceladas por causas externas		
	6	Action	Preguntar actividades pendientes canceladas	Preguntador actividades canceladas	Se preguntan las actividades pendientes canceladas		
	7	Percept	Respuesta actividades pendientes canceladas	Preguntador actividades canceladas	Se obtiene la respuesta de las actividades pendientes canceladas		
	8	Goal	Obtener actividades a reemplazar	Obtenedor actividades canceladas	Teniendo en cuenta las actividades explicitamente canceladas y las que no se pueden hacer por causas meteorologicas, obtener el conjunto de actividades a reemplazar.	Viajes confirmados	
	9	Goal	Obtener alternativa actividad	Obtenedor actividades canceladas, Planificador de viajes	Se obtienen las alternativas a una actividad		
	10	Action	Preguntar actividades	Preguntador actividades	Se preguntan actividades disponibles por hacer		
	11	Percept	Respuesta actividades	Preguntador actividades	Se obtienen las actividades disponibles a contratar		
	12	Goal	Registrar cambio actividad	Registrador de viajes	Se registran las actividades cambiadas		Viajes confirmados
	13	Goal	Informar cambio actividad usuario	Planificador de viajes	Se informa al usuario de las actividades que han cambiado		
	14	Action	Informar de cambio actividad usuario	Planificador de viajes	Se informa al usuario de las actividades que han cambiado		
Variation							

3.4.3 Preguntar satisfacción

Scenario Preguntar satisfacción en escenario

Name	Preguntar satisfacción en escenario					
Description	Este escenario ocurre cuando se debe preguntar por la valoración de un viaje ya ocurrido a los usuarios					
Priority	Not Specified					
Actors						
Initiated by	System					
Trigger						
Steps	#	Type	Name	Role	Description	Data used Data produced
	1	Percept	Contador interno	Obtenedor satisfacción	Se recibe una llamada por trigger de tiempo	
	2	Goal	Obtener viajes no valorados	Obtenedor satisfacción	Se obtiene toda la información del conjunto de viajes que han acabado recientemente y no han sido valorados.	Viajes finalizados
	3	Goal	Obtener satisfacción	Obtenedor satisfacción	Se debe recibir las valoraciones de los viajes finalizados sin valorar	
	4	Action	Preguntar satisfacción a usuario	Obtenedor satisfacción	Se piden valoraciones de los viajes finalizados a los usuarios	
	5	Percept	Respuesta satisfacción usuario	Registrador satisfacción	Un usuario valora un viaje realizado	
	6	Goal	Registrar satisfacción	Registrador satisfacción	Se registra la valoración de un usuario	
Variation						

3.4.4 Hacer recomendación

Scenario Hacer recomendación en escenario

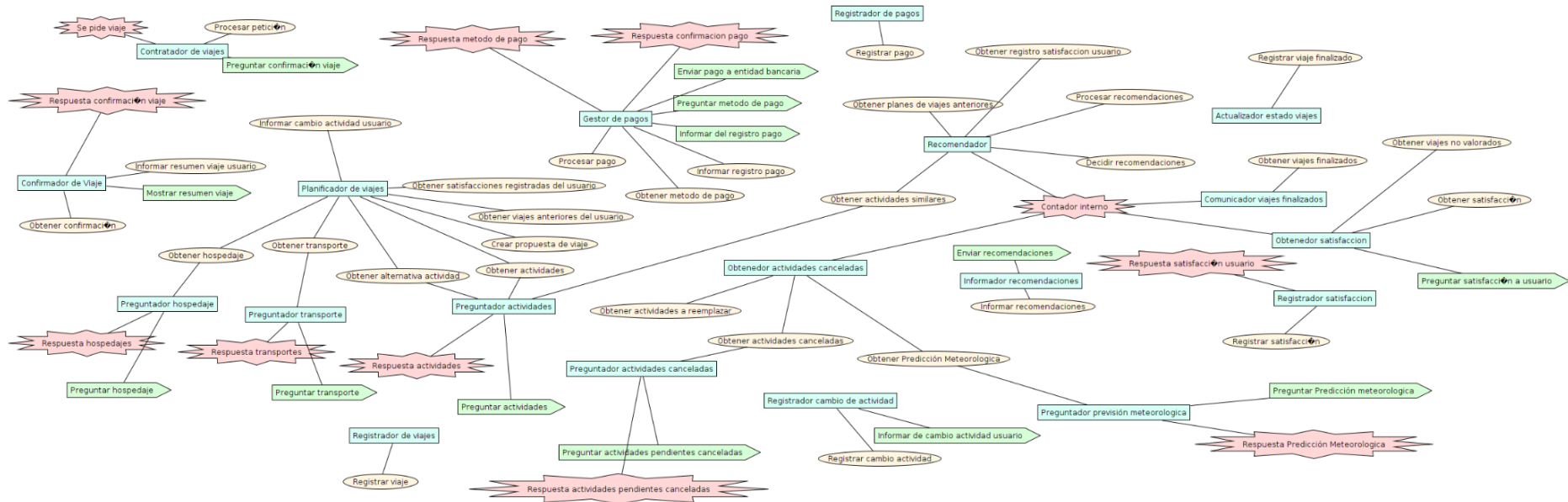
Name	Hacer recomendación en escenario						
Description	Se realizaz recomendaciones a los usuarios						
Priority	Not Specified						
Actors							
Initiated by	System						
Trigger							
Steps	#	Type	Name	Role	Description	Data used	Data produced
	1	Percept	Contador interno	Recomendador	Se recibe una llamada por trigger de tiempo		
	2	Goal	Obtener planes de viajes anteriores	Recomendador	Se deben obtener los planes de viaje anteriores del usuario	Viajes finalizados	
	3	Goal	Obtener registro satisfaccion usuario	Recomendador	Se obtienen las valoraciones de satisfaccion que ha hecho anteriormente el usuario	Satisfacciones	
	4	Goal	Procesar recomendaciones	Recomendador	Se deben obtener parametros comunes de la informacion de los viajes finalizados del usuario		
	5	Goal	Obtener actividades similares	Recomendador, Preguntador actividades	Se deben obtener actividades con similitudes a las especificadas		
	6	Action	Preguntar actividades	Preguntador actividades	Se preguntan a los agentes de actividades por actividades con parametros similares a los de las especificadas		
	7	Percept	Respuesta actividades	Preguntador actividades	Se obtiene la informacion con actividades similares a la de las especificadas		
	8	Goal	Decidir recomendaciones	Recomendador	Se deciden las actividades que seran recomendadas de un conjunto de ellas		
	9	Goal	Informar recomendaciones	Informador recomendaciones	Se deben mostrar a los usuarios las recomendaciones generadas por el sistema		
	10	Action	Enviar recomendaciones	Informador recomendaciones	Se informar a los usuarios de las recomendaciones hechas por el sistema para ellos		
Variation							

3.4.5 Actualizar estado viajes finalizados

Scenario Actualizar estado viajes finalizados scenario

Name	Actualizar estado viajes finalizados scenario					
Description	Registra los viajes que han acabado como viajes finalizados					
Priority	Not Specified					
Actors						
Initiated by	System					
Trigger						
Steps	#	Type	Name	Role	Description	Data used
	1	Percept	Contador interno	Actualizador estado viajes	Cada día a primera hora se marcan como finalizados los viajes que han acabado el día anterior	
	2	Goal	Obtener viajes finalizados	Actualizador estado viajes	Se obtienen de los viajes confirmados los que acabaron ayer	Viajes confirmados
	3	Goal	Registrar viaje finalizado	Actualizador estado viajes	Se eliminan de los viajes confirmados los que ya se han realizado y se registran como viajes finalizados	Viajes confirmados, Viajes finalizados
Variation						

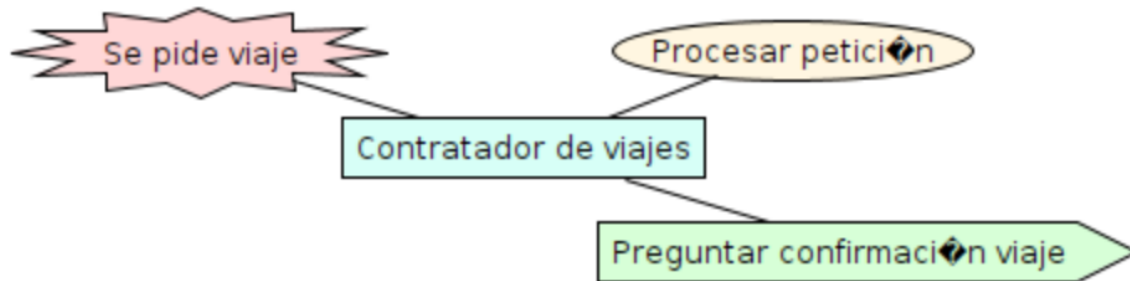
3.5 System Roles



Respecto a los roles del sistema, hemos decidido desarrollarlos siguiendo un esquema de “preguntador/obtenedor - registrador”. Así, separamos la lógica de aquellos componentes encargados de obtener la información del exterior de aquellos que se encargan de persistir la lógica en nuestro sistema. Esto se hace con el fin de obtener una especificación del sistema lo más desgranada posible para posteriormente tener una mejor comprensión de las responsabilidades de cada rol en la agrupación en agentes.

A continuación se describe cada uno de ellos en detalle.

