

# Modelo y desarrollo de W-planner: sistema multiagente on-line aplicado al turismo electrónico

Luis F. Castillo<sup>1</sup>, J. M. Corchado<sup>2</sup> y Manuel G. Bedia<sup>3</sup>

*Recepción: 03 de agosto de 2004 — Aceptación: 15 octubre de 2004*

*Se aceptan comentarios y/o discusiones al artículo*

---

## Resumen

Este trabajo introduce el llamado “problema del turista” y presenta una solución basada en un sistema multiagente. Un grupo de agentes que usan un sistema para identificar acciones y planes es capaz de encontrar el itinerario más adecuado para un turista de acuerdo con ciertas restricciones. Los artefactos “sin hilos” se usan para que el turista interactúe con el agente. Se utilizan técnicas de “cálculo variacional” durante el proceso para identificar el conjunto de soluciones posibles del problema y técnicas de “campos de Jacobi” para encontrar la solución “más fácilmente replanificable”. Este método analítico facilita la identificación de un itinerario turístico previamente y es capaz de modificar la ruta propuesta en tiempo de ejecución. Para finalizar se muestra el seguimiento de un caso de uso típico, en el cual un turista solicita al W-planner bajo ciertas condiciones la ruta más ajustada a sus requerimientos.

**Palabras claves:** agentes, sistemas multiagentes, soluciones inalámbricas, razonamiento basado en casos.

## Abstract

This paper introduces the “tourist problem” and presents an Multi-agent system based solution for it. A set of agents that uses a case-based reasoning system to identify actions and plans is capable of determining the most suitable itinerary with restrictions for a tourist. Wireless devices are used by the tourists to interact with the agent. Variational Calculus is used during the reasoning process to identify the set of possible problem solutions and Jacobi fields to find the most replanning-able solution. This analytical method facilitates the identification of a tourist itinerary in advance and is also capable of modifying the tourist route in execution time. To conclude, a case of typical use is shown, in which a tourist requests to the W-planner the most appropriate route that fits in well with the requirements.

**Key words:** agents, multiagent system, wireless solutions, case based reasoning.

---

<sup>1</sup> Doctor en Informática y Automática, lfcastil@autonoma.edu.co, profesor investigador, Universidad Autónoma de Manizales.

<sup>2</sup> Doctor, corchado@usal.es, profesor investigador, Universidad de Salamanca.

<sup>3</sup> Doctor en Informática, bedia@usal.es, investigador, Universidad de Salamanca.

## 1 Introducción

En este trabajo se plantea desarrollar un sistema multiagente [16] que utilice un mecanismo de razonamiento basado en casos para generar sus planes, y dar solución al problema de planificación de un turista que visita por primera vez una ciudad y que exige condiciones sobre la ruta solución (por ejemplo, lo que debe durar o lo que puede gastar). Se fundamentará este planteamiento en un soporte matemático basado en el cálculo variacional y en técnicas de Campos de Jacobi, que permitirá introducir un mecanismo para la planificación y re-planificación del agente en tiempo real.

En el momento actual, el formalismo y la implementación de agentes software y de sistemas multiagentes constituye el trabajo de investigación de numerosos científicos. Se ha demostrado en los últimos años que el modelo de agente más adecuado es aquel que estructura sus capacidades con base en información diferente, la cual se clasifica en tres tipos de instancias: conocimiento del sistema (Beliefs), objetivos que pretende alcanzar (Desires) y compromisos entre su conocimiento y el modo de alcanzar los objetivos (Intentions). Estos sistemas se conocen como agentes BDI. En [13] se establece un formalismo de agentes BDI [6] a partir de lógicas multi-modales pero encuentran gran diferencia entre la potencia de esas lógicas y lo que se requiere para la actuación práctica de tales sistemas.

En este estudio se presenta un modelo de trabajo que contempla un modo de implementación basado en técnicas analíticas. Además se muestra cómo la estructura de agentes a los que se implementa un motor de razonamiento basado en casos CBR (Case Based Reasoning) [1] puede acelerar sustancialmente el proceso de deliberación y resuelve los problemas relacionados con la capacidad de aprendizaje de los agentes: implementar agentes en la forma de sistemas CBR facilita su aprendizaje y adaptación [5].

Se comenzará revisando los conceptos generales sobre agentes, planificación, procesos CBR, y el cálculo variacional junto a técnicas de Jacobi que muestran como automatizar el razonamiento de agentes con las características mencionadas. Como se verá, estas herramientas matemáticas permiten automatizar el ciclo de razonamiento del agente, fundamentalmente durante la etapa de actualización, para garantizar un mecanismo de planificación y re-planificación eficiente en tiempo real.

Posteriormente, se presenta el esquema de funcionamiento del diseño propuesto, conocido como W-planner, para dar servicios en el sector del turismo electrónico, usando metodología de sistemas multiagentes que previamente se habrá introducido.

Para finalizar se mostrará el funcionamiento del sistema y las conclusiones generales.

## 2 Agente BDI-CBR aplicado al problema de planificación turística

Los sistemas de razonamiento basado en casos (CBR) se usan para resolver problemas mediante la adaptación de soluciones que fueron usadas en la resolución de problemas

similares. El proceso de trabajo de un sistema CBR [15] (figura 1) consiste, fundamentalmente, en la adaptación de viejas soluciones para construir planes asociados a nuevas experiencias, usando casos ya validados para plantear nuevas propuestas, para formular nuevas soluciones o para razonar de manera análoga en situaciones similares.

