Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro ISYS, Laboratorio de Inteligencia Artificial

Desarrollo de un agente para la captura de datos agronómicos desde dispositivos móviles utilizando la metodología JADE

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por el bachiller
José Rafael Subero Carrillo
para optar por al Título de
Licenciado en Computación

Tutores
Prof. Iván Flores
Prof. Haydemar Nuñez

Caracas, 2010

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designados por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado, presentado por el Bachiller José R. Subero C. C.I.: 18.304.158, con el título "Desarrollo de un agente para la captura de datos agronómicos desde dispositivos móviles utilizando la metodología JADE", a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue dicho trabajo por cada uno de los Miembros del Jurado, se fijó el día 6 de Mayo de 2010, a las 11:30 AM, para que el autor lo defienda en forma pública, en el aula de postgrado de la Escuela de Computación, mediante la exposición oral de su contenido, luego de la cual respondió satisfactoriamente a las preguntas que le fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió APROBARLO.

En fe de lo cual se levanta la presente Acta, en la Ciudad Universitaria de Caracas a los 6 días del mes de Mayo del año dos mil diez, dejándose también constancia de que actuó como Coordinador del Jurado el profesor Iván Flores.

Jurado Principal

_____	_____
Profesora Ana Morales	Profesor Jesús Viloria
_____	_____
Profesor Iván Flores (Tutor)	Profesora Haydemar Nuñez (Tutora)

Resumen

El estudio y clasificación de suelos resulta ser una actividad imprescindible para el manejo sustentable de este recurso en múltiples actividades humanas. Un problema al que se enfrentan los investigadores en esta área está relacionado con la manera cómo se organiza y gestiona la información que se genera desde las distintas fuentes utilizadas, tales como: imágenes satelitales, levantamiento de suelos, resultados de distintas pruebas de laboratorio, entre otras. Con el fin de apoyar a los profesionales en esta actividad, se ha propuesto un sistema basado en agentes inteligentes para el almacenamiento, gestión y clasificación de datos. En este sistema se contempla el diseño y construcción de un agente que permita la captura de datos desde un dispositivo móvil, que es el motivo del presente trabajo. El agente beneficia el proceso de recolección de datos, haciendo la captura y el registro de los datos de campo desde la misma aplicación; a diferencia de los procesos manuales, los cuales requieren realizar gastos adicionales de tiempo y recursos, para la validación y el almacenamiento de los datos. Además, el agente proporciona asistencia al especialista durante el proceso de captura.

Agradecimientos

Primeramente quisiera agradecerle a Dios por haberme formado como la persona que soy. Por todas las ayudas materiales y espirituales que a lo largo de mi carrera me ha brindado para seguir adelante con mis decisiones y sueños. En diversas oportunidades he sido testigo de su apoyo incondicional y de su amor reconfortante, no sólo a través de múltiples personas que han dejado su huella en mi vida, sino por su Santa Palabra, que me ha servido de sostén y sabiduría en momentos bien particulares de mi vida, en la que necesitado sentir su cercanía y presencia.

También agradecer a mis padres y familiares, que me han permitido dedicarme con confianza a mis estudios por su apoyo económico y amor incondicional. Gracias, por haber creído siempre en mí.

A todos mis amigos con los que es compartido todos estos años y han enriquecido de forma especial cada faceta de mi vida.

A todos mis profesores, preparadores, formadores que a lo largo de mi vida, me han sabido con dedicación y esmero enseñarme amar el camino del conocimiento y las virtudes.

A la Universidad Central de Venezuela, la casa que vence las sombras. Por todas las personas que en ella trabajan y enseñan, por abrirme las puertas a un conocimiento universal, no sólo a un nivel académico sino para la vida. Ruego a Dios que con el transcurso de los años siga siendo una universidad de clase mundial, orgullo para todos los venezolanos; distinguida no sólo por sus talentos arquitectónicos, sino por el talento humano que en ella se forman.

Quisiera agradecer también de manera especial a mis tutores Iván Flores y Haydemar Nuñez, que durante el desarrollo de esta investigación, han sabido orientarme y apoyarme hasta la consumación exitosa del mismo.

No quisiera tampoco dejar de mencionar el apoyo de la profesora Esmeralda Ramos, que ha podido seguir de cerca e involucrarse en el avance de este trabajo en múltiples oportunidades, contribuyendo con sus observaciones y apreciaciones en la calidad del mismo.

Faltan palabras para poder expresar realmente el aprecio y agradecimiento a la cantidad de personas que han contribuido de una forma u otra, a la distinción que hoy estoy compartiendo y celebrando junto a mis seres queridos. En sencillas palabras, gracias a todos.

Dedicatoria

Dedico este trabajo a Venezuela, patria de infinitas riquezas y conocimientos. Hogar y sueño de muchos hombres y mujeres que con mucha entrega y sacrificio han puesto su mirada expectante hacia el futuro, anhelando con ansía disfrutar los ricos frutos de la libertad y la virtud.

Me sumo al sueño de Bolívar, un hombre visionario, quién contempla con profunda admiración el futuro próspero que aguarda a nuestra patria, lleno de esplendor y majestad. Citando sus célebres palabras:

“Volando por entre las próximas edades, mi imaginación se fija en los siglos futuros, y observando desde allá, con admiración y pasmo, la prosperidad, el esplendor, la vida que ha recibido esta vasta región, me siento arrebatado y me parece que ya la veo en el corazón del universo, extendiéndose sobre sus dilatadas costas, entre esos océanos que la naturaleza había separado, y que nuestra Patria reúne con prolongados y anchurosos canales. Ya la veo servir de lazo, de centro, de emporio a la familia humana; ya la veo enviando a todos los recintos de la tierra los tesoros que abrigan sus montañas de plata y de oro; ya la veo distribuyendo por sus divinas plantas la salud y la vida a los hombres dolientes del antiguo universo; ya la veo comunicando sus preciosos secretos a los sabios que ignoran cuán superior es la suma de las luces a la suma de las riquezas que le ha prodigado la naturaleza. Ya la veo sentada sobre el trono de la libertad, empuñando el cetro de la justicia, coronada por la gloria, mostrando al mundo antiguo la majestad del mundo moderno.”

Discurso de Angostura, 15 de Febrero de 1819.

Índice General

Índice de figuras

Índice de tablas

Introducción

El estudio de suelos es una necesidad para la realización exitosa de diversas actividades humanas como: ganadería, agricultura, construcción, conservación, entre otras. Es por esto, que se realizan estudios para la clasificación de los suelos, con el fin de estimar su potencial productivo para distintas alternativas de aprovechamiento.

Uno de los problemas que se presenta en estos estudios, se debe a las distintas fuentes de información que son utilizadas en la recopilación de datos, por ejemplo: imágenes satelitales, levantamiento de suelos, resultados de pruebas experimentales, entre otros. Este proceso genera una gran cantidad de información, que dificulta la forma de almacenarla y consultarla eficientemente por parte de los investigadores.

Algunas investigaciones han resuelto estas dificultades parcialmente, logrando centralizar información de ciertas áreas geográficas específicas, que sirven de apoyo en la toma de decisiones sobre el uso de los suelos.

Para esta finalidad el Laboratorio de Inteligencia Artificial (LIA) de la Facultad de Ciencias y el Instituto de Edafología de la Facultad de Agronomía de la UCV, están interesados en desarrollar un sistema basado en agentes inteligentes para la gestión y clasificación de los suelos.

Este proyecto plantea la construcción de un agente de captura de datos de campo desde un dispositivo móvil. El agente permitirá a los especialistas edafólogos realizar la captura y el registro de datos obtenidos de los campos de levantamiento de suelos.

En este Trabajo Especial de Grado se explica el desarrollo del agente de captura de datos, utilizando la metodología Jade y la plataforma de aplicaciones móviles JavaME.

Este documento se encuentra dividido en las siguientes partes:

- **Capítulo 1 - Marco teórico**: presenta las teorías de la inteligencia artificial que dan soporte al diseño y construcción del agente móvil, así como las distintas tecnologías de comunicaciones móviles implicadas en su desarrollo.
- **Capítulo 2 - Marco aplicativo**: presenta cada una de las actividades, artefactos y resultados alcanzados en la construcción del agente móvil, así como las características que describen su uso por parte de los especialistas.

Por último, se exponen las conclusiones y recomendaciones para trabajos a desarrollar en el futuro, los cuales contribuirían a extender y mejorar la investigación realizada.

Capítulo 1: Marco teórico

En este capítulo se presentan los fundamentos teóricos relacionados con el paradigma basado en agentes inteligentes y las tecnologías de las comunicaciones móviles que fueron utilizados para la construcción del agente móvil.

1.1) Paradigma basado en agentes

Un agente inteligente es una entidad de software que, basándose en su propio conocimiento, realiza un conjunto de operaciones para satisfacer las necesidades de un usuario, o programa, o bien por iniciativa propia (Hípola y Vargas, 1999).

En su forma básica, un agente cuenta con sensores y actuadores que le permiten interactuar con su entorno. Las percepciones del agente son capturadas a través de sensores, y esta información es usada por el agente para describir el conocimiento de su entorno. Si el agente necesita realizar alguna acción específica sobre su entorno en beneficio propio, la realiza mediante el componente de actuadores (Balcázar, 2009).

En la Figura 1 se observa las funciones y las interacciones básicas de un agente.

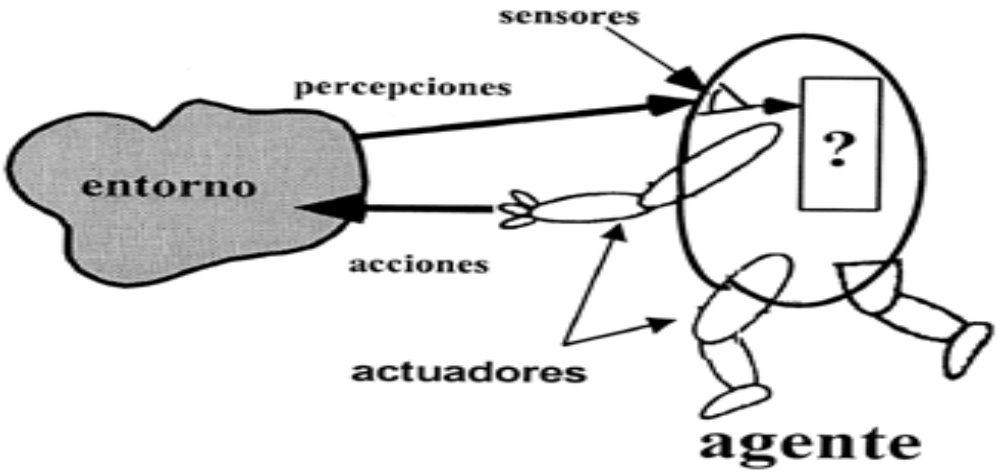


Figura 1: Visión esquemática de un agente (Balcázar, 2009)

Este arquetipo básico de un agente permite introducir algunas de las arquitecturas propuestas, para el diseño de un agente.

El primer enfoque propuesto es la arquitectura reactiva, que consiste en definir comportamientos o acciones de un agente de acuerdo en respuesta a las percepciones de su entorno. La estructura del agente debe ser la siguiente: sensores, actuadores y una representación básica del conocimiento para establecer la correspondencia entre las percepciones y las acciones (Brooks, 1991).

Un segundo enfoque propuesto es conocido como arquitectura deliberativa, que consiste en darle al agente la noción de un estado mental. Esto se consigue mediante la correspondencia de patrones simbólicos representativos del estado del agente, con relación a los estímulos percibidos de su entorno. Con esta definición se propuso la arquitectura BDI (*Besides Desires Intentions*). Esta arquitectura propuesta por Shoham (1993), consiste en representar el estado mental del agente mediante tres componentes: creencias, deseos e intenciones. Las creencias están relacionadas con el conocimiento a priori que tiene el agente del mundo que le rodea. Los deseos están relacionados con las metas que debe cumplir el agente. Las intenciones son las acciones que el agente puede realizar para lograr el cumplimiento de sus objetivos. Los tres componentes interactúan entre sí en la medida que se producen cambios en el estado mental del agente (Shoham, 1993).

Un último enfoque busca integrar los dos enfoques anteriores, y se le conoce con el nombre de arquitectura híbrida. Consiste en diseñar un agente compuesto por dos subsistemas: uno deliberativo, que contiene un módulo simbólico del entorno, desarrolla planes y toma las decisiones; y uno reactivo, que reacciona a los estímulos percibidos del ambiente. A menudo, el componente reactivo tiene cierto grado de precedencia sobre el deliberativo, para que el agente responda de forma inmediata sobre los eventos importantes producidos en el ambiente (Georgeff y lansky, 1987).

Para escoger el enfoque apropiado para el diseño del agente, es necesario conocer bien las características relevantes que exhibe el agente. Diversos autores definen un conjunto de características necesarias para considerar si un agente es realmente inteligente, algunas de las cuales son: (Castro y otros, 2005).

- **Autonomía**: capacidad del agente de actuar sin depender de nadie más.
- **Independencia:** libertad que tiene el agente para decidir sus propias acciones.
- **Robustez**: grado de tolerancia que tiene el agente sobre las percepciones del entorno.
- **Reactividad**: capacidad del agente para reaccionar a los cambios del entorno.
- **Proactividad**: capacidad del agente para tomar iniciativa y actuar de acuerdo a sus intereses.
- **Racionalidad**: capacidad del agente para plantearse objetivos en función de una meta esperada.
- **Movilidad**: capacidad del agente para trasladarse dentro de su entorno.
- **Adaptabilidad**: capacidad del agente para adecuar su estado interno y su comportamiento de acuerdo a las condiciones de su entorno.

- **Sociabilidad**: capacidad del agente para relacionarse con otros agentes, colaborando y cooperando entre sí para alcanzar una meta.
- **Veracidad**: Es una cualidad inherente a la sociabilidad. La información que proporcione el agente debe ser verdadera, pues es la base de una sana cooperación entre los agentes.
- **Benevolencia**: Es otra cualidad inherente a la sociabilidad. Ningún agente debe ser totalmente racional, ya que sus acciones pueden perjudicar el éxito de otros agentes.
- **Autodidacta**: Es la capacidad que tiene el agente de aprender de sus propias acciones. Entiéndase, la aceptación obtenida por el agente por una acción ejecutada en un momento dado, repercute en las acciones futuras que realiza el agente.

Una de estas características que ha tenido mayor relevancia en la construcción de sistemas de agentes es la sociabilidad. Los agentes pueden colaborar y cooperar en conjunto de forma tal que sus relaciones pueden describirse como una sociedad de agentes.

Para efectos de esta investigación, en el próximo apartado se explicará una forma de organización de agentes, conocido con el nombre de sistemas multiagentes.

1.1.1) Sistemas multiagentes

Un sistema multiagente (SMA) es un conjunto de agentes inteligentes que coordinan sus acciones, teniendo en cuenta sus habilidades y recursos, para darle solución a problemas individuales o grupales (Acosta, 2005).

Las características esenciales que distinguen a un SMA son las siguientes:

- **Heterogeneidad:** grado de similitud de los agentes.
- **Autoridad:** Es el mando o forma de coordinación entre los agentes. Puede ser centralizado (cuando un agente planifica las tareas de los demás) o distribuido (cuando todos los agentes conocen sus tareas de antemano).
- **Cooperación:** Se refiere al trabajo en conjunto de los agentes para alcanzar metas compartidas.
- **Comunicación:** Se refiere al lenguaje y protocolos que definen la comunicación entre agentes.

En un SMA que sigue un esquema de autoridad distribuida, de forma tal que cada agente conoce de antemano las tareas que realiza, y responde a las solicitudes de servicio de otros agentes, este sistema pudiera adaptarse a una arquitectura orientada a servicios. Este enfoque trata de considerar a las organizaciones como un SMA, cada departamento de la organización representa

una entidad o agente que provee servicios al resto de los departamentos de la organización; y del mismo modo, estos servicios podrían ser compartidos entre otras organizaciones. El flujo de información en estos sistemas se realizaría bajo un esquema de solicitud y adquisición de servicios (Herrera y otros, 2008).

Otra característica resaltante de estos sistemas, es el referente a la heterogeneidad. Cuando los agentes de un SMA se ejecutan en distintos ambientes, es necesario manejar el problema de compatibilidad que podría haber entre ellos, debido a las diferentes tecnologías de estos sistemas. Para este fin se utilizan las plataformas de sistemas multiagentes.

1.1.2) Plataformas de sistemas multiagentes

Las plataformas multiagentes permiten ejecutar los SMAs de forma transparente, delegando los aspectos técnicos y de configuración a la plataforma y no al programador. Son una capa de abstracción que controla el contexto de ejecución de los agentes, independientemente del sistema donde se ejecuten (Hayzelden y Bigham, 1999).

En el mercado se pueden encontrar un gran número de plataformas de SMAs, especializadas para distintos propósitos.

Para efectos de esta trabajo se hará mención de la plataforma JADE (*Java Development Framework*, 2010).

La plataforma JADE, es una plataforma de agentes con filosofía Open Source y se distribuye bajo licencia LGPL. Fue desarrollada por TILAB (*Telecom Italia Lab*) y su principal objetivo es desarrollar aplicaciones multiagentes distribuidas, basadas en arquitecturas de comunicaciones punto a punto. Su arquitectura es descentralizada y los puntos son autónomos, con capacidad de tomar la iniciativa en una comunicación, así como de prestar servicios a otros puntos.

Entre las ventajas técnicas que proporciona esta plataforma se mencionan las siguientes (García y otros, 2005):

- Ha sido desarrollado totalmente en Java y cumple con la especificación de FIPA, por lo que interactúa con cualquier plataforma de agentes que cumpla con este estándar.
- La comunicación entre agentes se realiza por medio del intercambio de mensajes asíncronos, permitiendo la independencia temporal entre los agentes que se están comunicando. La estructura de los mensajes se basa en el lenguaje ACL (*Agents Communication Languages*) definido por FIPA, además se cuenta con protocolos de interacción típicos para la ejecución de tareas comunes en diversas aplicaciones.

- Seguridad, utilizando mecanismos de autenticación y verificación de los

derechos asignados a los agentes. Esto permite a las aplicaciones verificar la identidad de los agentes y prevenir acciones no permitidas.

- Movilidad de código y de estado de ejecución permitiendo la distribución de funciones y de carga computacional.
- Escalabilidad, en ambientes con limitación de recursos, Jade permite que se ejecuten tareas paralelas en un mismo hilo de ejecución.

Existe también una versión de Jade reducida, que ofrece la posibilidad de ejecutar agentes en dispositivos móviles, conocido con el nombre de Jade Leap.

Se han realizado múltiples investigaciones con el uso de esta plataforma, una de ellas estuvo dirigida a plantear una propuesta de desarrollo de SMAs basada en JADE; con el fin de facilitar la construcción de estos sistemas. En el siguiente apartado se explica en detalle esta propuesta, que fue utilizada para el desarrollo del agente móvil.

1.1.3) Metodología de desarrollo basada en Jade

Una metodología de desarrollo es una guía de las actividades y prácticas recomendables al realizar un proceso de desarrollo de software. En el caso de los SMAs, también existen propuestas que ayudan a conceptualizar, diseñar y construir estos sistemas.

Una de estas propuestas es la metodología JADE. Ha sido presentada por Magid Nikrakde la Universidad de Murdoch al oeste de Australia (2006), como parte de su proyecto grado de PhD. La metodología toma en cuenta las actividades claves dentro del ciclo de desarrollo de SMAs, haciendo inclusión de las características propias del desarrollo en JADE. (Caire y otros, 2006).

El ciclo de vida de desarrollo propone esta metodología es semejante a la mayoría de los procesos de desarrollo, considerando las siguientes fases: planeación, análisis, diseño, implementación y pruebas. En la fase de planeación se estima la viabilidad del proyecto. En la fase de análisis se determinan los aspectos conceptuales y funcionales del dominio de la aplicación. En la fase de diseño se integran los modelos obtenidos de la fase de análisis, y se incluyen algunos criterios para la implementación de los agentes. La fase de implementación y pruebas la metodología hace la recomendación de usar las herramientas que proporciona JADE y Java, para el desarrollo y pruebas del sistema.

A continuación se explica en detalle las actividades que se consideran en cada una de las fases, los artefactos que se generan en cada una de ellas son explicados más adelante para facilitar la lectura.

- **Análisis**: En esta fase se busca modelar el SMA, de acuerdo a sus funcionalidades, recursos y agentes que interactúan en el dominio de la aplicación.

- **Construcción de modelo de casos de uso**: Se captura las potenciales funcionalidades del SMA, así como los escenarios de interacción que hay entre los distintos actores y el sistema.
- **Identificación de los tipos de agentes**: Se identifican los agentes de acuerdo al contexto y los recursos presentes en el sistema (normalmente, evidentes en los casos de uso). El artefacto que se genera en la actividad es el diagrama de agentes.
- **Identificación de las responsabilidades**: Se identifican las responsabilidades de cada agente de acuerdo a las interacciones con sus recursos. La captura de las responsabilidades se realiza mediante una tabla de responsabilidades por cada agente.
- **Identificación de interacciones**: Consiste en actualizar el diagrama de agentes y las tablas de responsabilidades con las interacciones que realizan los agentes.
- **Refinamiento de los agentes**: Se identifican nuevos agentes, interacciones y responsabilidades, según la exposición de servicios de los agentes (Entiéndase, interacciones adicionales entre los agentes). Durante esta actividad pueden añadirse nuevos agentes que faciliten la coordinación entre los agentes. Estos agentes pueden ser: agentes de soporte, si proporciona información utilizada por otros agentes; descubrimiento, si enlazan o descubren a otros agentes; administración y monitoreo, si guardan la traza del estado de otros agentes. En esta actividad también se agrega el *"Yellow Page"* (agente que expone los servicios en JADE) al diagrama de agentes.
- **Información de despliegue de agentes**: Se modela la información relacionada con la plataforma de los agentes. El artefacto generado en esta actividad, es un diagrama en el que se incluyen los agentes y su relación con los dispositivos físicos donde serán ejecutados.

- **Diseño**: En esta fase se busca integrar todos los artefactos obtenidos de la fase de análisis y refinarlos de acuerdo a las consideraciones de diseño.
  - **División/Mezcla/Renombre de agentes**: Se toman las decisiones sobre los agentes obtenidos de la fase de análisis, de acuerdo a los siguientes criterios: (a) evitar la duplicación de datos, (b) evitar la duplicación de código para el acceso a recursos, (c) evitar la división de agentes, al menos que se tenga una buena razón, (d) cada agente debería ser situado en una máquina, (e) evitar tener agentes grandes y complejos, (f) usar un agente traductor en caso de presentarse cobertura de código (se refiere a códigos protegidos de escritura). Además, debe mantenerse un balance entre muchos agentes simples y

pocos agentes complejos.

- **Especificación de las interacciones**: Se debe especificar una tabla de interacciones por cada tipo de agente, tomando en cuenta las tablas de responsabilidades. Las interacciones corresponden aquellas responsabilidades que necesitan la colaboración de otros agentes para su realización.
- **Definir protocolos de interacción**: Se agregan a la tabla de interacciones aquellas interacciones que hagan uso de alguno de los protocolos definidos en JADE.
- **Uso de plantillas de mensajes**: Se agregan a la tabla de interacciones, el nombre del formato de mensajes usado por cada interacción. Se añade mediante una nueva columna en la tabla.
- **Descripción del registro de servicios**: Se formaliza el nombramiento y registro de servicios en el agente "*Yellow Page*" (que es el agente que maneja la exposición de servicios en JADE).
- **Interacciones sobre agentes de recursos**: Se definen las interacciones sobre los agentes que hacen uso de algún tipo de recurso. Los recursos son considerados tanto pasivos (no generan eventos) como activos (generan eventos). Esta actividad se apoya en un diagrama de estado de la interacción sobre el estado del recurso.
- **Interacciones sobre agentes de usuario**: En esta actividad se definen aspectos relacionado con la interfaz de usuario (IU). Las IU suelen ser vistas también como un recurso activo, por lo que esta actividad pueden ser incluida en la actividad anterior dependiendo de su complejidad. Las IU pueden también clasificarse de acuerdo al origen de los datos y las tecnologías utilizadas para su construcción, de forma general pueden clasificarse de acuerdo a estos dos tipos: locales (El usuario visualiza los datos locales por medio del agente) o web GUI (cuando la interfaz de usuario se construye mediante las tecnologías de Internet). La metodología define un modelo de trabajo en cada caso, y los nuevos elementos deben añadirse a la arquitectura del sistema. Estos modelos de trabajo han sido omitidos porque no se utilizaron en el desarrollo del agente móvil.
- **Comportamientos internos del agente**: Se identifican los comportamientos de acuerdo a las responsabilidades e interacciones de cada agente, registrándolas mediante entradas de comportamientos en una tabla para cada uno de los agentes. En esta actividad es necesario que se identifique el ciclo de vida y el flujo de trabajo de los comportamientos, buscando correspondencia con los tipos de comportamientos definidos en JADE.

- **Definición de ontologías**: Se modela el conocimiento utilizado por los agentes. Este conocimiento puede ser representado mediante ontologías (formulación exhaustiva y rigurosa de un esquema conceptual dentro de uno o varios dominios dados, con la finalidad de facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades), las cuales se formalizan haciendo uso de un lenguaje ontológico. La ontología debe ser lo suficientemente simple y sencilla que le permita a los agentes trabajar y comunicarse.
- **Selección del contenido del lenguaje**: Consiste en seleccionar el lenguaje de los mensajes entre los agentes.

- **Implementación y pruebas**: Se propone el uso de herramientas de generación de código incluidas en JADE. Para las pruebas se recomienda el uso de técnicas de verificación que proporciona Java, por ejemplo: el uso de test unitarios, etc.

La metodología propone también algunos artefactos o instrumentos para utilizar en cada una de las actividades.

A continuación se explicará cada uno de los artefactos, haciendo uso del siguiente ejemplo (Caire y otros, 2006):

Ejemplo

Una operadora móvil ofrece un servicio de invitaciones a sus suscriptores:

- Invitar algunos amigos a ver una película.
- Recolectar las preferencias del que invita y de los invitados.
- Sugerir la mejor opción que se adapte a las preferencias del grupo.

Se asume que la operadora tiene contrato con todos los cines de varias ciudades. Existe un servicio proveedor que indica la información actualizada de la ciudad correspondiente, en base a la localización móvil de las personas. Este servicio actualiza la información con cada cine al evaluar las invitaciones.

- **Modelo de casos de uso**: es el primer artefacto de la fase de análisis, permite la captura de todas las funcionalidades del sistema, así como identificar los actores que intervienen.

  En la Figura 2, se observa el diagrama de casos de uso para el ejemplo mencionado.

