



UNIVERSIDAD DE ZARAGOZA
Centro Politécnico Superior



PROYECTO FIN DE CARRERA

Ingeniería Informática

Curso 2008/2009

Near Field Communication Application Provisioning Framework

Alumno: Andrea Martínez Mazo

Director: Jakob Saros – Ericsson AB, Luleå (Sweden)

Tutor: Kåre Synnes – Luleå Tekniska Universitet, Luleå (Sweden)

Ponente: Francisco Javier Fabra Caro – Universidad de Zaragoza

Zaragoza, Septiembre de 2009



NEAR FIELD COMMUNICATION APPLICATION FRAMEWORK

RESUMEN

Near Field Communication es una tecnología de comunicación que está ganando adeptos día a día. NFC posee características comunes a la tecnología Bluetooth: ambas son tecnologías de comunicación de corto alcance que funcionan sobre una conexión inalámbrica; aún con todo, NFC posee propiedades con las que aventaja a Bluetooth. Una de las más importantes radica en el tiempo de establecimiento de la conexión, el cual es significativamente menor en la tecnología NFC. Por todo ello, esta Comunicación de Corto Alcance se ha convertido en un medio fácil y seguro de transmisión de información. Su aplicación en el campo de la telefonía móvil abre nuevos caminos a servicios y transacciones de la vida cotidiana, que buscan facilitar y agilizar los procedimientos seguidos en la actualidad.

El proyecto con título NFC Application Provisioning Framework busca fomentar el uso de la tecnología descrita. La manera en que se desea conseguir este fin es facilitando un servicio al usuario, el cual le permitirá interactuar con cualquier etiqueta NFC, incluso cuando éstas, en un principio, no pudieran ser leídas por su teléfono móvil.

De esta forma, una única aplicación será la encargada de proveer todo tipo de servicios o aplicaciones requeridas para que un dispositivo móvil pueda comunicarse con dispositivos NFC y llevar a cabo transferencias de información entre ellos.



TABLA DE CONTENIDOS

Resumen	2
Tabla de contenidos	4
Tabla de figuras	6
1. Introducción	8
2. Análisis	14
2.1. Estudio previo	14
2.2. Metodología	14
2.3. Requisitos del sistema	16
2.4. Escenarios	17
2.5. Diagrama de Casos de Uso	17
2.6. Diagramas de actividades	19
2.7. Diagrama de Secuencia	20
2.8. Diagrama de clases	21
3. Diseño del sistema	22
3.1. Arquitectura física	22
3.2. Arquitectura lógica	23
3.3. Diseño de la Base de Datos	26
3.4. Interfaces de Usuario	27
4. Implementación	28
4.1. fases	28
4.2. Tecnologías y otros métodos	29
4.3. Estructura de la implementación	30
4.4. Tests de usabilidad	33
5. Conclusiones	34
Referencias bibliográficas	36



TABLA DE FIGURAS

Figura 1 – Diagrama Gantt, 1ª parte.....	15
Figura 2 – Diagrama Gantt, 2ª parte.....	15
Figura 3 – Diagrama de Casos de Uso	18
Figura 4 – Diagrama de actividades para el CU “GET GCRTD”	19
Figura 5 – Diagrama de actividades para el CU “SEARCH SERVICE”	19
Figura 6 – Activity diagram: Download service.....	19
Figura 7 – Diagrama de Secuencia	20
Figura 8 – Diagrama de Clases	21
Figura 9 – Arquitectura Física del sistema	23
Figura 10 – Record Handling Architecture.....	25
Figura 11 – Diagrama del modelo de datos.....	26



ABBREVIATIONS

AMS	Application Management System
API	Application Programming Interface
CDC	Connected Device Configuration
CLDC	Connected Limited Device Configuration
CSS	Cascade Style Sheets
DB	DataBase
EJB	Enterprise JavaBeans
GC	Generic Control
GCF	Generic Connection Framework
GCRTD	Generic Control Record Type Definition
GUI	Graphic User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
J2EE	Java 2 Platform, Enterprise Edition
J2ME	Java 2 Platform, Micro Edition
JAD	Java Application Descriptor
JAR	Java ARchive File
Java EE	Java Platform, Enterprise Edition
Java ME	Java Platform, Micro Edition
Java SE	Java Platform, Standard Edition
JSR	Java Specification Requests
JVM	Java Virtual Machine
KB	Kilobytes
Kbps	Kilobits per second
KVM	Kernel-based Virtual Machine
MB	Megabytes
Mbps	Megabits per second
MDIP	Mobile Information Device Profile
MHz	Megahertz
MIME	Multipurpose Internet Mail Extensions
NDEF	NFC Data Exchange Format
NFC	Near Field Communication

PC	Personal Computer
PDA	Personal Digital Assistant
PDF	Portable Document Format
RF	Radio Frequency
RFID	Radio Frequency Identification
SDK	Software Development Kit
RTD	Record Type Definition
SE	Secure Element
UC	User Case
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAR	Web application ARchive
XML	eXtensible Markup Language

1. INTRODUCCIÓN

La tecnología NFC o de corto alcance hace referencia a un protocolo basado en una interfaz inalámbrica de un alcance inferior a 10 centímetros, basada en estándares, simple y segura y que permite la interacción bi-direccional entre dispositivos electrónicos.

NFC está basado en tecnologías sin contacto por Radio frecuencia (RFID, definidas en el estándar ISO/IEC 14443), y por ello requiere un dispositivo emisor y un receptor. Trabaja en la banda de los 13'56 MHz, lo cual implica que no se le aplican restricciones y no se requieren licencias de uso. Aún así, cada país es libre de imponer sus propias limitaciones en las emisiones electromagnéticas para este bando de radio-frecuencia.

La tecnología NFC está suponiendo un cambio importante en el campo de las conexiones inalámbricas. Un gran número de empresas han iniciado hace algunos años ensayos y pruebas para incluir esta tecnología en sus transacciones habituales. Así, empresas como 7-eleven, el metro de Londres o el Estadio de Fútbol del Manchester City Football Club, han introducido ya la posibilidad de realizar pagos u otras operaciones mediante dispositivos con soporte para NFC.

Con el desarrollo del presente proyecto se ha querido indagar en esta puntera tecnología, colaborando así en su desarrollo y proponiendo una solución a un sector de la misma que, se ha considerado, no estaba cubierto de forma total por los estándares existentes.

*Para más información sobre la tecnología NFC, consultar la sección "A.1.1. NFC Technology" del "anexo A. Preliminary Study" al final de este documento.

ESTADO DE LAS SOLUCIONES NFC

En la actualidad las aplicaciones NFC para dispositivos móviles se desarrollan asumiendo que el dispositivo lector o receptor tiene la capacidad de interpretar y dar respuesta al mensaje enviado por el emisor de la transferencia. En caso de que esto no se cumpla, el dispositivo receptor ignorará la comunicación, puesto que no tiene recursos para poder establecerla y llevar a cambio el intercambio de información deseado. Este hecho implica que, previo a poder interactuar con cualquier tipo de dispositivo NFC, y más en concreto con etiquetas RFID, es preciso que en el dispositivo receptor se hayan

instalado las aplicaciones o servicios requeridos para que el proceso de transmisión se pueda hacer efectivo.

Como se puede deducir del contexto, esta forma de reacción ante dispositivos no-conocidos es desaconsejable, tanto para el usuario final, como para la organización que busca ofrecer una información al usuario mediante el uso de esta tecnología.

Así, el usuario final no tiene la oportunidad de interactuar con el dispositivo de origen, sin obtener a su vez, ningún tipo de información sobre el problema que causa esta incompatibilidad. El mensaje que puede visualizar, en un formato generalizado, es “Etiqueta de formato desconocido” o similar, sin obtener indicaciones o recomendaciones para poder solucionar el problema.

Por su parte, el comercial, empresa u organización que desea ofrecer un servicio a sus clientes mediante esta técnica, ve reducido el número de receptores, esto es, la cantidad de personas a las que llega su mensaje, puesto que, todos aquellos que no tengan la funcionalidad requerida en su teléfono móvil NFC, no podrán acceder a esta información.

OBJETIVOS Y ALCANCE

En el presente proyecto se busca dar una solución al problema expuesto anteriormente. Se estudiarán posibles acercamientos que puedan solventar la incompatibilidad entre ciertos dispositivos NFC emisores (etiquetas NFC) y dispositivos NFC receptores (dispositivos móviles).

Para ello, se van a utilizar como elementos de prueba una serie de etiquetas NFC y un dispositivo móvil con soporte para NFC. El producto final consistirá en una aplicación que pueda ser instalada en dicho dispositivo, de forma que lleve a cabo las operaciones necesarias para resolver así el problema mostrado.