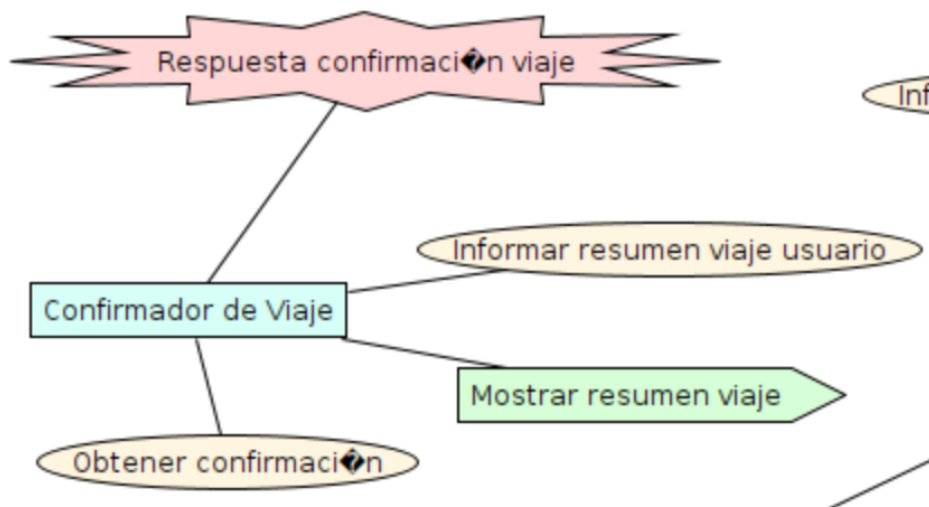
3.5.1 Contratador de viajes



Este es el rol que tiene como objetivo procesar las peticiones de viaje de los usuarios. Una vez obtiene los requisitos y limitaciones que pide el usuario, manda generar una propuesta de plan al rol *Planificador de viajes*. Cuando eventualmente este le responda, con una sugerencia de plan de viaje, *Contratador de viajes* le muestra el plan al usuario y le pide una confirmación.

Ahí se acaba el trabajo de *Contratador de viajes*, los siguientes pasos de la contratación del viaje serán responsabilidad de otros roles, favoreciendo así que al menos a nivel de roles cada uno se encargue de pocas tareas.

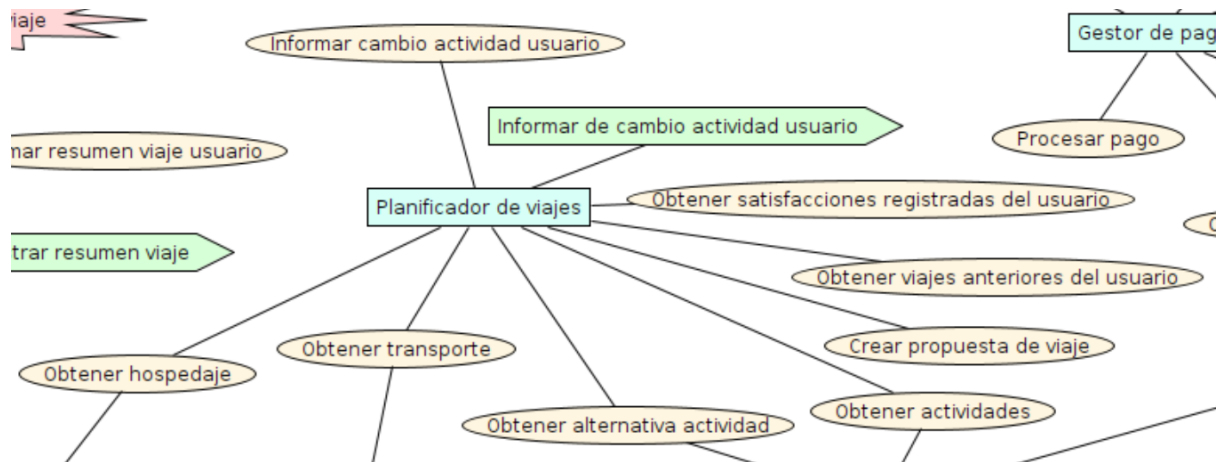
3.5.2 Confirmador de viaje



Este rol se encarga de continuar el proceso que *Planificador de viajes* ha iniciado. Una vez el cliente ha evaluado el plan de viaje que se le ha propuesto, este responde, y esa percepción es tratada por *Confirmador de viaje*. Si acepta, solo queda cobrar el viaje (que se delega a otro rol) y una vez hecho esto se muestra el resumen del viaje contratado al usuario.

La separación de los roles de *Contratador de viaje* y *Confirmador de viaje* nos ha parecido la opción más coherente, ya que el obtener la interacción con usuarios finales son puntos asíncronos, donde la respuesta no sabemos cuánto puede tardar en llegar (si es que llega) y separar responsabilidades evita crear cuellos de botella.

3.5.3 Planificador de viajes

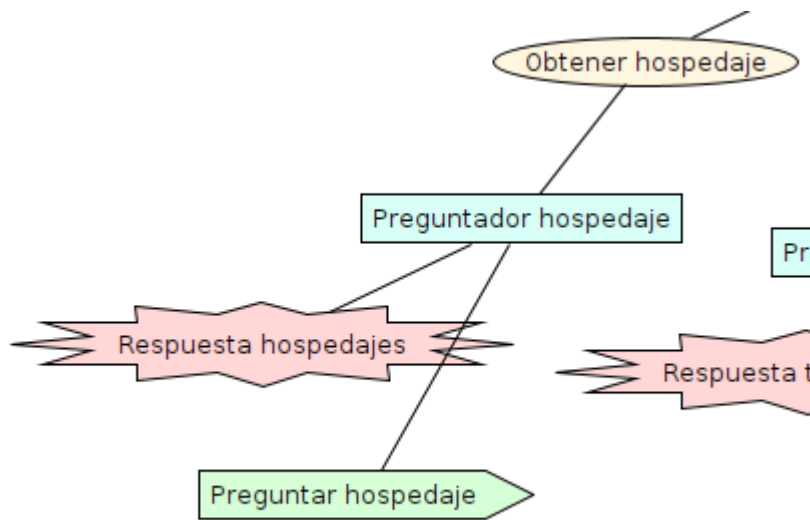


Este rol es el encargado de gestionar toda la lógica de negocio relacionada con crear los planes de viaje o modificar actividades de los ya existentes.

Cuando se le llama para crear un plan de viaje nuevo, es decir, un cliente hace una petición de que se genere un plan, *Planificador de viajes* obtiene todas las posibles opciones de actividades, transporte y hospedaje. Una vez hecho esto y teniendo en cuenta las restricciones y preferencias descritas en la petición, los viajes anteriores y valoraciones emitidas, genera el plan que mejor se adapta al cliente.

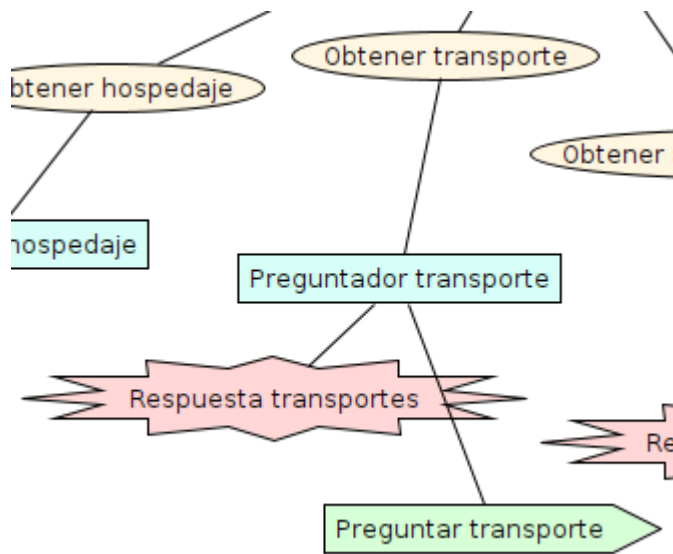
La otra situación donde actúa este rol es en el escenario de que haya actividades canceladas que requieren ser sustituidas. En este caso se le pasan el conjunto de las actividades a replanificar y este busca las mejores alternativas. Una vez hecho esto, también notifica al usuario de los cambios.

3.5.4 Preguntador de hospedaje



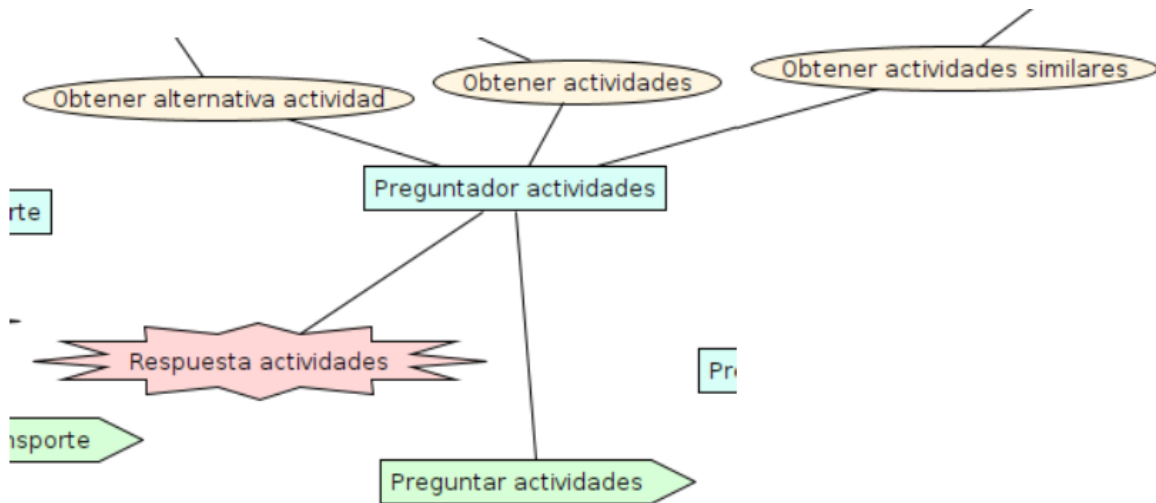
Este rol tiene como único objetivo interactuar con el actor externo que provee de los datos de hospedaje cuando se necesiten para generar un plan de viaje.