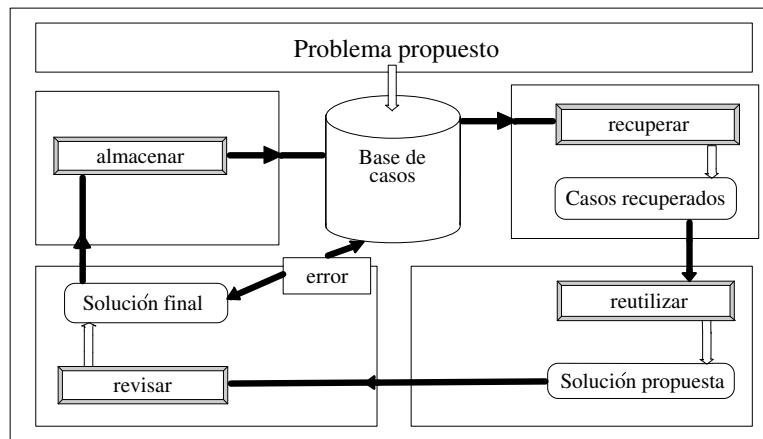


Figura 1: Ciclo de vida del CBR

En este trabajo se hace referencia a un problema de planificación turística en el que un usuario (turista) solicita un plan (ruta) de acuerdo con unos requerimientos iniciales (Costo, Tiempo de Visita, Hora de Inicio, etcétera) a un sistema que almacena un conjunto de planes definidos con anterioridad. La solución planteada debe cumplir con las restricciones impuestas. La elección de una arquitectura BDI-CBR se justifica en trabajos previos en los que se ha demostrado la conveniencia de esta combinación [7].

El sistema propuesto está compuesto por múltiples agentes que interactúan, coordinándose y dando una solución acorde con las necesidades. Esto permite elaborar una arquitectura distribuida que presenta propiedades deseables para enfrentarse a entornos típicos de planificación no-clásica como la escalabilidad, fiabilidad y reutilización de soluciones, que produce mejoras en el desempeño de la solución. La arquitectura multiagente se centra en un **agente planificador** [4] que se comunica con otros agentes para dar soluciones al usuario (representado por un dispositivo inalámbrico). La elección de un CBR como motor de razonamiento no es casual y se debe a las características intrínsecas de nuestro problema ya que, aunque se admiten cambios en el entorno que pueden modificar planes en ejecución, la estructura del entorno (monumentos a visitar, distribución de ellos, costo de la visita, tiempo promedio de la visita) permanece estática, lo que posibilita aprovechar planes que en el pasado han sido exitosos para otros turistas con condiciones similares. Este es esencialmente el modo de trabajo de un CBR.

### 3 Formalización del modelo teórico

El modelo propuesto está basado en una arquitectura BDI (Beliefs, Desires, Intentions - Deseos, creencias e Intenciones) [14]. A continuación se pasa a presentar con detalle el lenguaje utilizado para trabajar con estos tres componentes que conforman una estructura BDI.

- **Creencias del Sistema:** monumentos que los turistas pueden visitar

El entorno es el conjunto de monumentos, plazas, lugares públicos, museos, etcétera, que representan un interés turístico para aquellos que desean visitar una ciudad. Todos estos elementos de interés pueden ser caracterizados por un conjunto de atributos que los identifican. En este mundo, una creencia, por ejemplo, “Monumento”, es un vector de atributos del entorno que caracterizan el monumento. Una creencia particular, por ejemplo, “Catedral Nueva”, es un vector como el anterior que valora en sus variables (tabla 1).

**Tabla 1:** Creencia “Monumento” y creencia “Catedral Nueva”

| Creencia Monumento |                      | Creencia M1 |                     |
|--------------------|----------------------|-------------|---------------------|
| t1                 | nombre               | t1          | Catedral Nueva      |
| t2                 | horario              | t2          | {10:00-18:00} horas |
| t3                 | costo visita         | t3          | 3 euros             |
| t4                 | indicador de calidad | t4          | 1.5                 |
| t5                 | zona                 | t5          | 2 horas             |
| t6                 | tiempo visita        | t6          | tiempo visita       |

El sistema tendría tantas creencias como instancias de monumentos posea una ciudad, ejemplo tabla (2).

**Tabla 2:** Creencias M2, M3, M4

| Creencia M2 |                     | Creencia M3 |                     | Creencia M4 |                     |
|-------------|---------------------|-------------|---------------------|-------------|---------------------|
| t1          | Plaza Mayor         | t1          | Palacio de Anaya    | t1          | Museo del Automóvil |
| t2          | {00:00-24:00} horas | t2          | {08:00-21:00} horas | t2          | {09:00-15:00} horas |
| t3          | 0 euros             | t3          | 0 euros             | t3          | 4 euros             |
| t4          | 3                   | t4          | 2                   | t4          | 3                   |
| t5          | Centro              | t5          | Histórica           | t5          | Río                 |
| t6          | 1 hora              | t6          | 2 horas             | t6          | 2 horas             |

- **Deseos del Sistema:** requerimientos del turista

El sistema debe posibilitar sugerir planes a los turistas de acuerdo con las restricciones que ellos definan (sus deseos). La naturaleza de estas restricciones está delimitada por los atributos de las creencias ya que como una intención está compuesta por la conjunción de creencias, sus atributos tienden a caracterizar también las intenciones que lo componen.