CONTEXTO DEL PROYECTO

El desarrollo del proyecto se va a llevar a cabo dentro del ámbito empresarial. Ericsson AB, sede situada en Luleå (Suecia), es la empresa en la que se va a desarrollar el presente proyecto. De esta forma, los recursos necesarios para poder testar producto van a ser proporcionados por dicha empresa.

Una vez finalizado el proceso de desarrollo e implementación del sistema, el proyecto pasará a formar parte de una plataforma de mayor alcance, diseñada en el departamento de investigación de Ericsson AB, para gestionar diferentes usos de etiquetas NFC.

Powertags® es el nombre con el que ha bautizado la empresa a la funcionalidad que van a aportar las etiquetas NFC obtenidas en este proyecto. Dichas etiquetas proporcionarán al usuario la libertad de poder interactuar con etiquetas NFC con las que su dispositivo móvil, en un principio, no hubiera sido capaz de interactuar.

LIMITACIONES

El sistema definido en el proyecto cuenta con ciertas restricciones que, en caso de no ser cumplidas, pueden causar el mal funcionamiento o un funcionamiento no esperado del mismo. A continuación se especifican estas restricciones:

- El dispositivo móvil utilizado para el sistema debe contar con conexión inalámbrica a Internet. En caso de no contar con ella, el sistema no podrá terminar de la forma deseada, puesto que es requisito imprescindible la conexión del teléfono con otros componentes del sistema, para el intercambio de información y servicios. De la misma forma, el usuario deberá permitir el acceso a dicha conexión, para que el dispositivo pueda hacer uso de ella.
- La aplicación final obtenida debe estar previamente instalada en el dispositivo móvil para que el sistema funcione de la forma deseada.

PERSONAS INVOLUCRADAS

- Desarrolladora del proyecto:

Andrea Martínez Mazo (andreammazo@gmail.com)
Ingeniería Informática
Universidad de Zaragoza (CPS)
Luleå Tekniska Universitet

- Director del proyecto:

Jakob Saros (jakob.saros@ericsson.com)
Research Engineer
Service Layer Technology
Ericsson AB (Ericsson Research)
Luleå, Sweden

- Tutor del proyecto:

Kåre Synnes (Kare.Synnes@ltu.se)
Computer Science and Electrical Engineering unit
Information and Communication Technology department
Luleå Tekniska Universitet
Luleå, Sweden

- Ponente del proyecto:

Francisco Javier Fabra Caro (jfabra@unizar.es)
Área de Lenguajes y Sistemas Informáticos
Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza (CPS)
Zaragoza, Spain

CONTENIDO DEL PRESENTE DOCUMENTO

En los siguientes capítulos de este documento se va a describir el proceso seguido para el desarrollo del proyecto. La estructura sigue el orden de las fases que componen un proyecto software, puesto que así se puede seguir el desarrollo del proceso en orden cronológico, siguiendo etapa a etapa su estudio, diseño e implementación.



2. ANÁLISIS

2.1. ESTUDIO PREVIO

La primera fase a la que se enfrenta el proyecto es el estudio de las tecnologías implicadas en el mismo. Al inicio de este proyecto, NFC es una tecnología desconocida en gran parte por el alumno. Es por ello que se han dedicado en torno a 4 semanas para entender su funcionamiento, tecnologías que la implementan, dispositivos compatibles y otros aspectos relevantes para el proyecto.

*Para información más detallada sobre los conocimientos adquiridos durante esta fase de estudio, véase el “anexo A. Preliminary Study” al final del presente documento.

2.2. METODOLOGÍA

El ciclo de vida del proyecto se va a seguir usando la metodología de “Prototipado de Software”. Esta elección hace posible la detección de fallos en etapas tempranas, en la toma de requisitos del sistema.

Para lograr cumplir los objetivos del proyecto se van a seguir varias etapas:

- **Identificación de requisitos** –análisis
- **Estudio de posibles tecnologías** –análisis
- ↓ Especificaciones incompletas
- **Identificación y diseño de prototipos** –diseño

- **Desarrollo e implementación del primer prototipo** –implementación
- **Evaluación del prototipo** (usuario final) –pruebas y realimentación

↓ Especificaciones completas

- **Implementación del producto** (expansión gradual)

Una vez definidas las etapas que va a seguir el proyecto, se procede a establecer recursos temporales para cada una de ellas. De esta fase se ha obtenido el Diagrama Gantt que muestra las restricciones de tiempo establecidas para llevar a cabo el desarrollo del presente proyecto:

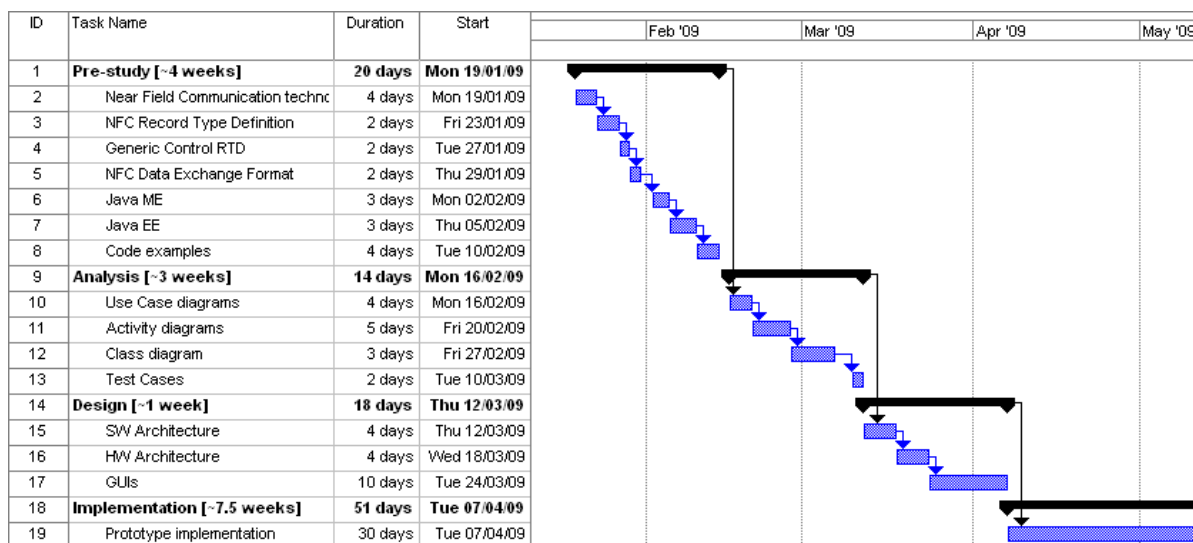


Figura 1 – Diagrama Gantt, 1ª parte

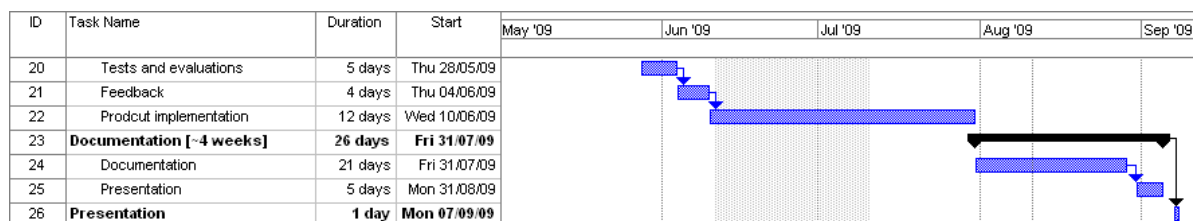


Figura 2 – Diagrama Gantt, 2ª parte

Como se puede ver en el diagrama anterior, el proyecto se inicia a finales de Enero de 2009 y finalizará a principios de Septiembre del mismo año, quedando algunas semanas libres de dedicación al mismo, por diferentes motivos.

2.3. REQUISITOS DEL SISTEMA

Se han estudiado las características que el sistema deberá tener para atender las necesidades establecidas. Así, se han obtenido los siguientes requisitos:

FUNCIONALIDAD

Se quiere obtener un proceso automatizado y transparente al usuario, en el que se minimice la interacción con el mismo, con el objetivo de facilitar así su uso y navegabilidad. Así, el proceso será iniciado como respuesta a un evento del usuario, en el que éste aproxime el dispositivo móvil a una etiqueta NFC.

Una vez iniciado el proceso, esto es, cuando suceda que un usuario inicia la interacción con una etiqueta NFC y su teléfono móvil no está preparado para dicha transacción de información, el usuario deberá permitir ciertas operaciones (como puede ser el acceso a Internet) y, tras ello, el servicio requerido para que el dispositivo móvil y la etiqueta puedan interaccionar, quedará instalado y listo para su ejecución.

INTERFACES DE USUARIO

Sencillas, comprensibles y que proporcionen una idea intuitiva de navegabilidad.