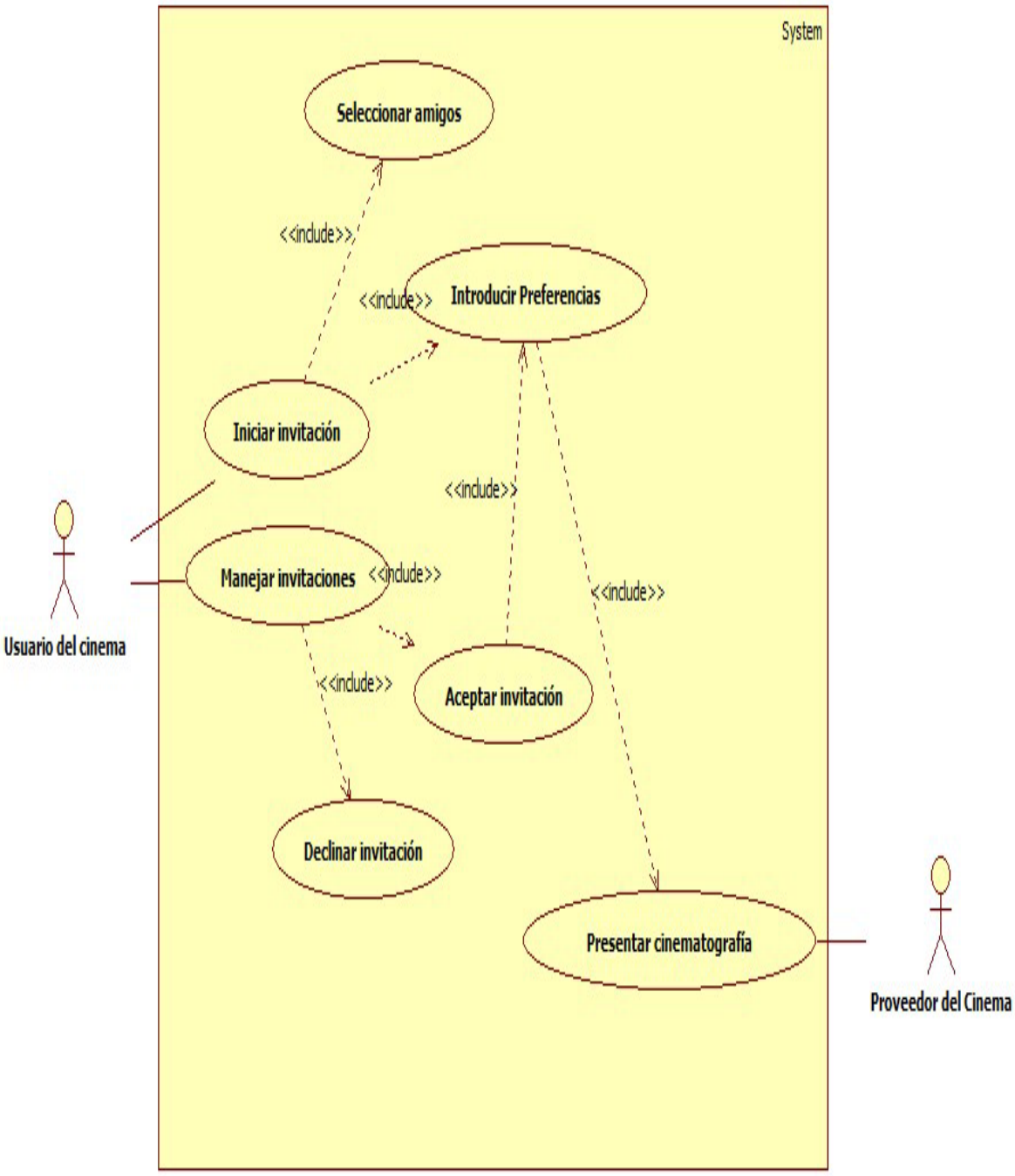


Figura 2: Ejemplo de diagrama de casos de uso (Caire y otros, 2006)

- **Diagrama de agentes**: permite describir las interacciones entre los agentes, los recursos y los actores del sistema.

  En la Figura 3, se observa el diagrama de agentes para el ejemplo mencionado.

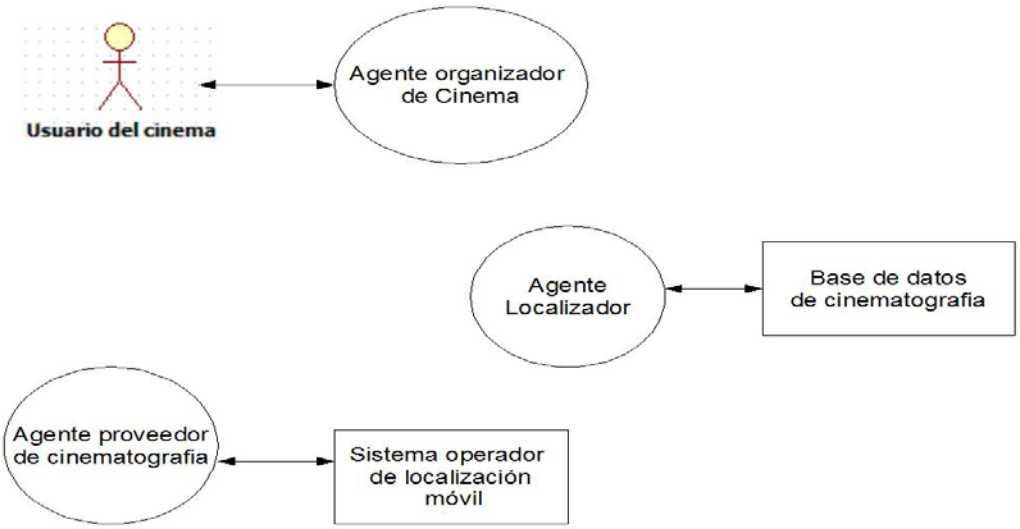


Figura 3: Ejemplo de diagrama de agentes (Caire y otros, 2006)

- **Tabla de responsabilidades**: permite identificar las propiedades estáticas y dinámicas (interacciones) de los agentes.

En la Tabla 1 se muestra el registro correspondiente al agente organizador de cinema de la tabla de responsabilidades del ejemplo mencionado:

Tabla 1: Ejemplo de tabla de responsabilidades

Tipo de agente	Responsabilidades
Agente organizador de cinema	• Servir las peticiones para iniciar las invitaciones desde el usuario del cinema. • Permitir al usuario del cinema seleccionar los amigos para invitar. • Permitir al usuario introducir las preferencias sobre películas y cines. • Presentar cinematografía. • Responder a las invitaciones desde otros agentes proveedores.

- **Diagrama de despliegue de agentes** (opcional): permite relacionar los agentes con la plataforma de agentes y los dispositivos físicos en el que se ejecutarán los agentes.

  En la Figura 4, se observa el diagrama de despliegue para el ejemplo mencionado.

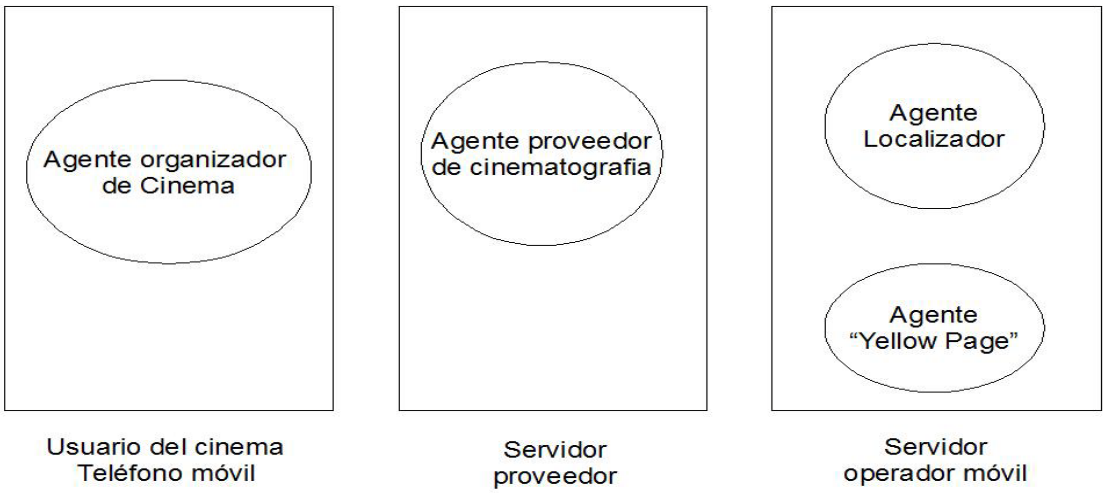


Figura 4: Diagrama de despliegue de agentes (Caire y otros, 2006)

- **Tabla de interacciones**: permite describir formalmente todas las interacciones entre los agentes y sus protocolos relacionados.

En la Tabla 2 se muestra un ejemplo de un registro en la tabla de interacciones correspondiente al agente organizador de cinema.

Tabla 2: Ejemplo de especificación de interacciones

Interacción	Resp.	PI	Rol	Con	Cuando	Plantilla
Invitar a otros agentes organizadores de cinemas	1	Contranet	I	Agente organizador de cinema	El usuario inicia una invitación	Conv-id
Responder a una invitación	5	Contranet	R	Agente organizador de cinema	Una invitación es recibida	Perf = CFP

La primera columna indica el nombre de la interacción; la segunda, indica la responsabilidad que produce la interacción (corresponde al identificador de la responsabilidad en la tabla de responsabilidad); la tercera, el protocolo de interacción utilizado; la cuarta, el rol que juega el agente considerado (I – iniciador, R – el que responde); la quinta, el nombre del agente con quien interactúa; la sexta, la situación que ocasiona la interacción; y la séptima columna, la plantilla del mensaje en la interacción (válido en el caso de Jade).

Finalmente, esta propuesta recomienda evitar las actividades que no sean relevantes dentro del proyecto, y modelar justamente lo que se espera desarrollar.

1.2) Comunicaciones móviles

Los apartados siguientes explican cada una de las tecnologías de comunicaciones implicadas en el desarrollo del agente móvil.

Para entablar una comunicación, es necesario que se sigan ciertas reglas comprendidas por los entes comunicantes, estas definen la forma en que se realizan las comunicaciones y se conocen con el nombre de protocolos de comunicación. Un protocolo de comunicación desde el punto de vista técnico puede definirse como: capa de intermedia que existe entre la capa de aplicaciones y la capa de transmisión de los dispositivos móviles, que permite abstraer a las aplicaciones de los aspectos técnicos de la transmisión. Esto facilita la creación de aplicaciones portables, favoreciendo una mejor calidad de servicios (Tanenbaum, 1997).

A continuación se hace mención de algunos de los protocolos más importantes utilizados en las comunicaciones móviles.

- **WAP (*Wireless Application Protocol*)**: Es una especificación que permite a los dispositivos inalámbricos ofrecer servicios de Internet. Se trata de un servidor dedicado de conexiones WAP que resuelve las peticiones desde los clientes (dispositivos móviles), busca el contenido en Internet, y emite una respuesta de regreso a la aplicación, normalmente interpretada por un navegador presentando la información correspondiente. La versión más reciente es WAP 2.0, el cual trabaja con XHTML-MP (XHTML Mobile Profile) (W@p Forum, 2002)
- **GPRS (*General Packet Radio Service*)**: Es una extensión del Sistema Global para Comunicaciones Móviles (*Global System for Mobile Communications* o GSM) para la transmisión de datos no conmutada (o por paquetes). Proporciona servicios tales como servicios WAP, mensajería de texto (SMS), mensajería multimedia (MMS), Internet, entre otros. Este protocolo facilitó la migración progresiva a las redes de tercera generación (Cysco System, 2000).

- **UMTS (*Universal Mobile Telecommunications System*)**: Es una de las tecnologías usadas para dispositivos móviles de tercera generación (3G, también llamado W-CDMA), sucesora de GSM (the UMTS Forum, 2003).

  Entre sus características se pueden mencionar:

  - Capacidades multimedia, una velocidad de acceso a Internet elevada, transmisión de audio y video en tiempo real con una calidad equiparable a la de las redes fijas.
  - Posibilidad de conectar los dispositivos directamente a un computador y compartirle su conexión a Internet.

- **EV-DO (*Evolution-Data Optimized o Evolution-Data Only*)**: Es un estándar del grupo 3GPP2 (*3rd Generation Partnership Project*) que pertenece a la familia CDMA2000 y ha sido adoptado por muchos proveedores a nivel mundial, sobre todo en el continente americano, particularmente por aquellos que ya contaban con redes IS-95/cdmaOne (en competencia con las redes GSM) (Motorola , 2006).
- **HSPA (*High-Speed Packet Access*)**: Es el resultado de la combinación de tecnologías posteriores y complementarias a la de tercera generación de telefonía móvil (3G), tales como 3.5G o HSDPA y 3.5G Plus, 3.75G o HSUPA. Su propósito principal es ayudar a la adaptación de las redes basadas en UMTS (W-CDMA) a las capacidades de transmisión esperadas de las redes 4G (Disna y Griparis, 2006).

Los fabricantes de dispositivos junto con los proveedores de redes móviles desarrollan tecnologías que permiten con el uso de estos protocolos integrar la mayor cantidad de servicios desde un mismo terminal móvil. Los dispositivos móviles fueron concebidos originalmente para permitir la comunicación de voz de forma inalámbrica. Pero con el paso de los años estos dispositivos han incrementado sus capacidades de procesamiento, incorporando el desarrollo de nuevos sistemas y aplicaciones.

Los teléfonos inteligentes (*smartphones*) son teléfonos celulares capaces de ejecutar aplicaciones que funcionan normalmente sobre un computador personal. Esta clase de dispositivos poseen capacidades limitadas en comparación a sus homólogos, pero cuentan con un sistema operativo que le permite administrar recursos y ejecutar aplicaciones. Este avance le permite a los desarrolladores de aplicaciones móviles abstraer la complejidad de interactuar directamente con los recursos del dispositivo, que normalmente requería una autorización del fabricante y una implementación de bajo nivel poco portable, por un desarrollo enfocado en el uso de APIs estandarizadas (*Application programming interface*). (Colaboradores de Wikipedia, 2009).

Otra estrategia que contribuye a la estandarización de APIs son las plataformas de aplicaciones. Estas plataformas permiten manejar el entorno de ejecución de las aplicaciones, independientemente del sistema operativo nativo. Las plataformas contribuyen al desarrollo de aplicaciones portables en un mayor número de dispositivos.

Algunas de las plataformas más populares en el mercado son las siguientes: Windows Mobile, Blackberry, iPhone, Symbian, Android, Brew, Flash Lite, Personal Java, J2ME y JavaFX Mobile. Para efectos del desarrollo del agente móvil, se procederá a explicar en detalle los recursos que proporciona la tecnología J2ME o JavaME (Java 2 MicroEdition) para el desarrollo de aplicaciones móviles.

1.2.1) Plataforma de aplicaciones móviles de Java (J2ME)

J2ME es la plataforma de Java para el ambiente de aplicaciones que funcionan en dispositivos móviles y embebidos. Fue desarrollado mediante el *Java Community Process* (JCP) bajo la especificación JSR 68 (*Java specification resource*). Además, se le añadieron otro número de especificaciones que contribuye a estandarización de la plataforma (Knudsen, 2007).

Una de estas especificaciones más importante es la JSR 37, que define MIDP (*Mobile Information Device Profile*) que es la base de las aplicaciones móviles de Java. Fue desarrollado por un consorcio de compañías que forman parte del JCP.

Las razones que han hecho de MIDP tan popular entre los fabricantes de dispositivos y operadores de red *Wireless*, se debe a la máquina virtual de Java (JVM). Las aplicaciones nativas ejecutan directamente las instrucciones sobre el procesador del dispositivo, por lo que comprometen la estabilidad del sistema en caso de algún error; mientras que las aplicaciones de Java solo comprometen a la máquina virtual, dejando el resto del dispositivo en funcionamiento. Otra de las ventajas, es que los fabricantes de dispositivos pueden ofrecer a los usuarios la carga de aplicaciones *Third-party* (instalación de software de terceros) de forma segura y autorizada (Miembros J2ME Grasia, 2009).

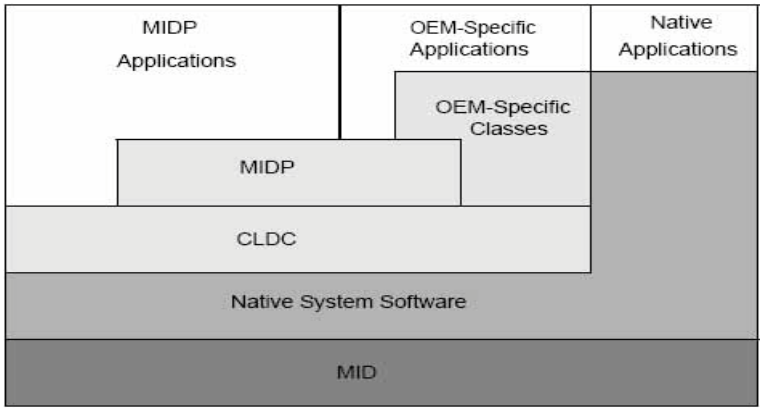


Figura 5: Arquitectura MIDP (Miembros J2ME Grasia, 2009)

En la Figura 5, se observa la arquitectura de la plataforma de aplicaciones basadas en la tecnología MIDP. La máquina virtual de Java interactúa con el sistema operativo nativo del dispositivo.

JavaME organizó su máquina virtual en dos tipos de configuraciones: CLDC (*Connected Limited Device ConFiguration*), dirigida a teléfonos inteligentes con capacidades limitadas de procesamiento; y CDC (*Connected Device ConFiguration*), dirigida a dispositivos móviles de mayor procesamiento, como por ejemplo un PDA (Sun Microsystems, 2009a).

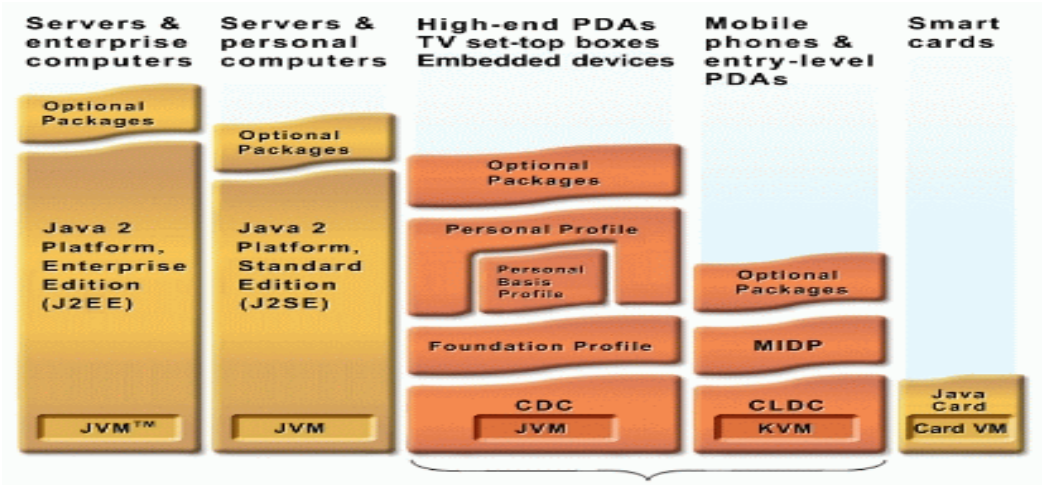


Figura 6: Configuraciones y perfiles JavaME (JavaME Technology, 2009)

En la Figura 6, se observan las configuraciones y perfiles de JavaME, los cuales fueron organizados en base a las capacidades y limitaciones de los dispositivos. Cada una de estas configuraciones utiliza su máquina virtual correspondiente.

La arquitectura que describe la plataforma móvil de Java, forma parte de un abanico de especificaciones y servicios, conocido con el nombre de *Mobile Service Architecture* (MSA). MSA define dos capas: la primera está diseñada para dispositivos con capacidades limitadas que no son capaces de soportar todas las características de MSA; la segunda, cuenta con la funcionalidad completa y especializada para dispositivos con capacidades especiales como GPS, servicios web, Bluetooth, y otras APIs que ofrecen servicios como: encriptación, manejador de contenidos MIME, transacciones de pago, etc (Sun Microsystems, 2009b).

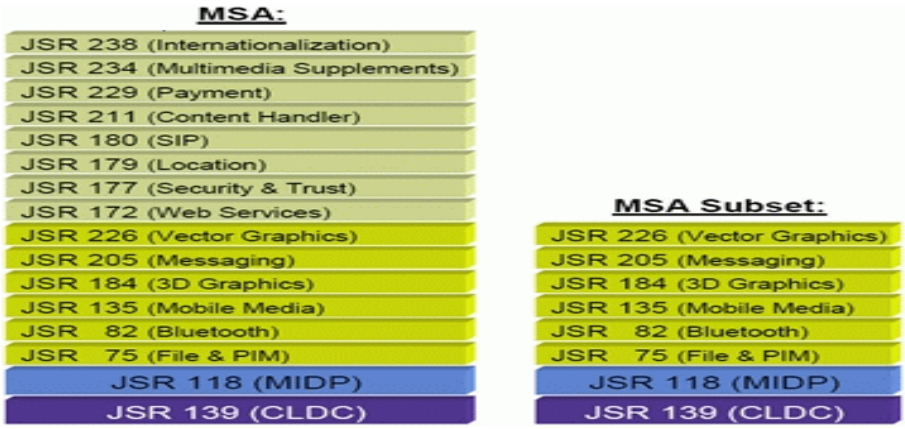


Figura 7: Especificación MSA (Sun Microsystems, 2009c)

En la Figura 7, se observa la arquitectura MSA de Sun Microsystems, quién espera completar la especificación de MSA 1.0 antes del lanzamiento de MIDP 3.0 en los próximos años. (Knudsen, 2007).

Las aplicaciones MIDP están diseñadas especialmente para funcionar en ambientes de dispositivos móviles. Estas aplicaciones son conocidas con el nombre de Midlet. Un Midlet se define a partir de una clase de Java que hereda de la clase Midlet, definida en el paquete javax.microedition.midlet. La clase Midlet define los siguientes métodos que deben ser sobreescritos: startApp (que es llamado al iniciar la aplicación), pauseApp (que es llamado en caso de que se produzca un evento de importancia en el sistema, por ejemplo: una llamada telefónica) y destroyApp (que es llamado cuando se cierra la aplicación para liberar los recursos de la aplicación).

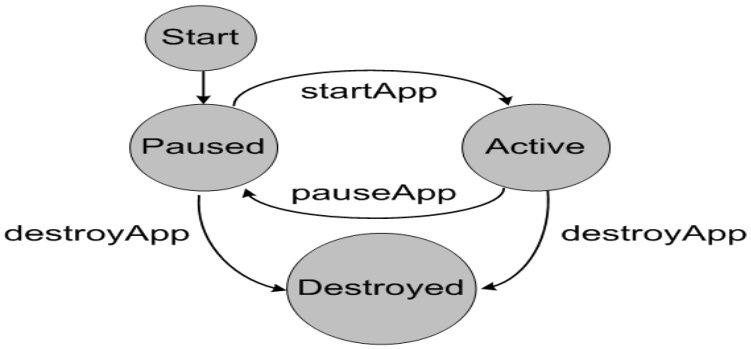


Figura 8: Ciclo de vida JavaME (Sun Microsystem Inc, 2009d)

En la Figura 8, se observa el modelo del ciclo de vida de una aplicación MIDP, la transición entre un estado y otro es controlada por el AMS (*Aplication Management Software*), pero es responsabilidad del programador especificar el comportamiento de la aplicación al cambiar de estado.

El ciclo de vida de los Midlet es controlado por el AMS que maneja todas los Midlet suites (paquete de aplicación que incluye uno o más Midlet) instalados en el dispositivo. Cuando el usuario inicia una aplicación MIDP, el AMS crea una instancia del Midlet y llama a su método startApp. Cuando la aplicación se cierra, el AMS llama al método destroyApp liberando los recursos de la aplicación. En caso de necesitarlo, los Midlet pueden cambiar de estado explícitamente por medio de múltiples hilos de ejecución, haciendo uso de los siguientes métodos: notifyPaused(), notifyDestroyed(), resumeRequest().

Para crear un proyecto de Midlet es necesario realizar los siguientes pasos:

1. **Compilar**: consiste en generar el Java code asociado con la suite del Midlet.
2. **Pre verificar**: consiste en verificar si cada clase cargada en la máquina virtual de Java está correctamente formada.
3. **Empaquetar**: consiste en guardar cada una de las clases pre verificadas adentro de un paquete .JAR de Java. Además, se incluyen el manifiesto de archivo JAR (.mf) y un archivo JAD, utilizados para especificar algunos atributos de seguridad.

JavaME distribuye sus propias herramientas para la construcción de aplicaciones MIDP (Sun Microsystems, 2009c):

- **Sun Java Wireless Toolkit for CLDC**: conocido formalmente como Java 2 Platform Micro Edition (J2ME) Wireless Toolkit, es una caja de herramientas para el desarrollo de aplicaciones wireless que estén basadas en la plataforma JavaME para CLDC y MIDP. Incluye los siguientes elementos: ambientes de emulación, optimización de rendimiento, características de personalización, documentación y ejemplos que los desarrolladores necesitarán para la creación

de aplicaciones.

- **Sun Java Toolkit for CDC**: es un SDK que ayuda a los desarrolladores de aplicaciones a comenzar rápidamente el desarrollo de aplicaciones CDC y PBP (Personal Basic Profile). Las herramientas CDC que están incluidas son: un emulador, una barra de herramientas para la compilación, empaquetamiento, documentación, y ejemplos para ayudar a los desarrolladores a comenzar.
- **Sun Java ME Software Development Kit (SDK)**: es la siguiente generación de los anteriores toolkits que ya se mencionaron. Es la herramienta más reciente para el desarrollo de aplicaciones móviles, combinando un completo soporte para crear y hacer pruebas de aplicaciones JavaME. Incluye un emulador con habilidades tales como emulación OTA y soporte GPS, sensores móviles, manejador de perfiles, entre otros.

Los IDEs que pueden utilizar para desarrollar aplicaciones móviles de Java son los siguientes:

- **Netbeans Mobility Pack**: es un IDE Open Source para el lenguaje de programación en Java. Las ventajas que ofrece este IDE para el desarrollo de aplicaciones, es el uso editor visual llamado One Jazzy, con el que se pueden crear las interfaces de las aplicaciones móviles y el flujo de interacciones entre estas. Esto acelera enormemente la creación de las GUIs (Graphic user interface). (Netbeans.org, 2009).
- **EclipseME**: Se puede usar EclipseME para crear Midlet Suites. Entre las características que tiene este plugin se mencionan las siguientes: uso de Wireless toolkit, creación de J2ME Midlet suite, creación de Midlet, editor para el JAD, pre verificación automática de archivos, soporte con emuladores J2ME, empaquetamiento y despliegue por medio del mecanismo OTA (*Over to Air*) y el uso de un ofuscator (optimizador de código) llamado ProGuard. (Eclipseme.org, 2009).