3.5.5 Preguntador de transporte



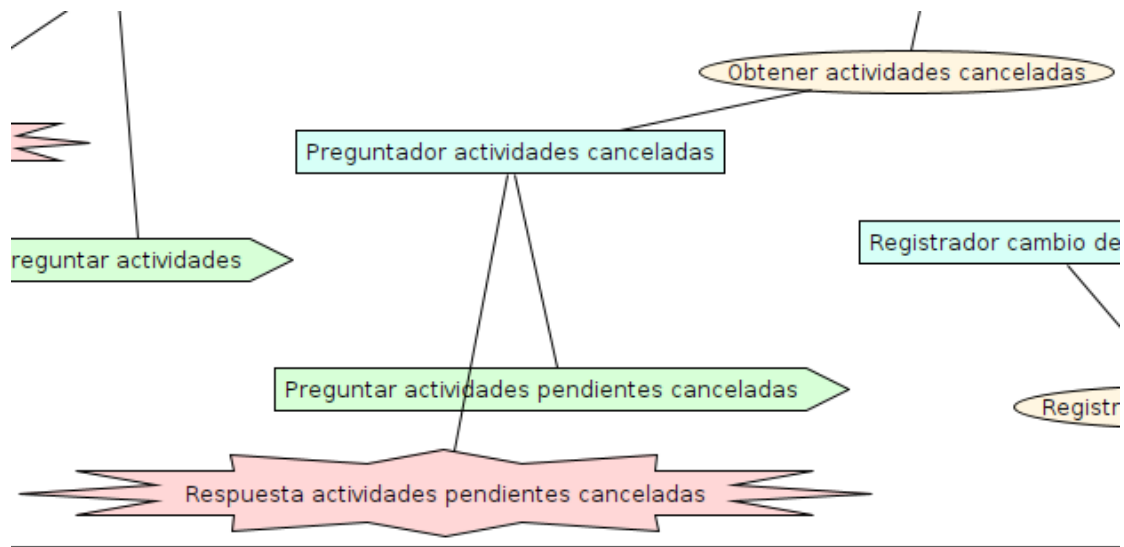
Este rol tiene como único objetivo interactuar con el actor externo que provee de los datos de transporte cuando se necesiten para generar un plan de viaje.

3.5.6 Preguntador de actividades



Este rol tiene como único objetivo interactuar con el actor externo que provee de los datos de actividades cuando se necesiten para generar un plan de viaje, sustituir actividades canceladas o obtener actividades similares para recomendar a los usuarios.

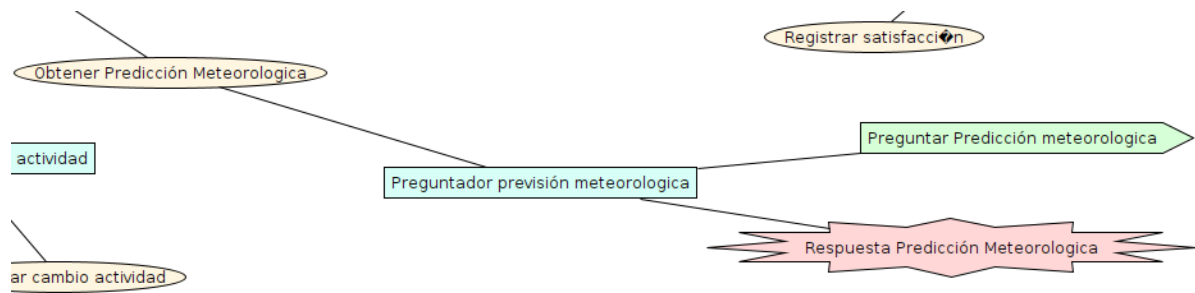
3.5.7 Preguntador de actividades canceladas



Este rol tiene como objetivo obtener las actividades que han sido canceladas al actor externo, para que puedan ser sustituidas. Estas actividades incluyen las que son explícitamente canceladas por el organizador, como pueden ser un concierto, festival o evento de cualquier tipo.

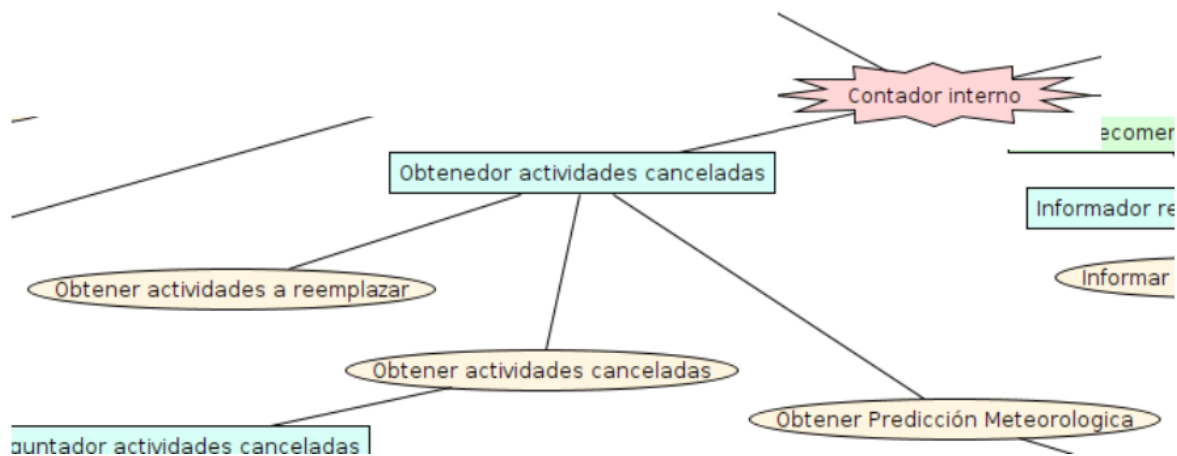
Sus funcionalidades y manera de trabajar es muy similar a *Preguntador de actividades*, pero decidimos separar estos roles siguiendo el principio de obtener un diseño del sistema lo más modular posible.

Preguntador previsión meteorológica



Este rol tiene como único objetivo interactuar con el actor externo que provee de los datos de la previsión meteorológica. Estos datos posteriormente son usados por el *Obtenedor de actividades canceladas* para obtener las actividades que no se van a poder realizar por inclemencias meteorológicas.

Obtenedor de actividades canceladas



Este rol es el encargado de obtener el conjunto de actividades que tienen que ser reemplazadas. Estas son:

- Actividades explícitamente canceladas por los organizadores y que son halladas por el rol *Preguntador de actividades canceladas*.
- Actividades que no se van a poder realizar por culpa del tiempo atmosférico. Aquí se usa a *Preguntador tiempo atmosférico* para conocer la previsión meteorológica de los próximos días. Con esta información se analiza todo el conjunto de actividades programadas para dichos días y se computan las que tienen que ser canceladas.

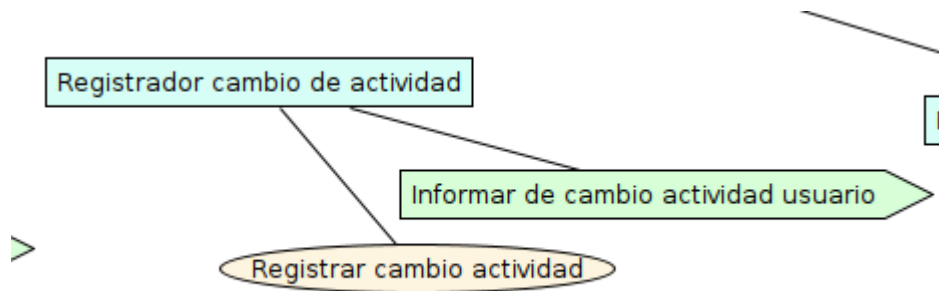
Este rol es activado por un contador interno que se dispara periódicamente.

3.5.8 Registrador de viajes



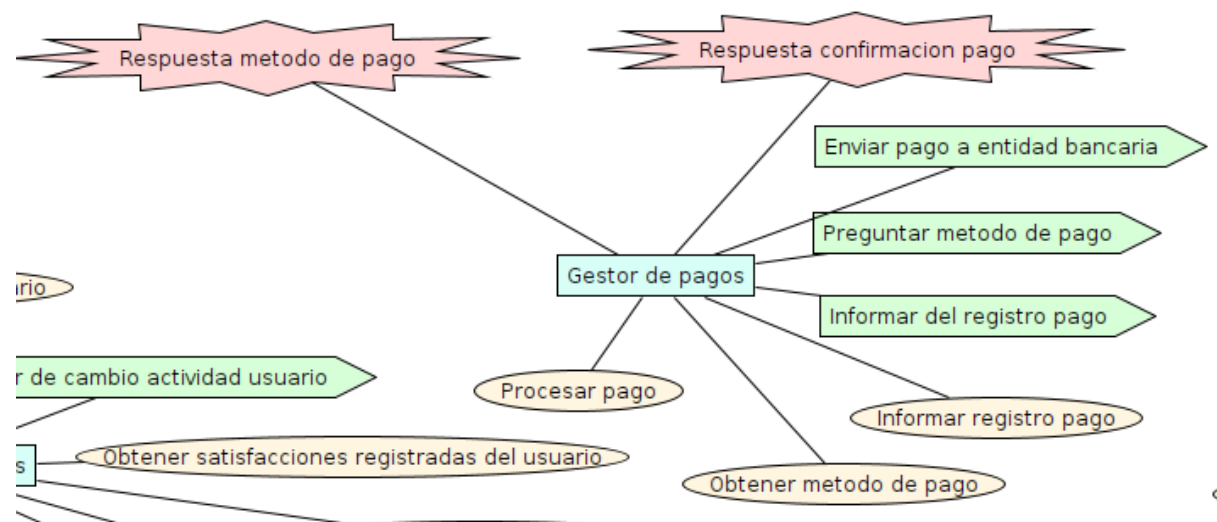
Este es uno de los roles mencionados en la introducción de este apartado que tiene como función registrar los viajes planeados en la base de datos. De esta forma se separa la lógica de procesamiento del proceso de persistencia de los datos.