- **Intenciones del Sistema:** rutas posibles para visitar la ciudad

Para el caso de estudio, las intenciones corresponden a los planes o rutas que el usuario debe seguir, conservando las restricciones que se han definido en cada una de las creencias.

Por ejemplo, si un turista solicita una ruta que cumpla los siguientes requisitos: visitar un número de monumentos mínimo ( $P \geq 3$ ), no gastar más que un valor máximo de ( $C \leq 40$ ), comenzar la visita a una hora definida ( $HI = 10$ ) y emplear en la visita un tiempo máximo ( $TP \leq 10$ ), entonces, dadas unas ciertas creencias, el sistema podría sugerir la siguiente intención caracterizada por unos parámetros globales:

$$i_1 = M1 \wedge D(\text{Histórica, Centro}) \wedge M2 \wedge D(\text{Centro, Histórica}) \wedge M3 \wedge D(\text{Histórica, Río}) \wedge M4.$$

La tabla (3) representa la estructura de la intención  $i_1$  tomando como base los recursos acumulados.

**Tabla 3:** Acumulados Objetivos Recursos Intención

| Horario (hr) | 10-12 | 12-13  | 13-14 | 14-15  | 15-16 | 16-18  | 18-20 | Parámetros globales |    |
|--------------|-------|--------|-------|--------|-------|--------|-------|---------------------|----|
| Intención    | M1    | D(H,C) | M2    | D(C,H) | M3    | D(H,R) | M4    | Número de monum.    | 4  |
| Costo(euros) | 3     | 0      | 0     | 0      | 0     | 0      | 4     | Costo(euros)        | 7  |
| Tiempo (hr)  | 2     | 1      | 1     | 1      | 2     | 1      | 2     | Tiempo visita (TP)  | 10 |
| Calidad      | 1,5   | -      | 3     | -      | 2     | -      | 3     | Hora de inicio (HI) | 10 |

Estos valores permiten identificar si la intención se ajusta con los requisitos o no es adecuada en función de los intereses del usuario.

#### 4 Etapas del motor CBR para el problema de planificación turística

Hasta este momento se han esbozado las componentes de la arquitectura BDI de una manera aislada, pero para que en nuestro sistema tengan una utilidad es necesario interrelacionarlos con el propósito de satisfacer las necesidades del turista. Con el objetivo de cumplir los deseos, este proceso se lleva a cabo a través de la implementación de un CBR, donde los casos corresponden a las intenciones generadas en el pasado. Las etapas que constituyen el ciclo de vida del CBR que se está implementando son las siguientes:

1. **Recuperación:** el sistema selecciona aquellos casos cuyo valor de caracterización se encuentra dentro de los límites de los requerimientos definidos por el turista. En otros CBR la etapa de recuperación se reduce a elegir un único caso; en el sistema propuesto se selecciona un conjunto de casos posibles.
2. **Reutilización:** esta etapa es el núcleo del CBR, se fundamenta y aplica de acuerdo con el modelo definido por [3]. Tiene como datos de entrada el conjunto de casos generados en la etapa de recuperación y el objetivo es determinar de este conjunto de opciones posibles cuál es aquel en que en caso de verse interrumpido en su ejecución, tiene asociado un número mayor de alternativas con las que puede replanificar su ruta. Este cálculo es complejo matemáticamente y el método de desarrollo para su determinación se basa en herramientas matemáticas de cálculo variacional [10] que permiten determinar cuál es el conjunto de planes más adecuados de acuerdo con los objetivos planteados. Para explicar la idea matemática se va a enunciar un conjunto de pasos que deben ser ejecutados de manera secuencial, para llegar a elegir aquel caso que para los requerimientos del usuario corresponde a un plan fácilmente replanificable. Estos se explicarán a continuación:
  - (a) **Proyección de casos en el espacio de requisitos:** para cada caso, se deben identificar las creencias que lo conforman, además se debe representar cada creencia como una n-tupla (de acuerdo con el número de atributos que se definan), y realizar para la secuencia de creencias una secuencia de n-tuplas, de manera que los recursos se vayan acumulando a medida que se avanza en el plan. Por ejemplo, para una intención con los datos de las creencias reflejados en la figura (2), el proceso de proyección estaría modelado así:

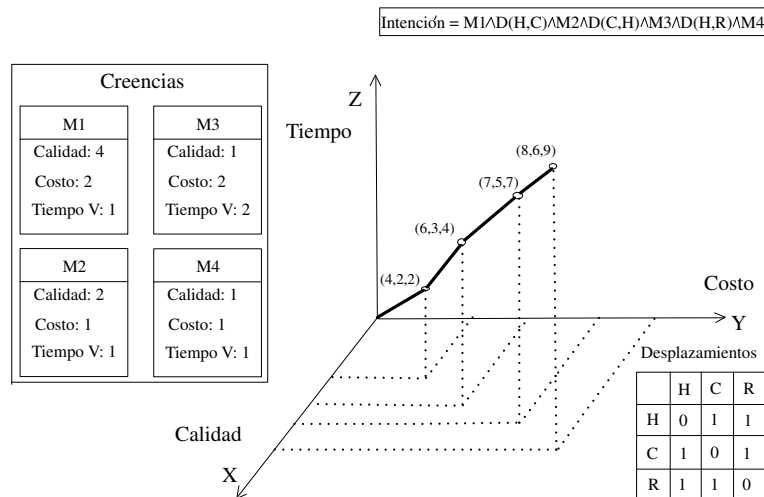
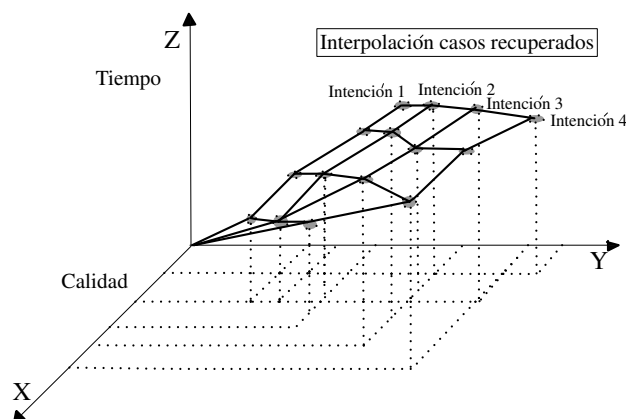


Figura 2: Proyección de las creencias en las intenciones

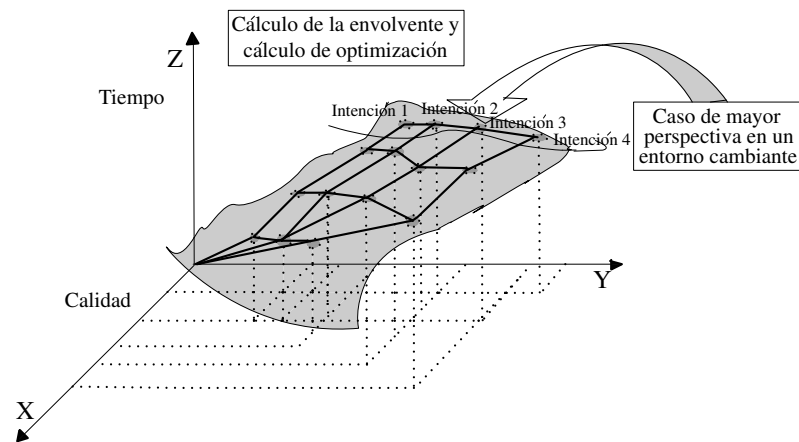
- (b) **Interpolación de los casos recuperados:** el primer paso consiste en crear una malla que contenga todos los puntos que representan las creencias de los casos recuperados. A continuación y a través de estrategias de interpolación se debe obtener (figura 3):



**Figura 3:** Interpolación de las intenciones

- (i) La envolvente de la malla, necesaria para la aplicación de cálculo de optimización explicado en el siguiente paso, y
  - (ii) La secuencia de estados finales que cumplen con los requisitos impuestos por el turista.
- (c) **Aplicación de cálculo de optimización:** a partir de la estructura de la envolvente y de la secuencia de estados finales, se pasa a calcular el conjunto de soluciones óptimas relativas a los casos recuperados. En concreto las técnicas de optimización utilizadas están basadas en cálculo variacional, porque [2] comparando diferentes estrategias, ha concluido que su utilización en entornos dinámicos es el que mejor desempeño ha obtenido.
- (d) **Elección del caso de mayor perspectiva en un entorno cambiante:** del conjunto de soluciones óptimas obtenidas en la etapa anterior, se determina cuál de ellas tiene a su alrededor mayor número de alternativas entre las que elegir, para poder reanudar la planificación en caso de verse interrumpida (figura 4).

La elección de aquella que tiene a su alrededor más opciones para replanificar en el caso de que el plan previsto no funcione se determina a partir de técnicas de Campos de Jacobi. Esta herramienta permite calcular para cada solución posible el campo de opciones alternativas que lleva asociado. La ejecución de estas técnicas permite conocer en tiempo real la mejor de las soluciones en términos de su capacidad de replanificación.



**Figura 4:** Envoltente de las intenciones y elección caso mayor perspectiva

3. **Revisión:** normalmente la etapa de revisión en un CBR consiste en la supervisión por parte de un humano que valora si el caso es adecuado para el problema en estudio. En el sistema esta etapa se realiza de manera automática por el propio mecanismo de replanificación, evaluando con un valor menor aquellos casos que han fallado durante su ejecución.
4. **Almacenado:** esta etapa se especifica en dos mecanismos diferentes. El primero de ellos corresponde a la actualización del peso de la intención de acuerdo con el éxito o porcentaje de fallo. El segundo depende del proceso de replanificación que conlleva a la generación de nuevas intenciones formadas a partir de la intención original y la nueva intención propuesta en el momento de la interrupción.

Hasta aquí se han descrito las claves del modelo teórico que sustentan el desarrollo de la aplicación. En las secciones que siguen se mostrará el proceso de diseño e implementación, además de la metodología de desarrollo empleada.