No todos los emuladores de dispositivos móviles se encuentran incluidos con el SDK de JavaME, los desarrolladores pueden descargarlos directamente de las páginas de los fabricantes e integrarlos al SDK de JavaME, generalmente, sin costo adicional ni adquisión de licencias.

Capítulo 2: Marco aplicativo

En este capítulo se presentan los antecedentes que motivaron esta investigación y los resultados obtenidos al culminar este trabajo, que consistió en el desarrollo del agente móvil para la captura de datos de campo.

2.1) Planteamiento del problema

Existe una gran necesidad de obtener información para la caracterización de los suelos, con el fin de valorar sus beneficios y conocer sus limitaciones. En este sentido, muchas investigaciones están dirigidas a la determinación de la capacidad de uso de una zona determinada, que permita una planificación adecuada de las posibles actividades agropecuarias (agricultura, ganadería, forestación, etc.) que pueden realizarse sobre el terreno (Viloria, Viloria y Nuñez, 2008).

Para realizar esta clasificación es necesario recabar la información característica, obtenida de los campos de levantamiento de suelos. Luego, haciendo uso de los métodos de clasificación, es posible determinar de forma genérica las características de los suelos. Una vez clasificados los suelos, esta información es utilizada para realizar predicciones más precisas y útiles sobre los usos específicos de los suelos, como por ejemplo, la disponibilidad en calidad, cantidad y distribución de este recurso.

Uno de los problemas que se presentan al realizar estos estudios, es el relacionado con la manera de cómo se organiza la información relativa a los suelos, dada las distintas fuentes de información que son recopiladas: imágenes satelitales, datos de levantamiento de suelos, resultados de pruebas, etc. La naturaleza de estos datos, dificulta el almacenamiento y consulta eficiente de la información por parte de los investigadores.

Para esta finalidad el Laboratorio de Inteligencia Artificial (LIA) de la Facultad de Ciencias y el Instituto de Edafología de la Facultad de Agronomía, ambas instituciones de la UCV; proponen desarrollar un sistema basado en agentes inteligentes para la gestión y clasificación de los suelos (Angulo y Vásquez, 2009).

En la Figura 9, puede verse la arquitectura propuesta para este sistema basado en agentes, y se describen cada uno de sus componentes.

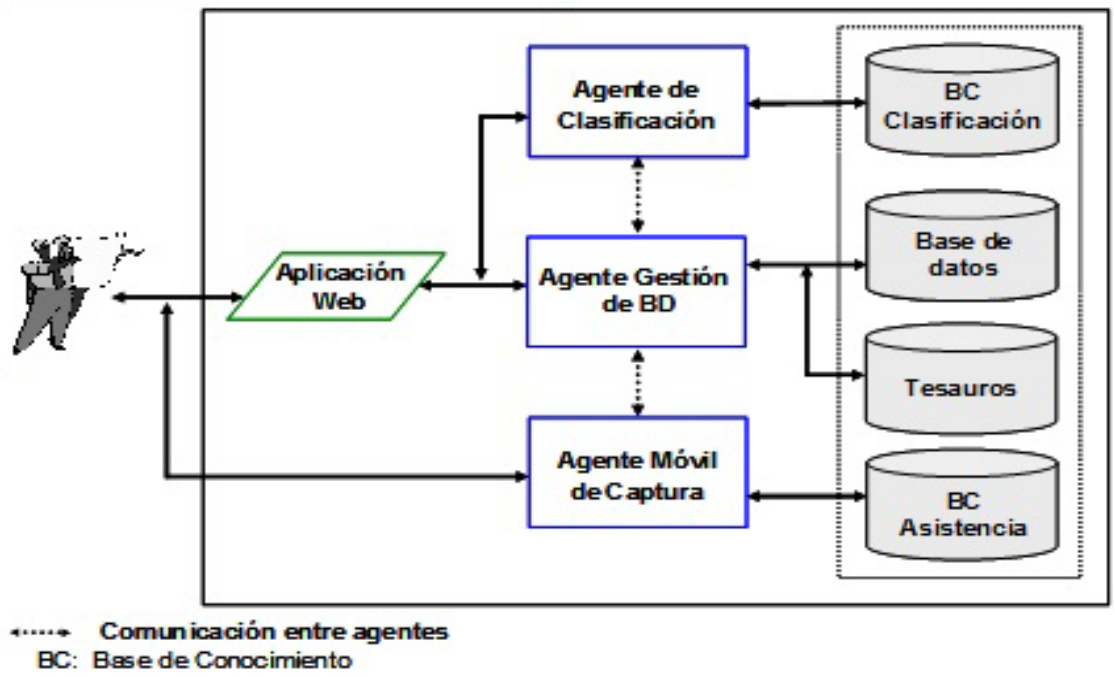


Figura 9: Arquitectura de sistema basado en agentes inteligentes (Angulo y Vásquez, 2009).

- **Agente de gestión de base de datos**: Este agente reactivo tiene como objetivo gestionar la base de datos, a partir de la información que le sea suministrada por el usuario a través de la interfaz, o por el agente de captura de datos a través de dispositivos móviles. Adicionalmente, el agente manipula un tesauro de los términos más representativos del dominio de suelos, el cual utiliza para ofrecer información complementaria al usuario que así lo necesite.
- **Agente móvil de captura de datos**: Este agente reactivo contará con una base de conocimiento que le permita asistir al usuario en la captura de datos cuando se realice el levantamiento de suelo en una zona determinada. Para facilitar su labor, se ejecutará en un dispositivo móvil.
- **Agente de clasificación**: A través de diferentes modelos de clasificación, los cuales serán construidos utilizando técnicas de minería de datos, este agente deliberativo será capaz de realizar una clasificación de la zona de estudio, según las categorías que definan los expertos. Actualmente, se cuenta con una base de conocimiento que puede realizar una clasificación de los suelos según su capacidad de uso.

El componente que se refiere al agente de gestión de base de datos ha sido desarrollado por los estudiantes Tomás Ángulo y Jonathan Vásquez, y se encuentra actualmente en funcionamiento en LIA. Este sistema permite la captura y almacenamiento de datos de campo desde una aplicación web, conocido con el nombre de Siscla (Angulo y Vásquez, 2009).

Una de las recomendaciones que surgió de este trabajo, fue el desarrollo de un agente móvil de captura de datos, de forma que pueda integrarse junto a este sistema para el almacenamiento de los datos de campo.

La construcción del agente móvil plantea la siguiente interrogante: ¿Cómo hacer uso de las tecnologías de comunicaciones y aplicaciones móviles, para el desarrollo de este agente?

Por lo expuesto anteriormente, se propone:

Propuesta de solución: Desarrollar un agente que permita la captura de datos por parte de los especialistas edafólogos desde un dispositivo móvil. Este agente proporcionará una guía en el proceso de captura, asistiendo al especialista y verificando la validez de los datos capturados. Finalmente, este agente debe comunicarse con el sistema Siscla, a través del agente de gestión de base de datos para el almacenamiento de los datos de campo.

Para esta solución se plantearon los siguientes objetivos:

2.2) Objetivos

General

Desarrollar un agente para la captura de datos de agronomía desde un dispositivo móvil, que pueda integrarse conjuntamente con el agente de gestión de base de datos.

Específicos

1. Diseñar el agente móvil utilizando la metodología JADE.
2. Modelar el conocimiento del agente móvil.
3. Construir la interfaz de usuario del agente móvil.
4. Desarrollar el agente móvil haciendo uso de la tecnología JavaME.

En los apartados siguientes se menciona el proceso de diseño y construcción del agente, así como el desarrollo de la aplicación.

2.3) Diseño y construcción del agente móvil

El agente móvil fue modelado usando la metodología JADE (mencionada en el capítulo I). Esta metodología se utilizó con la finalidad de realizar comparaciones futuras con otras metodologías, por parte del grupo de trabajo de LIA.

Se realizaron las actividades siguiendo el orden de desarrollo que propone la metodología y utilizando los artefactos que recomienda, aunque se descartaron aquellas actividades que tuvieron poca relevancia en el desarrollo.

A continuación se procederá a explicar cómo fue el proceso de desarrollo seguido para el diseño y construcción del agente móvil.

2.3.1) Fase de planeación

Esta fase consistió en la estimación de la viabilidad del proyecto y las características del sistema. Para ello se escogió realizar los siguientes artefactos:

- **Cronograma de actividades del proyecto**: Consistió en realizar una evaluación personal de la complejidad y tiempo de cada una de las actividades a realizarse para el desarrollo del sistema. Esta información se muestra en las Tablas 3 y 4 respectivamente.
- **Levantamiento de requerimientos**: Consistió en la captura temprana de los requerimientos del sistema. La captura de requerimientos que se muestra a continuación, fue el resultado de una serie de reuniones con los tutores, y conversaciones con el profesor Jesús Viloria (especialista en el área de edafología), sobre los requerimientos de la aplicación.

La captura de requerimientos fue organizado con la siguiente información:

Requerimientos funcionales

- Hacer la captura de datos de campo desde un dispositivo móvil.
- Autenticar la identidad del especialista que suministra los datos.
- Guiar al usuario en el proceso de captura, mediante pasos (estilo wizard).
- Asistir al usuario en el proceso de captura, aclarando dudas de terminología, haciendo recomendaciones, autocompletando valores de registro.
- Verificar la integridad y veracidad de los datos capturados.
- Permitir al usuario administrar cada una de las capturas realizadas desde el dispositivo.
- Permitir el registro de datos capturados en el sistema Siscla (en comunicación con el agente de gestión de base de datos).

Requerimientos no funcionales

- La interfaz usuario debe ser usable para dispositivos móviles.
- El tamaño de la aplicación debe ser optimizado para facilitar su distribución (mediante el uso de un ofuscator).
- La aplicación debe ser robusta en cuanto a la conectividad (ser de

utilidad cuando el dispositivo tiene conexión, y cuando no).

- El modelo de conocimiento de los agentes debe ser simple y fácil de implementar.

Tabla 3: Cronograma de actividades

Etapa	Actividad	Artefacto	Complejidad
Planeación	Estimación de viabilidad del proyecto	Cronograma de actividades	Baja
		Levantamiento de requerimientos	
Análisis	Realizar casos de uso	Casos de uso del sistema	Baja
	Identificación de tipos de agentes	Diagrama de agentes	Baja
	Identificación de responsabilidades	Tabla de responsabilidades por agente	Media
	Identificación de relaciones	Actualización de tablas de responsabilidades	Baja
	Refinamiento de los agentes	Actualización del diagrama de agentes	Baja
	Información de despliegue de agentes	Diagrama de despliegue de agentes	Baja
Diseño	Refinamiento de diseño de agentes	Ninguno (Actualización de la tabla de responsabilidades y el diagrama de agentes)	Media
	Especificación de interacciones de agentes	Tabla de interacción por agente	Baja
	Interacciones de los agentes con recursos	Actualización de tabla de interacciones	Media
	Interacciones de los agentes con usuarios	Actualización de tabla de interacciones y modelo de interfaz	Media
	Comportamiento de agentes	Tabla de comportamientos por agente	Baja
	Definición de ontología	Modelo de conocimiento	Media
Implementación	Desarrollo aplicación móvil basada en agentes	Aplicación móvil Agentes implementados	Alta
Pruebas	-Verificación de requerimientos -Pruebas en dispositivo	Prototipos de prueba	Media

Tabla 4: Tabla de estimación de proyecto

Etapa	Horas dedicación	Tiempo estimado	Recomendación
Análisis	180 horas	3 semanas - 1 mes	Tiempo ajustado
Diseño			
Implementación	480 horas	2 meses	Tiempo ajustado
Testing	240 horas	1 mes	-
Documentación	180 horas	3 semanas - 1 mes	-

La estimación de esta tabla, fue realizada en base a 10 horas diarias y 6 días a la semana de trabajo.

La duración de esta fase fue de una semana aproximadamente.

2.3.2) Fase de análisis

La fase de análisis consistió en la conceptualización del agente móvil, y se realizaron los siguientes artefactos: modelo de casos de uso, diagrama de agentes, diagrama de despliegue y tabla de responsabilidades.

El modelo de casos de uso es el primer artefacto que propone realizar formalmente la metodología JADE, en el cual se modela las funcionalidades del sistema.

Las funcionalidades fueron organizadas en los siguientes tres casos de uso, los cuales corresponden al nivel 0 de la aplicación: ingresar datos, registrar datos, consultar el estado de registro de datos.

Ingresar datos consiste en el proceso de captura de los datos que corresponde a un perfil de suelo; este proceso de captura tiene una particularidad, y es que debe proporcionar una asistencia al usuario, si el usuario lo necesita. Además, si el usuario no ha completado el proceso de captura, debe proporcionarle la flexibilidad al usuario, de continuar el proceso de captura en otro momento.

Registrar datos consiste en el almacenamiento de los datos capturados, que puede ser de dos tipos: almacenar localmente, que consiste en guardar la captura en el dispositivo móvil; almacenar externamente, que consiste en guardar la captura en el sistema Siscla. Cada vez que un almacenamiento externo finaliza correctamente, toda la información asociada con dicha captura de datos es borrada; esto se debe a que el almacenamiento en el dispositivo es concebido como memoria temporal que brinda apoyo en el proceso de captura y registro de datos.

Finalmente, se añadió una funcionalidad al usuario que consiste en sincronizar aquellos datos de campo que no se pudieron registrar en el Sistema Siscla (ejemplo: ausencia de señal, error de transmisión, olvido de autenticación, etc) y han sido guardados temporalmente en el dispositivo móvil. Esta sincronización no es más que el registro sucesivo de los datos de campo.

En la Figura 10 se muestra el diagrama de casos de uso de la aplicación. La especificación formal de los casos de uso se presenta en la Tabla 5.

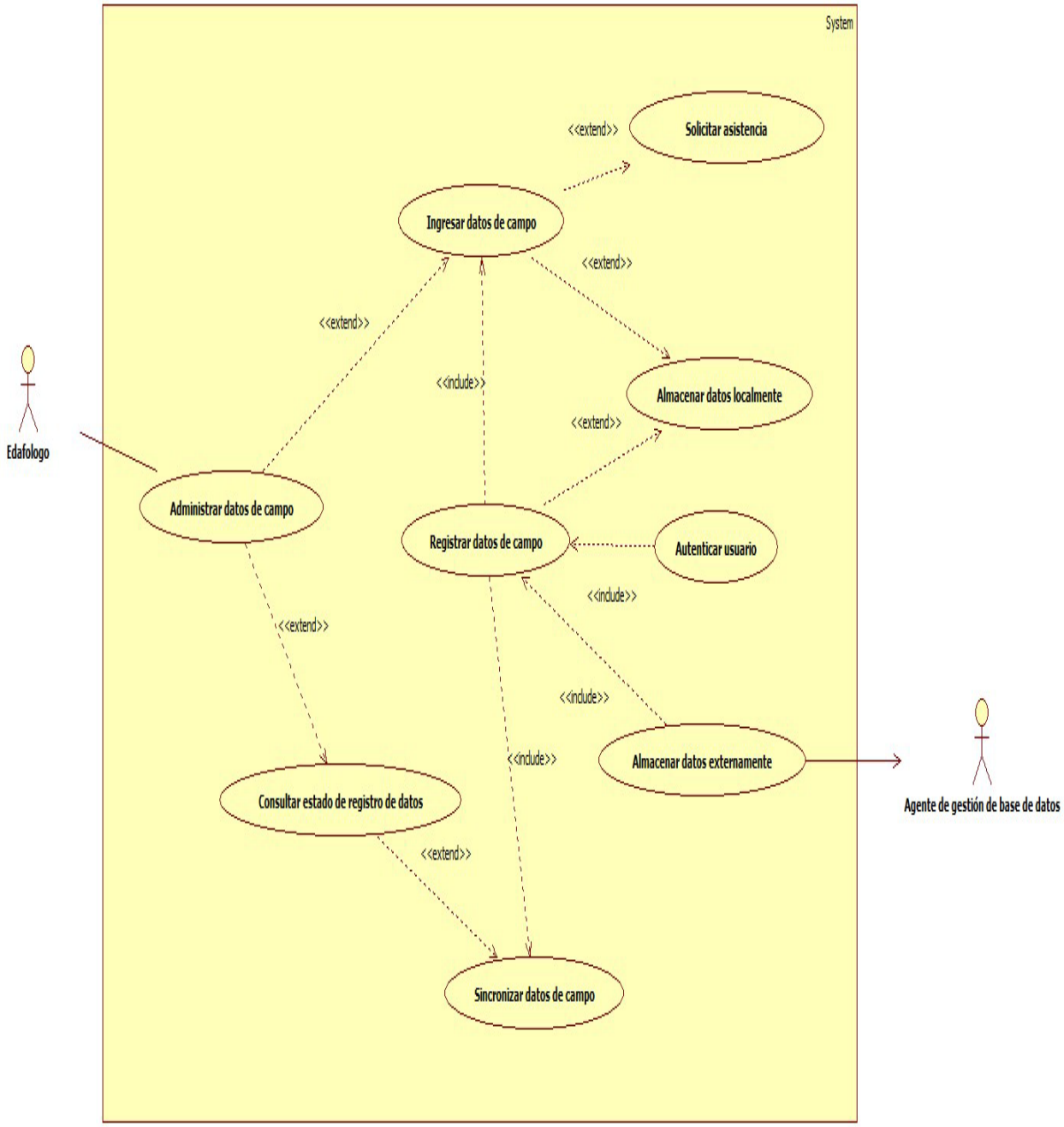


Figura 10: Casos de uso de la aplicación móvil

Tabla 5: Especificación de casos de uso

Caso de uso	Nivel	Descripción	Pre condiciones	Post condiciones	Flujo básico	Flujo alternativo	Extensiones
Administrar datos de campo	0	Consiste en administrar las capturas de campo realizadas desde el dispositivo.	Ninguno	Ninguna	El usuario elige ingresar datos de campo o ver el estado de capturas por registrar.	El usuario puede cerrar la aplicación	-ingresar datos de campo -consultar datos de campo
Ingresar datos de campo	1	Consiste en permitir al usuario completar una captura de datos de campo y registrarla.	Ninguno	Captura registrada	El usuario elige realizar una nueva captura o continuar una captura anterior.	El especialista puede cancelar la captura, sin perjudicar la aplicación. Si hay problemas durante el registro de datos, se almacena la captura localmente.	-Asistencia en el proceso de captura datos. -Almacenar la captura localmente
Consultar estado de registro de datos	1	Consiste en indicar las capturas que están pendientes por registrar, es decir, las que están guardadas localmente.	Ninguno	Se listan las capturas que están pendientes por registrar.	El usuario elige consultar estado de registro de datos.	Si no hay capturas pendientes por registrar, se le informa al usuario.	Sincronización de capturas no registradas.
Solicitar asistencia	2	Consiste en asistir el proceso de captura de datos: explicando conceptos, dando recomendaciones y corrigiendo errores.	Iniciar el proceso de captura.	Ninguna	Mientras el especialista es guiado en el proceso de captura, la asistencia va facilitando la captura.	Ninguna	Ninguna

Caso de uso	Nivel	Descripción	Pre condiciones	Post condiciones	Flujo básico	Flujo alternativo	Extensiones
Registrar datos de campo	2	Consiste en registrar los datos capturados, en el sistema de Siscla.	El especialista debe autenticarse.	La captura se almacena en Siscla	Se autentica el especialista y se lleva a cabo el registro de datos.	En caso de no haber conexión, los datos se almacenan localmente, hasta que se vuelva a tener conexión. Si el usuario no puede autenticarse, se le permite reintentar la operación o abortar el registro.	-Almacenar la captura localmente
Sincronizar datos de campo	2	Consiste en registrar las capturas que no se han podido registrar en Siscla.	Debe existir capturas pendientes por registrar. El especialista debe autenticarse	Se eliminan del dispositivo las capturas que fueron registradas	Se autentica al usuario y se lleva a cabo la sincronización.	En caso de algún error durante el registro, se le informa al usuario sin perjudicar el estado de las capturas.	Ninguna
Almacenar datos localmente	3	Consiste en almacenar los datos de captura en la memoria del dispositivo.	Los datos ingresados fueron verificados.	Los datos son almacenados en el dispositivo.	Se almacenan los datos localmente, en caso que no se pueda externamente.	En caso de error, se le informa al usuario la razón de la falla.	Ninguna
Almacenar datos externamente	3	Consiste en almacenar los datos de captura en el sistema de Siscla.	Los datos de captura están listos para enviarse El especialista debe autenticarse.	Los datos fueron registrados en Siscla. Se elimina cualquier almacenamiento temporal asociado a la captura.	Se autentica al usuario y se procede al registro de la captura.	En caso de error, se le informa al usuario la razón de la falla, y se le permite intentar de nuevo o almacenar los datos localmente.	Ninguna
Autenticar usuario	3	Consiste en la verificar la identidad del especialista.	Disponer de conexión móvil.	El especialista es autorizado.	Se captura los datos de cuenta y se validan en el sistema de clasificación de suelos.	Ninguno	Ninguno

Una de las particulares que tiene la aplicación es la forma en que se realiza el proceso de captura de datos. Para ver la secuencia de operaciones que realiza la aplicación en este proceso se elaboró el siguiente diagrama de secuencias. Ver Figura 11).

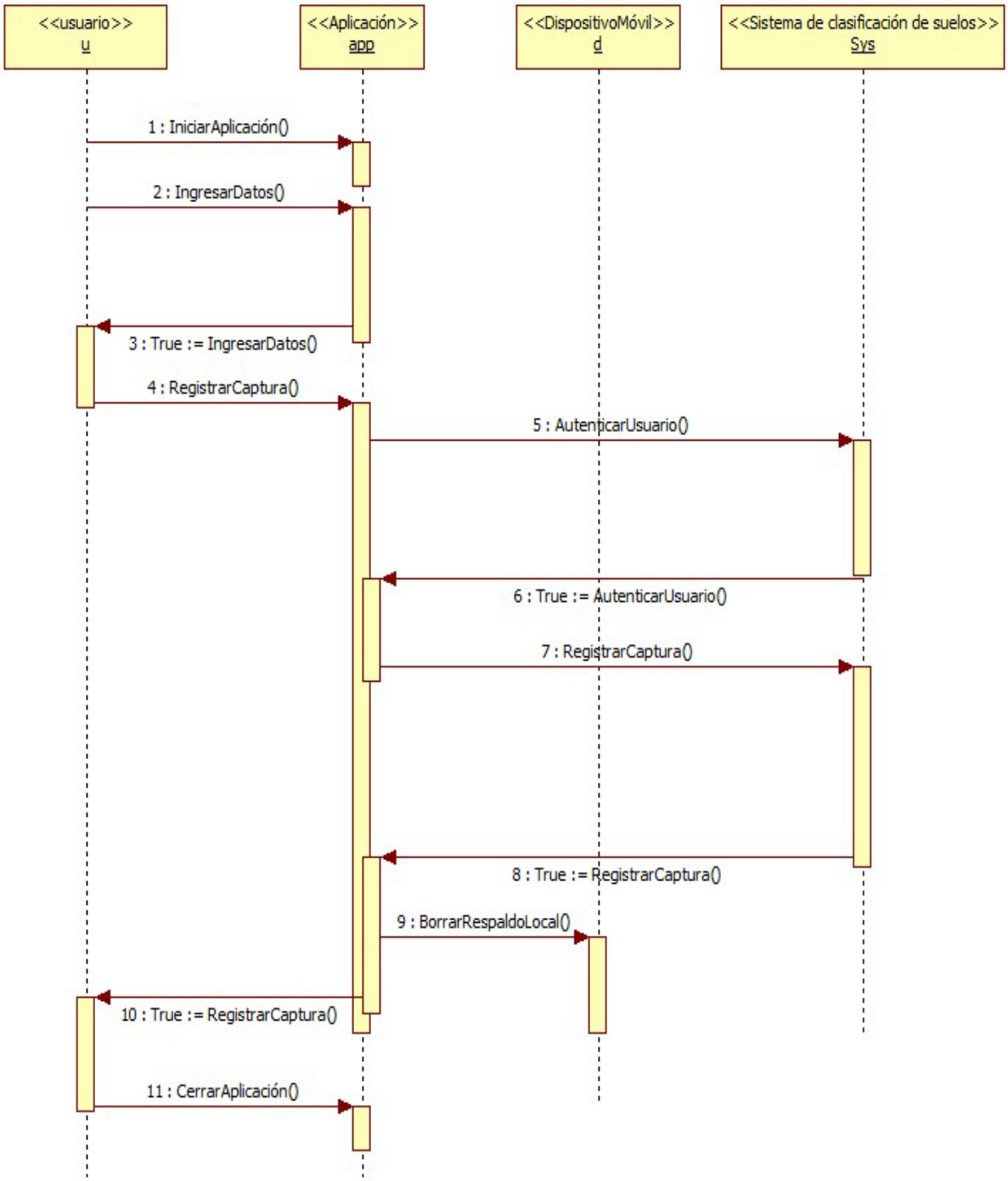


Figura 11: Diagrama de secuencias de la aplicación móvil

En la figura anterior se describe el flujo normal de operaciones requeridas para completar un proceso de captura de datos. Cada vez que el usuario completa una captura de datos, el sistema procederá a llevar el registro en el sistema Siscla; esto sucede siempre que el usuario se encuentre autorizado por la autenticación. Cuando una captura se registra exitosamente en el sistema de clasificación de suelos, la aplicación borrará automáticamente la presencia de cualquier almacenamiento local asociado a dicha captura. En caso de producirse alguna eventualidad durante este flujo de operaciones que describe el proceso de captura, el usuario podrá guardar la captura localmente.

Luego de hacer el modelo de casos de uso, se comenzó a modelar concretamente los agentes del sistema según la metodología JADE. El primer artefacto que propone la metodología para este fin es el diagrama de agentes, en el cual se identifican los agentes responsables de administrar y gestionar los recursos del sistema. Se identificaron los siguientes cuatro agentes: el Agente-captura, que es el responsable de coordinar todo el proceso de captura de datos; el Agente-IU, responsable de interactuar con el usuario mediante una interfaz de usuario; el agente Agente-Bdlocal, responsable de administrar el almacenamiento en el dispositivo móvil; y el Agente-Conexión, el cual se encarga de establecer las conexiones externas de la aplicación; en este caso, con el sistema Siscla.

El diagrama de agente respectivo se muestra a continuación en la Figura 12.