3.5.9 Registrador de cambio de actividad



Otro rol del tipo registrador. Este tiene como objetivo registrar los cambios de actividad producidos por las cancelaciones en la base de datos del sistema. Una vez hecho esto informa al usuario.

3.5.10 Gestor de pagos



Este rol es el encargado de la parte de gestión de los pagos en el proceso de contratación del viaje.

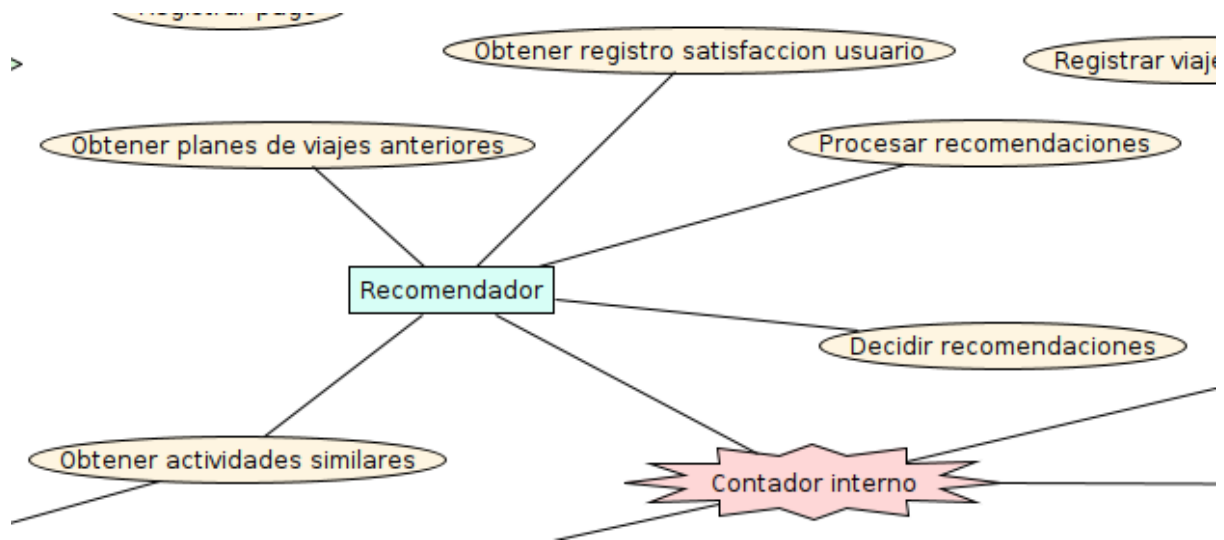
Una vez el cliente acepta la propuesta de plan del sistema, el *Confirmador de viajes* invoca al *Gestor de pagos*. Este le pide los datos bancarios al cliente, los valida y ejecuta el cobro. Una vez el agente externo que provee la conexión con la entidad bancaria comunica el resultado de la transacción, *Gestor de pagos* notifica dicho resultado al cliente y a *Confirmador de viajes*, para que finalice con la contratación del viaje.

3.5.11 Registrador de pagos



Otro rol del tipo registrador. Este tiene como objetivo registrar los cambios de actividad producidos por las cancelaciones en la base de datos del sistema.

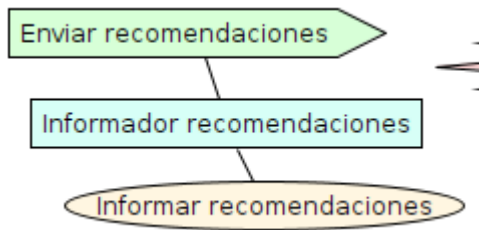
3.5.12 Recomendador



Este rol es el encargado de ejecutar toda la lógica de negocio relacionada con las recomendaciones.

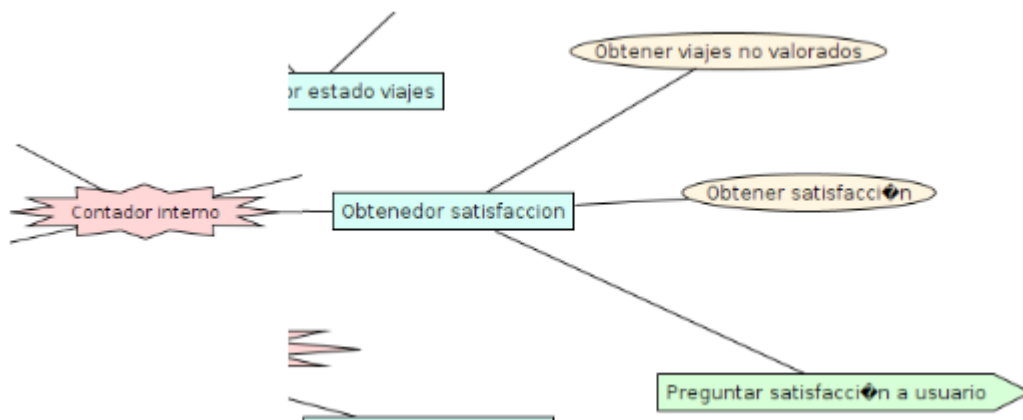
Primeramente obtiene los viajes anteriores y las valoraciones de satisfacción de esos planes (si existen). Después procesa toda esa información para obtener patrones comunes y los gustos de cada usuario. Con esto le pide a *Preguntador de actividades* que le proporcione actividades que se van a hacer en un futuro y coincidan con las conclusiones que ha obtenido respecto a los intereses de los usuarios. Finalmente decide qué recomendaciones son las más óptimas para enviar a los clientes

3.5.13 Informador de recomendaciones



Este rol tiene como único objetivo enviar las recomendaciones de planes de viaje a los clientes una vez el rol *Recomendador* ha generado las sugerencias.

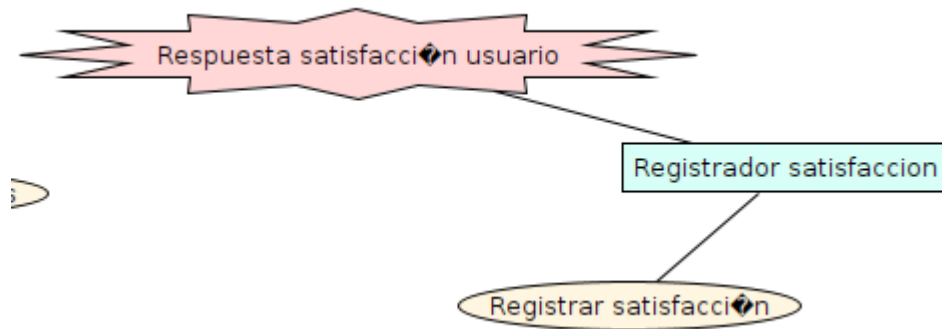
3.5.14 Obtenedor de satisfacción



Este rol es el encargado de obtener la satisfacción de los clientes del plan generado por el sistema una vez el viaje se ha realizado. Para eso primeramente obtiene de la base de datos los viajes que han finalizado recientemente y no han sido valorados. Para cada uno de estos, el sistema contacta con el cliente para pedirle una valoración de las actividades realizadas.

Este rol lo activa un contador interno que se dispara periódicamente.

3.5.15 Registrador de satisfacción



Otro rol del tipo registrador. Este tiene como objetivo registrar las valoraciones de satisfacción emitidas por los clientes en la base de datos del sistema.

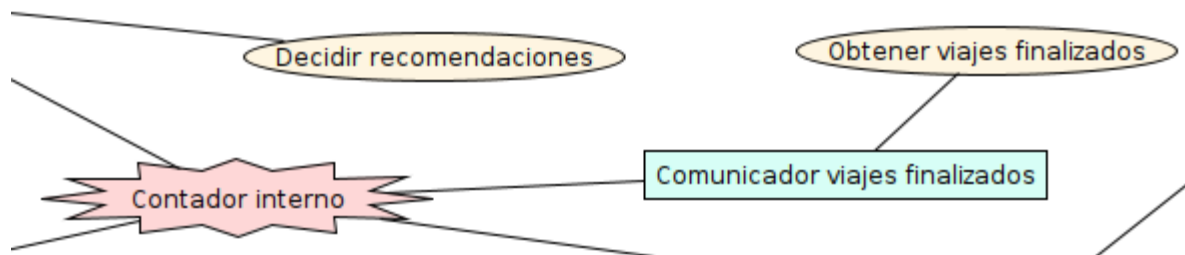
3.5.16 Actualizador estado viajes



Este rol tiene como objetivo marcar los viajes que han finalizado para poder ser procesados luego en las recomendaciones, para pedir satisfacción, etc.

Este rol lo activa el *Comunicador viajes finalizados* cuando le envíe el respectivo mensaje con los viajes finalizados.