DISPOSITIVOS

Otro de los requisitos impuestos es el uso de los dispositivos con los que se realizarán las pruebas. Éstas se llevarán a cabo con etiquetas RFID (UPM Rafsec) y con un teléfono móvil con soporte para NFC (Nokia 6131 NFC).

REQUISITOS NO IMPUESTOS POR EL SISTEMA

Conforme se ha ido analizando el sistema y sus requerimientos, se ha encontrado cierta funcionalidad que no es imprescindible para que el sistema funcione, pero sí aconsejable y deseable para que así, todos los aspectos que interfieren en el proceso queden cubiertos.

Es por ello que se van a añadir las nuevas características estudiadas al sistema, y con ello se amplían ligeramente los requisitos del mismo. A continuación están descritas las funcionalidades y propiedades que implementarán estos requisitos no impuestos por el propio sistema:

- Aplicación que permita escribir en etiquetas NFC: Se desea obtener un modo de escribir la información deseada en etiquetas NFC vacías. De esta forma, se podrá programar el sistema, para que los dispositivos móviles puedan detectar la etiqueta de la forma esperada.
- Aplicación web de subida de servicios: Es conveniente tener un sistema web (para que pueda ser accedido de forma global), el cual permita subir servicios al servidor del sistema, de forma que, tras esto, queden listos para ser descargados por los usuarios.

2.4. ESCENARIOS

Tras el estudio de tecnologías y captura de requisitos, se han pensado una serie de escenarios en los que se puede ver cómo se podría aplicar el sistema que se va a desarrollar, en contextos de la vida real, y cómo afectaría éste en la interacción entre usuarios y dispositivos NFC. Esta fase ayuda a visualizar la funcionalidad que ofrecerá el sistema y algunos aspectos de interacción que pueden aclarar aspectos hasta ahora no tenidos en cuenta.

A continuación se enumeran los escenarios que se han definido. Todos ellos están pensados para que se sitúen etiquetas NFC en distintas localizaciones, con distintos objetivos.

- **Pósters inteligentes (Smart Posters) en el cine:** Etiquetas NFC integradas en carteles anunciadores de películas, que aportarán información de la película que publicitada (horario, público apto, sinopsis...).
- **Noticias / titulares / información meteorológica:** Etiquetas NFC situadas en áreas céntricas de la ciudad. Al tocarlas, se podrán consultar las noticias destacadas del día (en forma, por ejemplo, de titulares) u otra información como la predicción meteorológica.
- **Información turística:** Las etiquetas NFC se situarán en oficinas de turismo. El usuario podrá obtener información turística relevante.
- **Adquisición de billetes last-minute:** Etiquetas NFC situadas en estaciones trenes y autobuses, cerca de paneles informativos. El usuario podrá adquirir billetes último-minuto de forma rápida y sin esperar colas, con tan sólo acercar su teléfono al trayecto deseado.

* Los escenarios arriba expuestos están explicados con mayor detalle en la sección "B.1. Scenarios" del "anexo B. Analysis" de este documento.

2.5. DIAGRAMA DE CASOS DE USO

Una vez identificados los requisitos del proyecto, se ha elaborado el Diagrama de Casos de Uso. En él se pueden identificar dos casos de uso principales:

- **RECOGER INFORMACIÓN (GET DATA):** Incluye las operaciones necesarias para obtener los parámetros de configuración del dispositivo móvil y de la etiqueta NFC.
- **PROCESAR SERVICIO (PROCESS SERVICE):** Comprende las operaciones necesarias para la búsqueda del servicio apropiado y su posterior descarga e instalación en el dispositivo.

En el sistema se han identificado seis actores. Cinco de ellos son dispositivos físicos encargados de procesar diversas operaciones. El sexto actor es el usuario final.

- **ETIQUETA NFC (NFC TAG):** Dispositivo que contiene la información a ser leída por el dispositivo NFC del usuario final.
- **DISPOSITIVO MÓVIL (MOBILE DEVICE):** Dispositivo que gestionará el proceso. En él quedará instalado el servicio que se descargue, de forma que pueda, una vez hecho esto, interactuar de la forma esperada con la etiqueta.
- **SERVIDOR DE LOOKUP (LOOKUP SERVICE):** Máquina encargada de buscar el servicio adecuado que cumpla con los requisitos de la etiqueta NFC. Esta máquina consultará los servicios disponibles en el Registro de Aplicaciones, y los comparará con los parámetros obtenidos de la etiqueta y el dispositivo móvil.
- **SERVIDOR DE APLICACIONES (APPLICATION SERVER):** Máquina donde se tiene almacenados los servicios, listos para ser descargados e instalados en los dispositivos móviles de los usuarios.
- **REGISTRO DE APLICACIONES (APPLICATION REGISTRY):** Máquina en la que se tiene almacenada la información referente a los distintos servicios almacenados en el Servidor de Aplicaciones.
- **USUARIO FINAL (END USER):** Es la persona que inicia el proceso, llevando a cabo el evento de tocar con su dispositivo móvil la etiqueta NFC.

A continuación se muestra el diagrama de Casos de Uso, formado por los componentes arriba descritos:

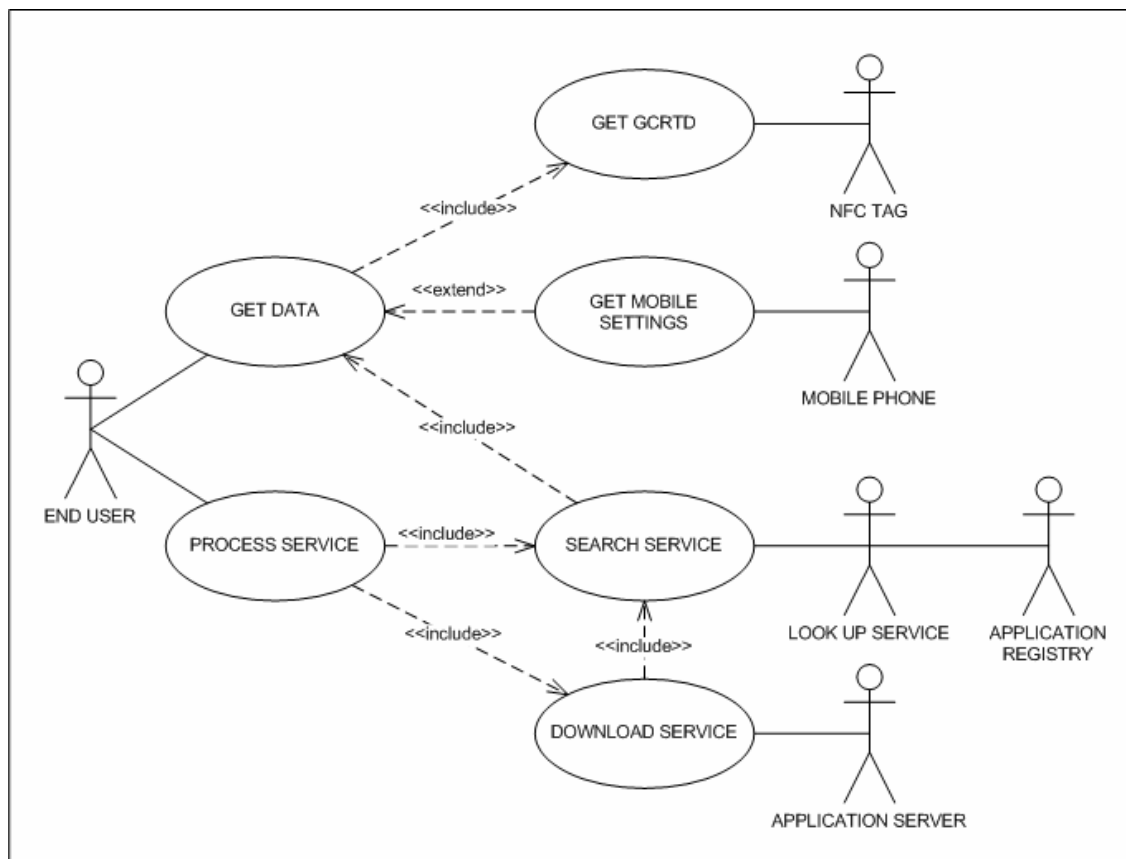


Figura 3 – Diagrama de Casos de Uso

2.6. DIAGRAMAS DE ACTIVIDADES

Un Diagrama de Actividades especifica un Caso de uso. Es por ello que se han seleccionado los casos de uso que tienen mayor complejidad y se han detallado mediante diagramas de actividades, los cuales se pueden ver a continuación:

- Especificación del CU “GET GCRTD”: El Caso de Uso especifica las operaciones necesarias para leer la información (Generis Control Record Type Definition) de la etiqueta NFC.