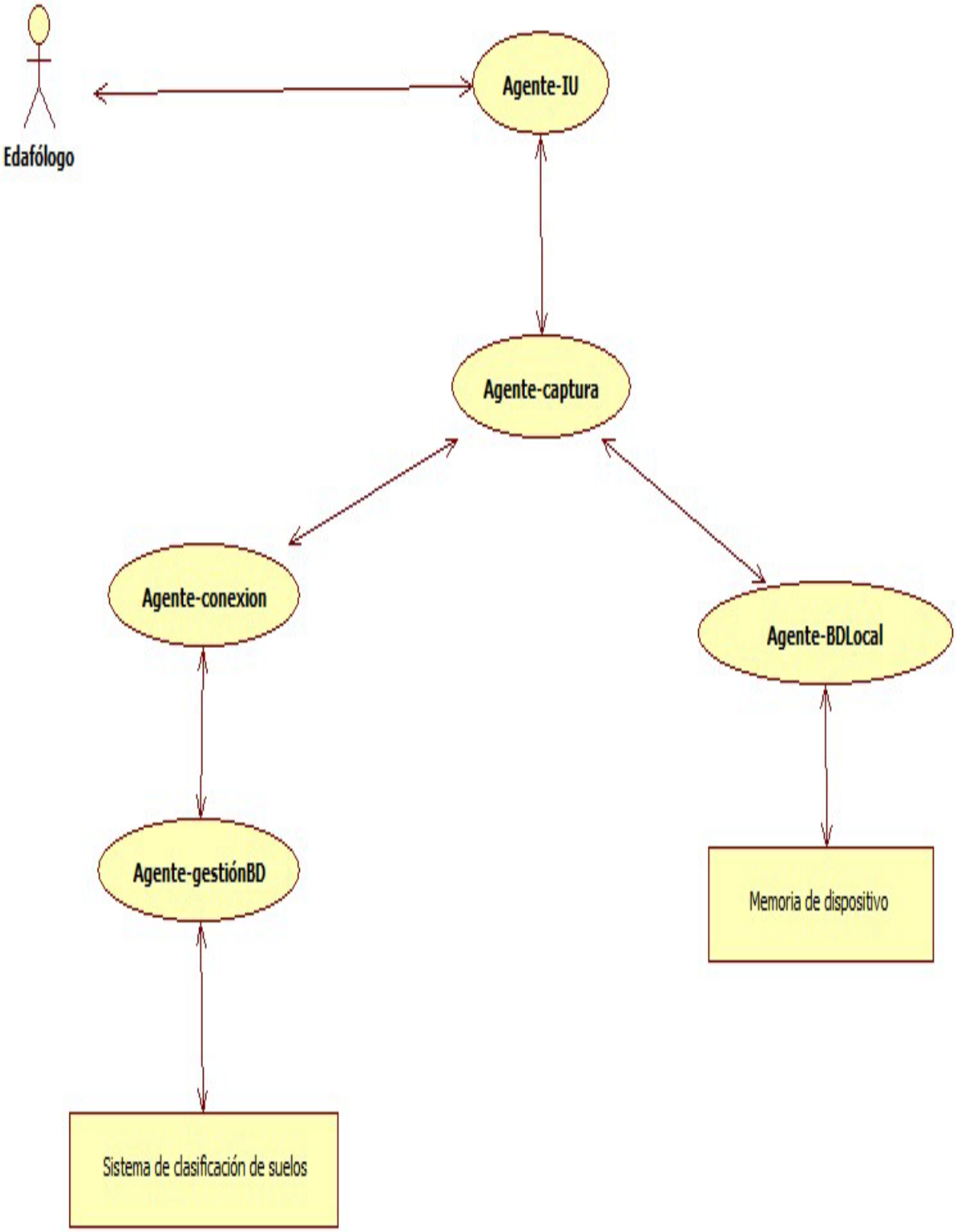


Figura 12: Diagrama de agentes de la aplicación

8

El siguiente artefacto realizado fue la tabla de responsabilidades de cada agente. Las responsabilidades fueron incluidas inicialmente en la tabla son aquellas relacionadas con la administración de recursos.

En actividades posteriores, esta tabla se refina para incluir aquellas responsabilidades asociadas a las interacciones entre los agentes y la administración de recursos especializados.

Por la extensión de la tabla de responsabilidades se ha querido omitirla para referenciarla en el Anexo 1.

El siguiente artefacto realizado fue el diagrama de despliegue, que consistió en representar a los agentes en su ambiente de ejecución correspondiente. La información que se obtiene de este diagrama, es usada para anticipar las características técnicas requeridas en el desarrollo de estos agentes.

El diagrama de despliegue de la aplicación se muestra a continuación en la Figura 13.

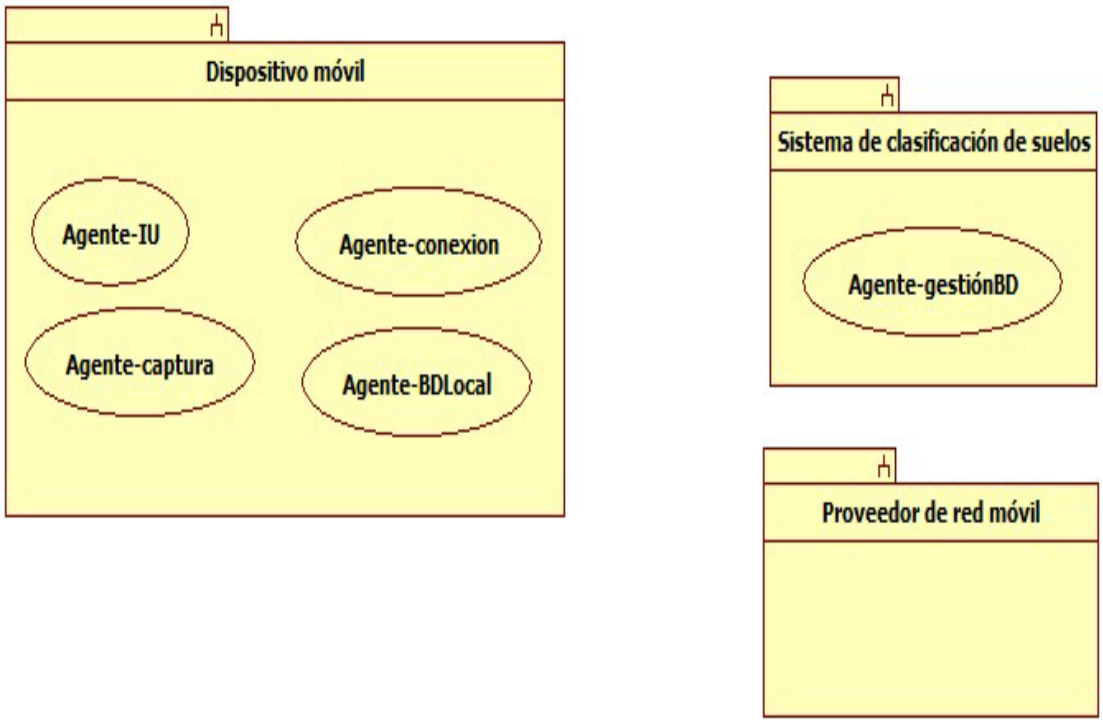


Figura 13: Diagrama de despliegue de la aplicación móvil

Todas las actividades que contempla la metodología JADE para esta fase, fueron realizadas. Ningún artefacto fue omitido. La duración de esta fase fue de tres semanas aproximadamente.

Algunos comentarios adicionales sobre esta fase se mencionan las siguientes:

Contemplar los casos de uso dentro de la metodología, ayuda a la descripción funcional de los agentes, además de ser una práctica comúnmente utilizada en desarrollo de sistemas tradicionales.

A pesar de que en esta fase no se dispone aún de información referente a la comunicación entre los agentes, esta fase modela de forma simple las características que definen a los agentes, permitiendo anticipar tempranamente la complejidad técnica y funcional inherente de cada uno de ellos.

2.3.3) Fase de diseño

La fase de diseño se caracterizó por la refinación de los agentes y la integración de los modelos. Además, se modeló el conocimiento del dominio de la aplicación y la interfaz de usuario.

Los artefactos obtenidos en esta fase fueron los siguientes: especificación de interacciones, la tabla de comportamientos para cada agente, el modelo de conocimiento y los prototipos de interfaz.

El primer artefacto desarrollado fue la especificación de interacciones. Para definir las interacciones fue necesario estudiar las responsabilidades adicionales entre los agentes de carácter colaborativo, es decir, que requerían de la participación de otros agentes para su realización. Las interacciones observadas fueron añadidas también en el diagrama de agentes. Además, se añadieron responsabilidades nuevas a los agentes, los cuales no habían sido identificados en la fase de análisis.

Al modelar las interacciones, se presentó cierta dificultad con las interacciones entre el Agente-Captura y el Agente-Interfaz, pues el número de interacciones reales entre estos dos agentes dependía del número de interfaces de la aplicación, que para ese momento no era conocido. También se presentó dificultad al modelar las interacciones entre el Agente-Captura y el Agente-Bdlocal, pues el número de interacciones reales entre estos dos agentes dependía no sólo del proceso de almacenamiento en sí, sino también de las características técnicas que proporciona la tecnología para la simplificación del software desarrollado; que en el caso de

este proyecto, se tradujo en un mayor número de interacciones entre estos dos agentes.

Se ha querido omitir la especificación de las interacciones debido a su extensión, y se ha colocado en el Anexo 2.

Después de realizar la especificación de interacciones, se procedió a realizar la tabla de comportamientos de cada agente. Se utilizaron tanto el diagrama de responsabilidades como la especificación de interacciones para definir los comportamientos de cada agente. Se conservó el atributo de la tabla referido al tipo de comportamiento según JADE, debido a que esta información podría ser de utilidad en la implementación de los comportamientos. Además, se añadieron dos atributos adicionales a la tabla que fueron los siguientes: el estado final y parámetros del comportamiento. Con el primero se expresa el estado final de la aplicación, después de que el agente realice el comportamiento; y el segundo, define el tipo de mensaje que recibe el comportamiento. Vale destacar que la metodología JADE propone un instrumento para especificar los mensajes, pero no se utilizó porque estaba fuertemente vinculado con la implementación de mensajes en la plataforma JADE.

En la próxima página se muestra la Tabla 6 que fue la tabla de comportamientos resultante de esta actividad.

Tabla 6: Tabla de comportamientos

Tipo de agente	Comportamiento	Tipo de comportamiento	Id_Resp. mapeada	Interacción mapeada	Parámetros	EdoFinal
Agente-IU	Cargar_interfaz	SimpleBehavior	1	CargaIUIngresarDatos CargaIUMenuCapturas CargaIUCapturasPorTerminar CargaIUAutenticación CargaIUEstadoCapturas CargaIUFalla CargaIUPregunta CargaIUEspera CargaIUÉxito	Interfaz a cargar (object)	Nueva interfaz desplegada
	Cargar_interfaz_anterior	SimpleBehavior	2	Ninguna	Ninguno	Interfaz anterior cargada
	Interfaz_ayuda	SimpleBehavior	3	Ninguna	Ninguno	Interfaz de ayuda cargada
	Interfaz_bienvenida	SimpleBehavior	3	Ninguna	Ninguno	Inicio de la aplicación
	Interfaz_salida	SimpleBehavior	5	Ninguna	Ninguno	Cierre de la aplicación
	Interfaz_pausa	SimpleBehavior	6	Ninguna	Ninguno	Aplicación pausada
Agente-Captura	Registrar_captura_externamente	OneShotBehaviour	1	-Registro de datos externamente -Conexión cancelada	Registro de captura (object)	Datos de campo almacenados en el sistema de clasificación de suelos
	Registrar_captura_localmente	OneShotBehaviour	2	-Registro de datos localmente -Eliminación de datos localmente -Operación cancelada	Registro de captura (object)	Datos de campo almacenados en la memoria del dispositivo
	Recuperar_captura	OneShotBehaviour	3	Recuperación de datos	Index (integer)	Carga de captura en la interfaz de ingresar datos.
	Autenticar	OneShotBehaviour	4	Autenticación de usuario	Datos de autenticación (object)	Estado de autenticación (boolean)
	Sincronizar_capturas	OneShotBehaviour	5	-Registro de datos externamente -Eliminación de datos localmente	Ninguno	Estado de sincronización (boolean)
	Interfaz_menúCapturas	SimpleBehavior	6	CargaIUMenuCapturas	Ninguno	Selección del tipo de captura que se desea iniciar.

Tipo de agente	Comportamiento	Tipo de comportamiento	Id_Resp. mapeada	Interacción mapeada	Parámetros	EdoFinal
	Interfaz_capturasPorTerminar	SimpleBehavior	7	CargaIUCapturasPorTerminar	Ninguno	Carga de captura por terminar seleccionada
	Interfaz_Wizard	SimpleBehavior	8	CargaIUIngresarDatos	Paso (Integer)	Obtiene interfaz correspondiente a la captura de datos
	Interfaz_autenticación	SimpleBehavior	9	CargaIUAutenticación	Ninguno	Obtiene la interfaz de autenticación para cargarla
	Interfaz_estadoCapturas	SimpleBehavior	10	CargaIUEstadoCapturas	Ninguno	Obtiene la interfaz de estado de capturas.
	Interfaz_pregunta	SimpleBehavior	11	CargaIUPregunta	Operación (String)	Consulta de usuario sobre operación
	Interfaz_falla	SimpleBehavior	12	CargaIUFalla	Operación (String)	Usuario informado de la falla
	Interfaz_espera	SimpleBehavior	13	CargaIUEspera	Operación (String)	Usuario informado por la espera de una operación
	Interfaz_éxito	SimpleBehavior	14	CargaIUExito	Operación (String)	Usuario informado por el éxito de una operación
Agente-BDLocal	Registrar_captura	OneShotBehaviour	1	Registro de datos localmente	Registro de captura (object)	Datos almacenados localmente
	Recuperar_captura	OneShotBehaviour	2	Recuperación de datos localmente	Index (Integer)	Registro de captura (object)
	Eliminar_captura	OneShotBehaviour	3	Eliminación de datos localmente	Index (Integer)	
	Cancelar_operación	OneShotBehaviour	4	Operación cancelada	Ninguno	Operación local cancelada
	Nombres_capturas	SimpleBehavior	5	Nombres de capturas almacenadas	TipoCaptura (String)	Lista de nombres
	Fechas_capturas	SimpleBehavior	6	Fechas de capturas almacenadas	TipoCaptura (String)	Lista de fechas
	Índice_captura	SimpleBehavior	7	Índice de captura	NombreCaptura (String)	Índice (Integer)
	Capturas_porTipo	SimpleBehavior	8	Capturas por tipo	TipoCaptura (String)	Lista de capturas según el tipo

Tipo de agente	Comportamiento	Tipo de comportamiento	Id_Resp. mapeada	Interacción mapeada	Parámetros	EdoFinal
Agente-gestiónBD	Registrar_captura	OneShotBehaviour	1	Registro de datos externamente	Registro de captura (object)	Datos almacenados en sistema de clasificación de suelos
	Validar_autenticación	OneShotBehaviour	2	Validación de autenticación	Datos de autenticación (object)	Estado de autenticación (boolean)
Agente-Conexión	Registrar_captura	OneShotBehaviour	1	Registro de datos externamente -Registro de datos externamente (Siscla)	Registro de captura (object)	Estado del registro de captura (boolean)
	Autenticar_usuario	OneShotBehaviour	2	Autenticación de usuario Autenticación de usuario (Siscla)	Datos de autenticación (object)	Estado de autenticación (boolean)
	Cancelar_conexión	OneShotBehaviour	3	Conexión cancelada	Ninguno	Conexión externa terminada

Con la tabla de comportamientos de cada agente, los agentes pueden implementarse mediante un lenguaje de programación.

Las siguientes actividades realizadas durante esta fase consistieron en modelar el conocimiento del dominio de la aplicación y la interfaz de usuario, los cuales serán explicados en detalle en subcapítulos posteriores.

Algunos comentarios adicionales de esta fase, se mencionan las siguientes:

La especificación de interacciones presentó cierto grado dificultad debido al esfuerzo de abstracción necesario para relacionar las responsabilidades de carácter colaborativo definida entre los agentes. También vale mencionar que múltiples artefactos utilizados en esta fase, necesitaron ser modificados para adaptarse a las características del proyecto.

2.3.4) Fase de implementación

Consistió en la construcción funcional del agente móvil. Para esta fase la metodología JADE propone seguir las actividades tradicionales de desarrollo, por lo que se adoptó un desarrollo basado en prototipaje hasta obtener un prototipo completamente funcional. Cada uno de los incrementos de la aplicación era liberado y corregido en un tiempo no mayor de 2 semanas.

Las herramientas utilizadas para el desarrollo fueron las siguientes:

- Netbeans 6.8 Full Versión.
- Plataforma de Java Micro Edition SDK 3.0.
- Sun Java Wireless Toolkit 2.5.2 para CLDC.
- Servidor Web Apache Tomcat 6.0.
- Servidor de base de datos Mysql 5.0.

La plataforma de Java Micro Edition SDK y Sun Java Wireless toolkit proporcionaron una serie de emuladores con distintas características físicas, técnicas y de interacción, que fueron utilizados durante el desarrollo de la aplicación.

Las tecnologías implicadas durante el desarrollo del agente móvil, fueron las siguientes:

- En la construcción de la interfaz de usuario se utilizó la tecnología Java2D para la construcción de items personalizables, y LCDIU del paquete J2ME para la construcción de los formularios.
- Para manejar el almacenamiento local, se utilizó la tecnología RMS del paquete J2ME, con el fin de evitar los inconvenientes relacionados con problemas de compatibilidad de los sistemas archivos.
- Para la comunicación entre aplicaciones, se utilizó la API para servicios web de J2ME. Las características de estos servicios implementados serán descritos en apartados posteriores.

Durante el desarrollo del agente móvil, se realizaron una serie de incrementos significativos de la aplicación, los cuales han sido añadidos en el CD anexo de este informe. Las características de estos incrementos realizados se describen a continuación:

- **Prototipo IU**: fue el primer prototipo desarrollado de la aplicación, muestra la implementación de las interfaces a utilizar en la aplicación. Esta liberación proporcionó observaciones a la guía de estilo y a las de interfaces de la aplicación.
- **Prototipo IU-RMS**: fue el siguiente incremento de la aplicación, en la se incluyó el manejo de eventos en las interfaces y la lógica de

almacenamiento en el dispositivo local. Esta liberación proporcionó mejoras en la construcción de interfaces de la aplicación y detalles de la sincronización de datos almacenados localmente.

- **Prototipo versión funcional**: fue el primer prototipo completamente funcional de la aplicación, incluye los aspectos de la comunicación web con el sistema Siscla, así como las interfaces faltantes de la aplicación.

Después de haber obtenido una la versión funcional de la aplicación, se pasó a la fase de pruebas para corregir las fallas de software y funcionamiento en la aplicación.

Algunos comentarios adicionales durante la fase desarrollo, se mencionan a continuación:

El adiestramiento en el uso de la tecnología JavaME requirió un tiempo significativo durante el desarrollo. No se encontró una API detallada de las funciones y uso de esta plataforma, además mucha de la información recopilada era referente a tutoriales y experiencias específicas de otros desarrolladores sobre algún aspecto de la tecnología. Por otro lado, la implementación de las especificaciones JSR, que componen la arquitectura móvil de servicios de JavaME es definida por cada fabricante, lo que complica más la búsqueda de información sobre el uso de esta plataforma.

Las herramientas de generación de código que proveen algunos IDE de desarrollo, por ejemplo Netbeans que fue usado para este proyecto, permitió la generación automática de código de algunos componentes de software, por ejemplo la generación de servicios web. Otras en cambio no fueron utilizadas, como por ejemplo One Jazzy para la generación visual de formularios J2ME, por considerarse poco práctico para la generación del gran número de formularios utilizados en la aplicación, los cuales requerían también incluir además algunas características personalizadas.

Otra dificultad que se presentó durante el desarrollo fue la abstracción lógica requerida en algunos componentes de software desarrollados, como por ejemplo, el uso de hilos concurrentes, la carga de imágenes, la generación de formularios, etc. En los cuales se aplicaron una serie de patrones de diseño para facilitar la interacción entre estos componentes, y así contribuir a la alta cohesión y el bajo acoplamiento de la aplicación.

La duración de la fase de desarrollo fue de aproximadamente seis semanas.

2.3.5) Fase de pruebas

Esta fase consistió en detectar y corregir fallas en el funcionamiento del agente móvil. Los casos de prueba fueron construidos haciendo uso del modelo de casos de uso. Además, se hicieron una serie de recorridos en el uso de la aplicación con el fin de corregir fallas de presentación de la interfaz, entre otras.

Se escogieron dos prototipos funcionales para la realización de las pruebas:

- **Prototipo basado en emuladores J2ME**: Consistió en escoger uno de los emuladores que proporciona la J2ME SDK, que ofreciera una presentación usable y confortable para el uso de la aplicación.
- **Prototipo basado en dispositivos reales**: Consistió en escoger un dispositivo de prueba, para poner en funcionamiento el agente de móvil y adaptarlo para su utilización en dicho dispositivo. Se escogió el uso del siguiente dispositivo para la realización de las pruebas Nokia 5230 XPressMusic.

Las pruebas realizadas se expresan a continuación:

Resultado de las pruebas

- **Pruebas por casos de uso**: Estas pruebas consistieron en probar el funcionamiento de la aplicación, definiendo casos de pruebas en función de los casos de uso de la aplicación. Las Tablas 7, 8, 9, 10 hacen referencia a cada una de de estas pruebas realizadas. Más adelante, se comentan algunas observaciones observadas durante las pruebas.

Tabla 7: Autenticar usuario

Clase de prueba	Tipo de pruebas	Prueba	Estado esperado	Estado obtenido	Verificación
Autenticación fallida	Autenticación con problemas de conexión	Desconectar dispositivo. Luego Autenticar	Se muestra Aviso de falla	Se muestra Aviso de falla	✓
		Desconectar el servidor. Luego Autenticar	Se muestra Aviso de falla	Se muestra Aviso de falla	✓
	Proporcionar datos de autenticación no válidos.	Autenticar con login y password no válidos.	Se muestra Aviso de falla	Se muestra Aviso de falla	✓

Tabla 8: Ingresar datos de campo

Clase de prueba	Tipo de pruebas	Prueba	Estado esperado	Estado obtenido	Verificación
Problemas durante la captura de datos.	Captura sin datos obligatorios	Sin datos agrologo, perfil, utm(e), utm(n). utm(s) Luego Registrar captura	Se muestra Aviso de falla	Se muestra Aviso de falla	✓
		Sin datos agrologo, perfil, utm(e), utm(n). utm(s) Luego Detener captura	Se muestra el Aviso de falla	Se muestra el Aviso de falla	✓
	Captura con ningún perfil asociado.	Captura terminada sin perfiles Registrar captura	Se muestra el Aviso exitoso de Registro (en Siscla)	Se muestra el Aviso exitoso de Registro (en Siscla)	✓
	Captura con múltiples perfiles asociados.	Captura terminada con perfiles 4. Luego Registrar captura	Se muestra el Aviso exitoso de Registro (en Siscla)	Se muestra el Aviso exitoso de Registro (en Siscla)	✓
	Captura con valores nulos o no definidos.	Captura con valores en blanco, excepto los obligatorios.	Se muestra el Aviso exitoso de Registro (en Siscla)	Se muestra el Aviso exitoso de Registro (en Siscla)	✓

Tabla 9: Registrar datos de campo

Clase de prueba	Tipo de pruebas	Prueba	Estado esperado	Estado obtenido	Verificación
Problemas durante el registro de datos	Falla en el registro externo de los datos.	-Desconectar dispositivo. -Registrar captura.	Se muestra Aviso de falla	Se muestra Aviso de falla	✓
		-Desconectar el servidor. -Registrar captura	Se muestra Aviso de falla	Se muestra Aviso de falla	✓
	Falla en el registro local de los datos	-Error de programación -Registro local	Se muestra Aviso de falla	Se muestra Aviso de falla	✓

Tabla 10: Consultar estado de registro de datos

Clase de prueba	Tipo de pruebas	Prueba	Estado esperado	Estado obtenido	Verificación
Problemas al mostrar las capturas pendientes en el listado	Número de capturas pendientes no esperadas	-Sin captura	Listado indica que no capturas pendientes	Listado indica que no capturas pendientes	✓
		-5 capturas.	Listado indica las 5 capturas	Listado indica las 5 capturas	✓
Problemas en la sincronización de capturas.	Falla de envío de múltiples capturas.	-Enviar 5 capturas sin perfiles.	-Registro exitoso (en Siscla) y el Almacenamiento temporal borrado.	Registro exitoso (en Siscla) y el Almacenamiento temporal borrado.	✓
		-Enviar 5 capturas con perfiles.	Registro exitoso (en Siscla) y el Almacenamiento temporal borrado.	-Registro exitoso (en Siscla) y el Almacenamiento temporal borrado.	✓
	Falla de conexión, durante el envío de capturas.	Desconectar dispositivo. -Enviar capturas.	Se muestra Aviso de falla	Se muestra Aviso de falla	✓

- **Pruebas por recorrido de escenarios de uso:** Estas pruebas consistieron en realizar recorridos por la aplicación para corregir fallas en la interfaz, entre otros. Los resultados son mostrados en la Tabla 11.

Tabla 11: Escenarios de uso

Escenario de uso	Tipo de prueba	Prueba	Estado esperado	Estado obtenido	Verificación
Recuperar una captura por terminar	Falla en la recuperación	-Realizar una captura y detenerla -Recuperar una captura.	Carga correcta de captura	Carga correcta de captura	✓
		-Realizar una captura y detenerla -Cerrar la aplicación -Recuperar la captura	Carga correcta de captura	Carga correcta de captura	✓
		-Realizar una captura detenerla. -Recuperar una captura y volverla a detener -Recuperar captura.	Carga correcta de captura	Carga correcta de captura	✓
		-5 capturas por terminar -Recuperar un captura	Carga correcta de captura	Carga correcta de captura	✓
Iniciar una nueva captura	Falla al iniciar una nueva captura	-Iniciar una nueva captura.	Carga correcta de captura vacía	Carga correcta de captura vacía	✓
		-Iniciar una nueva captura, detenerla e iniciar una nueva captura.	Carga correcta de captura vacía	Carga correcta de captura vacía	✓
		-Iniciar una nueva captura, completarla e iniciar una nueva captura.	Carga correcta de captura vacía	Carga correcta de captura vacía	✓

Al finalizar las pruebas se constató dos problemas sin solución en el Nokia 5320. El primero se presentó en los listados con Items (en los listados de capturas por terminar, capturas pendientes por registrar y administrar horizontes de una captura), los cuales al añadir un número determinado de Items al formulario, la aplicación se congelaba y al tiempo se cerraba. Haciendo distintas pruebas, se concluyo que se debe justamente al número de CustomItems que añadidos a un formulario. El segundo problema se presenó tanto en el Nokia como en el prototipo con emuladores, siempre que se realiza una conexión http, con el objeto httpconection a través de la función open, en la clase MobileClient; si el servidor no está disponible, la aplicación se congela por un momento a pesar de que la conexión se encuentre en un hilo de ejecución separado de la aplicación base, y retorna el control al producirse una excepción por error de conexión.