## 5 Implementación

Para crear un sistema multiagente que proporciona un servicio de planificación aplicado al turismo y pueda afrontar las características de un entorno dinámico, se desarrolló un modelo basado en cálculo variacional que permite seleccionar la ruta con mayor expectativa para la replanificación (SRMER), contemplando que el diseño de la solución deba permitir una comunicación a través de agentes que se encuentren en dispositivos móviles. Para esto fue necesario seguir la metodología GAIA [17] que concibiera el sistema como una conjunción de agentes y luego llegar a definir la estructura, interacción y comunicación de los agentes a través de la plataforma JADE-LEAP. La metodología GAIA proporciona



un conjunto de pasos incrementales donde el aspecto trascendental es la construcción de sistemas basados en agentes como un proceso de diseño organizacional.

### 5.1 Metodología GAIA

Para analizar y diseñar el sistema multiagente que permite la planificación con un sistema de razonamiento basado en casos, se definieron los requerimientos, roles, permisos, responsabilidades, protocolos y actividades, para proceder luego a esquematizar los tipos de agentes, servicios y comunicación que caracterizan a GAIA (figura 5).

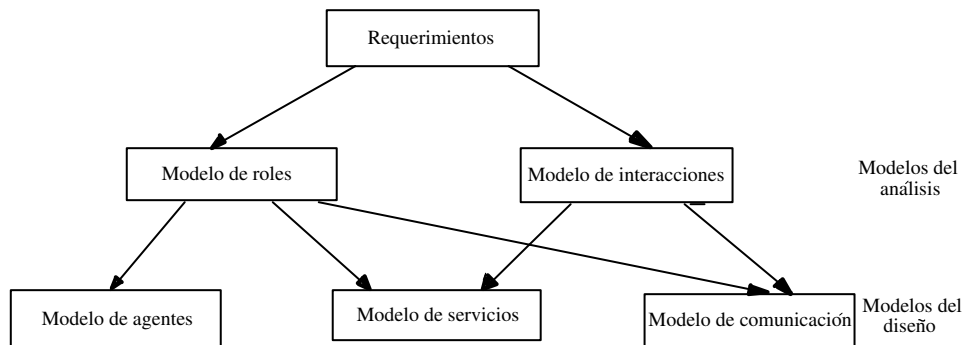


Figura 5: Relaciones existentes entre los modelos de GAIA

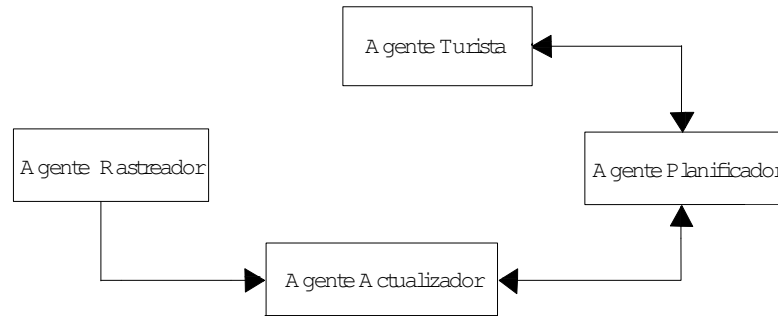
**5.1.1 Fase de Análisis.** Uno de los aspectos más relevantes del análisis en GAIA consiste en la definición de los roles, los cuales pueden irse depurando en varias iteraciones hasta alcanzar el grado de especificación deseado. Se realizaron dos ciclos de iteración, llegando a identificar los roles explicados en la tabla (4) que caracterizan al sistema, detallados de acuerdo con su definición en el ciclo de desarrollo.

Tabla 4: Roles primera y segunda iteración

| Primera iteración        | Segunda iteración   |
|--------------------------|---|
| Rol usuario              | Rol usuario   |
| Rol planificador         | Rol seleccionador ruta con mejor expectativa de replanificación |
|                          | Rol validador monumento   |
|                          | Rol replanificador  |
| Rol gestor información   | Rol enviar listado de monumentos                                |
|                          | Rol actualizar-monumentos                                       |
| Rol usuario-actualizador | Rol usuario-actualizador  |

**5.1.2 Fase de diseño.** Uno de los modelos más sencillos que ayuda a definir la interacción entre los agentes es el Modelo de Comunicación, el cual simplemente define los enlaces de comunicación que existen entre los tipos de agentes. Ellos no definen que mensajes son enviados o cuando los mensajes son enviados, sólo indican los caminos de comunicación que existen, figura (6).

**Modelo de comunicación**



**Figura 6:** Modelo de comunicación del sistema (GAIA)

El sistema multiagente esta formado por:

- **Agente Planificador**, basado en una arquitectura BDI que permite analizar de las diferentes alternativas tomando como base en un entorno dinámico, cuál es la ruta más fácilmente replanificable. Esta arquitectura permite que las creencias que representan a los sitios turísticos que se pueden visitar en una ciudad (en este caso Salamanca), se encuentren parametrizadas de manera que puedan ser fácilmente adaptables a otra ciudad, a través de un algoritmo que genera las intenciones (posibles rutas) que son susceptibles de ser visitadas por el usuario. Estas rutas son los casos que se recuperan para analizar cuál es el de mayor expectativa de éxito de acuerdo con las condiciones definidas por el turista a través del dispositivo móvil.
- **Agente Turista**, se encuentra instalado en un dispositivo móvil y su objetivo es el de capturar las preferencias del turista de acuerdo con el costo, tiempo de visita y hora de inicio. Debe establecer una comunicación con el Agente Planificador quien, según estas restricciones, le enviará la ruta con mayor probabilidad de éxito.
- **Agente Actualizador**, encargado de modificar el estado de las creencias (Monumentos) y las intenciones (rutas), previa solicitud del Agente Rastreador. Como resultado de esta actualización se guarda un registro de los cambios solicitados por los agentes.
- **Agente Rastreador**, es un agente que se encuentra en un dispositivo móvil y que permite enviar una solicitud al Agente Actualizador para conocer el estado de las

creencias y si lo desea cambiar dicho estado, dando a su vez lugar al cambio de las intenciones que contengan esas creencias.