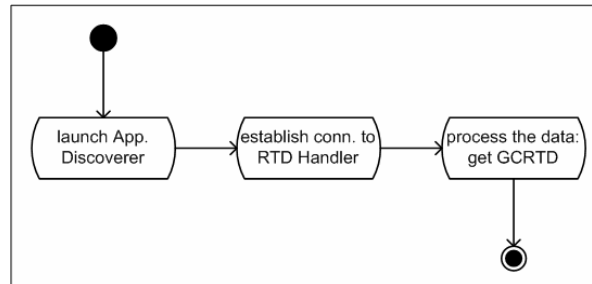


Figura 4 – Diagrama de actividades para el CU “GET GCRTD”

- Especificación del CU “SEARCH SERVICE”: Este diagrama muestra las operaciones que se llevan a cabo para buscar el servicio más apropiado que se adapte a la etiqueta NFC consultada. La búsqueda se realiza a partir de parámetros del teléfono y parámetros obtenidos de la etiqueta.

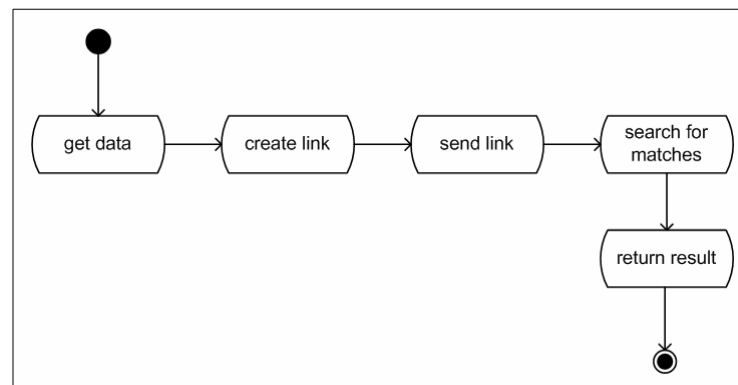


Figura 5 – Diagrama de actividades para el CU “SEARCH SERVICE”

Especificación del CU “DOWNLOAD SERVICE”: El diagrama muestra las operaciones requeridas para descargar el servicio requerido por la etiqueta NFC.

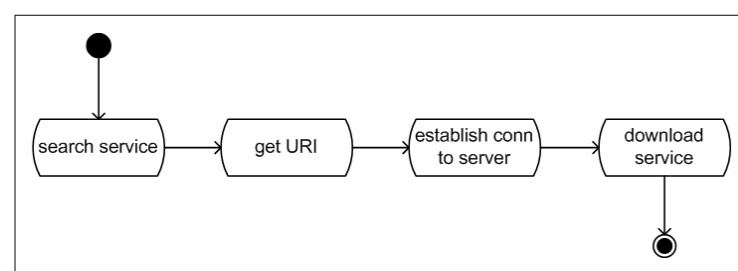


Figura 6 – Activity diagram: Download service

2.7. DIAGRAMA DE SECUENCIA

Un Diagrama de Secuencia muestra cómo se suceden las distintas operaciones a lo largo del proceso, qué métodos se llaman en cada etapa y qué información se transmite en ellos, cuánto dura la vida de cada uno de los objetos y cómo se comunican entre ellos.

El Diagrama de Secuencia de nuestro sistema está compuesto por seis objetos principales. El objeto denominado GUI es el objeto que desencadena el proceso. En él se producen las primeras acciones, y es él el que recibe la información final. Éste objeto es el que representa la interacción con el usuario, el encargado de recoger la información necesaria para iniciar el proceso de búsqueda del servicio y el encargado de notificarle antes los posibles eventos que se sucedan.

El Diagrama de Secuencia que modela el sistema principal que estamos desarrollando se puede ver a continuación:

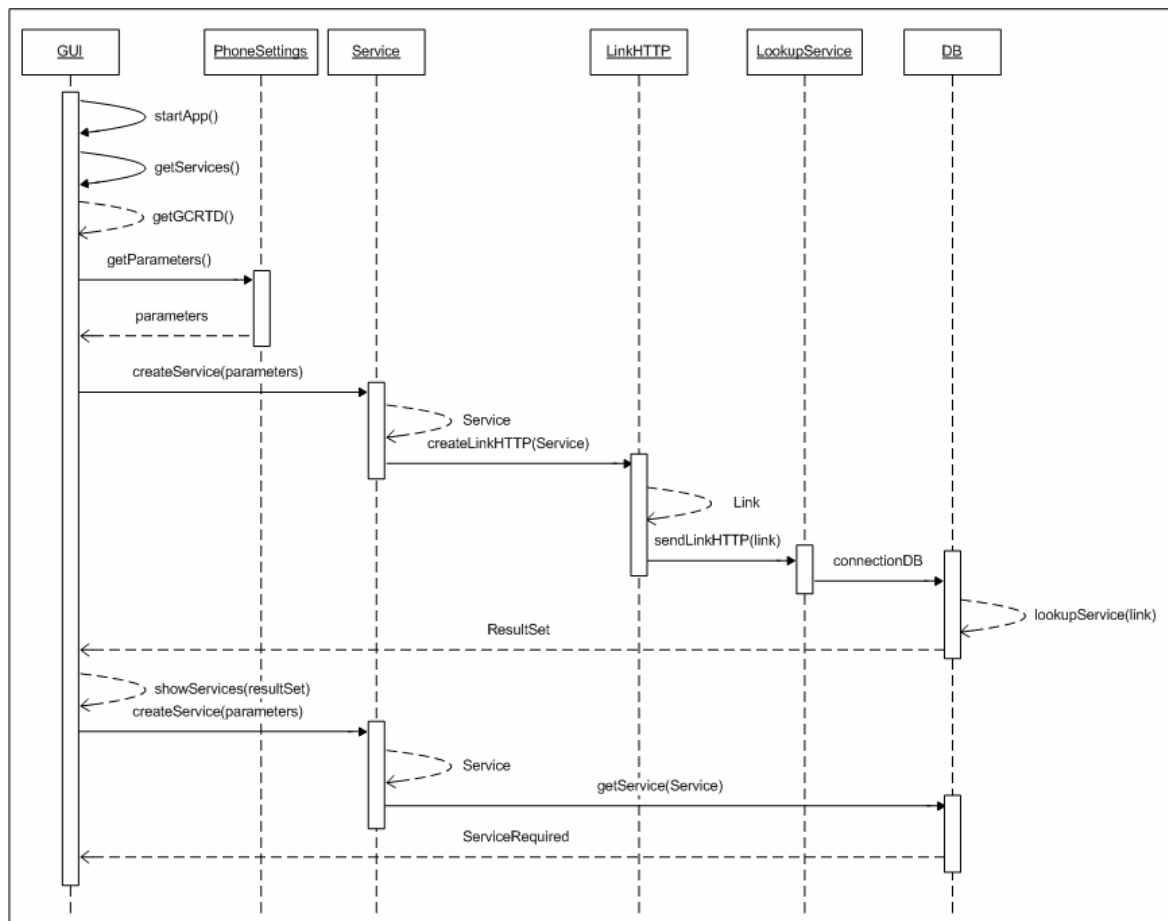


Figura 7 – Diagrama de Secuencia

2.8. DIAGRAMA DE CLASES

Por último, se ha creado el Diagrama de Clases, una estructura estática que describe el sistema mostrando sus clases, las relaciones entre ellas y, en este caso, las operaciones que cada una de ellas implementa. Podemos ver el diagrama a continuación:

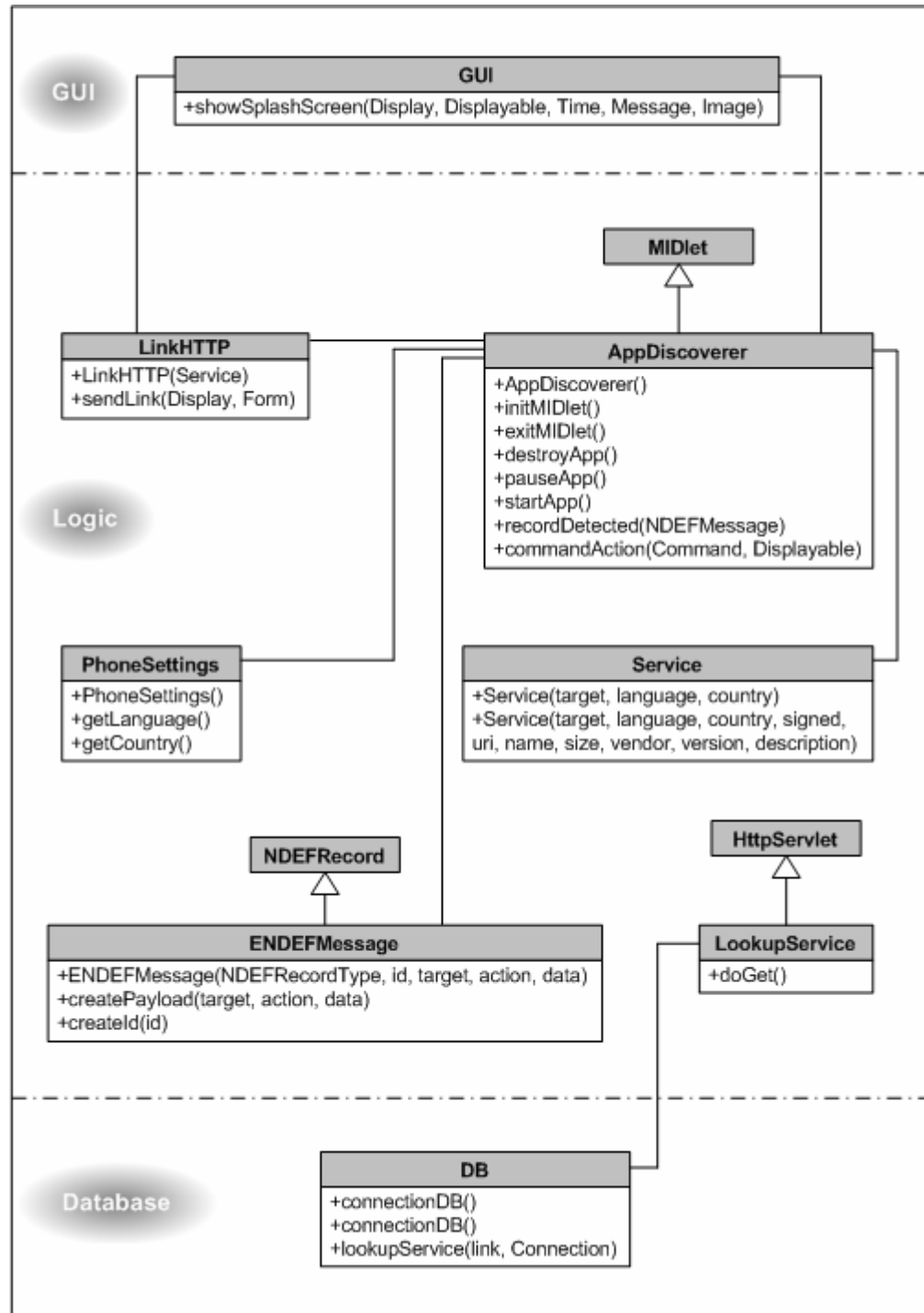


Figura 8 – Diagrama de Clases

3. DISEÑO DEL SISTEMA

3.1. ARQUITECTURA FÍSICA

La arquitectura física de un sistema está compuesta por los elementos físicos que la componen. Dentro de nuestro sistema se han identificado seis elementos que conforman esta arquitectura:

- **Etiqueta NFC:** Etiqueta RFID que contiene la información que debe ser mostrada en el dispositivo móvil del usuario final. Al ser detectada por el dispositivo, se activa e inicia la transferencia de información, enviando al teléfono móvil un `NDEFMessage` con el contenido deseado.
- **Dispositivo móvil con soporte NFC:** Es el dispositivo que inicia el proceso al aproximarse a una etiqueta NFC a menos de 10 centímetros de distancia. Una vez que la etiqueta ha sido reconocida y su información se ha leído, el teléfono comprueba si tiene el servicio requerido para procesar dicha información. Si no lo tiene disponible, se lanza la `ApplicationDiscoverer`, y con ello, se envía una petición al servidor junto con los parámetros leídos del móvil e información de la etiqueta. Este dispositivo debe estar provisto de acceso inalámbrico a Internet.
- **Servidor de Lookup:** Servidor Web que recibe las peticiones HTTP que los dispositivos móviles lanzan. En él están alojados los Java servlets que se ejecutan al detectar una petición HTTP. Para que el proceso pueda desarrollarse de la forma deseada, este servidor debe tener acceso al Registro de Aplicaciones.
- **Registro de Aplicaciones:** Base de datos que almacena información acerca de los servicios disponibles para ser descargados. Recibe peticiones SQL desde el Servidor de Lookup, enviando por su parte la respuesta a dichas peticiones.
- **Servidor de Aplicaciones:** Servidor donde se alojan los servicios para ser descargados. Cuando el dispositivo móvil obtiene la URI del servicio requerido, se establece una conexión entre dicho dispositivo y el servidor de Aplicaciones, para que se pueda descargar así el servicio.

- **Ordenador Personal:** U otro dispositivo con acceso a Internet, es el dispositivo desde el que se visualizará la página Web en la que se introducirá información sobre el servicio a subir, y se cargará el propio servicio. Este dispositivo establecerá una conexión con el Servidor de Lookup, que será el encargado de almacenar los servicios y su información el destino correspondiente.

El esquema de la arquitectura física del sistema se puede ver a continuación:

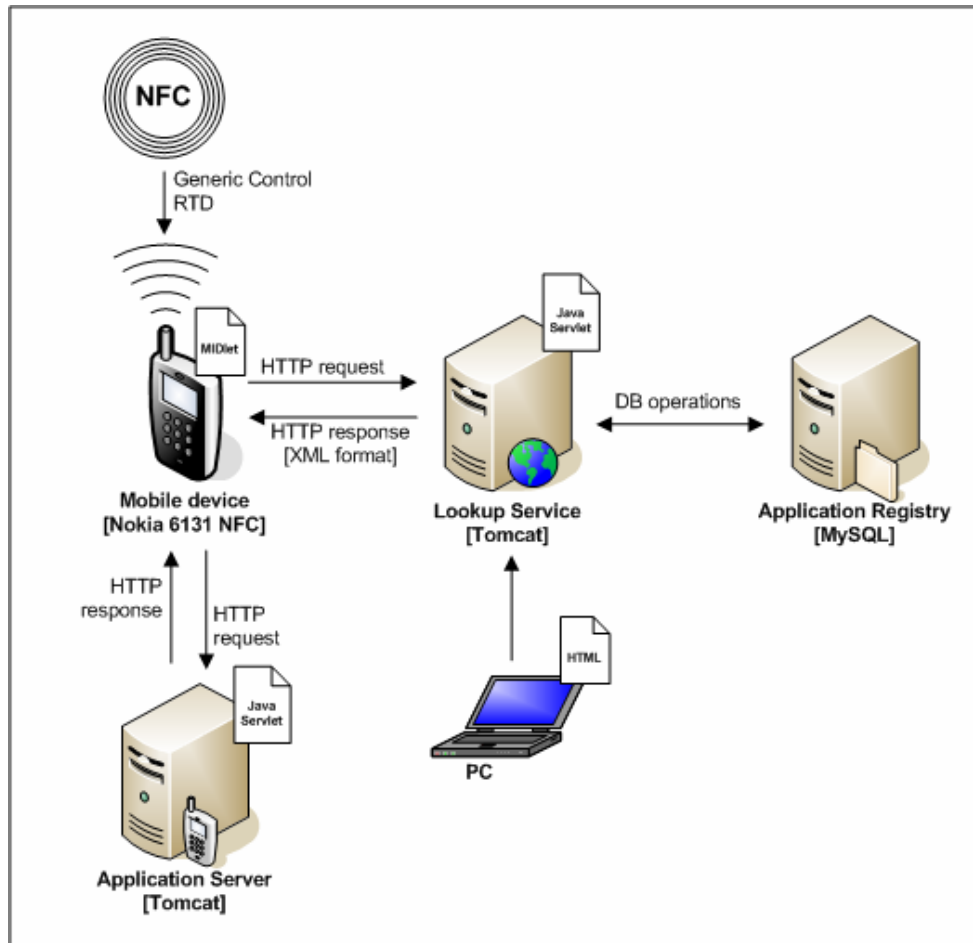


Figura 9 – Arquitectura Física del sistema

3.2. ARQUITECTURA LÓGICA

El sistema a desarrollar en este proyecto se ha dividido en tres sub-sistemas.

El primero de ellos se corresponde con la aplicación más importante que se ha denominado *ApplicationDiscoverer*. Ésta será la encargada de auto-lanzarse cuando una etiqueta NFC, no reconocida por el dispositivo móvil, se detecte. A partir de ahí, se realizará el proceso de búsqueda de la aplicación, en el Servidor de Registros, a través de Servidor de Lookup, se obtendrán las localizaciones de los servicios aconsejados por

el sistema, el usuario deberá entonces seleccionar una opción, y este servicio escogido se descargará e instalará en el dispositivo móvil del usuario final.

Un segundo sub-sistema es el definido para poder escribir el `NDEFMessage` deseado en una etiqueta NFC vacía.

El último de los sub-sistemas definidos es el compuesto por la página Web y los servidores de bases de datos (Servidor de Registro y Servidor de Aplicaciones). Este sistema permite introducir servicios y su información correspondiente mediante una interfaz Web.