Buscando referencias en Internet, se encontró que Nokia presenta múltiples problemas de implementación para Midp 2.0. Para el caso de la función open usando httpconection, se corroboró un bug en el método que no permite cancelar la conexión, dejando a la aplicación en lectura indefinida.

Además de estas pruebas, se logró probar la aplicación: LG KP-540 y Nokia E71, en los que se evidencia los efectos de la fragmentación en la aplicación, específicamente, en la presentación de la interfaz.

La duración de la fase de pruebas fue de aproximadamente tres semanas.

Habiendo culminado todo el proceso de desarrollo del agente móvil, se puede destacar lo siguiente:

La metodología JADE se ajustó en las actividades y prácticas para el diseño y construcción del agente móvil. La distribución de tareas en una sociedad de cuatro agentes, facilitó la implementación de las funcionalidades del sistema y el desarrollo de algunos componentes de software adicionales que sirven de apoyo en sus operaciones.

La plataforma de aplicaciones J2ME por su parte, permite al agente móvil ajustarse para futuras ampliaciones y extensiones. Esto se debe a que el código de la aplicación es portable para una gran variedad de dispositivos y tecnologías.

2.4) Modelo de conocimiento del agente móvil

Una parte importante en el diseño del agente, fue representar el conocimiento del dominio de la aplicación. El conocimiento que maneja este agente corresponde

a los datos de campo que son recopilados por los edafólogos para la caracterización del suelo.

La metodología JADE no propone artefactos para modelar el conocimiento de los agentes, se limita simplemente a recomendar una actividad para modelar las ontologías del dominio. Para este propósito se escogió modelar el conocimiento del agente móvil mediante un esquema de representación de redes semánticas, usando la herramienta Compendium (CompendiumInstitude, 2010).

Para construir la red semántica se utilizaron las siguientes fuentes de información:

- **Modelo de datos del agente de gestión de base de datos**: Representa el conocimiento y la información gestionada por este agente para el proceso de captura y clasificación de suelos. Esta información ha sido validada por especialistas edafólogos y es la referencia más fidedigna sobre el dominio de la aplicación.

- **Plantilla de campo**: Son plantillas textuales que utilizan los edafólogos para la captura de datos en los sitios de levantamiento de suelo. Estas plantillas fueron facilitadas al grupo de trabajo de LIA, por parte de miembros de la Facultad de Agronomía. La información que proporcionó esta plantilla fue validada con el modelo de datos que utiliza el agente de gestión de base de datos.

- **Chuleta de campo**: Es una guía referencial de los conceptos utilizados en las plantillas de campo. Esta información sirvió para revisar el modelo de conocimiento utilizado en el proceso de captura de datos de campo.

- **Guía para la definición de pedones**: Es una guía más completa que la anterior, que describe el conjunto de datos utilizados en la caracterización de un perfil de suelo. Esta información también sirvió para revisar el modelo de conocimiento utilizado en el proceso de captura de datos de campo.

- **Diccionario de datos del sistema Siscla**: Es un documento que explica en detalle el modelo de conocimiento utilizado por el agente de gestión de base de datos. Esta información fue útil para compararla con la información que proporcionaron las plantillas de campo.

Para tener referencia de estos documentos se recomienda solicitarlos al grupo de trabajo de LIA, o consultarlo en las referencias de este trabajo.

En la Figura 14 se muestra la red semántica resultante de esta actividad. La especificación de los atributos de clase se ha incluido en la Tabla 12 para facilitar la lectura de la red.

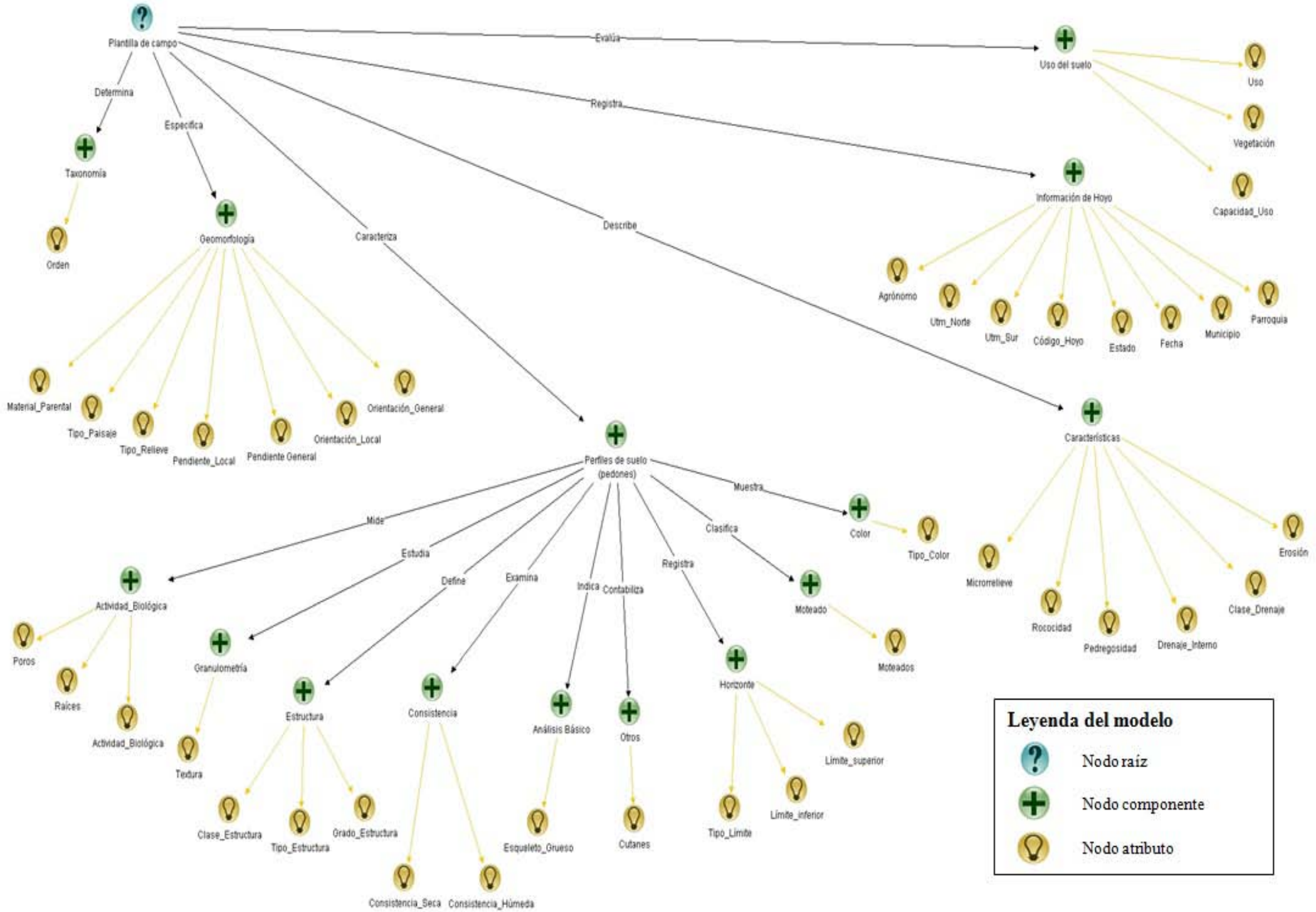


Figura 14: Modelo de conocimiento de la aplicación móvil

Tabla 12: Especificación de valores de atributos

Nombre	Valor	Pertenece a concepto
Clase_drenaje	[Muy pobremente drenado, pobremente drenado, imperfectamente drenado, moderadamente bien drenado, bien drenado, drenaje algo drenado, excesivamente drenado]	Características generales
Drenaje_interno	[Muy lento, lento, moderado, rápido]	Características generales
Erosión	[Sin evidencia, ligera, moderada, fuerte, severa]	Características generales
Microrelieve	[Liso, concavo depresión, monticular o convexo, irregular, gilgai]	Características generales
Pedregosidad	[Sin piedras menor 0,01%, moderadamente pedregoso 0,01 - 0,1%, pedregoso 0,1 - 3%, muy pedregoso 3-15%, excesivamente pedregoso 5-90%, terreno ripioso mayor a 90%]	Características generales
Rocosidad	[Ninguna o muy poca, moderadamente rocosa, muy rocosa, extremadamente rocosa, afloramiento rocoso]	Características generales
Pendiente_general	[0-3%, 3-5%, 5-8%, 8-16%, mayor a 16%]	Geomorfología
Orientación_general	[Norte-sur, norte-este, norte-oeste, sur-norte, sur-este, sur-oeste, este-norte, este-sur, este-oeste, oeste-norte, oeste-sur, oeste-este]	Geomorfología
Pendiente_local	[0-3%, 3-5%, 5-8%, 8-16%, mayor a 16%]	Geomorfología
Orientación_local	[Norte-sur, norte-este, norte-oeste, sur-norte, sur-este, sur-oeste, este-norte, este-sur, este-oeste, oeste-norte, oeste-sur, oeste-este]	Geomorfología
Forma_terreno	[Banco alto, banco medio, bajio, estero, abanico, glacis, terraza lacustrina 1er nivel, banco bajo, terraza lacustrina 2do nivel, terraza lacustrina 3er nivel, terraza lacustrina 4to nivel, terraza lacustrina 3ro - 4to nivel, depósitos de creciente, cauce rellenado, cima, lomas, terraza lacustrina]	Geomorfología
Material Parental	[Aluvial, Aluvial sobre Lacustrino, Lacustrino, Lacustrino sobre Aluvial, Coluvial]	Geomorfología
Tipo_paisaje	[Planicie, piedemonte, valle, altiplanicie, planicie y valles]	Geomorfología
Tipo_relieve	[Plano inclinado, vega, llanura aluvial, llanura lacustrina, pantano lacustrino, delta, mesa, valle coluvial-aluvial, vallecito aluvial-mesa]	Geomorfología
Agrónomo	String	Información del hoyo
Codigo_Hoyo	String: (5 letras)	Información del hoyo
Estado	[estados en venezuela]	Información del hoyo
Fecha	Date	Información del hoyo
Localidad	String	Información del hoyo
Municipio	[municipios en venezuela]	Información del hoyo
UTM(E)	Entero sin signo	Información del hoyo
UTM(N)	Entero sin signo	Información del hoyo
Actividad biológica	[Abundante, frecuente, poca, nula]	Perfiles de suelo - Actividad biológica
Poros	[pocas menor a 1 raíz / cm2, frecuentes 1-5 raíces /cm2, muchas mayor a 5 raíces / cm2]	Perfiles de suelo - Actividad biológica
Raíces	[pocas menor a 1 raíz / cm2, frecuentes 1-5 raíces /cm2, muchas mayor a 5 raíces / cm2]	Perfiles de suelo - Actividad biológica
Esqueleto_grueso	% [0-100]	Perfiles de suelo - Análisis básico

Nombre	Valor	Pertenece a concepto
Profundidad	[mayor a 100, 50-100, 25-50,0-25]	Perfiles de suelo - Característica
Tipo_color	[Seco, Húmedo]	Perfiles de suelo - Color
Consistencia_seca	[Suelta blanda, ligeramente dura, dura, muy dura, extremadamente dura]	Perfiles de suelo - Consistencia
Consistencia_húmeda	[Muy friable, friable, firme, muy firme, extremadamente dura]	Perfiles de suelo - Consistencia
Clase_estructura	[Primaria, secundaria]	Perfiles de suelo - Estructura
Grado_estructura	[Sin estructura, débil, moderado, fuerte]	Perfiles de suelo - Estructura
Tipo_estructura	[Maciva o macisa, blocosa angular, blocosa sungular, primástica, laminar, poliedrones, granos simples]	Perfiles de suelo - Estructura
Textura	[Arenoso, arcilloso, franco, esquelético]	Perfiles de suelo - Granulometría
Tipo_límite	[Plano, ondulado, irregular]	Perfiles de suelo - Horizonte
Límite_superior	Entero sin signo	Perfiles de suelo - Horizonte
Límite_inferior	Entero sin signo	Perfiles de suelo - Horizonte
Moteados	[moteado 1, moteado 2]	Perfiles de suelo - Moteado
Cutanes	Entero sin signo	Perfiles de suelo - Otros
Orden	[Alfisoles, entisoles, inceptisoles, molisoles, oxisoles, ultisoles, vertisoles]	Taxonomía
Capacidad_uso	String	Uso del suelo
Uso	String	Uso del suelo
Vegetación	String	Uso del suelo

El modelo de conocimiento fue utilizado para definir los campos utilizados en el proceso de captura y registro de datos. Vale destacar que ninguno de los agentes definidos usa explícitamente este conocimiento, sino que a través de algunos componentes de software adicionales, los agentes hacen uso implícito de este conocimiento.

Este modelo fue realizado en la fase de diseño, y necesito de al menos dos semanas de trabajo.

2.5) Construcción de la interfaz de usuario del agente móvil

La construcción de la interfaz de usuario fue una actividad que fue realizada en la fase de diseño e implementación del agente móvil. Consistió en modelar las interfaces de la aplicación, haciendo uso del modelo de conocimiento de la aplicación y las funcionalidades definidas en los casos de uso.

El primer artefacto utilizado para modelar la interfaz, fue realizar unos bocetos en papel de cada una de las interfaces, e identificar los recorridos cognitivos entre ellas. Luego, se incluyeron algunos patrones de interacción en el proceso de captura de datos, como por ejemplo: ayuda, wizard, mensajes de información, entre otros. Además, se siguió los aspectos definidos en la guía de estilo utilizada en el sistema Siscla para conservar los principios de consistencia.

Los bocetos o prototipos en papel, así como la guía de estilo utilizada en la aplicación, han sido incluidos en el Anexo 3 y 4 respectivamente.

Demostración del sistema: Se ha querido incluir en este apartado un recorrido por las interfaces de la aplicación desarrollada. Las siguientes imágenes fueron tomadas haciendo uso del emulador "DefaultFxTouchPhone1" de la plataforma de Java Micro Edition SDK 3.0, para describir el funcionamiento y uso de la aplicación. De la Figura 15 a la 25 pueden verse cada una de estas interfaces.

- **Inicio de aplicación:**

Al iniciarse la aplicación, se cargan todos los componentes del sistema y se muestra la interfaz de entrada que es "Principal". Desde esta interfaz el usuario tiene acceso a las funcionalidades del sistema, en este caso, a "captura datos" (que inicia el proceso de captura) y a "estado capturas", donde se muestran las capturas que no se han podido registrar.

Figura 15: Inicio de aplicación

- **Menú de capturas:**

Cuando el usuario elige “captura datos”, se carga la interfaz de menú de capturas, donde el usuario puede seleccionar el tipo de captura que desea realizar. Cuenta con dos opciones, la primera consiste en iniciar una nueva captura y la segunda consiste en recuperar una captura anterior para completarla. Haciendo click en alguna de las opciones, se carga el proceso de captura correspondiente

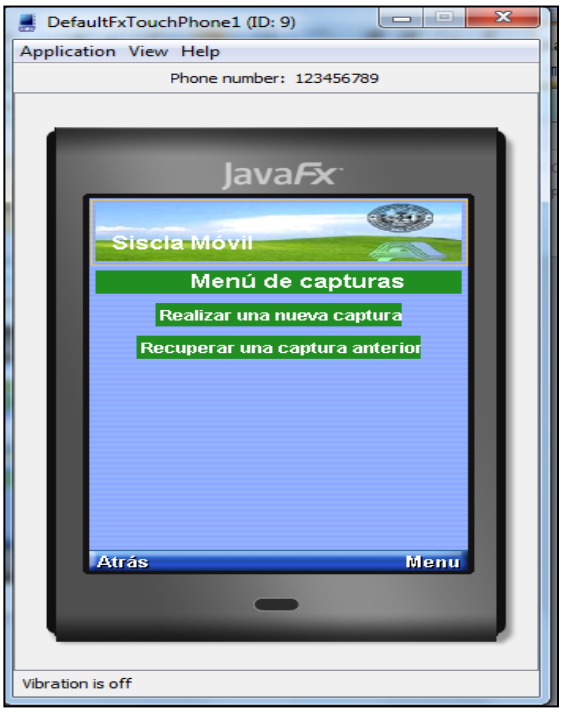


Figura 16: Menú de capturas

- **Recuperar una capturas anterior:**

En la interfaz de “Recuperar una captura anterior”, el usuario podrá seleccionar una de las capturas que no se han completado. Dándole en la “lupa”, el sistema carga la captura correspondiente y reanuda el proceso de captura.

Figura 17: Recuperación de capturas

- **Captura de datos:**

Sea que el usuario elija comenzar una nueva captura o cargue una anterior, se inicia el proceso de captura, el cual se encuentra estructurado en pasos relacionados con la información del hoyo. El primer paso, consiste en información básica que describe el hoyo respectivo; el segundo paso, consiste en la descripción geomorfológica de la zona; el tercer paso, consiste en las características verificables en el lugar del hoyo; el cuarto paso, consiste en la información taxonómica; el quinto paso, consiste en la capacidad de uso que evidencia el suelo; el sexto paso contiene la información asociada a los perfiles de suelo registrado del hoyo; el séptimo paso es finalmente el registro de la captura.

Figura 18: Captura de datos

- **Administrar perfiles:**

Para añadir uno o más perfiles, la aplicación permite agregar, modificar y eliminar tantos perfiles se necesite registrar del hoyo respectivo.

Figura 19: Administrar perfiles

- **Detener una captura:**

Si un usuario no puede completar una captura por las razones que fuesen, el sistema le permite detener el proceso de captura, donde se le preguntará si desea guardar la captura realizada, ignorarla o simplemente cancelar la operación.

Figura 20: Detener una captura

- **Registro de captura:**

El último paso del proceso de captura, consiste en el registro de los datos en el sistema Siscla. Haciendo click en "Registrar", se llevará a cabo el registro donde se autenticará al usuario y se procederá al registro de la captura correspondiente. En caso de producirse algún error durante la operación, la aplicación le permitirá guardar la captura en el dispositivo para enviarse en otro momento.

Figura 21: Registrar una captura

- **Almacenar una captura localmente:**

La interfaz de falla permite dos acciones: “reintentar” la operación que ha fallado, que generalmente corresponderá a una falla de registro; o “continuar”, que normalmente estará relacionado al resguardo de los datos en el dispositivo.

Figura 22: Almacenar una captura localmente

- **Ver estado de capturas:**

Siempre que una captura falle durante el registro, y el usuario desee guardar la captura, ésta se verá reflejada en la interfaz de “Estado de capturas”, la cual se puede acceder a través de la interfaz “Principal”. En caso de que haya alguna captura esperando por registro, el usuario puede hacer click en el botón “Sincronizar”, que iniciará el proceso de envío de las capturas al sistema de clasificación de suelos. Esta operación hace uso de la operación de registro, por lo que solicitará la autenticación del usuario antes del envío de datos correspondiente. Si alguna falla se produce durante el envío de capturas, las capturas que no hayan podido ser registradas no serán borradas del dispositivo.

Figura 23: Estado de registro de capturas

- **Sincronización exitosa:**

En caso que durante el proceso de envío de capturas no haya tenido ninguna falla, así como en otras operaciones de la aplicación, se cargará una interfaz de aviso informando del éxito de la operación. Si el usuario le da click al botón “continuar” la aplicación regresará a la interfaz “Principal”.

Figura 24: Sincronización de capturas

- **Ayuda de aplicación:**

Una de las opciones contenidas en el menú de comandos ubicado en la parte inferior de la pantalla, se trata de una interfaz de ayuda, dónde se ofrece información adicional de cómo usar la aplicación, además otras opciones adicionales, como regresar al menú “Principal” o salir de la aplicación.

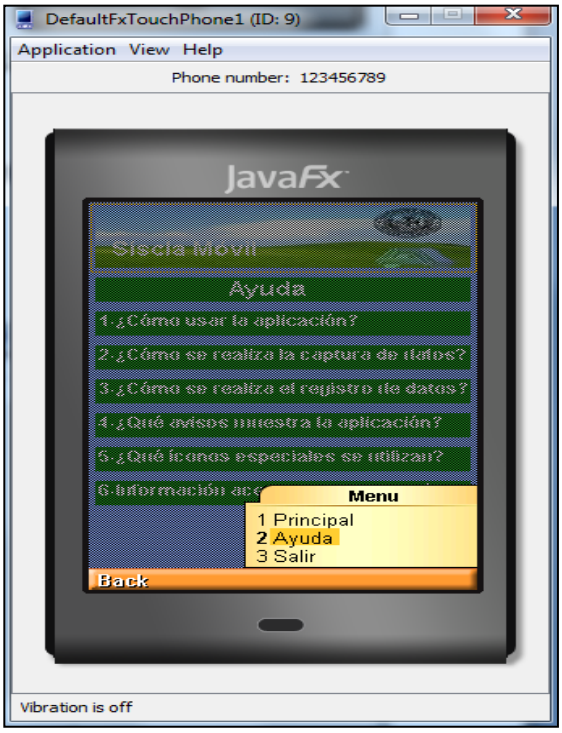


Figura 25: Ayuda

2.6) Componentes de software implementados

Adicionalmente al diseño y construcción de los agentes fue necesario modelar y construir algunos componentes de software adicionales. En este apartado se describen estos componentes organizándolos en los siguientes grupos:

- **Aplicación**: representa las clases principales de la aplicación. Se tiene una clase Midlet, que representa la entidad de ejecución de la aplicación JavaME; la clase Servicios, que proporciona el acceso a los agentes que proporcionan el funcionamiento a toda la aplicación. Vale destacar que la implementación de cada agente se realizó mediante un singleton, para controlar el número de instancia de estos agentes a una sola instancia. De esta forma se aprovechan mejor los recursos del sistema y facilita la sincronización entre los agentes. El diagrama de clases de estos componentes lo pueden ver en la Figura 26.

- **Agente-IU**: representa la clase Agente-IU junto con algunos componentes auxiliares. El Agente-IU tiene la responsabilidad de cargar las interfaces en el dispositivo, por lo que tiene acceso a la clase Midlet. Además, utiliza algunas de las interfaces implementadas en la clase Interfaz de usuario, el cual ha sido modelado mediante un singleton para reutilizar las interfaces creadas en la aplicación. El diagrama de clases de estos componentes lo pueden ver en la Figura 27.

- **Agente-Captura**: representa la clase Agente-Captura, que es el agente responsable atender todas las peticiones en la aplicación. Este agente es el responsable de la carga de interfaces relacionadas con la captura de datos haciendo uso de la clase Interfaz de usuario. Además, es el responsable de ejecutar cualquier operación en la aplicación invocando los servicios que proporcionan los demás agentes. Este agente hace uso de hilos de ejecución que sincronizados a través de la clase Hilos_app; de esta manera las operaciones se realizan en segundo plano, sin comprometer el hilo base de la aplicación. El diagrama de clases de estos componentes lo pueden ver en la Figura 28.

- **Agente-BDLocal**: representa la clase Agente-BDLocal, que es el encargado de todas las operaciones relacionadas con el almacenamiento local en el dispositivo, haciendo uso de la clase interfaz bdlocal. El diagrama de clases de estos componentes lo pueden ver en la Figura 29.

- **Agente-Conexión**: representa la clase Agente-Conexión, que es el encargado de todas las operaciones relacionadas con el almacenamiento externo de los datos de campo, haciendo uso de la clase MobileClient. La clase MobileClient es la responsable de la comunicación mediante servicios web con el sistema Siscla. El diagrama de clases de estos componentes lo pueden ver en la Figura 30.

- **ComponentesVarios-IU**: representa las clases relacionadas con la construcción de interfaces. Se incluyen las siguientes clases: interfaz de usuario, wizard, imagen, localidad y customitem. La clase interfaz de usuario es la encargada de construir todas las interfaces de la aplicación; la clase wizard, encapsula todas las interfaces relacionadas con el proceso de captura de datos; la clase imagen, es la responsable de cargar adecuadamente las imágenes usadas en la aplicación; la clase customitem, es la clase genérica que representa a todos los ítems personalizados utilizados en la construcción de las interfaces; finalmente, la clase localidad ayuda en la carga dinámica de estados, municipios y parroquias. Vale mencionar que la clase imagen y la clase interfaz de usuario se han implementado como entidades singleton, con el fin de reutilizar sus componentes, imágenes e interfaces, que han sido previamente instanciadas en la aplicación. El diagrama de clases de estos componentes lo pueden ver en la Figura 31.

- **CustomItems**: representa las clases implementadas para la personalización de los ítems utilizados en las interfaces. La primera clase que mencionaremos es la clase form, la cual representa la clase genérica utilizada en J2ME para la creación de interfaces. A este componente se le agregan los ítems y customitems de la aplicación. Un customitem es una clase genérica de J2ME que permite personalizar mediante canvas (tecnología gráfica de Java2D) los ítems en un formulario. Para esta intención se implementaron las siguientes clases: aviso_ci, banner_ci, fin_wizard_ci, inicio_ci, big_texto_ci, ítem_ci, button_ci, metafora_ci, pausa_ci, recomendacion_ci, salida_ci, titulo_wizard_ci, titulo_ci, estadocaptura_ci, capturadatos_ci; las cuales fueron reutilizadas en distintas interfaces de la aplicación. El diagrama de clases de estos componentes lo pueden ver en la Figura 32.

- **ComponentesVarios-Almacenamiento**: representa las clases relacionadas con el almacenamiento de los datos de campo. Se incluyen las siguientes clases: capturas, perfiles, interfaz_bdlocal y almacenamiento. Para manipular los datos de una captura se implementó la clase capturas y perfiles respectivamente. Para llevar un registro de las capturas almacenadas localmente se creó la clase almacenamiento. El diagrama de clases de estos componentes lo pueden ver en la Figura 33.

Una descripción más detalla del uso de estas clases se puede encontrar en el código fuente documentado de la aplicación, que se encuentra en el CD anexo entregado con este documento.

2.7) Plataforma tecnológica

El prototipo del agente móvil desarrollado en J2ME, funciona en cualquier dispositivo móvil que incorpore la máquina virtual de Java, con las siguientes características:

- Cldc 1.1 o superior.
- Midp 2.0 o superior.

La interfaz de usuario fue construida para verse apropiadamente en dispositivos con pantallas de 240x300 pixeles de resolución o superior, aunque podría verse afectada por efectos de la fragmentación.

En el CD anexo se incluye una versión de la aplicación adaptada para modelos Nokia S60 para propósitos de prueba. Adicionalmente, se ha añadido una aplicación web que simula el almacenamiento del sistema Siscla, el cual tiene las siguientes características:

- Uso de servidor web Tomcat 6, o cualquier servidor compatible con JDK 5 o superior.