Debido a que el diseño en la metodología GAIA presenta un nivel alto de abstracción se complementó con AUML (Agent-based Unified Modeling Language) [11], [12] que permite refinar los sistemas basados en agentes de acuerdo con el protocolo de interacción de Agentes (AIP); para ello se utilizaron los diagramas de secuencia, de colaboración y de actividad.

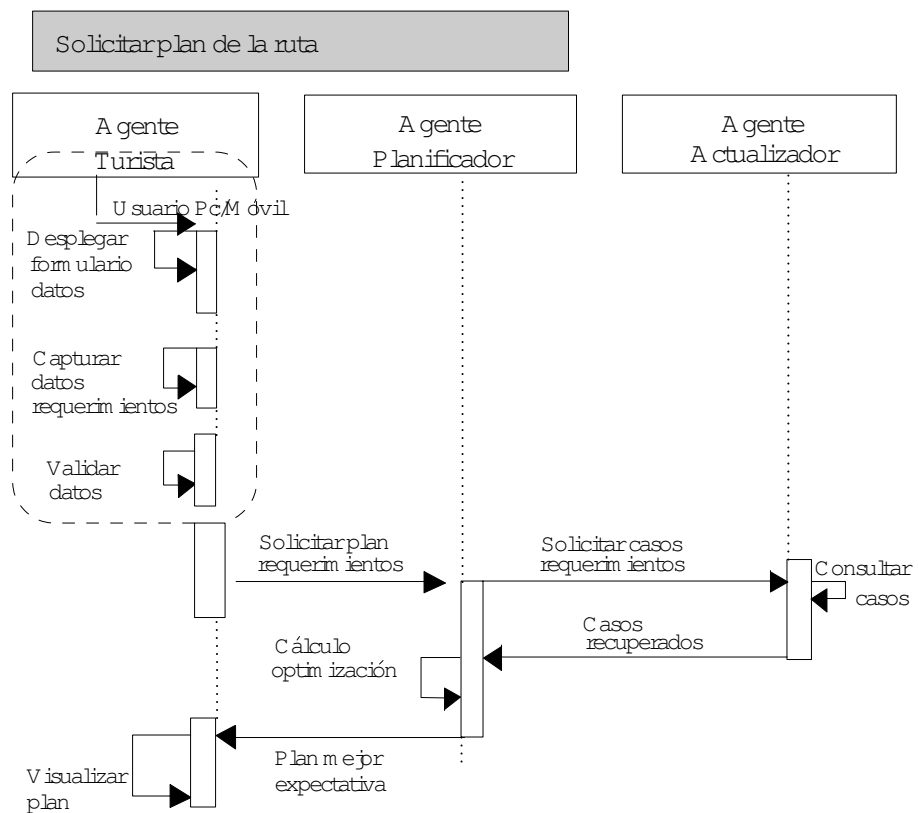
### 5.1.3 Refinamiento del modelo con AUML

En este artículo se representa a través de un diagrama de secuencia los protocolos de comunicación para el proceso de “Solicitar Plan de la Ruta”, y este proceso se complementa con la capa de representación del procesamiento interno de los agentes, en la cual se busca obtener una representación en detalle de lo que ocurre dentro de un agente.

#### Procesamiento interno

La representación interna del procesamiento de un agente, ha sido implementada utilizando los comportamientos en **JADE-LEAP**, la lógica de dichos comportamientos se ve reflejada por los algoritmos que permiten dar soporte a todos los roles definidos mediante GAIA. En la figura (7) se visualiza un algoritmo que calcula la mejor ruta que se le enviará al turista de acuerdo con las peticiones del usuario. Este corresponde sólo a uno de los procesos.

1. Datos de entrada: requerimientos del usuario (Costo, Tiempo, Visita, Calidad, Hora de inicio).
2. Recuperar los casos (Rutas) que cumplan con los requerimientos. Para esta selección se inicia prevaleciendo los casos con mayor número de Monumentos.
3. Proyectar los atributos de cada Monumento de cada Ruta en un espacio de requisitos representados por tres variables  $(x, y, z)$  correspondientes (calidad, costo, tiempo visita). Estos valores son acumulativos dependiendo de cada Ruta.
4. Realizar un cálculo de interpolación espacial para el conjunto de puntos obtenidos en el paso (3) para identificar la envolvente a la malla.
5. Realizar un cálculo de interpolación lineal para el conjunto de puntos finales de cada ruta que permite obtener el total de estados posibles que cumplirían los requisitos del usuario.
6. Calcular de manera teórica y con los resultados de los pasos (4) y (5) la Ruta que cumpliendo los requisitos del usuario estaría mejor preparada para ser replanificada en una interrupción. La ruta teórica es la ruta promedio de los puntos de la interpolación lineal calculada sobre la envolvente de la interpolación espacial [2]. El interés de este cálculo se encuentra al conocer cuál de las rutas ya ejecutadas y que se han recuperado, se acerca a este resultado teórico.