Por otro lado, la arquitectura lógica del sistema está estructurada en tres capas. Así, cada una de ellas (capa de presentación, capa de lógica de negocio y la capa de datos) puede mantenerse de forma individual, y alojarse en un módulo independiente al resto de capas. Esta arquitectura proporciona escalabilidad y aumenta el rendimiento del sistema.

A continuación se va a explicar el papel de cada uno de los sub-sistemas, y las estructuras de capas que los conforman.

- **Application Discoverer:** La aplicación se despliega en una arquitectura de tres capas:
 - Datos: está distribuida en dos fragmentos: el Servidor de Aplicaciones y el Registro de Aplicaciones. Éste último está implementado en una base de datos MySQL, en la que se almacena toda la información correspondiente a los servicios disponibles en el Servidor de Aplicaciones. Por su parte, el Servidor de Aplicaciones está desplegado en un directorio del servidor, en el que se encuentran los servicios, dispuestos a ser descargados por los usuarios. Éste está implementado en un servidor Apache Tomcat.
 - Lógica de negocio: Está distribuida entre el Servidor de Lookup y el dispositivo móvil. Los Java servlets y las operaciones de acceso a la base de datos se encuentran en el servidor. Por su parte, estructuras de manejo e intercambio de datos, quedan situadas en el lado del cliente. La parte correspondiente al lado del servidor (Servidor de Lookup) está alojada en un servidor Apache Tomcat.

En el presente proyecto, la capa de datos y la capa de lógica están desplegadas en la misma máquina, por motivos de disponibilidad de recursos, pero éstas podrían migrarse a diferentes máquinas, teniendo que cambiar, únicamente, las referencias físicas a dichas máquinas, como son las URIs y direcciones IP de las cadenas de conexión..

 - Capa de presentación: Se aloja en el dispositivo móvil. Está formada por clases de creación de interfaces gráficas y un `MIDlet`. Esta capa contiene las operaciones referentes al aspecto visual de la aplicación y a su interacción con el usuario. También contiene las operaciones requeridas de detección de etiquetas NFC y la gestión de la transmisión de información, puesto que éstas son clases propias de la clase `MIDlet`.
- **WriteNDEFMessage:** La aplicación se distribuye en dos capas, puesto que no requiere almacenamiento de datos:
 - Lógica: Está formada por las clases que forman las estructuras de datos necesarias para llevar a cabo el proceso de escritura. Éstas son las

- o propias para el manejo de mensajes NDEF y los registros que los forman. Se encuentra desplegada en el dispositivo móvil.
 - o Presentación: Contiene el diseño de las interfaces gráficas y las operaciones requeridas para detectar etiquetas NFC vacías. Éstas últimas están implementadas en un MIDlet. Al igual que la capa de lógica de negocio, ésta se está desplegada en el dispositivo móvil.
- **Aplicación Web:** Se podría decir que esta aplicación, en cierta forma, alimenta a la `ApplicationDiscoverer`, puesto que es el medio para subir los servicios y que queden dispuestos a ser descargados. Es por ello que ambos sub-sistemas comparten la capa de datos. La Aplicación Web está formada por tres capas.
 - o Datos: Misma estructura de la `ApplicationDiscoverer`. Por tanto, está formada por el Registro de Aplicaciones y el Servidor de Aplicaciones.
 - o Lógica de negocio: La forman las clases que implementan las estructuras de datos necesarias para llevar a cabo la transferencia de información, y los Java servlets alojados en el Servidor de Lookup, desplegado en un servidor Apache Tomcat.
 - o Presentación: Implementada con código HTML y hojas de estilo CSS. Se muestra en el lado del cliente, a través de un navegador Web.

Algunos de los conceptos definidos en la Arquitectura Lógica se pueden ver en la figura mostrada en la sección anterior (Figura 9 – Arquitectura Física del sistema).

ARQUITECTURA RECORD HANDLING

Una arquitectura sobre la que es preciso hacer referencia es la llamada *Record Handling Architecture*. Ésta arquitectura debe estar implementada en el dispositivo móvil del usuario final para que el sistema (más en concreto la `ApplicationDiscoverer` y la aplicación `WriteNDEFMessage`) pueda funcionar de la forma deseada. Es la encargada de lanzar la `ApplicationDiscoverer` en el caso de que el servicio requerido por la etiqueta NFC no se encuentre disponible en el teléfono móvil.

A continuación podemos ver cómo afecta la instalación y registro de la `ApplicationDiscoverer` a la arquitectura Record Handling:

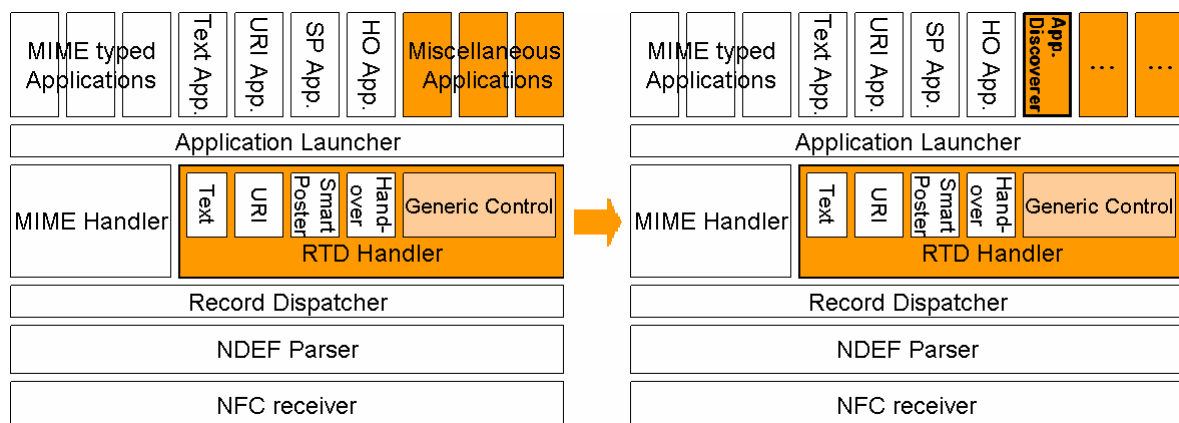


Figura 10 – Record Handling Architecture, antes y después de la instalación de la `ApplicationDiscoverer`

Si nos fijamos en la parte superior derecha de la figura anterior, podemos ver el efecto que tiene la instalación y registro, mediante el método `PushRegistry`, en esta arquitectura. De esta forma, la `ApplicationDiscoverer` queda preparada y a la escucha de detectar etiquetas no reconocidas por otras aplicaciones, momento en el cual será lanzada, y ejecutará su proceso de búsqueda del servicio requerido.

* Para obtener información detallada de esta arquitectura se puede consultar la sección “A.2. Background” del “anexo A. Preliminary Study” y el apartado titulado “Record Handling Architecture”, en la sección “C.2. SW Architecture” del “anexo C. System Design”, al final de este documento.

3.3. DISEÑO DE LA BASE DE DATOS

El diseño del modelo de datos ha quedado reflejado en una única tabla, dado a la sencillez de la información a almacenar. La tabla lleva por nombre “services (servicios)”, y mantendrá la información de los servicios disponibles en el Servidor de Aplicaciones. Esta tabla tiene como cable primaria la unión de cuatro campos: name (nombre) + version (versión) + language (lenguaje) + country (país), todos ellos hacen referencia a un servicio.

El diseño del modelo de datos se puede ver en la siguiente figura:

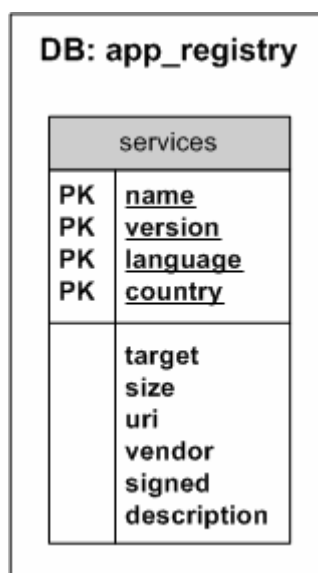


Figura 11 – Diagrama del modelo de datos

3.4. INTERFACES DE USUARIO

Como ya hemos en apartados anteriores, la metodología seguida en el proyecto es una metodología guiada por prototipos. Es por ello que las interfaces gráficas han tenido gran importancia a lo largo de todo el desarrollo del proyecto y éstas han guiado, en gran medida, la marcha del mismo.

En las etapas iniciales, se desarrollaron varios prototipos para el sistema, de forma que pudiéramos hacernos una idea de cómo iba a ser el resultado final. Una vez obtenidos estos prototipos, y habiendo seleccionado uno de ellos como el más adecuado para nuestro propósito, nos pusimos en contacto con Didier Chincholle, un especialista en Diseño e Interacción, del departamento de Interacción y Usabilidad de Ericsson Research, en su sede de Estocolmo. Éste nos guió y nos dio ciertas pautas que deben cumplir las aplicaciones destinadas a usuarios finales. De esta forma, a partir del feedback obtenido, se creó un segundo prototipo, que volvió a ser evaluado, y nuevamente valorado por el especialista en Interacción y Usabilidad.