- JDK 5 o superior, instalado en el servidor.

Dado que la aplicación móvil especifica la dirección web del servidor al que se conecta mediante servicios web, se recomienda seguir alguna de las dos alternativas que se mencionan continuación:

- **Primera alternativa – Recompilando la aplicación JavaME**: para hacer esto recomendamos instalar las herramientas que se especificaron en la fase de desarrollo del agente móvil. Luego, se carga el código fuente de la aplicación, se resuelve cualquier dependencia no resuelta que

probablemente se relaciona con el nombre plataforma JavaME utilizada. Luego dirigirse a Source Package -> WebClientServices -> MobileClient.java del proyecto y modificar el URL que se indica en el constructor de la clase al nuevo URL del servidor. Recompilar el código de la aplicación. Dirigirse a la carpeta “dist” del proyecto y cargar los archivos .jar y .jad en el dispositivo para su instalación; o en caso desee distribuirse en un servidor para su distribución, se modifica en el archivo JAD el url que tendrá el archivo .Jar, y se copian ambos archivos en el servidor de modo que puedan ser referenciados y descargados.

- **Segunda alternativa – Utilizando direcciones dinámicas para el servidor**: para esto se contrata un servicio DNS o se descarga una aplicación para DNS dinámico en el servidor. El nuevo dominio debe indicarse en la aplicación siguiendo los pasos indicados en la primera alternativa. Cada vez que se modifique la dirección del servidor, no hará falta recompilar la aplicación, pues solo es necesario modificar la dirección indicada en el dominio DNS.

Las instrucciones para la instalación de todas las herramientas que fueron utilizadas en el desarrollo del agente móvil, son las siguientes:

- Descargar e instalar Netbeans. Recuerde seleccionar en la instalación, la instalación de la plataforma JavaME y el servidor Tomcat 6.
- Descargar e instalar Sun Java Wireless Toolkit.
- Agregar Sun Java Wireless Toolkit a las plataformas de Java que usa Netbeans, eso lo pueden hacer en “Herramientas->Plataformas de Java”.
- Ubicar el código fuente de la aplicación en la carpeta de proyectos definida en Netbeans, y abrir el proyecto desde Netbeans.
- En caso de que el proyecto se abra con errores, es probable que sea necesario, adecuar los nombres de las plataformas de JavaME, o si se quiere, crear un nuevo proyecto y agregar solo el código fuente de la aplicación.

Nota: Todas las herramientas utilizadas y el código fuente de la aplicación lo pueden conseguir en el CD anexo que fue entregado junto a este informe.