3.5.17 Comunicador viajes finalizados



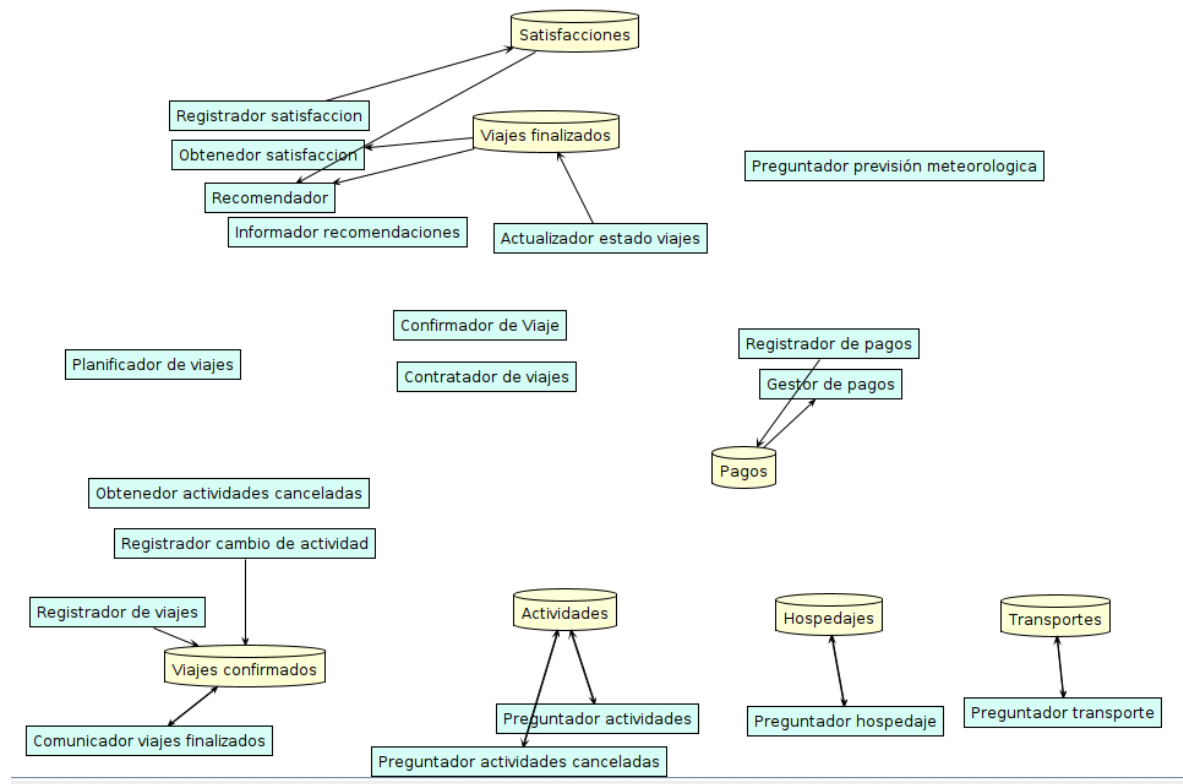
Este rol tiene como objetivo buscar los viajes que han finalizado recientemente y enviárselos a *Actualizador estado viajes* para que los procese adecuadamente. A pesar de estar muy interrelacionado con dicho rol, se han separado porque interactúan con información diferente. Uno tratará los viajes confirmados no realizados y el otro viajes que ya han pasado.

Este rol lo dispara un contador interno del sistema que se activa cada día.

4 Architectural Design

4.1 Data Coupling

A continuación se muestra la separación de fuentes de datos y que roles que tienen acceso a ellas.



4.1.1 Viajes confirmados

La base de datos de viajes confirmados almacena todos los datos de los viajes que han sido planificados por el sistema y confirmados por los respectivos usuarios pero que todavía no se han realizado.

Los roles que escriben en esta son el registrador de viajes al obtener un viaje recién pagado y el registrador de cambios de actividades al modificar un viaje que ya estaba registrado pero necesita un cambio en caso de cancelación de alguna actividad.

Con la nueva versión del diseño que hemos hecho recientemente, hemos desacoplado el *actualizador de viajes* de que tuviera acceso a varias bases de datos y por lo tanto ahora solo se encarga de escribir las actualizaciones en viajes finalizados, que recibirá vía mensaje. Para ello ha sido necesario la creación de *Comunicador viajes finalizados*, que se encarga de buscar estos viajes y traspasarlos.

4.1.2 Viajes finalizados

La BBDD de viajes finalizados contiene todos los viajes planificados por el sistema que ya han sido realizados.

En esta base de datos únicamente escribe el rol de actualizador del estado de viajes al recibir vía mensaje los viajes recientemente finalizados. Leen el obtenedor de satisfacción, para poder preguntar a los usuarios sobre sus experiencias en los viajes realizados, y el rol de recomendador para reajustar los perfiles de gustos de los clientes y ofrecer mejores recomendaciones.

4.1.3 Pagos

La base de datos de pagos almacena todos los registros de pagos que se realizan para pagar los viajes planificados por el sistema que son aceptados y pagados y es escrita únicamente por el registrador de pagos.

4.1.4 Satisfacciones

Las satisfacciones són las valoraciones que los usuarios que han ido a viajes planificados por el sistema otorgan a estos una vez los han realizado.

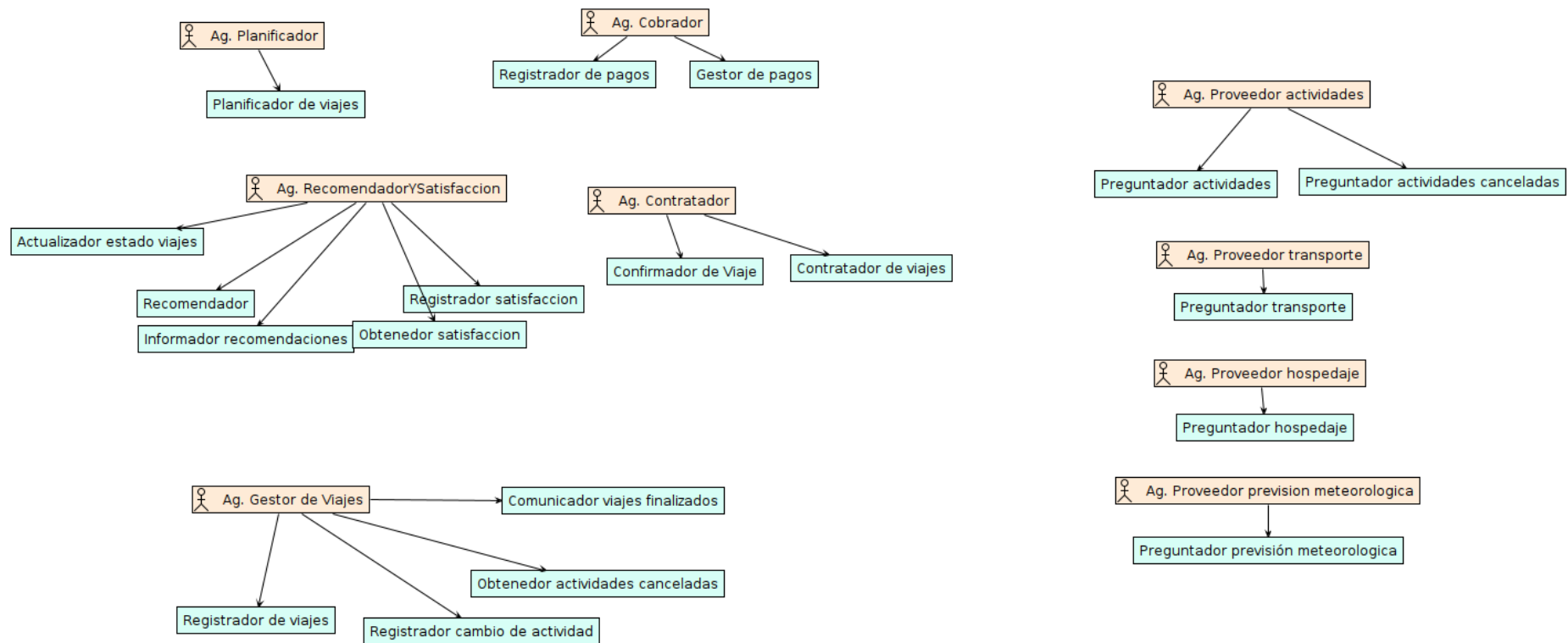
Así, esta base de datos es escrita por el rol encargado de registrar estas satisfacciones y es leída tanto por el rol de planificador de viajes como por el recomendador para que puedan ajustar sus outputs teniendo en cuenta las preferencias de los usuarios.

4.1.5 Actividades, Hospedajes y Transportes

Estas tres bases de datos son utilizadas tanto en escritura como en lectura por los roles encargados de obtener los respectivos datos como caché de almacenamiento de las respuestas obtenidas del exterior. Así, estos pueden evitar tener que realizar preguntas redundantes, mitigando la sobrecarga de la red y aumentando la velocidad de respuesta del sistema.

4.2 Agent-Role Grouping

La agrupación en agentes se puede ver en el diagrama.



A continuación se comentan las decisiones de este diseño y por qué se han agrupado de esta forma.

4.2.1 Ag. Contratador

El agente *Contratador* agrupa *Confirmador de Viajes* y *Contratador de viajes*. Ambos son los encargados de interaccionar con el usuario cuando este le pide un plan de viaje. No tiene acceso a ninguna fuente de datos, toda la información la obtiene vía mensajes.

4.2.2 Ag. Cobrador

Este agente integra los roles conceptualmente relacionados con el pago de los viajes contratados, en este caso *Registrador de Pagos* y *Gestor de Pagos*. Es el único poseedor de acceso a la fuente de datos de *Pagos*.

Realiza su trabajo siempre a petición del *Contratador de Viajes*, cuando éste le pide el pago de un viaje.

4.2.3 Ag. Recomendador Y Satisfacción

Este agente integra los roles conceptualmente relacionados con la generación y envío de recomendaciones de planes de viaje, en este caso *Recomendador* e *Informador de Recomendaciones*. También los roles conceptualmente relacionados con las valoraciones de satisfacción de los viajes finalizados, en este caso *Registrador de Satisfacción* y *Obtenedor de Satisfacción*.