**Figura 7:** Solicitar plan de la ruta (Protocolo comunicaci n)

7. Calcular la Ruta m s cercana al resultado te rico: para cada ruta se establece la distancia eucl dea con respecto a la ruta te rica encontrada en el paso (6).
8. Elegir la Ruta de las recuperadas cuya distancia eucl dea promedio es la menor.
9. El sistema autom ticamente incluye un m todo para darle mayor prioridad a las rutas que en el pasado han tenido mayor  xito, para ello se defini  un porcentaje de aceptaci n (0,10 definido por convenio) debajo del cual, la ruta elegida no s lo debe ser la m s cercana a la te rica sino la de mayor probabilidad de  xito.

Esta nueva era de los agentes ha propiciado que muchos investigadores e instituciones, hayan conformado a FIPA [8], con el prop sito de definir est ndares y protocolos que permitan la promoci n y difusi n de agentes inteligentes, a trav s de la comunicaci n e interoperabilidad entre plataformas.

## 5.2 Agentes con JADE-LEAP

En la actualidad existe una plataforma denominada LEAP (*Lightweight Extensible Agent Platform*), desarrollado por un consorcio de instituciones [9], la cual es definida como la precursora de la segunda generación de plataformas que cumplen con la FIPA, con la novedad que permite a los investigadores migrar agentes y sistemas multiagentes a dispositivos móviles. JADE-LEAP contiene un conjunto de librerías en las cuales se pueden definir agentes y comportamientos, además de implementar posibilidades de comunicación siguiendo el lenguaje ACL especificado por FIPA, en el desarrollo de este proyecto se utilizó esta plataforma como pieza fundamental para la definición y comunicación entre los agentes debido a que al estar definida sobre el lenguaje de programación puede ser fácilmente adaptable a los dispositivos inalámbricos que soporten este lenguaje.

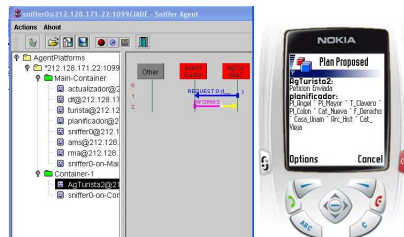
### 5.2.1 Características de JADE-LEAP

Algunas características de LEAP son:

1. Se ejecuta tanto en PC como en dispositivos con recursos limitados como son PDA o en teléfonos basados en Java.
2. Permite incorporarse a redes inalámbricas como TCP sobre GSM y redes LAN inalámbricas basadas en IEEE 802.11.
3. Explora la portabilidad de Java, para instalarse en sistemas operativos que lo soporten.

## 5.3 Interfaces de la aplicación

Para el desarrollo y prueba de la aplicación se utilizó un teléfono celular Nokia 7650 con bluetooth y conexión GPRS para poder acceder al computador Host. En el Host se encuentra instalado JADE-LEAP y la versión de java sdk 1.4.2 (figura 8).



**Figura 8:** Comunicación de los agentes en JADE-LEAP

En la figura (9) se visualiza el midlet que se instala en el dispositivo móvil y que permite ver la ruta con mayor posibilidades de éxito.



Figura 9: Midlet seguimiento de la ruta

## 6 Ejemplos de uso



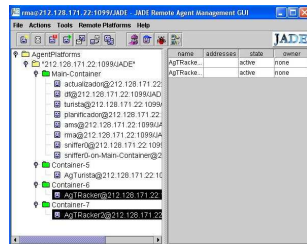
1) Ejecución de la aplicación



2) Registro en el sistema



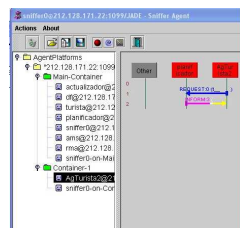
3) Formulario inicial



4) Registro del agente en la plataforma JADE-LEAP



5) Registrar las preferencias del turista. El agente Turista se encarga de capturar estas preferencias y enviar una petición al Agente Actualizador



6) Proceso comunicativo a través de mensajes ACL entre el Agente Turista y el Agente Planificador



7) Opciones generales de planificación y consulta de monumentos



8) Mapa de Salamanca y los monumentos que debe visitar



9) Recursos acumulados durante la ejecución del plan

## 7 Resultados

- El sistema permite solicitudes de planificación desde un PC o un dispositivo inalámbrico (por ejemplo un teléfono móvil).
- El sistema aprovecha información del pasado que le sea de utilidad para definir los planes de los nuevos usuarios.
- El sistema permite al usuario que defina unas condiciones bajo las cuales los planes serían considerados satisfactorios.
- El sistema proporciona al usuario el plan que se ajuste con las condiciones y tiene la característica de ser uno de los planes con mayor probabilidad de éxito en un entorno cambiante.
- El sistema permite al usuario que una vez haya recibido el plan, inicie su ejecución validando cada uno de los elementos que constituyen el plan.
- El sistema permite al usuario la consulta de los recursos que ha utilizado durante la ejecución del plan.
- El sistema permite al usuario solicitar una replanificación de la ruta, enviando planes alternativos tomando como base los monumentos visitados, y los recursos acumulados frente a los requerimientos iniciales definidos.
- El Sistema permite que múltiples usuarios soliciten planes, además de identificar el estado actual de la ruta de cada usuario.
- El Sistema permite que en el dispositivo móvil aparezca la gráfica de la ruta que el turista debe seguir.