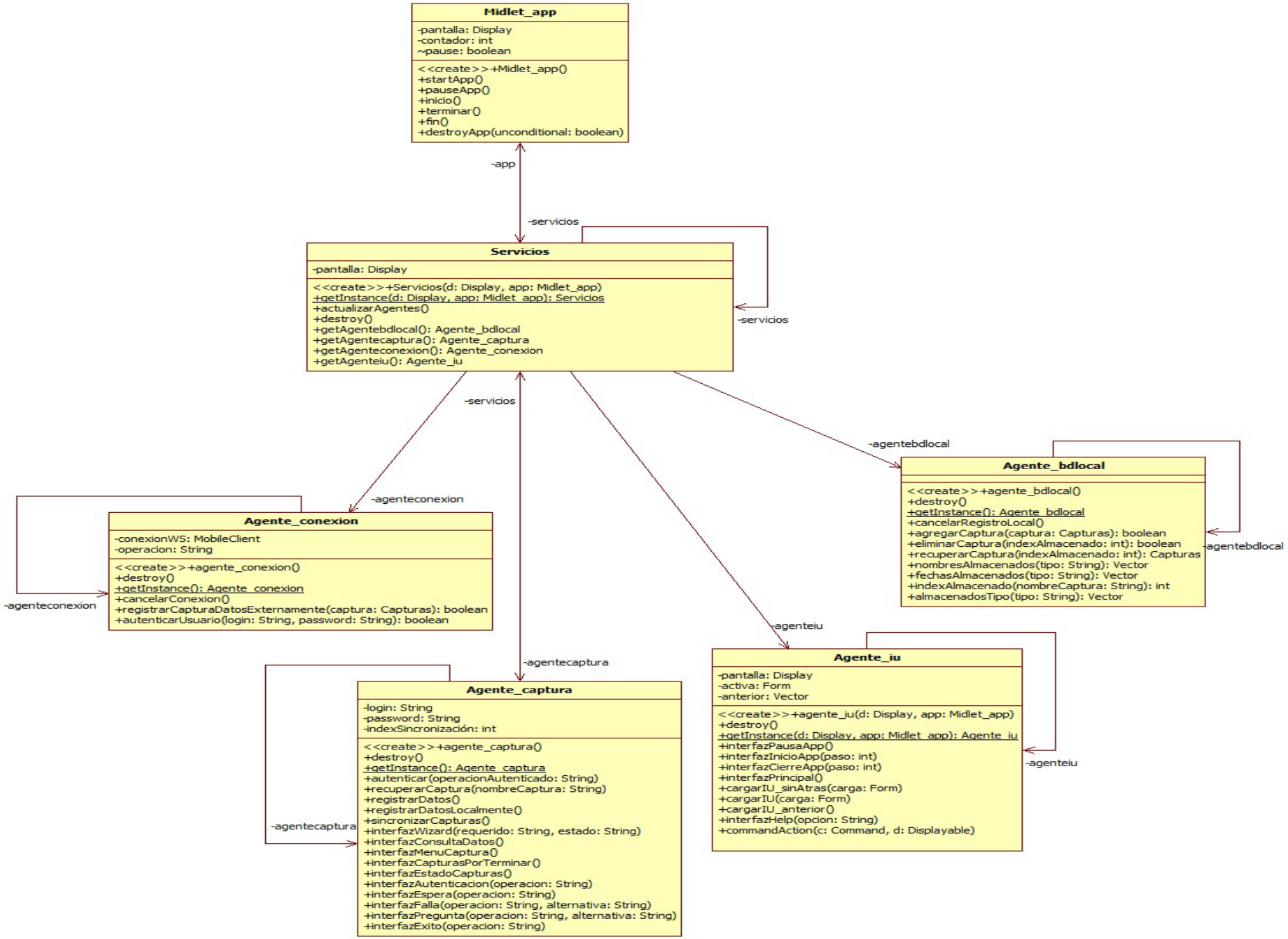


Figura 26: Aplicación

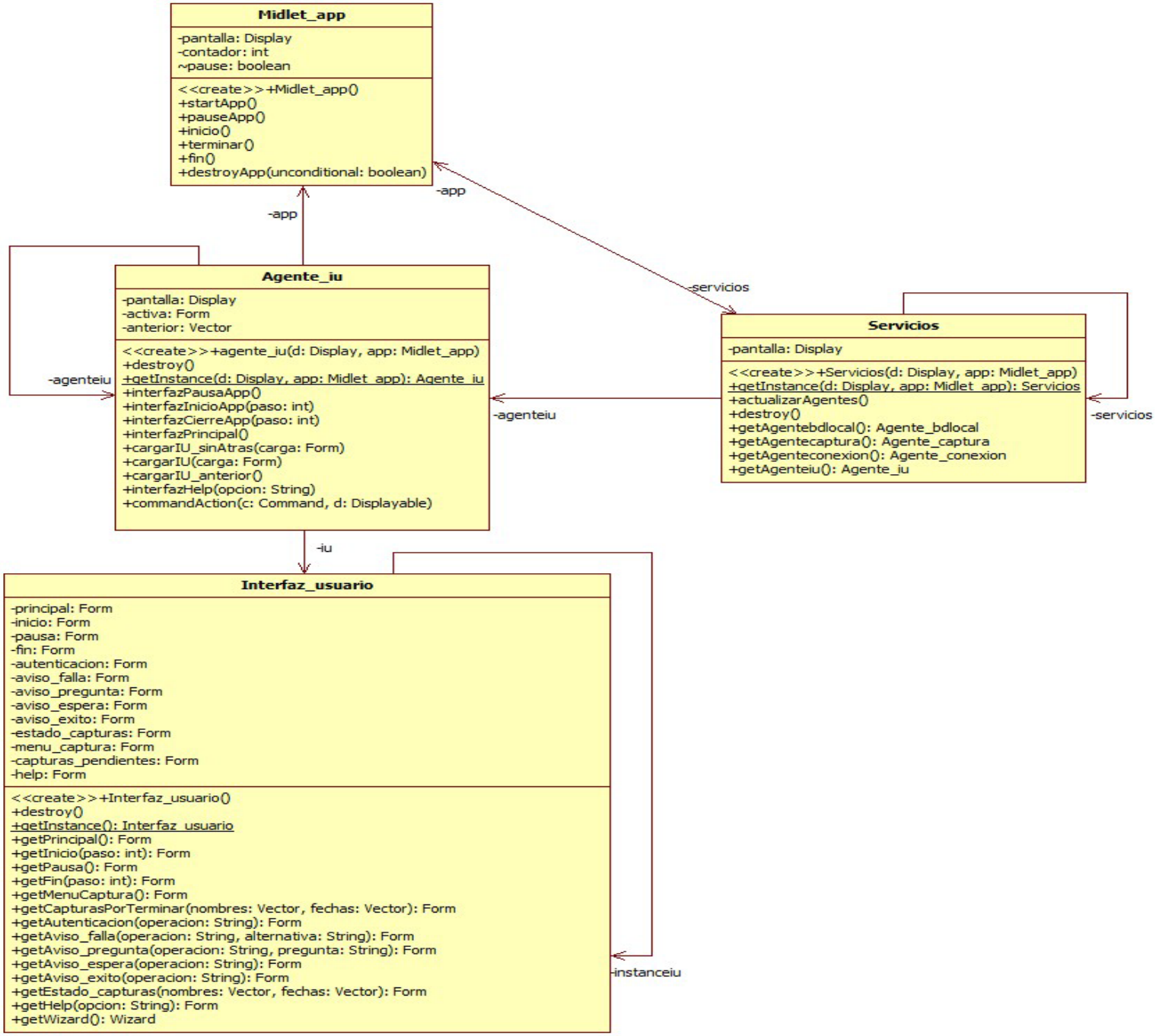


Figura 27: Agente-IU

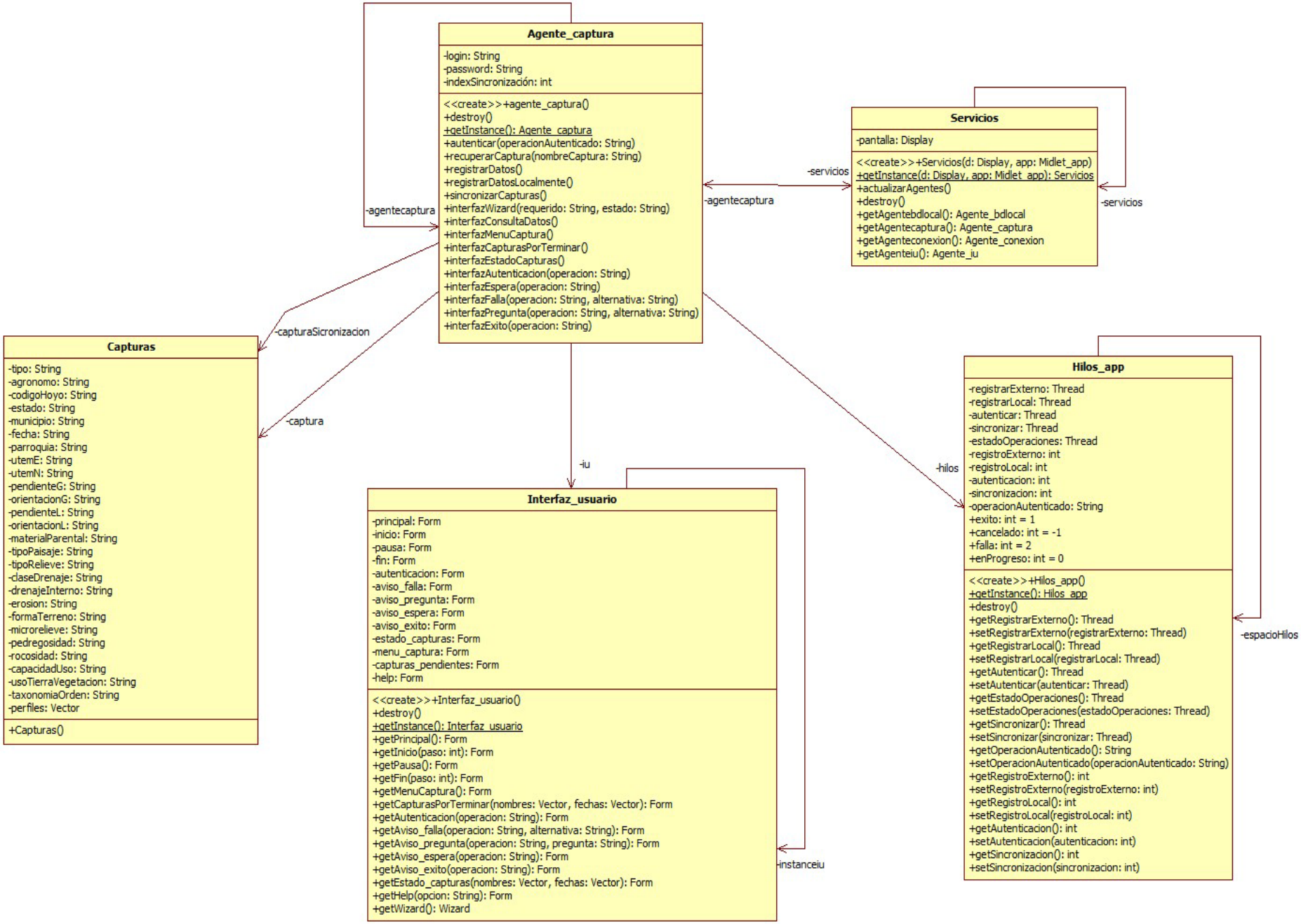


Figura 28: Agente-Captura

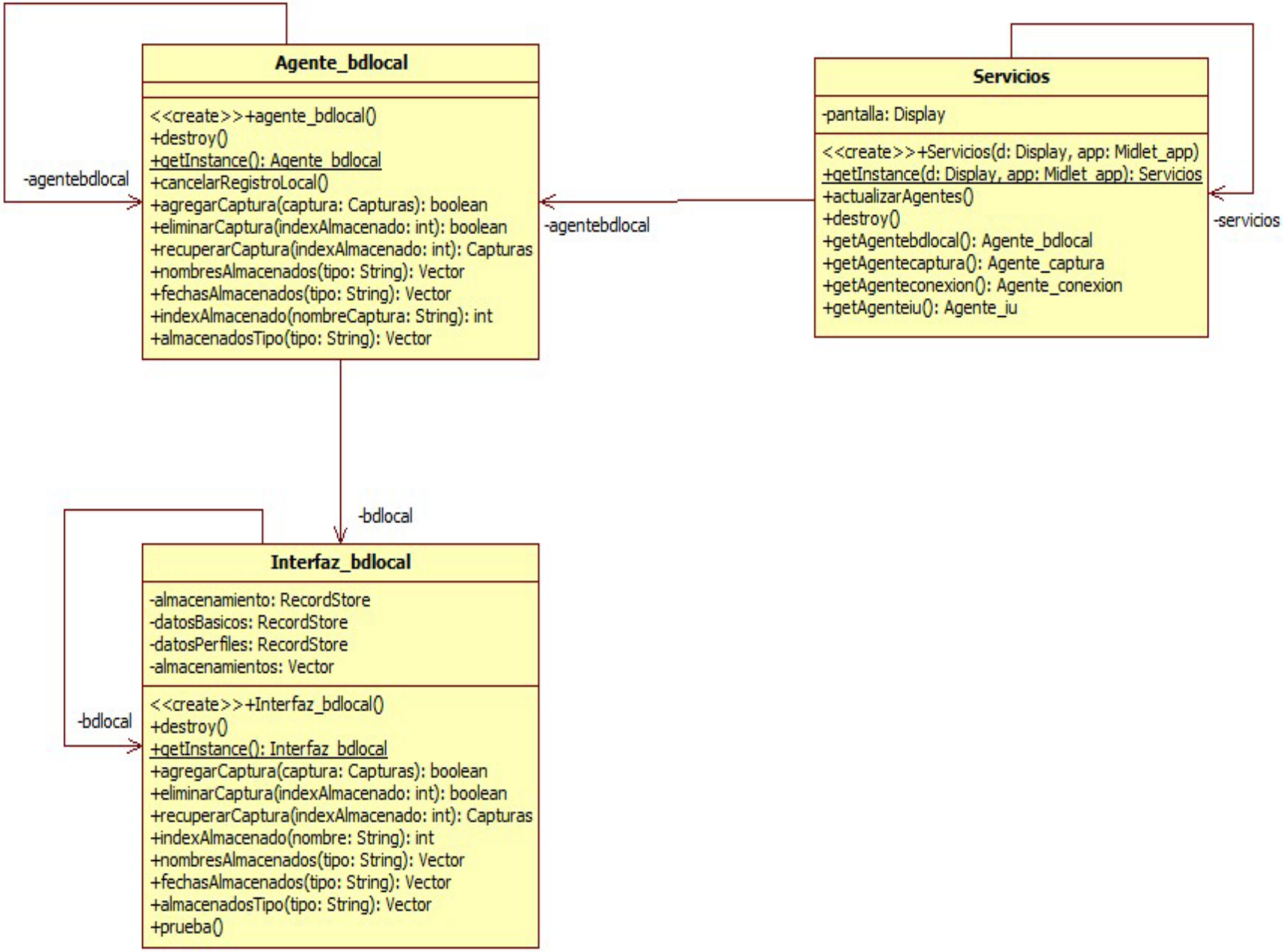


Figura 29: Agente-Bdlocal

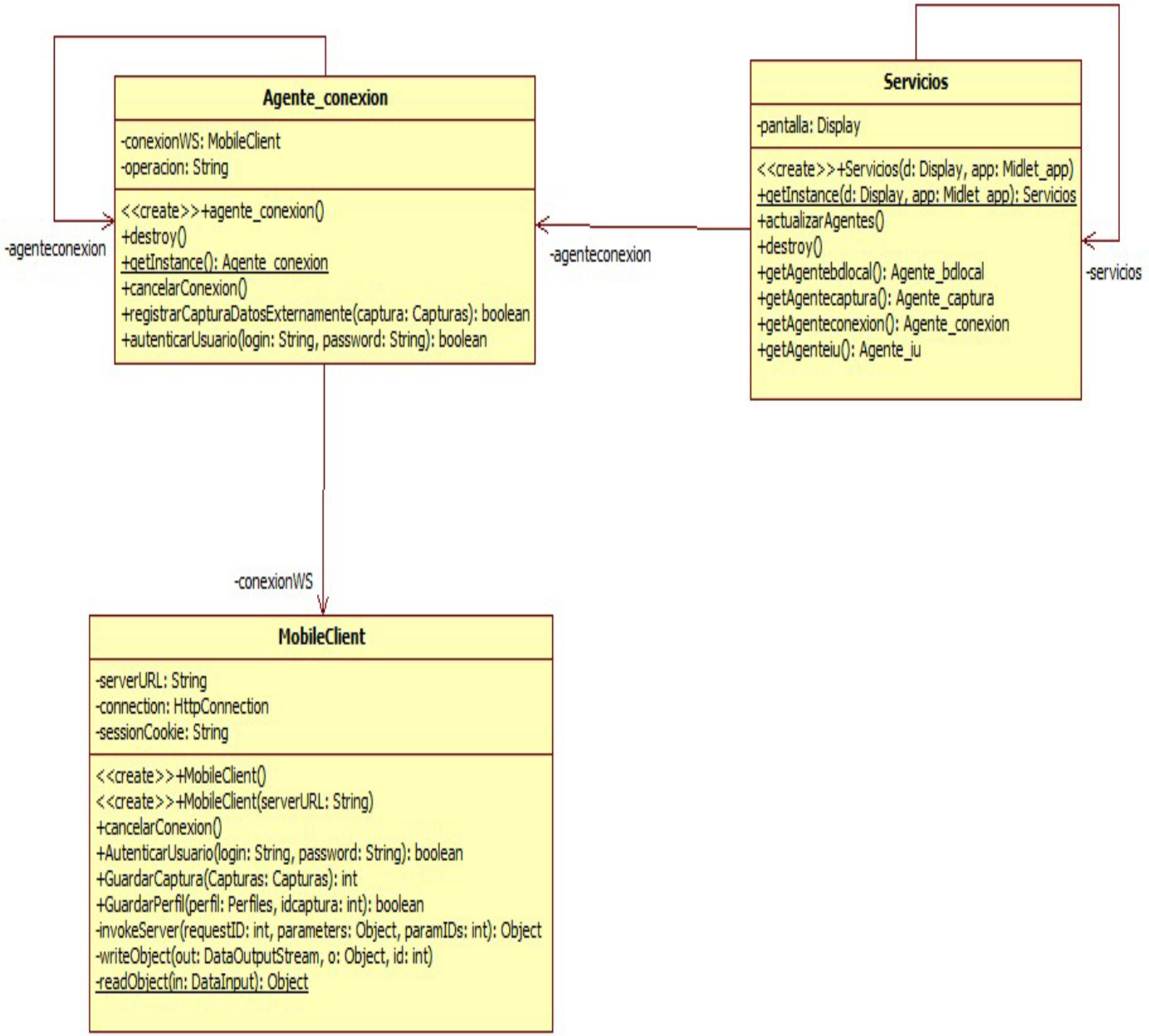


Figura 30: Agente-Conexión

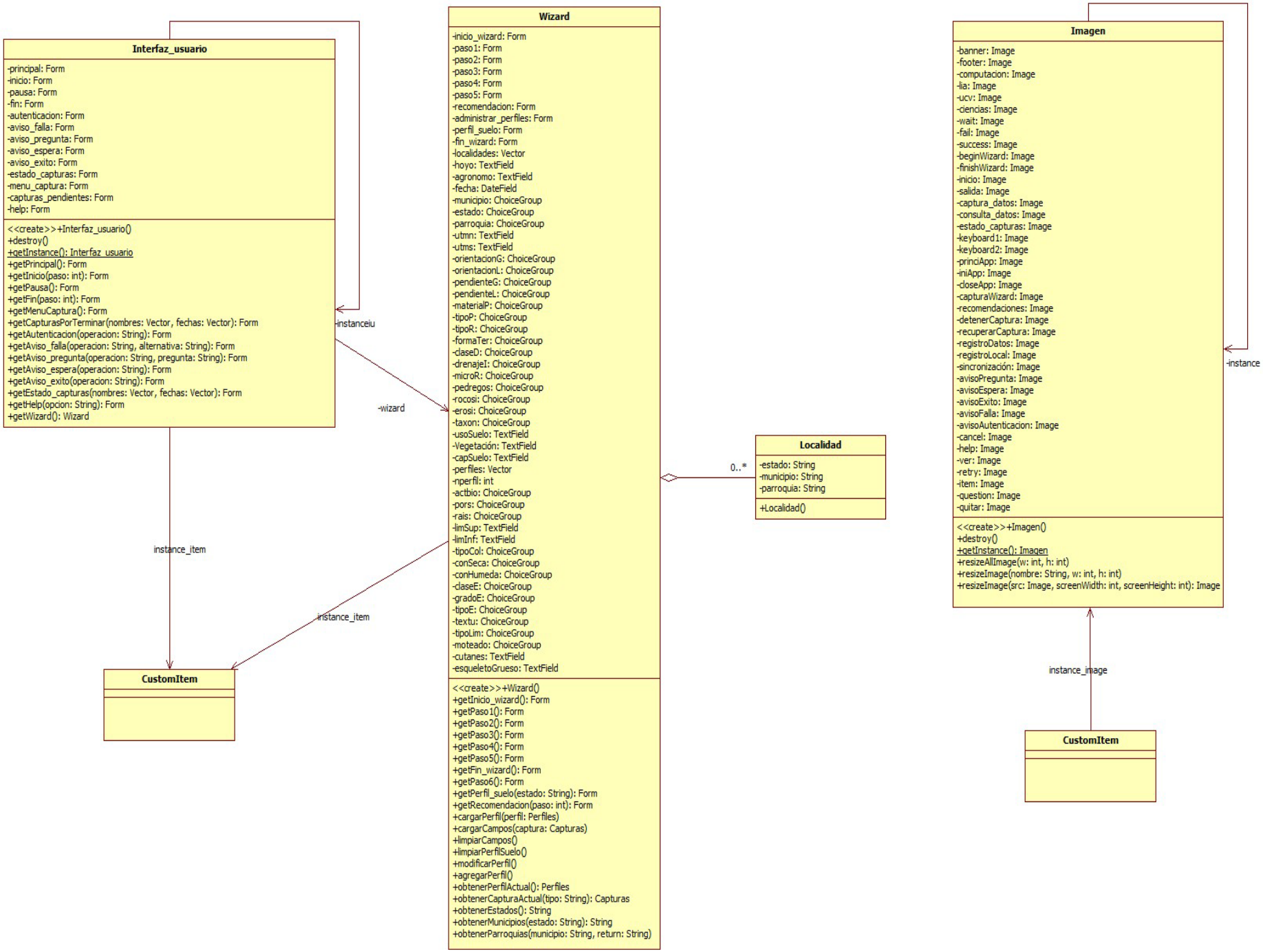


Figura 31: Componentes varios- IU

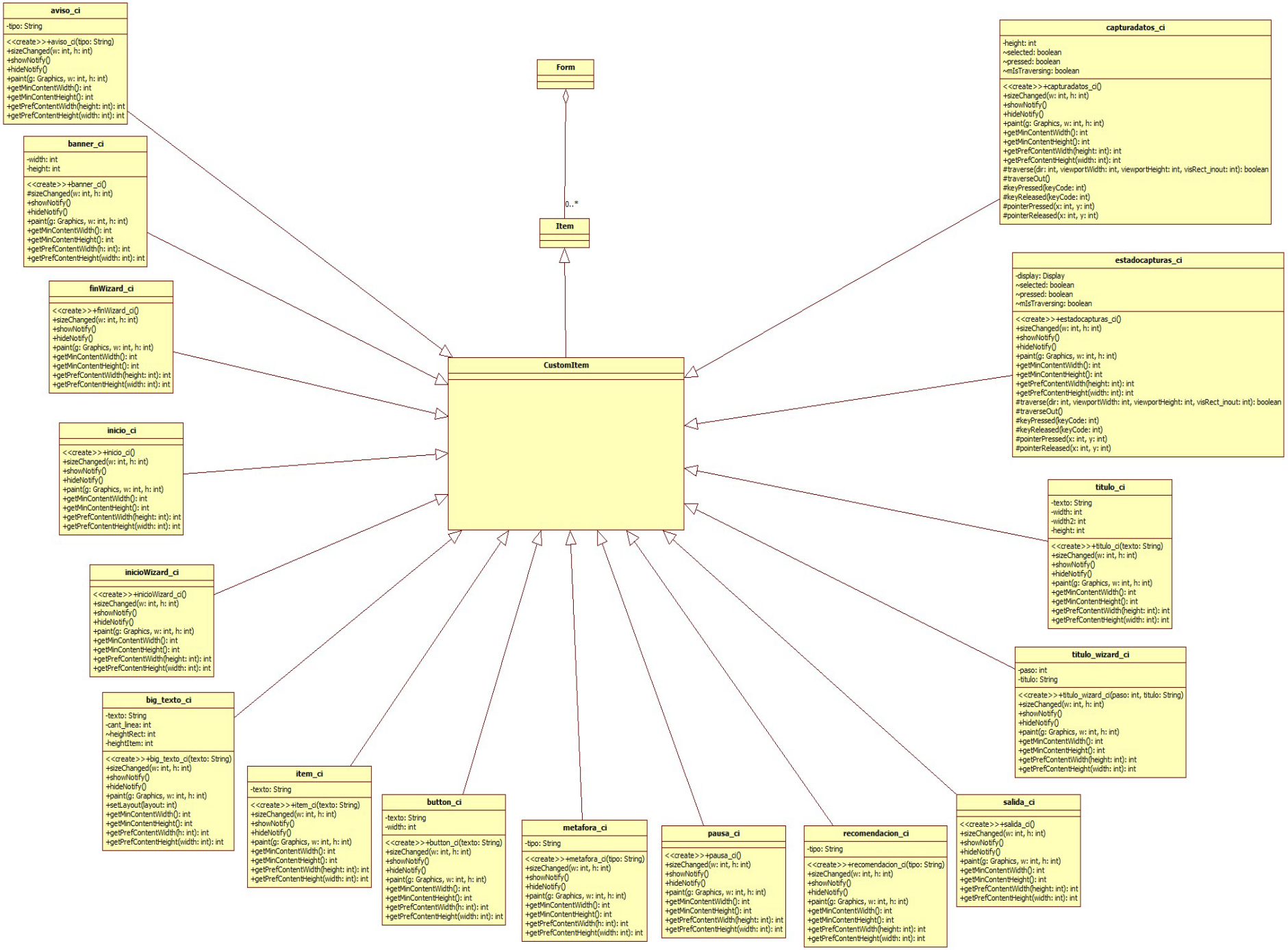


Figura 32: CustomItems

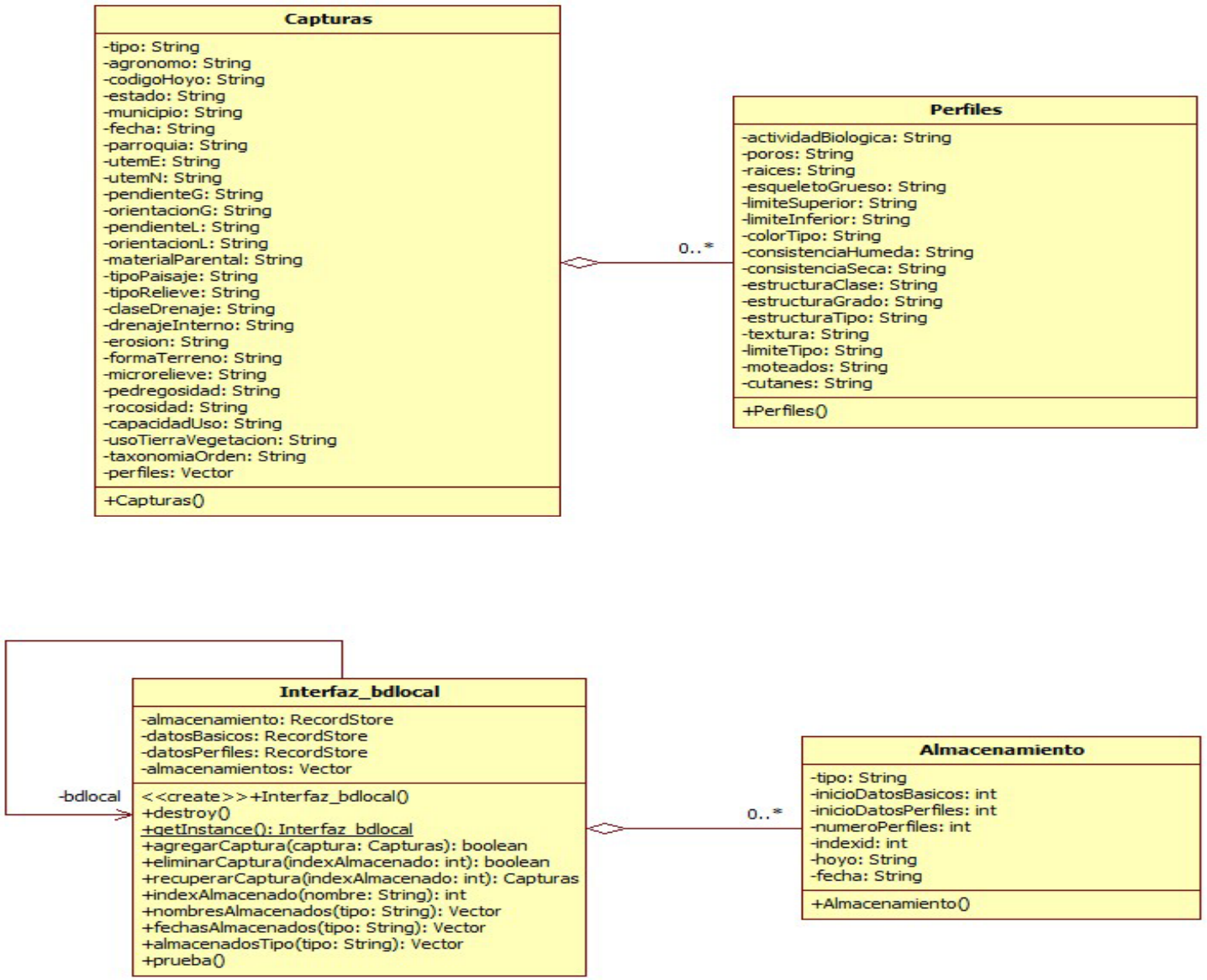


Figura 33: Componentes varios-Almacenamiento

Conclusiones

La construcción de este agente móvil que fue propósito de este trabajo, produjo los siguientes resultados y observaciones, que deben ser objeto de estudio para futuros trabajos de investigación relacionados con el área de agentes y el uso de las tecnologías de comunicaciones móviles.

Haciendo una síntesis de los logros alcanzados con este trabajo:

El agente móvil fue diseñado haciendo uso de la metodología Jade, descartando algunas de las actividades recomendadas por considerarse de poca relevancia para este proyecto.

Se realizó un modelo de conocimiento del dominio de la aplicación, con el fin de representar el conocimiento de los edafólogos utilizado en el proceso de captura de datos de campo.

Se modeló la interfaz de la aplicación haciendo uso del conocimiento del dominio, prototipos en papel, uso de patrones de interacción y una guía de estilo, para conservar los principios de consistencia con el sistema Siscla.

Se implementaron los agentes y se desarrollaron algunos componentes de software adicionales, usando la tecnología JavaME.

Finalmente se realizaron algunas pruebas del agente móvil en algunos dispositivos móviles, con el fin de comprobar la portabilidad de la aplicación.

Algunos de los aportes que contribuye la realización de este proyecto son las siguientes:

El desarrollo del agente móvil haciendo uso de la metodología JADE, tuvo repercusiones positivas en la aplicación obtenida. El diseño de los agentes puede resultar un tanto rápido e intuitivo con esta metodología, si se conocen bien los recursos de la aplicación y las responsabilidades necesarias para gestionarlos. La metodología JADE permitió distribuir la estructura del agente móvil en una sociedad de cuatro agentes, los cuales interactúan entre sí de forma indirecta mediante una instancia “Servicios”. Todos los agentes son de carácter reactivo. El agente captura cumple una función indispensable en la coordinación de los agentes, porque a través de él todas las peticiones de la aplicación son resueltas, invocando los servicios ofrecidos por los otros agentes si la petición solicitada en la aplicación así lo requiere.

Desde el punto de vista del entorno de ejecución, cada una de las operaciones de la aplicación se ejecuta en un hilo de ejecución a parte, en el que los agentes realizan sus acciones correspondientes en colaboración con los demás.

Por otro lado, el uso de la metodología presentó algunos inconvenientes. Algunas de las actividades y artefactos propuestos estaban fuertemente relacionados con el desarrollo JADE, los cuales tuvieron que ser adaptados, sustituidos y en algunos casos descartados. Además, hay fases y actividades en la metodología que carecen de artefactos, limitándose a recomendar simplemente lo que se debería hacer. Esto hace parecer a la metodología poco rigurosa y exhaustiva, lo que podría ser perjudicial en algunos proyectos de desarrollo.

Sobre el uso de la tecnología JavaME para la implementación del agente móvil surgen los siguientes comentarios.

J2ME es una plataforma que busca la estandarización, lo que contribuye al desarrollo de aplicaciones que funcionen en una mayor cantidad de dispositivo. Sin embargo el problema que presenta esta plataforma, es que no es completamente estable, lo que dificulta desarrollar aplicaciones que no se vean afectados por los efectos de la fragmentación entre dispositivos. Durante las pruebas en algunos dispositivos, se constató diferencias desde el punto de vista de interfaz, lo que sugiere que las aplicaciones desarrolladas en esta plataforma, necesitan un mantenimiento adicional para garantizar el comportamiento adecuado de la aplicación en nuevos dispositivos.

Recomendaciones

En este apartado se mencionan algunas ideas y recomendaciones que pudieran ser aprovechadas en futuras investigaciones relacionadas con este trabajo.

La definición de los agentes usando la metodología JADE fue especialmente útil para definir la arquitectura de la aplicación, por lo que se recomienda su uso en otros proyectos basados en agentes inteligentes. Además, se puede usar este trabajo como referencia de la aplicación de esta metodología, con el objeto de compararla con otras metodologías.

Referente al uso de las comunicaciones móviles, se recomienda incorporar un área de investigación dedicado a este tipo de tecnologías, pues los problemas técnicos y de ingeniería de software que caracterizan el desarrollo de estas aplicaciones, pueden atraer el interés y la atención académica para la generación de nuevos conocimientos.

En el caso del uso de la tecnología JavaME, es importante definir en etapas tempranas de desarrollo, el dispositivo o clase de dispositivos donde se espera utilizar la aplicación. Esto con el fin de adquirir las herramientas y el adiestramiento necesarios para el desarrollo de la aplicación respectiva. Esta recomendación contribuye a reducir el impacto que tendría la fragmentación durante el proceso de desarrollo.

Con respecto al agente móvil desarrollado, se recomienda específicamente lo siguiente:

- Definir un dispositivo, o clase de dispositivos que podrían ser utilizados por los usuarios, y adaptar los cambios pertinentes de interfaz en la aplicación para su uso. La plataforma Nokia presenta algunos problemas en la implementación Midp 2.0, por lo que se recomienda seguir investigando en la plataforma más adecuada para esta aplicación.
- Integrar concretamente el agente móvil con el agente de gestión de base de datos, lo que seguramente necesitará de una revisión concerniente al modelo de conocimiento del agente móvil y la comunicación entre estos dos agentes.
- Añadirle nuevas funcionalidades a la aplicación, por ejemplo, la consulta de datos de campo desde el dispositivo móvil.

Adicionalmente, durante una presentación de la aplicación en presencia del profesor Jesús Viloria, se hicieron las siguientes observaciones para la aplicación:

- Permitir una carga de datos por defecto al realizar una nueva captura, tomando en cuenta los datos en capturas anteriores, por ejemplo: la información básica del hoyo y los datos de identificación del especialista.
- Se recomendó añadir al modelo de conocimiento y a los campos en el proceso de captura, un valor “otro”, en caso de que ninguno de los valores especificado en la aplicación se ajusten a la observación del edafólogo.
- Se recomendó colocar los campos de coordenadas UTM relacionadas a la ubicación del hoyo como obligatorios. Además, se recomendó investigar la posibilitad de utilizar el GPS de los dispositivos, para la carga automática de estos campos, conservando la posibilidad de que puedan ser modificados por el edafólogo.
- Permitir distintas formas para la descarga de las captura de datos, como por ejemplo: bluetooth, conexión con cable, archivos, etc. Con el fin de proporcionar distintas vías para la recuperación de los datos capturados.

Finalmente, se comentó la posibilidad de estudiar el desarrollo de un sistema para la Facultad de Agronomía, aprovechando las investigaciones relacionadas con el sistema Siscla.

Referencias

Acosta, G. P. (2005). Estado del Arte Agentes Informáticos, Sistemas Multi-agentes informáticos e Inteligencia Artificial. Colombia: Pontificia Universidad Javeriana. Disponible en: http://pegasus.javeriana.edu.co/~groupage/archivos/entregables/estado_arte/IA/ESTADOARTE_V0.7_IA.pdf

Angulo, T. y Vásquez J. (2009). Organización, Gestión y Análisis de Datos de Levantamiento de Suelos Mediante un Agente de Software". Trabajo Especial de Grado, Universidad Central de Venezuela, pp. 5-17, 37-41.

Balcazar, G V. (2009). Agentes Inteligentes: El siguiente paso en la Inteligencia Artificial. Bolivia: Universidad Pública del alto. Disponible en: http://bravokcha.blogspot.com/

Brooks, R. A. (1991). Intelligence whitout representation. Artificial Intelligence, pp 139-159.

Caire, G. , Nikrazla, M. y Bahria, P. (2006). A Methodology for the Analysis and Design of Multi-Agent Systems using Jade. Australia: Universidad de Murdoch. Disponible en: http://jade.tilab.com/doc/tutorials/Jade_methodology_website_version.pdf

Castro, A. , Castro, N. , García, G. , Hernandéz, M. y Torres, M. (2005). Agentes Inteligentes. México: Instituto Tecnológico de Nuevo Laredo. Disponible en: http://www.itnuevolaredo.edu.mx/maestros/sis_com/takeyas/Apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/Agentes_Inteligentes/Agentes_Inteligentes(2005-II).pdf

Colaboradores de Wikipedia. (2009). Smartphone. Wikipedia la Enciclopedia Libre. Disponible en: http://es.wikipedia.org/wiki/Tel%C3%A9fono_inteligente

CompendiumInstitute. (2010). Descargas. Disponible en: http://compendium.open.ac.uk/institute/download/download.htm

Cysco System (2000). GPRS White Paper. Cysco System Inc. Disponible en: http://www.cisco.com/warp/public/cc/so/neso/gprs/gprs_wp.pdf

Disnal, E. y Griparis, T. (2006). High-speed Packet Access Technologies for UMTS networks. Bechtel Corporation. Disponible en: http://www.bechtel.com/communications/assets/files/TechnicalJournals/June2006/Article09.pdf

Eclipseme.org (2009). EclipseME. http://www.eclipseme.org derechos reservados. Disponible en: http://eclipseme.org/docs/index.html

García, A. , Solarte, Z. M. , Castillo, C. y Vásquez E. (2005). Agentes en Computación Móvil. Colombia: Universidad Autónoma de Occidente, Grupo de Investigación en Telemática. Disponible en: http://www.umanizales.edu.co/programs/ingenieria/ventana/ventana12/articulo12.pdf

Georgeff, M. P. y Lansky, A. L. (1987). Reactive reasoning and planning. USA: Proceedings of the Sixth National Conference on Artificial Intelligence, pp. 677-682.

Hayzelden, A. L. y Bigham, J. (1999). Software Agents for Future Communications Systems. Springer-Verlag. Abril, 1999. ISBN 3-540-65578-6

Herrera, V.V. , Bepperling, A. , Lobov, A. , Smit, H. , Colombo, A.W. y Lastra, J. (2008). Integration of Multi-Agent Systems and Service-Oriented Architecture for industrial automation. Finlandia: Tampere University of Technology. Disponible en: http://ieeexplore.ieee.org/xpl/RecentCon.jsp?punumber=4603797

Hípola, P y Vargas, B. (1999). Agentes inteligentes: definición y tipología. Los agentes de información. España: Universidad de Granada. Disponible en: http://www.elprofesionaldelainformacion.com/contenidos/1999/abril/agentes_inteligentes_definicion_y_tipologia_los_agentes_de_informacion.html

Java Development Framework. (2010). Java Development Framework is a Open Source Platform for peer-to-peer agents based applications. Disponible en: http://jade.tilab.com/

Knudsen, J. (2007). Kicking Butt with MIDP and MSA. Sun Microsystems Inc, pp. 3-45.

Miembros J2ME Grasia (2009). JavaME Grasia: MIDP. España: Universidad Complutense de Madrid. Disponible en: http://grasia.fdi.ucm.es/j2me/_J2METech/MIDP.html

Motorola (2006). CDMA 2000 EV-DO Revision B. Motorola Inc. Disponible en: http://www.motorola.com/staticfiles/Business/Solutions/Industry%20Solutions/Service%20Providers/Wireless%20Operators/Cellular%20Networks/CDMA/RAN/_Documents/static%20files/cdma-dorb-paper.pdf?localeId=33

Netbeans.org. (2009). Java ME and JavaFX Mobile Application Technology Trail. http://www.netbeans.org All rights reserved. Disponible en: http://www.netbeans.org/kb/trails/mobility.html

Shoham, Y. (1993). Agent-oriented programming. Artificial Intelligence, Vol. 60, Nº. 1, pp. 51-92.

Sun Microsystems(2009a). JavaME Technology. Sun Microsystems Inc. Disponible en: http://java.sun.com/javame/technology/index.jsp

Sun Microsystems(2009b). The Mobile Service Architecture Specification. Sun Microsystems Inc. Disponible en: http://developers.sun.com/mobility/midp/articles/msaintro/

Sun Microsystems (2009c). JavaME Emulator Tookits. Sun Microsystems Inc. Disponible en: http://java.sun.com/javame/sdk/index.jsp

Tanenbaum, A. (1997). Redes de Computadoras. Mexico. Prentice-Hall Hispanoamericana, 3a edition.

the UMTS Forum (2003). MOBILE EVOLUTION Shaping the Future. UMTS FORUM. Disponible en: http://www.umts-forum.org/component/option,com_docman/task,doc_download/gid,1634/

Viloria, A. , Viloria, J. y Nuñez, H. (2008). Aplicación de técnicas de inteligencia artificial para la estructuración de modelos de clasificación de paisajes y predicción de atributos de suelos a partir de imágenes satelitales y modelos digitales de elevación. Jornadas de investigación y extensión. Facultad de Ciencias. Universidad Central de Venezuela.

Viloria, J. , Estrada, C. y J. C. Rey. (1998). SISDELAV: Sistema de información de suelos de la depresión del lago de Valencia. Venesuelos, 6: pp. 2 - 9.

W@p Forum (2002). Wireless Application Protocol WAP 2.0. Wireless Application Protocol Forum Ltd. Disponible en: http://www.wapforum.org/what/WAPWhite_Paper1.pdf

Anexos

Los siguientes artefactos fueron omitidos en el marco aplicativo debido a su extensión:

1. Tablas de responsabilidades

Tipo de agente	Id_resp.	Nombre	Descripción
Agente-Captura	1	Registrar datos de campo	Consiste en registrar los datos de una captura en el sistema de Siscla.
	2	Registrar datos de campo localmente	Consiste en registrar los datos de una captura en el almacenamiento local del dispositivo.
	3	Recuperar datos de campo	Consiste en recuperar una captura del almacenamiento local del dispositivo móvil.
	4	Autenticar usuario	Consiste en verificar la autenticación con el sistema de clasificación de suelos.
	5	Sincronizar capturas	Consiste en registrar las capturas que no se han podido registrar en el sistema de Siscla.
	6	Generar interfaz de menú captura	Es la interfaz que permite escoger entre una nueva captura o completar una captura anterior.
	7	Generar interfaz de capturas por terminar	Es la interfaz que permite listar las capturas anteriores que no se han completado.
	8	Generar interfaz para ingresar datos de campo	Es la interfaz por la cual se ingresan los datos de campo.
	9	Generar interfaz de estado de capturas	Es la interfaz que permite listar las capturas pendientes para ser registradas en Siscla.
	10	Generar interfaz para autenticación	Es la interfaz por la que el usuario autentica a la aplicación.
	11	Generar interfaz de falla	Es la interfaz que reporta el error producido por una operación determinada.
	12	Generar interfaz de espera	Es la interfaz que reporta la espera de una operación determinada
	13	Generar interfaz de éxito	Es la interfaz que reporta la culminación exitosa de una operación determinada
	14	Generar interfaz de pregunta	Es la interfaz que permite la consulta al usuario, sobre una operación en proceso

Tipo de agente	Id_resp.	Nombre	Descripción
Agente-BDLocal	1	Registrar captura localmente	Consiste el registrar los datos de captura temporalmente en la memoria del dispositivo.
	2	Recuperar captura localmente	Consiste en recuperar los datos de captura almacenados en el dispositivo.
	3	Eliminar captura localmente	Consiste en borrar datos registrado en la memoria del dispositivo
	4	Cancelar registro local	Consiste en abortar una operación de registro de almacenamiento local.
	5	Obtener nombres de capturas almacenadas	Consiste en obtener los nombres de las capturas almacenadas.
	6	Obtener fechas de capturas almacenadas	Consiste en obtener las fechas de las capturas almacenadas.
	7	Obtener el índice de captura almacenada	Consiste en obtener el index de almacenamiento una captura.
	8	Obtener capturas de acuerdo al tipo	Consiste en obtener un conjunto de capturas de acuerdo a su tipo.

Tipo de agente	Id_resp.	Nombre	Descripción
Agente-GestiónBD	1	Registrar la captura de datos móvil	Consiste en recibir y almacenar los datos de captura desde un dispositivo móvil, dentro del sistema de clasificación de suelos.
	2	Autenticar usuario	Consiste en validar los datos de autenticación de un especialista desde un dispositivo móvil.

Tipo de agente	Id_resp.	Nombre	Descripción
Agente-Conexión	1	Registrar captura externamente	Consiste en registrar los datos asociados a una captura en el sistema de Siscla.
	2	Autenticar usuario	Consiste en autenticar los datos de un usuario en el sistema con el sistema de Siscla.
	3	Cancelar conexión	Consiste en abortar una operación con el sistema de Siscla.

Tipo de agente	Id_resp.	Nombre	Descripción
Agente-IU	1	CargarIU	Consiste en cargar una interfaz de usuario, en el contenedor visual de la aplicación (Midlet).
	2	CargarIU_anterior	Consiste en carga la interfaz que fue cargada previamente por la aplicación.
	3	Presentar interfaz de ayuda	Es la interfaz que proporciona ayuda al usuario para el uso de la aplicación
	4	Presentar interfaz de bienvenida	Es la interfaz inicial de la aplicación
	5	Presentar interfaz de salida	Es la interfaz mientras se cierra la aplicación
	6	Presentar interfaz de pausa	Es la interfaz cuando la aplicación pasa a estar en segundo plano por algún evento.

2. Especificación de interacciones

Agente-IU

Interacción	Resp. (Id)	PI	Rol	Con	Cuando
CargaIUIngresarDatos	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Se inicia la captura de datos
CargaIUCapturasPorTerminar	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se listan las capturas anteriores que no se han completado.
CargarIUMenuCapturas	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando el usuario elige sobre realizar una nueva captura o completar una anterior.
CargaIUAutenticación	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	La aplicación valida la identidad del usuario
CargaIUFalla	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se produce algún error en la aplicación
CargaIUEspera	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Mientras se espera el resultado de una operación
CargaIUExito	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se procesa con éxito una operación
CargaIUPregunta	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se consulta al usuario sobre una operación.
CargaIUEstadoCapturas	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando el usuario desea saber cuáles capturas de datos no han sido cargadas en el sistema externo.

Agente-gestiónBD

Interacción	Resp. (Id)	PI	Rol	Con	Cuando
Registro de datos externamente	1	Solicitud (Request)	Efector (Responser)	Agente-Conexión	El agente captura quiere registrar los datos
Validación de autenticación	2	Solicitud (Request)	Efector (Responser)	Agente-Conexión	Cuando se validan la identidad del usuario, a través de los datos suministrados

Agente-BDLocal

Interacción	Resp. (Id)	PI	Rol	Con	Cuando
Registro de datos localmente	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se requiere guardar una captura localmente
Eliminación de datos localmente	3	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se requiere eliminar una captura del almacenamiento local.
Recuperación de datos	2	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se necesita recuperar una captura desde el almacenamiento local.
Nombres de capturas almacenadas	4	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se recuperan los nombres de las capturas almacenadas localmente.
Fechas de capturas almacenadas	5	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se recuperan las fechas de las capturas almacenadas localmente.
Índice de captura	6	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se verifica el índice de una captura almacenada.
Capturas por tipo	7	Solicitud (Request)	Efector (Responser)	Agente-Captura	Cuando se recuperan capturas almacenadas localmente de acuerdo a su tipo.
Operación cancelada	8	Solicitud (Request)	Efector (Responser)	Agente-Captura	Al cancelarse una operación de almacenamiento interno.