Se han decidido juntar en este único agente, *RecomendadorYSatisfacción* ya que ambos requieren únicamente la información de viajes ya pasados y necesitan compartir mucha información entre sí (para recomendar se requiere acceder a las satisfacciones emitidas por el usuario).

4.2.4 Ag. Gestor de Viajes

Este agente contiene los roles encargados de la gestión de persistencia y coherencia de los viajes confirmados que todavía no se han realizado. Contiene los roles de *Comunicador viajes finalizados*, *Registrador de Viajes*, *Registador cambio de actividad* y *Obtener actividades canceladas*.

Se han juntado de este modo porque todos requieren acceso a la fuente de datos *Viajes Confirmados*.

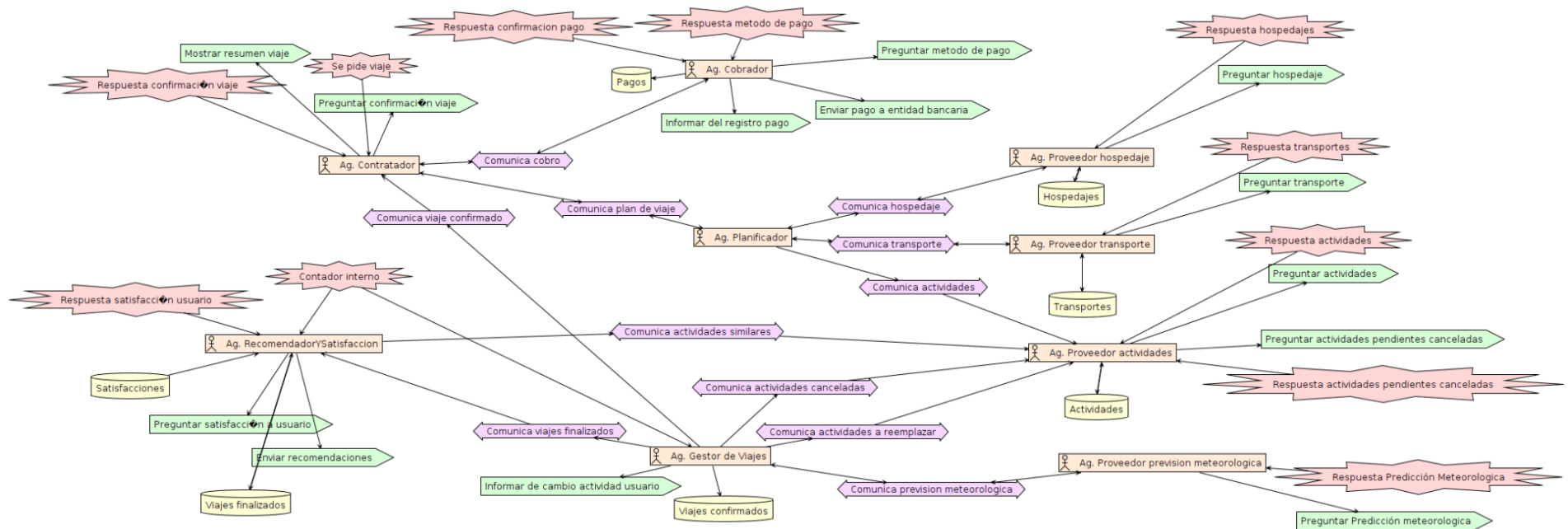
4.2.5 Agentes Proveedores (actividades, transporte, hospedaje, previsión meteorológica)

Son los agentes encargados de comunicarse con el exterior para obtener datos de y poder crear y actualizar los planes de viaje. Son los encargados de procesar esta información e introducirla dentro del sistema en un formato válido para poder ser interpretado por el resto de agentes.

Creemos que tiene sentido que para cada categoría de datos haya un agente distinto responsable de ella.

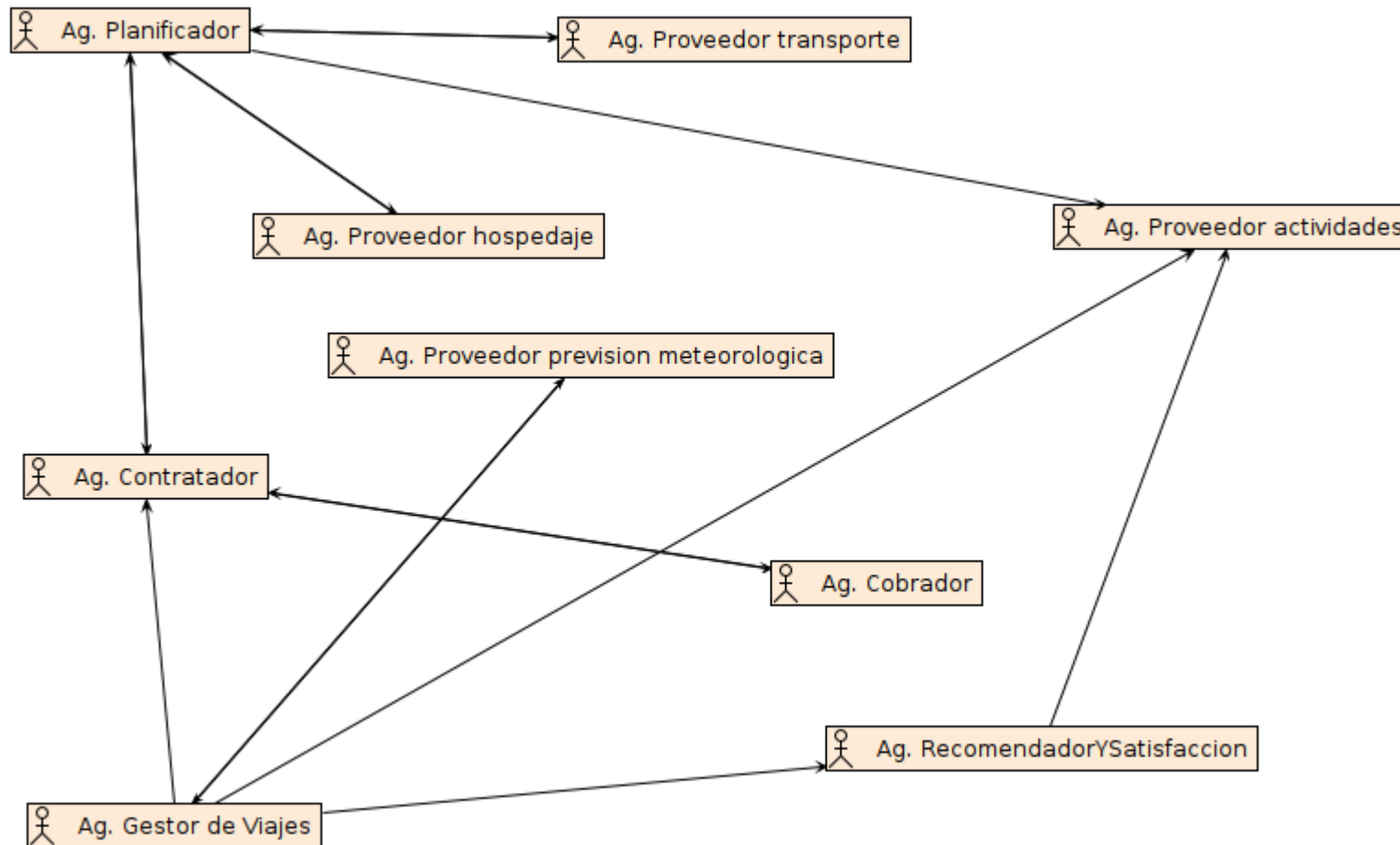
4.3 System Overview

Finalmente, la descripción general de la arquitectura del sistema se puede ver a continuación, donde se ilustran las diferentes percepciones que reciben los agentes, las acciones que realizan, las bases de datos de las que leen y los mensajes que se envían entre ellos para realizar sus tareas.



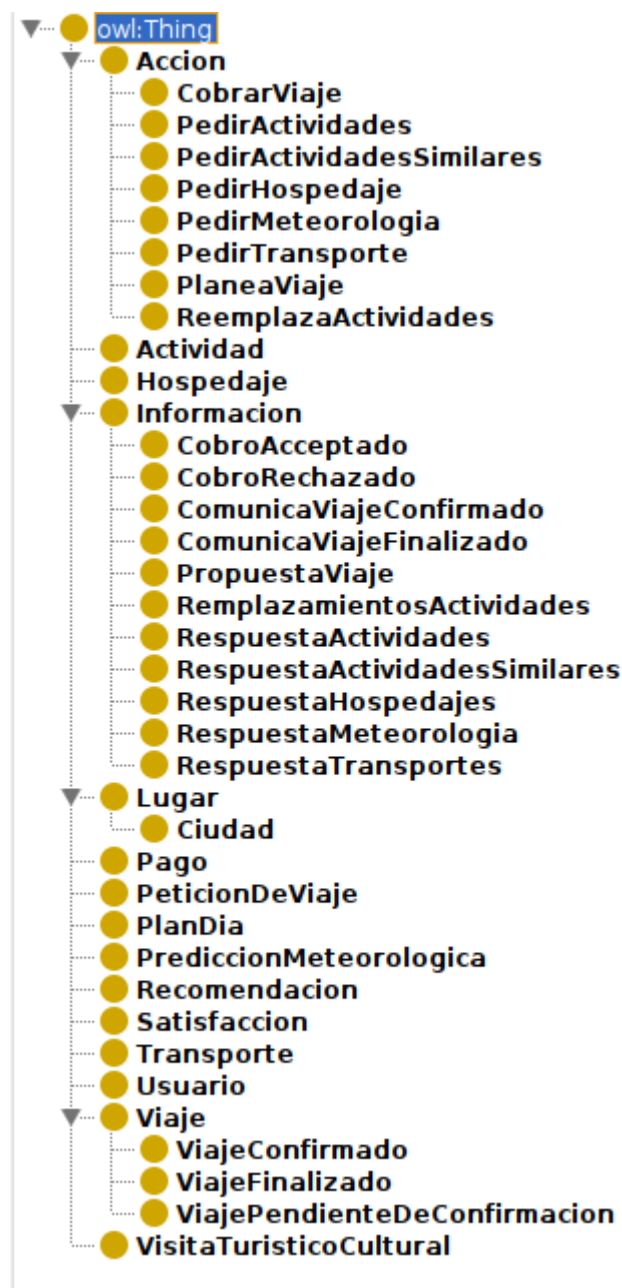
4.4 Agent Acquaintance

Para llevar a cabo las actividades que nuestro sistema debe desempeñar, estas son las comunicaciones necesarias entre agentes.