Una vez aplicadas las últimas modificaciones, el prototipo estaba listo para ser testado y evaluado por usuarios finales reales, y obtener así una nueva fuente de información y opiniones. Es en este momento cuando se lleva a cabo la primera fase de encuestas, que fueron respuestas tras probar el prototipo en un dispositivo real. Este prototipo no contaba con funcionalidad, siendo exclusivamente una secuencia de pantallas, las mismas (o muy similares) que poseería la aplicación final.

Se llevaron a cabo los últimos retoques, teniendo en cuenta los comentarios obtenidos de esta evaluación de usuarios, y se obtuvo de esta manera el prototipo final, el cual sería implementado y provisto de la funcionalidad requerida.

4. IMPLEMENTACIÓN

4.1. FASES

La implementación del sistema se ha llevado a cabo una vez que el prototipo final quedó definido. Así, el orden seguido ha sido el siguiente:

- En primer lugar se implementó la aplicación `WriteNDEFMessage`. La funcionalidad obtenida es un `MIDlet` que, al ejecutarse, queda en espera de detectar una etiqueta vacía, y una vez localizada, escribe en ella el mensaje NDEF correspondiente. Esto permite tener todo preparado para poder continuar con la siguiente fase de la implementación puesto que se pueden escribir las etiquetas NFC de la forma deseada, algo necesario para asegurarnos la correcta lectura de las mismas.
- La segunda fase de implementación se corresponde con el desarrollo de la `ApplicationDiscoverer`. La funcionalidad implementada está formada por el registro de la aplicación en el móvil (mediante el método `PushRegistry`), la detección de etiquetas NFC, lectura de la información existente en las mismas, y gestión de dicha información. La gestión de la información incluye las operaciones de obtención de los datos necesarios de la etiqueta y del dispositivo móvil, la creación y envío de un link hacia el servidor (mediante el protocolo `http` y los servicios Web RESTful, la activación del servlet apropiado y búsqueda del servicio, a partir de los parámetros pasados en el link, en el Registro de Aplicaciones, el envío de la respuesta al dispositivo móvil en formato XML, y la posterior descarga del servicio elegido por el usuario final desde el Servidor de Aplicaciones
- La última fase de implementación consiste en crear una página Web que solicite el servicio que se desee que sea subido al Servidor de Aplicaciones, y solicite a su vez la correspondiente información del mismo (nombre, versión...).

4.2. TECNOLOGÍAS Y OTROS MÉTODOS

En el desarrollo del presente proyecto se han utilizado diversas tecnologías. A continuación se van a enumerar las más importantes, junto con una explicación de la funcionalidad que se ha implementado con ellas:

- **Java:** Se ha escogido este lenguaje de programación porque es el más adecuado para obtener los resultados deseados. Este lenguaje proporciona interfaces de gestión de etiquetas NFC –tanto escritura como lectura-, estructuras de manejo de datos para dichas etiquetas, gestión de transferencia de datos entre etiquetas NFC y otros dispositivos con soporte para NFC... Otra de las ventajas que proporciona Java dentro de su plataforma Java ME, es la interfaz que ofrece para crear aplicaciones para teléfonos móviles, o también llamados MIDlets. El lenguaje da soporte para la creación de MIDlets y su posterior gestión a lo largo del ciclo de vida de los mismos. Java, dentro de su plataforma Java EE, ofrece la posibilidad de crear servlets, que darán cobertura a peticiones recibidas en el lado del servidor y generarán las respuestas requeridas en cada caso.
 - **Servidor > MySQL y Apache Tomcat:** Si unimos a las posibilidades aportadas por Java, las funcionalidades de un servidor Apache Tomcat y una base de datos MySQL, obtenemos así todo lo necesario para la gestión del sistema del lado del servidor. Apache Tomcat alojará los servlets implementados, y MySQL la información almacenada, quedando así cubiertas las necesidades del sistema.
 - **Cliente > HTML y MIDlets:** Para implementar el lado del cliente, se ha utilizado HTML y hojas de estilos CSS para crear la página Web, y Java MIDlets, como ya se ha comentado, para la parte correspondiente al dispositivo móvil
- **API MIDP (Mobile Information Device Profile):** Es una plataforma dentro de Java ME. Provee soporte para interfaces gráficas, operaciones de red y operaciones de almacenamiento.
- **API CLDC (Connected Limited Device Configuration):** Provee una interfaz para gestionar aspectos de conexiones wireless.
- **API JSR-257 Contactless Communication:** Es el API que permite que una aplicación NFC, como lo es la `ApplicationDiscoverer`, pueda controlar la interfaz NFC. Esto permite integrar la tecnología NFC en entornos de desarrollo Java.
- **XML:** Se ha utilizado este formato para enviar datos desde el Servidor de Lookup al dispositivo móvil. En concreto, la información transmitida corresponde a los servicios encontrados disponibles para ser instalados.
- **Near Field Communication:** Es la tecnología sobre la que se desarrolla la totalidad del proyecto.
- **Especificaciones del NFC Forum:** El NFC Forum, como ya se comentó en los capítulos iniciales, han definido distintas especificaciones para hacer compatibles las distintas aplicaciones NFC diseñadas. Nuestro proyecto cumple con estas especificaciones y las ha seguido en todo momento, para así hacerlo también compatible con otros servicios NFC.

- **ISO 3166-1 e ISO 369-1:** Estándares que definen códigos para denominar países y sus lenguajes. Se han seguido ambos estándares para tener una única nomenclatura de dichos conceptos. Los parámetros lenguaje y país son parámetros que se leen del dispositivo móvil y se usan luego en la búsqueda del servicio apropiado.
- **PushRegistry:** Es la técnica utilizada para registrar la `ApplicationDiscoverer` en el dispositivo móvil. De esta manera, la aplicación es lanzada cada vez que se detecte una etiqueta NFC que coincida con la cadena de conexión establecida en el registro de la aplicación, mediante `PushRegistry`.
- **Servicios Web RESTful:** Arquitectura software que provee operaciones de envío de peticiones/respuestas entre un cliente y un servidor. Para el presente proyecto, las peticiones se han basado en el protocolo HTTP.

* Para ver ejemplos de código de las tecnologías anteriormente descritas, consultar el “anexo D. Implementation” al final del presente documento.

4.3. ESTRUCTURA DE LA IMPLEMENTACIÓN

Tomando como base el Diagrama de Clases elaborado en la fase de análisis, se han implementado los módulos que componen el sistema, siguiendo, como se ha explicado en la sección previa, la división del mismo en tres sub-sistemas bien definidos. A continuación se va a mostrar la estructura final que se ha obtenido en la implementación.

WRITENDEFMESSAGE

El primero de los subsistemas está desplegado en un único componente hardware: el dispositivo móvil con soporte para NFC, usado en el proyecto. Una etiqueta NFC también es necesaria para poder probar el funcionamiento del sub-sistema.

En cuanto a la estructura software, el sub-sistema está formado por un `MIDlet` encargado de detectar una etiqueta NFC vacía, y escribir en ella el mensaje NDEF que contiene la información necesaria para que pueda ser detectada posteriormente. Éste `MIDlet` está implementado sobre la plataforma Java ME. La clase que lo implementa extiende, como es natural, de la clase `MIDlet`, e implementa el interfaz `TargetListener`, que forma parte del paquete `javax.microedition.contactless`, perteneciente al API JSR-257 Contactless Communication. La implementación del método `targetDetected(TargetProperties[] properties)`, aporta la funcionalidad más destacable en este sub-sistema: una vez que el `MIDlet` esté en ejecución, éste método (`targetDetected`) se ejecutará en el momento en que una etiqueta NFC sea detectada. Así, el código implementado dentro del método provee operaciones de creación de un mensaje NDEF (mediante una estructura creada específicamente para este propósito, llamada `ENDEFRecord` –Extended NDEFRecord-) y escritura del mismo en la etiqueta, una vez que ésta ha sido detectada.

La clase `ENDEFRecord` extiende de `NDEFRecord`. Un objeto de la clase `NDEFRecord` está formado por un tres campos: `NDEFRecordType` –tipo de registro (en nuestro caso: `EXTERNAL_RTD`) y cadena de conexión (`urn:nfc:ext:ericsson.com:demo`)-, un segundo campo de tipo `byte[]` que representa el identificador del registro, y un tercer campo de tipo `byte[]` que representa el `payload` o información útil que contiene el registro. La estructura creada en la clase `ENDEFRecord` amplía esta disposición de información, creando, a su vez, una nueva estructura para el campo `payload`, que queda establecido como la unión de un campo `target`, un campo `action`, y un campo `data`. De esta manera, podemos escribir y leer información de las etiquetas (mensajes NDEF, formados por registros ENDEF), conociendo su estructura interna y siendo capaces de extraer la información que queremos gestionar.