## 8 Conclusiones

La planificación basada en cálculo variacional en conjunción con un motor CBR, facilita un modelo de razonamiento que permite encontrar planes óptimos fijadas unas restricciones previas. La implementación de este modelo siguiendo las etapas de la metodología

GAIA nos permite dotar al agente principal de autonomía y de aprendizaje, condiciones esenciales de la naturaleza de los agentes. GAIA aunque tiene muchas ventajas, carece de un nivel concreto en el diseño de la solución, y para solventar esta carencia se utilizó AUML que proporciona un nivel de detalle mayor. Esta combinación constituyó una guía acertada para explicar el problema en estudio.

El complemento que las técnicas de “Campos de Jacobi” suponen al cálculo variacional permite determinar cual de los planes posibles es aquel que tiene mayor capacidad de replanificación en el caso de que se vea interrumpido en su desarrollo.

Los usuarios que acceden a esta tecnología inalámbrica desean utilizarla para mejorar sus actividades cotidianas. La versión de JADE-LEAP que se utiliza para dispositivos móviles tiene un proceso de optimización del código que disminuye el tamaño de las librerías que se instalan en los equipos. El desarrollo del sistema multiagente con JADE y JAVA ha permitido que la aplicación utilice librerías para el cálculo de funciones matemáticas de interpolación desarrollados por grupos externos al “Grupo de Sistemas Informáticos Inteligentes” (GSII) de la Universidad de Salamanca (Librerías GNU jspline).

## Referencias

- [1] A. Aamodt y E. Plaza *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*, AICOM, Vol. 7, March 1994, 39-59.
- [2] Manuel Bedia, *Fundamentos Cognitivos para el diseño de arquitecturas de agentes planificadores en contextos dinámicos de acción*, Phd Tesis, Universidad de Salamanca, Departamento de Informática y Automática, Salamanca España, 2004.
- [3] Manuel Bedia y J. M. Corchado, *A Planning Strategy Based on Variational Calculus for Deliberative Agents*, Computing and Information Systems Journal, Vol. 10, No. 1, ISBN: 1352-9404 (2002), 2-14.
- [4] R. Bergmann y W. Wilke *Learning Abstract Planning Cases*, En N. Lavrac and S. Wrobel (Eds.), Machine Learning: ECML-95, 8th European Conference on Machine Learning, Heraclion, Greece, April 1995. Number 912 in Lecture Notes in Artificial Intelligence, Berlin, Springer (1995), 55-76.
- [5] R. Bergmann y W. Wilke *Towards a New Formal Model of Transformational Adaptation in Case-Based Reasoning*, European Conference in Artificial Intelligence (ECAI'98), John Wiley and Sons (1998), 224-236.
- [6] M. E. Bratman *Intentions, Plans and Practical Reason*, Harvard University Press, Cambridge, M.A, 1987.
- [7] J. M. Corchado y R. Laza *Construction of BDI Agents from CBR systems*, First German Workshop on Experience Management, Berlin, ISBN 3-88579-340-7, ISN 1617-5468, Vol. 1, March 2002, 47-60.



- [8] Foundation for Intelligent Physical Agents, <http://www.fipa.org>.
- [9] Lightweight Extensible Agent Platform, <http://leap.crm-paris.com/>.
- [10] P. Morse y H. Feshbach *Methods of Theoretical Physics*, McGraw-Hill, New York, 1953.
- [11] J. Odell, V. D. Parunak, B. Bauer. *Representing Agent Interactions Protocols*, En UML First Internacional Workshop on Agent Oriented Software Engineering (AOSE 2000), 2000.
- [12] V. D. Odell Parunak y B. Bauer, *Extending UML for Agents*, Proceedings of the Agent-Oriented Information System Workshop at the 17th Nacional Conference on Artificial Intelligence, 2000.
- [13] A. S. Rao y M. P. Georgeff *BDI Agents: From Theory to Practice*, First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, USA (1995), 312-319.
- [14] Rao y Georgeff. *Modeling rational agents within a BDI-architecture*. En J. Allen, R. Fikes, y E. Sandewall, editors, Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann Publishers, San Mateo, CA, 1991.
- [15] Watson I. y Marir F. *Case-Based Reasoning: A Review*. Cambridge University Press. The knowledge Engineering Review, **9**(3) (1994), 249-293.
- [16] M. Wooldridge y N. R. Jennings *Agent Theories, Architectures, and Languages: A Survey*. Procs. ECAI-94 Workshop on Agent Theories, Architectures, and Languages (1994), 317-361.
- [17] M. Wooldrige, N. R. Jennings y D. Kinny. *The Gaia Methodology for Agent-Oriented Analysis and Design, Autonomous Agents and Multi-Agent Systems*, Vol. 3, Kluwer Academic Publishers, The Netherlands (2000), 285-312.