5 Descripción ontología

Para la descripción del sistema usando una ontología basada en notación rdf, se usan las siguientes clases:



De la clase Acción heredan todas clases cuyo cometido es una petición de información.

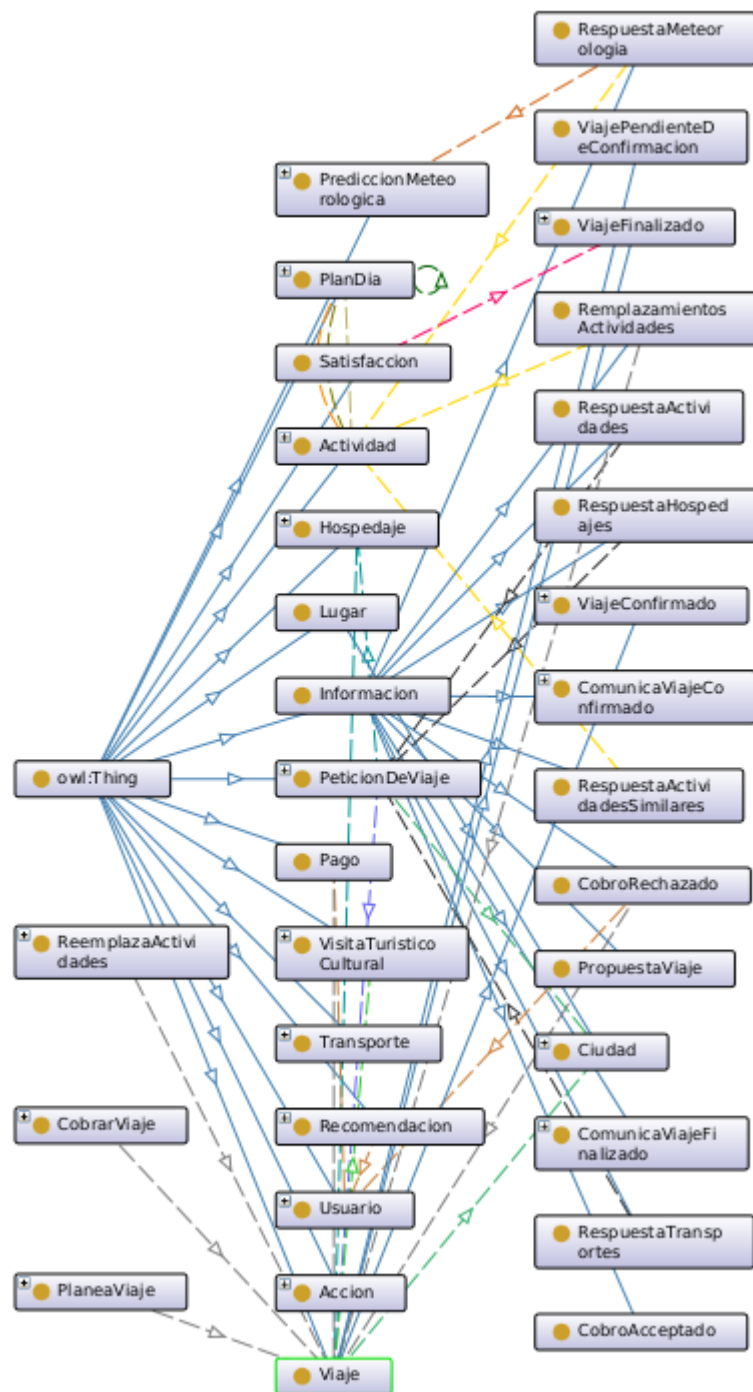
De la clase Información heredan las clases que representan mensajes de información o respuestas a acciones

Tenemos que el transcurso general de los elementos es:

PeticiónDeViaje se envía y responde con PlaneaViaje y PropuestaViaje. En ese proceso se comunica con Actividades, Transportes y Hospedaje para generar la PropuestaViaje. Una vez recibida, pasa al estatus de ViajePendienteDeConfirmación (si el usuario la acepta, está pendiente de confirmación bancaria). Seguidamente, se resuelve su estado con la acción CobrarViaje y

su resultado (Aceptado o Rechazado). En caso de que se cobre el viaje, correctamente, se envía mediante ComunicaViajeConfirmado. En ese estado, el ViajeConfirmado puede ser modificado debido a consultas meteorológicas hasta que finaliza y es usado posteriormente para recomendaciones y satisfacciones

Los detalles concretos se encuentran especificados mediante rdfs tags en el fichero de la ontología.



6 Nivel avanzado

Hemos implementado los siguientes extras en nuestra práctica:

6.1 Agentes de información con caché

Los agentes proveedores de información (AgenteProveedorHospedaje, AgenteProveedorActividades, AgenteProveedorTransporte) disponen de una caché donde se guarda la información de las búsquedas realizadas al exterior. Así, primeramente siempre se realiza una búsqueda en esta, y si no resulta exitosa, se activa una llamada a proveedores externos para poder resolver la petición recibida al mismo tiempo que se solicita información extra para poder guardar información de cara al futuro y ahorrar llamadas.

6.2 Gestión de pago:

Para la gestión de pagos se implementa un Agente Banco que se encuentra en una dirección fija, emulando lo que sería una API bancaria externa. En su interfaz, (DS-Hostname:9008/iface) se puede observar un diccionario con cuentas bancarias inicializadas y números de tarjeta en las cuales se pueden realizar cobros. Si los fondos de una cuenta son insuficientes para pagar un viaje, la transacción no es exitosa.

Estos pagos siempre se realizan al recibir una propuesta de viaje y es necesario realizar un pago exitoso para poder pasar una propuesta de viaje a viaje confirmado.

6.3 Perfilado de usuarios y recomendaciones

Contamos con un sistema de perfilado de usuarios basado en el nombre de usuario introducido en la petición inicial de viaje.

Cuando el sistema detecta que un viaje contratado ya ha finalizado, es decir que la fecha del vuelo de vuelo es anterior a la actual, este se mueve a una base de datos específica, donde se procede a procesar. Una vez aquí, si nos conectamos a la interfaz del AgenteRecomendadorYSatisfaccion(DS-Hostname:9011/iface) e introducimos un usuario, se puede dar una valoración de las actividades

recomendadas. Se pueden hacer tantas encuestas como viajes ya realizados que no han sido valorados previamente. Con esto el sistema almacena para cada usuario un perfil con la relación entre tipos de actividades que más le gustan.

A partir de este perfil de gustos, una vez al día el sistema activa un proceso donde para cada usuario que haya introducido *feedback* de sus viajes al menos una vez, le recomienda un plan de una semana con las actividades que más se ajustan a sus gustos. En nuestra implementación, este plan se almacena en el directorio `.ecsd/recomendaciones/<nombre_usuario>`.

6.4 Planificación para más de una ciudad

El sistema permite escoger entre más de una ciudad de salida y un conjunto de dos ciudades de llegada, Barcelona y Londres.

Esta funcionalidad está implementada de manera que al hacer la selección inicial entre Barcelona y Londres, los datos traducen el valor de la selección a su código IATA correspondiente, “BCN” o “LON” respectivamente.

Mediante este código podemos hacer las peticiones correspondientes a los servicios de amadeus de transporte y hospedaje ya que utilizan esta codificación para determinar el lugar donde buscar los servicios. Para las actividades, estas se buscan por pares de longitud/latitud, por lo que el mismo agente `AgenteProveedorActividades`, traduce de manera manual los valores de cada código a las coordenadas geográficas correspondientes y así obtener las actividades para cada ciudad. Posteriormente, las actividades guardan de manera asociada la longitud y la latitud a la que pertenecen para poder facilitar las búsquedas `sparkl` posteriores.

6.5 Servicio de tiempo meteorológico para modificar planes entregados

Implementamos un servicio de predicción meteorológico que permite modificar las actividades de planes confirmados según el tiempo.

Este funciona de tal manera que el agente `AgenteProveedorPrevisiónMeteorológica`, realiza una llamada al día (por la mañana) para obtener la predicción meteorológica por horas a servicios externos. Este agente obtiene la predicción para las horas 10, 18 y 23 y las traduce en tres franjas (`franja_mañana`, `franja_tarde`, `franja_noche`) con un booleano con si lloverá o no para cada una.

Esta información se transmite entonces al agente `AgenteGestorDeViajes`, que busca todas las actividades para ese día en las franjas horarias con lluvia, y en caso de haberlas, solicita nuevas actividades sustitutivas para cada una al agente `AgenteProveedorActividades`. Especificando dos parámetros, uno con la información que menciona que deben ser actividades cubiertas y otro con la especificación de la ciudad donde se deben llevar a cabo para que concuerden con la localización del viaje al que le pertenece la actividad cancelada que se está sustituyendo..