Agente-Captura

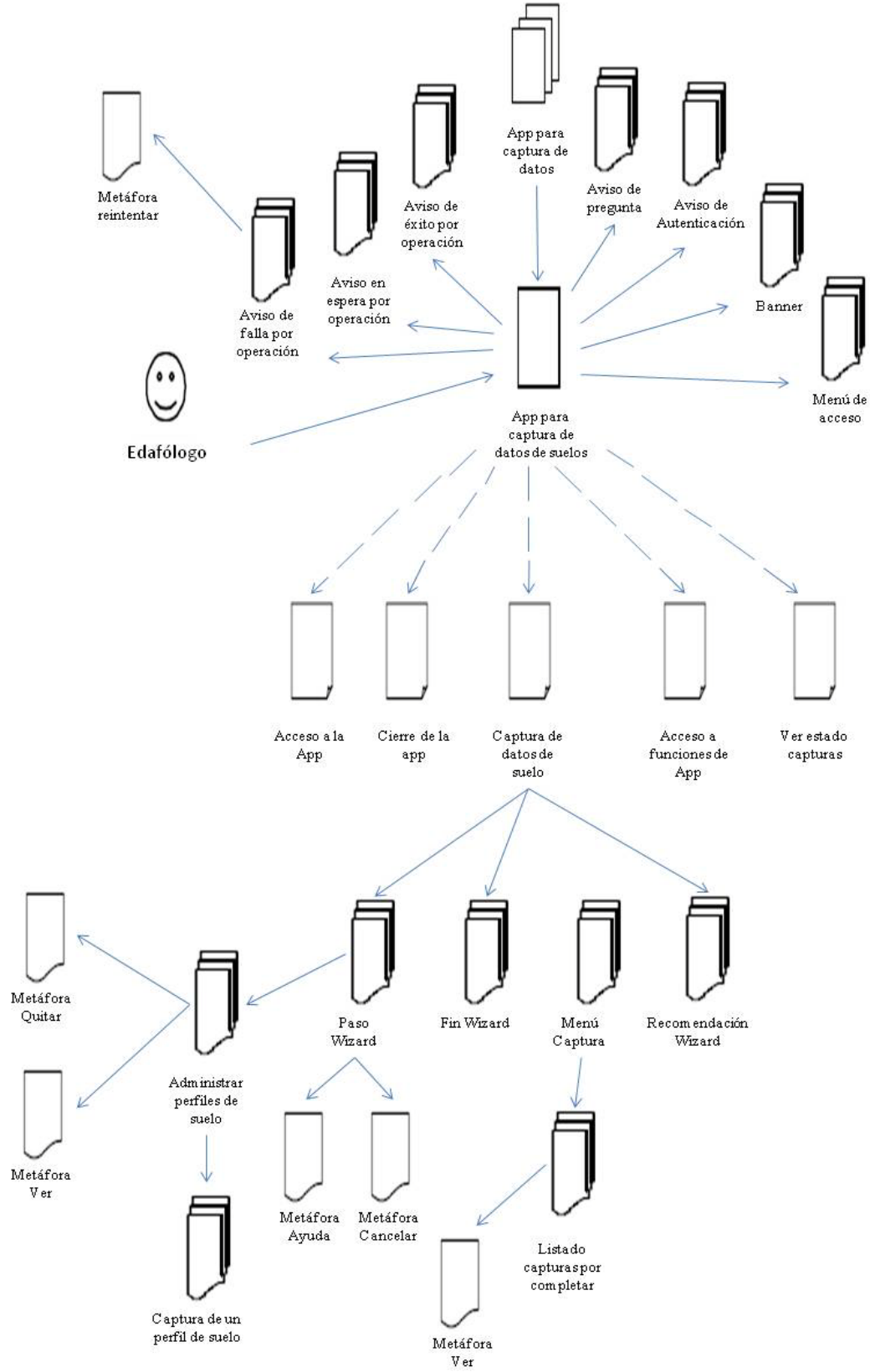
Interacción	Resp. (Id)	PI	Rol	Con	Cuando
CargaIUIngresarDatos	6	Solicitud (Request)	Iniciador (Initiator)	Agente-IU	-Cada vez que el usuario avanza en los pasos de captura de datos. -Se genera alguna asistencia o evento de información en la captura.
CargaIUCapturasPorTerminar	7	Solicitud (Request)	Iniciador (Initiator)	Agente-IU	Cuando se listan las capturas anteriores que no se han completado.
CargarIUMenuCapturas	8	Solicitud (Request)	Iniciador (Initiator)	Agente-IU	Cuando el usuario elige sobre realizar una nueva captura o completar una anterior.
CargaIUAutenticación	9	Solicitud (Request)	Iniciador (Initiator)	Agente-IU	La aplicación valida la identidad del usuario
CargaIUFalla	10	Solicitud (Request)	Iniciador (Initiator)	Agente-IU	Cuando se produce algún error en la aplicación
CargaIUEspera	11	Solicitud (Request)	Iniciador (Initiator)	Agente-IU	Mientras se espera el resultado de una operación
CargaIUExito	12	Solicitud (Request)	Iniciador (Initiator)	Agente-IU	Cuando se procesa con éxito una operación
CargaIUPregunta	13	Solicitud (Request)	Iniciador (Initiator)	Agente-IU	Cuando se consulta al usuario sobre una operación.
CargaIUEstadoCapturas	14	Solicitud (Request)	Iniciador (Initiator)	Agente-IU	Cuando el usuario desea saber cuáles capturas de datos no han sido cargadas en el sistema externo.
Registro de datos externamente	1	Solicitud (Request)	Iniciador (Initiator)	Agente-Conexion	Se desea registrar los datos en el sistema de clasificación de suelos.
Registro de datos localmente	2	Solicitud (Request)	Iniciador (Initiator)	Agente-BDLocal	Cuando se requiere guardar una captura localmente
Eliminación de datos localmente	1,2	Solicitud (Request)	Iniciador (Initiator)	Agente-BDLocal	Cuando se requiere eliminar una captura del almacenamiento local.
Recuperación de datos	3	Solicitud (Request)	Iniciador (Initiator)	Agente-BDLocal	Cuando se necesita recuperar una captura desde el almacenamiento local.
Nombres de capturas almacenadas	8,14	Solicitud (Request)	Iniciador (Initiator)	Agente-BDLocal	Cuando se recuperan los nombres de las capturas almacenadas localmente.
Fechas de capturas almacenadas	8,14	Solicitud (Request)	Iniciador (Initiator)	Agente-BDLocal	Cuando se recuperan las fechas de las capturas almacenadas localmente.

Interacción	Resp. (Id)	PI	Rol	Con	Cuando
Índice de almacenado	2	Solicitud (Request)	Iniciador (Initiator)	Agente-BDLocal	Cuando se verifica el índice de una captura almacenada.
Capturas por tipo	5	Solicitud (Request)	Iniciador (Initiator)	Agente-BDLocal	Cuando se recuperan capturas almacenadas localmente de acuerdo a su tipo.
Autenticación de usuario	4	Solicitud (Request)	Iniciador (Initiator)	Agente-Conexión	Al realizarse una autenticación de usuario.
Conexión cancelada	1	Solicitud (Request)	Iniciador (Initiator)	Agente-Conexión	Al cancelarse una operación de almacenamiento externo.
Operación cancelada	2	Solicitud (Request)	Iniciador (Initiator)	Agente-BDLocal	Al cancelarse una operación de almacenamiento interno.

Agente-Conexión

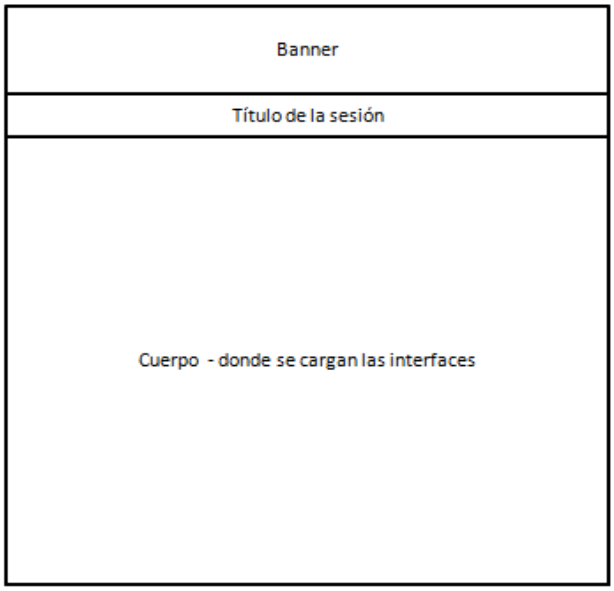
Interacción	Resp. (Id)	PI	Rol	Con	Cuando
Registro de datos externamente	1	Solicitud (Request)	Efector (Responser)	Agente-Captura	Se desea registrar los datos en el sistema de clasificación de suelos.
Autenticación de usuario	2	Solicitud (Request)	Efector (Responser)	Agente- Captura	Al realizarse una autenticación de usuario.
Registro de datos externamente (Siscla)	1	Solicitud (Request)	Efector (Responser)	Agente-gestiónBD	Se desea registrar los datos en el sistema de clasificación de suelos.
Autenticación de usuario (Siscla)	2	Solicitud (Request)	Efector (Responser)	Agente-gestiónBD	Al realizarse una autenticación de usuario.
Conexión cancelada	3	Solicitud (Request)	Efector (Responser)	Agente- Captura	Al cancelarse una operación de almacenamiento externo.

3. Prototipos de interfaz

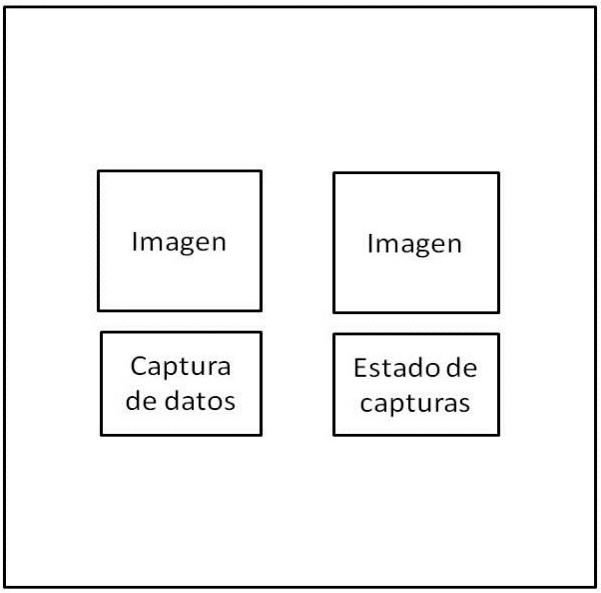


Bocetos en papel

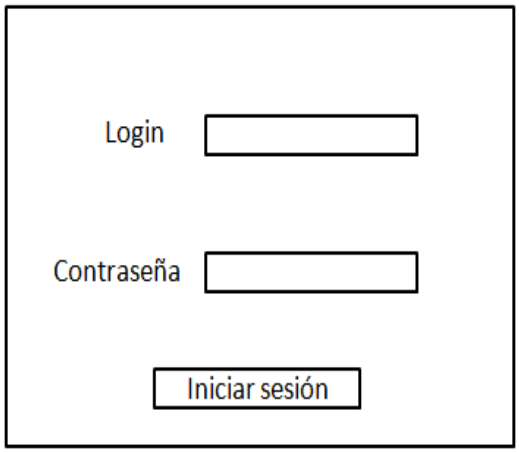
Este boceto refleja el patrón de sistema utilizado en la aplicación. Todas las interfaces se caracterizan por tener la siguiente estructura. Un banner con el logotipo de la aplicación, el título de la sesión en que se encuentra y el cuerpo que contiene el contenido de la sesión.



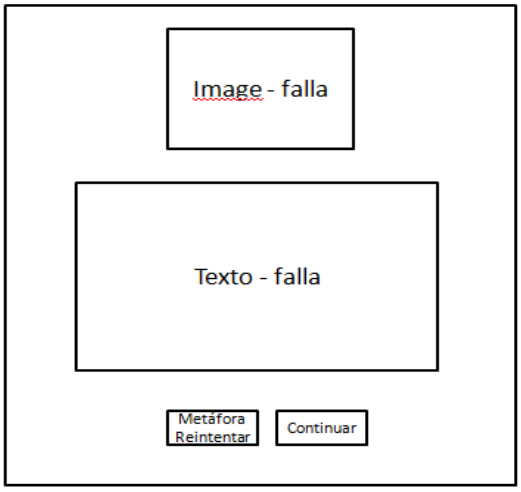
Este boceto refleja el mecanismo de interación utilizado en el menú principal. Las funcionalidades básicas se agregan en forma de columnas representados por una metáfora visual y el nombre de la sesión. Al hacer click en esta especie de botón personalizado, se carga la interfaz correspondiente a la sesión solicitada.



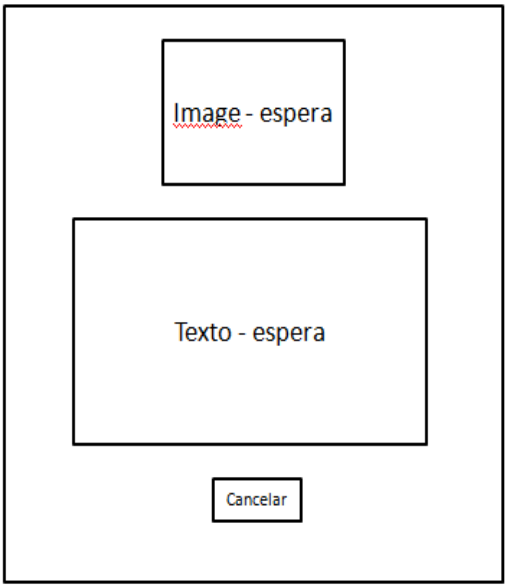
Este boceto refleja el patrón de autenticación utilizado para la autorización del usuario desde el sistema Siscla.



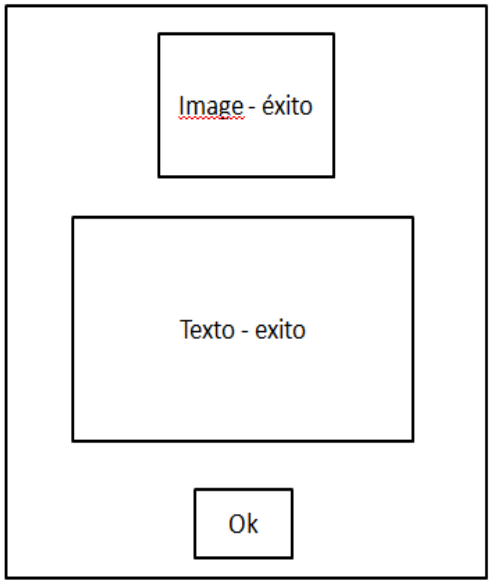
Este boceto refleja el aviso utilizado en caso de producirse alguna falla al procesarse alguna operación en la aplicación. Se caracteriza por una imagen logo, un texto que informa la falla producida, y unos botones de interacción que ayudaran al usuario a recuperarse de la falla de la operación.



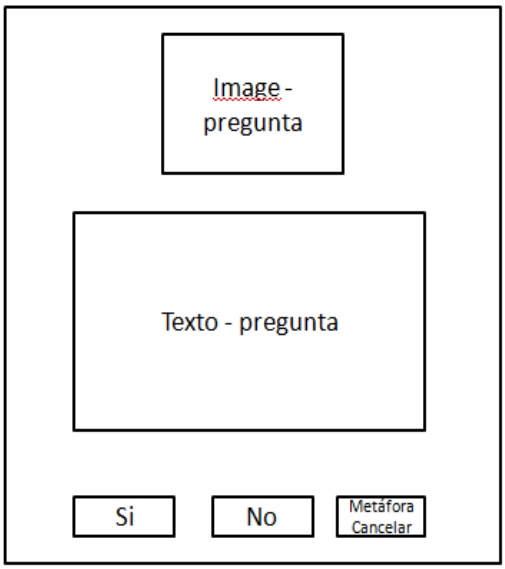
Este boceto refleja el aviso utilizado mientras se espera por el resultado de una operación. El usuario puede cancelar la operación en caso de que la operación se demore más de lo esperado, regresando a la interfaz anterior.



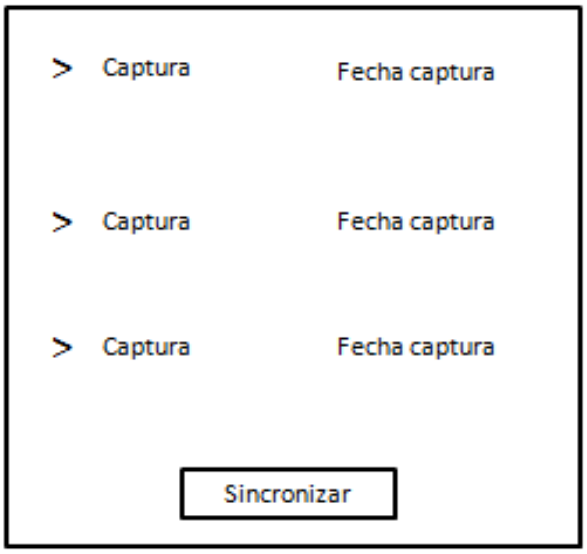
Este boceto refleja el aviso utilizado al completarse exitosamente una operación en la aplicación. El usuario regresará al menú principal después de hacer click en el botón.



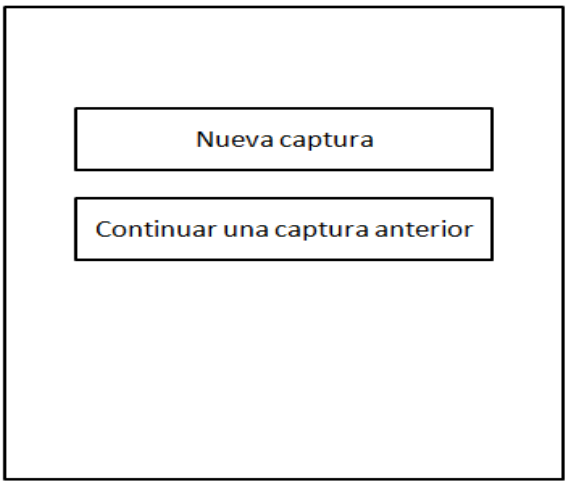
Este boceto refleja el aviso de pregunta cuando la aplicación requiere alguna información adicional del usuario, después de solicitarse alguna operación. El usuario puede aceptar lo que la aplicación le pregunta, negarlo o simplemente cancelar la operación, que devolverá la aplicación a la interfaz anterior.



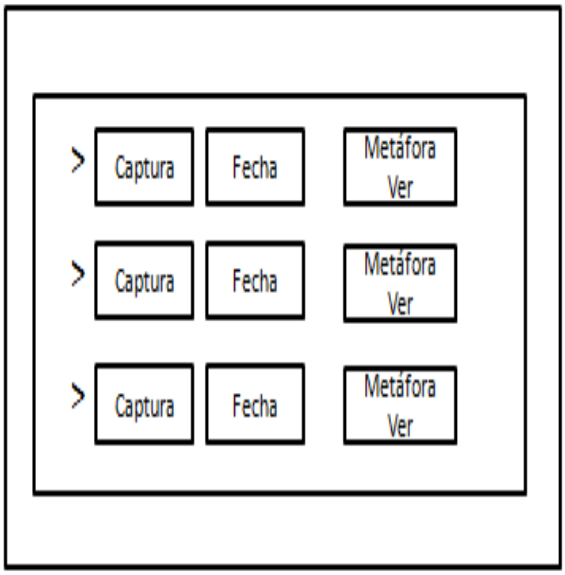
Este boceto refleja la lista de capturas pendientes por registrar en el sistema Siscla. Esta interfaz se carga al hacer click en la opción “Estado de capturas”



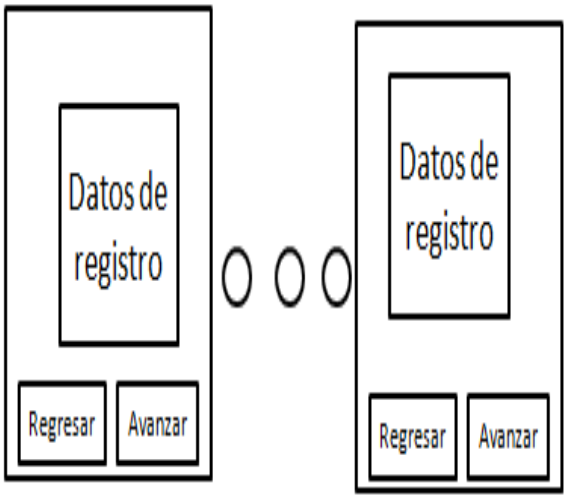
Este boceto refleja las distintas opciones para el proceso de captura de datos, al que se tiene acceso haciendo click en el botón “Captura de datos”. Las opciones definidas para la aplicación son: nueva captura o continuar una captura anterior.



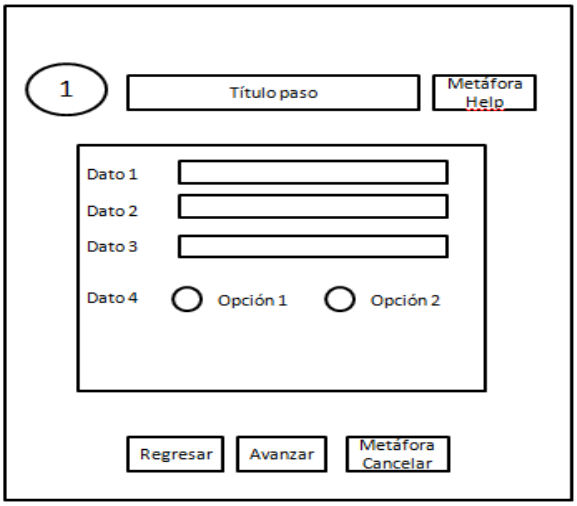
Este boceto refleja la lista de capturas pendientes por completar. El usuario puede reanudar el proceso de captura respectivo haciendo click en la metáfora ver.



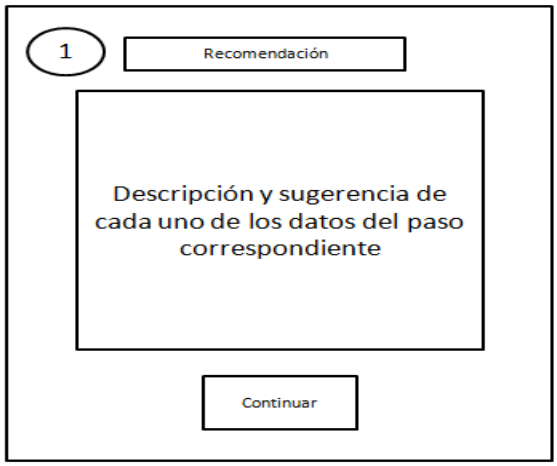
Este boceto refleja representa al patrón de tipo wizard utilizado para el proceso de captura de datos. Se caracteriza en una secuencia de pasos sucesivos en el que el usuario puede avanzar o regresar en cualquiera de los pasos en el que se encuentre.



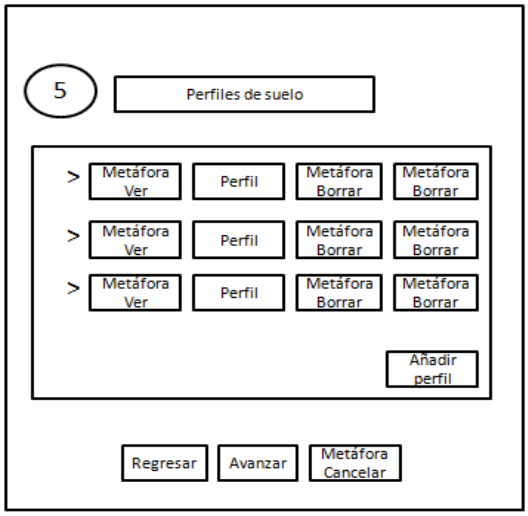
Cada paso del wizard, se caracteriza por un conjunto de campos para la recolección de datos, los cuales son utilizados en la medida que reduzcan la carga cognitiva del usuario; entiéndase el uso de listas desplegables, radio buttons, etc. Además se incluye una metáfora de ayuda, que proporciona una asistencia al usuario del paso respectivo, y la metáfora cancelar que detiene el proceso de captura de datos.



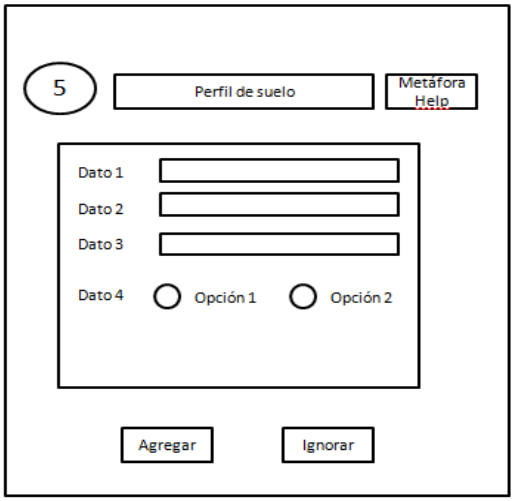
Este boceto refleja las recomendaciones que se ofrecen en cada uno de los pasos del wizard. Consiste en unos textos informativos y descriptivos de cada unos de los campos del paso respectivo.



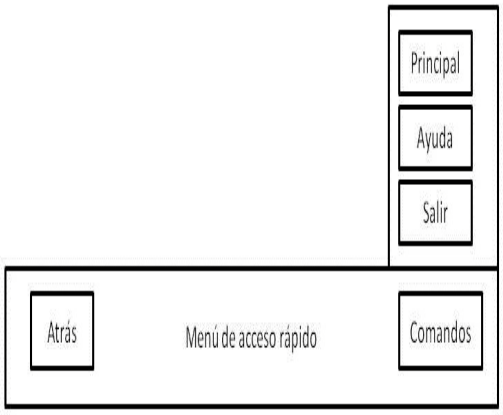
Este boceto refleja el patrón utilizado para realizar el registro de horizontes. Consiste en una lista de elementos que se van agregando cada vez que se añade un horizonte asociado.



La información relacionada con cada horizonte, se rellena con un formulado semejante al utilizado en cada uno de los pasos del wizard. El usuario puede agregar el paso o simplemente ignorarlo. En caso de que el horizonte ya fue creado, el botón agregar cambia a modificar.



Este boceto refleja el menú de acceso rápido de navegación. Las opciones incluidas en este menú son las siguientes: Atrás, que sirve para regresar a la interfaz anterior; y los comandos: principal, ayuda y salir.



4. Guía de estilo

Se utilizan la guía de estilo utilizada en el sistema Siscla, con el fin de mantener consistencia.

Norma	Especificación
Banner	
Enlaces en banner y pie de pagina	Color de fuente blanco
Título banner	Color de fuente: 227,231,110
Títulos	Color de fuente blanco; Color de fondo verde: 68,109,7 - 79,136,3 - 91,153,15 - 210,226,136 -
Contenido	Color de fuente negro; Color de fondo verde claro: 228,236,227 - 143,188,143 - 180,238,180
Botones	Estilo por defecto html
Elementos de menú	Viñeta de elemento (>); Títulos: subrayados, fuente verde;
Labels	Color de Fuente negro;
Campos de texto	Color de fondo blanco
Tablas	Encabezado: fuente blanco, fondo verde: 68,109,7 Celdas: fuente negro, fondo blanco; Sin bordes internos ni externos.

Además se especifican las siguientes normas adicionales.

Norma	Especificación
Botones	Color de fondo: Verde: 68,109,7 Color de fuente: Blanco: 255,255,255
Indicador del paso wizard	Color de fondo: Verde: 68,109,7 Color de fuente: Blanco: 255,255,255.
Formularios	Color de fuente negro: 0,0,0
Fondo de aplicación	Color de fondo: Según el dispositivo móvil.
Items	Color de fondo: Verde: 68,109,7 Color de fuente: Blanco: 255,255,255.
Imagen de espera	
Imagen de falla	
Imagen de éxito	

Norma	Especificación
Imagen de pregunta	
Imagen de inicio	
Imagen de salida	
Imagen captura de datos	
Imagen estado de capturas	
Imagen Fin wizard	
Propagandas (Acerca de la aplicación - Ayuda)