7 Detalles implementación

7.1 Actualización del tiempo solo para el mismo día, sin caché

Los agentes proveedores de información que acceden al exterior todos implementan una caché local que almacena información de las peticiones realizadas para poder reutilizarla en el futuro y ahorrar costes, todos menos menos AgenteProveedorPrevisiónMeteorológica.

Esto se ha decidido hacer así por diversos motivos. El primero de ellos siendo que el servicio solo realiza una petición al día, por lo que no resulta en un coste tan alto ni se beneficia en tanta medida de tener una caché como lo hacen el resto, que realizan múltiples peticiones demandando la misma información por segundo en un entorno real.

El segundo motivo tiene que ver con la misma naturaleza de la información representada. Las predicciones no dejan de ser probabilidades matemáticas con apuestas sobre el futuro, y por lo tanto hacer una petición y guardar información de una predicción una semana vista tiene mucho menos sentido que hacerla para el mismo día.

Así, AgenteProveedorPrevisiónMeteorológica sólo realiza una petición al día y no registra la información en ninguna caché local.

7.2 Actividades lúdicas únicas aptas para climas adversos

Para poder llevar a cabo la substitución de actividades según el clima hemos tenido que decidir qué actividades se consideran aptas para realizarse por lluvia y cuáles no.

Así, de todos los tipos de actividades, por la naturaleza y definición de amadeus, únicamente las de tipo lúdico están compuestas por actividades de tipo “shopping”, lo que nos asegura que en un muy alto porcentaje este tipo de actividades serán cubiertas.

Por lo tanto, todas las actividades que se realicen en un clima de lluvia, serán de tipo lúdico.

7.3 Actualización del tiempo en intervalos de 24h

La actualización de las actividades según la predicción meteorológica se lleva a cabo cada 24h por la mañana, ya que es la única que combina una predicción lo suficientemente fiable del clima meteorológico junto con un tiempo de respuesta adecuado para el usuario.

De esta manera nos aseguramos de que ni el usuario cancele planes que no deberían ser cancelados por una mala predicción meteorológica, ni tampoco que llegue tarde a los nuevos planes establecidos a causa de una actualización de los planes demasiado ajustada.

7.4 Cuando cambia estado viajes

Respecto a la actualización del estado de los viajes de confirmado a finalizado, de esto se encarga el agente `AgenteGestorDeViajes` cada 24h por la noche mediante un trigger automático, donde envía por mensaje toda la información de los viajes a `AgenteRecomendadorYSatisfaccion` y la elimina de sus bases de datos.

De esta manera podemos llevar a cabo las computaciones costosas cuando el sistema tiene menos demanda, moviendo así todos los viajes finalizados por la noche y creando todas las recomendaciones adecuadas a partir de estos nuevos datos mientras el sistema se encuentra en su mínimo uso para poder tener la máxima potencia durante el día.

8 Planificación y repartición del trabajo

La planificación y ejecución del trabajo ha seguido la guía inicial planteada por la asignatura donde, si bien puede haber habido desviaciones de un par de días entre ajustes y correcciones por feedback, se ha seguido la siguiente distribución temporal:

- **Primera Fase: 27 de febrero a 31 de marzo** Primera especificación del sistema y su diseño arquitectónico.
- **Segunda Fase: 1 de abril a 8 de mayo:** Primera corrección de la especificación del sistema y el diseño arquitectónico conjuntamente con el primer diseño de la ontología.
- **Tercera Fase: 9 de mayo a 29 de mayo:** Ampliación del diseño del sistema y la corrección de la ontología. Al mismo tiempo en esta fase se inicia la implementación del sistema.
- **Cuarta Fase: 29 de mayo a 5 de junio:** Finalización de la implementación del sistema, los juegos de prueba y toda la documentación asociada.

Respecto a la repartición del trabajo, esta se ha distribuido acorde a la siguiente tabla. En esta, una X indica un alto grado de implicación en el desarrollo de la tarea en cuestión, ya que si bien todos hemos hecho de todo, en algunas la implicación ha podido ser más secundaria con tareas como actuar de corrector o de apoyo a esta pero en bastante menor grado.

TAREA	GENÍS GUTIÉRREZ	CARLES PEDRALS	DANIEL GALLARDO
Especificación del sistema	X	X	X
Corrección especificación del sistema	X	X	X
Diseño arquitectónico	X	X	X
Corrección diseño arquitectónico		X	X

Diseño detallado	X	X	X
Corrección diseño detallado		X	X
Documentación	X		X
Ontología		X	
Corrección ontología	X	X	X
AgenteDirectorio	X	X	X
AgenteContratador	X	X	X
AgentePlanificador	X	X	X
AgenteProveedorActividades	X		
AgenteProveedorTransporte			X
AgenteProveedorHospedaje		X	
AgenteProveedorPrevisiónMeteorológica	X		
AgenteGestorDeViajes	X		
AgenteBanco		X	
AgenteCobrador		X	
AgenteRecomendadorY Satisfacción			X
cuentas_bancarias.html		X	
forzar_lluvia.html	X		
iface.html	X		
n_viajes_finalizados.html	X		
propuesta_viaje.html	X	X	X
ver_cambios_viaje.html	X		
viaje_confirmado.html	X		

9 Evaluación resultados

Después de debatir y razonar entre nosotros sobre los resultados alcanzados finalmente en nuestra entrega, hemos llegado a las conclusiones que se discuten en este apartado.

Primero de todo destacar que estamos más que satisfechos con los resultados obtenidos. Nuestra práctica cumple a la perfección las expectativas base que se requerían para la entrega con un sistema capaz de crear de manera distribuida una planificación de viaje a demanda con transporte, alojamiento y discriminación de actividades según preferencia para una ciudad.

Además, hemos conseguido ir más allá y aumentar la propuesta de valor del sistema con actividades opcionales extra que extienden la funcionalidad de este de la siguiente manera:

- Demanda de pago del viaje antes de su confirmación mediante tarjeta bancaria y comprobación de veracidad contra agente bancario.
- Creación de caché en los agentes de información que realizan llamadas a sistemas externos para aligerar la carga de trabajo de estos y evitar errores de tipo código 429 (exceso de llamadas a servicio). Así, estos agentes únicamente realizan búsquedas al exterior cuando la información que se les solicita no se encuentra en sus memorias, y en caso de realizar estas llamadas, aprovechan para solicitar más información de la necesaria en una única llamada reduciendo así los costes de interacción y la sobrecarga del sistema.
- Perfilado de usuarios mediante nombre de usuario que permite identificar a los solicitantes de los viajes y crear un sistema de recomendación de viajes según sus experiencias y feedback pasado.
- Actualización de viajes planificados según el tiempo meteorológico para cambiar actividades que se imposibilitarían por el clima a otras realizables en ese momento y notificación a los usuarios correspondientes.
- Planificación de viajes para más de una ciudad

Por lo tanto, podemos ver como efectivamente nuestro sistema cumple con las expectativas y extiende su valor mediante las actividades extra. Aún así, también es cierto que nos habría gustado poder implementar algunas funcionalidades más para acabar de perfeccionar el sistema y cerrar así el ciclo de desarrollo de manera total.

Si hubiéramos gozado de más tiempo o hubiéramos tenido una mejor gestión de este, nos habría gustado poder extender y solucionar las limitaciones siguientes:

- El agenteProveedorActividades realiza las llamadas externas según el tipo de actividad, lo que hace que tener que tener que llenar la caché con actividades de todos los tipos requiera de un mínimo de 3 llamadas. Una solución más eficiente habría sido efectuar una llamada en batch y luego de manera interna separar y etiquetar las actividades de manera singular, reduciendo así hasta un 66% las llamadas totales de este agente a un coste de mayor complejidad interna, pero con claros beneficios respecto al gasto total.
- El perfilado de usuarios se identifica únicamente con el nombre de usuarios. Añadir un sistema simple de identificación mediante usuario - contraseña habría ayudado a aportar más realismo y seguridad a la aplicación.
- Permitimos la creación de planes para más de una ciudad, en concreto, Londres y Barcelona. Aún así, con más tiempo podríamos haber implementado muchas ciudades más ya que la implementación actual permite añadir más localizaciones con muy pocos cambios. En efecto, mediante la utilización de un servicio de traducción de códigos IATA a coordenadas geográficas podríamos convertir el sistema a un servicio de planificación de viajes global.
- La modificación de actividades según el clima tiene dos “limitaciones” considerables. La primera es que solo permite cambiar actividades según la predicción del mismo día, lo cual puede ser positivo, ya que es la más correcta y evita cambiar actividades a más de dos días vista que podrían no ser correctas. Aún así, tal vez un sistema de alerta, en vez de modificación, para predicciones climáticas adversas para los días distantes habría sido un buen añadido. La segunda limitación está relacionada con lo que consideramos “condiciones climatológicas adversas”, ya que solo la lluvia o

las nevadas ocasionan cambios actualmente. Una mejora si hubiéramos contado con más tiempo podría haber sido el cambio de actividades en caso de calor extremo o alertas de calor.

- Por último, una buena mejora respecto a las recomendaciones de viajes habría sido la implementación de algún sistema de recomendación colaborativa mediante el algoritmo k-means por ejemplo, y no solo la recomendación respecto a viajes similares pasados del mismo usuario como realizamos actualmente.

Finalmente, podemos concluir que nuestro sistema cumple con todos los requisitos requeridos para ser considerado excelente en el contexto del aprendizaje de sistemas distribuidos y el desarrollo académico donde, con muchas horas detrás de él, resulta en un sistema fuerte, robusto y completo. Pero ciertamente, para poder llevarlo a producción y competir en el mercado a niveles profesionales, aunque no es el objetivo, requeriría de ciertas funcionalidades extra que le permitieran tener ventajas competitivas respecto a la competencia.