Puesto que la transmisión de información debe hacerse mediante estructuras de bytes, también se han creado métodos específicos para convertir una secuencia de objetos en bytes, y viceversa. La implementación de esta funcionalidad puede consultarse en mayor profundidad en el “anexo D. Implementation”, al final de este documento.

Cabe puntualizar el porqué se ha creado esta versión extendida de un `NDEFRecord`. La respuesta reside en la necesidad de transmitir un mensaje NDEF formado por un registro, el cual cumpla con las especificaciones establecidas para esta tecnología. Así, el campo `payload` de un registro genérico está comprendido por los tres campos vistos en el párrafo anterior: `target`, `action`, `data`.

APPLICATIONDISCOVERER

El segundo de los sub-sistemas implementados es el que ha requerido un mayor esfuerzo y el que mayor número de componentes requiere para su correcto funcionamiento. Todo ello se describe en los siguientes párrafos.

`ApplicationDiscoverer` es un sistema cliente-servidor. En el lado del cliente, se cuenta con un `MIDlet` encargado de dirigir el proceso, implementado, al igual que el `MIDlet` anteriormente descrito, sobre la plataforma Java ME. Éste está desplegado en un dispositivo móvil. Una de las primeras acciones que deben llevarse a cabo es el registro del mismo en el dispositivo. Dada la configuración establecida en el `MIDlet`, esto se consigue durante el proceso de instalación de la aplicación, sin necesidad de configuraciones externas, por parte del usuario final. El registro del `MIDlet` se lleva a cabo mediante el método `PushRegistry`, el cual requiere como parámetros una cadena de conexión, el nombre del `MIDlet` y el grupo de usuarios a los que se les permite el uso de la aplicación. Lo que se consigue con este registro es dejar al `MIDlet` preparado para que, en el momento en que se lea una etiqueta que contiene la cadena establecida en el momento del registro, la aplicación sea lanzada de forma automática.

Tras el registro, la siguiente función a implementar es la detección de etiquetas. Para ello, la clase `ApplicationDiscoverer` debe implementar la interfaz `NDEFRecordListener`. La interfaz cuenta con el método `recordDetected(NDEFMessage ndefMessage)`, que se ejecuta al detectarse un registro (esto es, una etiqueta que contiene un `NDEFMessage` formado por registros). Esto es posible puesto que en el método constructor del `MIDlet` se ha añadido un `NDEFRecordListener` que permite la detección del registro. Así, hasta este punto tenemos implementada la detección de la etiqueta (gracias al método `PushRegistry`, que queda establecido al instalarse la aplicación) y la detección de la información en ella contenida (gracias al `Listener` y al método `recordDetected`).

A continuación, y tomando como base el mensaje NDEF obtenido de la etiqueta (nos lo proporciona el método `recordDetected(NDEFMessage ndefMessage)`), podemos extraer la información que necesitamos para la gestión del proceso. Esto se consigue desmontando la estructura de bytes creada por la aplicación `WriteNDEFMessage`. Lo que nos interesa en este momento es la información almacenada en el campo `target`. Junto a este dato, se añaden ciertos parámetros que recogemos del dispositivo móvil, como son el lenguaje y el país en el que se encuentra registrado el teléfono, y una vez obtenida esta información, enviamos una petición al Servidor de Lookup, en forma de link y mediante el protocolo HTTP (usando los métodos que implementa la arquitectura RESTful Web Services). La información se envía usando la conexión por defecto del teléfono móvil, que como es natural, debe ser una conexión WiFi.

El Servidor de Lookup tiene implementado un servlet, llamado `LookupService`. Éste está desarrollado sobre la plataforma Java EE. El servlet obtiene la petición enviada por el dispositivo móvil y comienza su proceso de búsqueda del servicio requerido. Para ello, establece una conexión con el Registro de Aplicaciones, el cual almacena la información referente a los servicios disponibles, y lleva a cabo una serie de operaciones contra la base de datos. El Servidor de Lookup obtiene los resultados, pudiendo éstos ser cero, uno o más servicios. Para poder gestionar este número variable de servicios, se ha optado por crear una estructura XML, la cual se envía en forma de cadena desde el Servidor de Lookup al dispositivo móvil del usuario final. Los resultados obtenidos se obtienen así en el dispositivo que inició el proceso, y tras ello, se procede a la interpretación y visualización de los mismos. Esto se consigue utilizando las funcionalidades que ofrece el paquete `kXML`, el cual se ha incluido en este sistema.

El usuario tiene entonces la posibilidad de seleccionar uno de los servicios mostrados, pudiendo obtener información detallada de cada uno de ellos. Una vez elegida una de las opciones mostradas, se procede a la descarga del servicio. Un nuevo componente entra en juego, el Servidor de Aplicaciones, el cual tiene almacenados los servicios disponibles para ser descargados. Así, se establece una nueva conexión entre el dispositivo móvil y el servidor, y mediante el método `platformRequest(String url)`, perteneciente al paquete `javax.microedition.midlet.MIDlet`, el servicio se descarga al teléfono del usuario. Tras esto, el servicio podrá ser instalado (de forma automática, previa confirmación del usuario) y ejecutado, obteniendo así el objetivo deseado: la interacción con la etiqueta NFC leída al inicio del proceso.

Al hablar de la `ApplicationDiscover`, es imprescindible destacar la rapidez con la que el servicio es llevado a cabo, la cual, si no tenemos en cuenta el tiempo destinado por el usuario a la selección del servicio (puesto que éste es un tiempo variable y no controlable por el sistema), se ejecuta de forma total en un tiempo inferior a 1 minuto. Este hecho permite que el usuario no sienta la necesidad de abortar el proceso y la operación se pueda culminar de forma satisfactoria.

PÁGINA WEB

El último sub-sistema implementado tiene una arquitectura cliente-servidor. El cliente puede ser cualquier dispositivo con acceso a Internet mediante un navegador Web. En él se mostrará la página implementada, la cual ha sido creada con código HTML y validaciones Javascript. Éstas aportan la funcionalidad requerida para asegurarnos que el usuario introduce la información adecuada en cada caso.

Tras introducir dicha información y seleccionar la localización del fichero o los ficheros que contienen el servicio deseado, se procede a almacenar este contenido en

los servidores del sistema. Así, la petición es recogida por el Servidor de Lookup – implementado sobre la plataforma Java EE-, el cual responde mediante un servlet creado para este fin, llamado Upload. El servlet procesará la información introducida y la almacenará en el Registro de Aplicaciones. De la misma forma, recogerá los archivos cargados y los almacenará en el Servidor de Aplicaciones.

4.4. TESTS DE USABILIDAD

En un sistema como el que se está desarrollando en este proyecto, es aconsejable que los usuarios finales formen parte del proceso de desarrollo, pudiendo aportar sus ideas y mejoras, puesto que, una vez implementado el sistema, serán ellos los que van a hacer uso del mismo.

Es por ello que se ha creado una serie de pruebas de usabilidad, para las que se ha seleccionado a un grupo de usuarios que cumplen el perfil de usuarios finales. Estas pruebas se han llevado a cabo en dos fases del proyecto: cuando se tuvo un primer prototipo del sistema, y cuando el sistema estuvo finalizado. La primera de ellas proporcionó feedback al sistema, pudiendo así modificar, mejorar y adaptar las debilidades encontradas. La segunda fase de evaluación indicó si los objetivos de cara al usuario final y a la interacción del mismo con la aplicación se cumplieron, y si el usuario ha quedado satisfecho con el sistema desarrollado.

Las conclusiones obtenidas tras realizar los tests de usabilidad establecidos, se pueden leer a continuación.

TESTS DE USABILIDAD REALIZADOS TRAS OBTENER UN PRIMER PROTOTIPO

Una vez obtenido un primer prototipo (previamente estudiado y evaluado por la parte desarrolladora), se llevó a cabo una primera fase de evaluación y, a pesar de que en ella se intentó crear un escenario real, en el que el usuario se pudiera situar y ambientar, esto no se llegó a conseguir, puesto que una gran parte de los evaluadores no consiguieron situarse en el contexto del problema.

Al margen de esta desventaja y lo que ello pudo afectar en relación con otros aspectos, el resto de cuestiones planteadas en el test de usabilidad dieron resultados positivos al prototipo obtenido.

TESTS DE USABILIDAD REALIZADOS TRAS OBTENER EL PRODUCTO FINAL

Una segunda fase de evaluación llevada a cabo al final de la etapa de implementación, aportó conclusiones positivas, dejando patente un claro interés por parte de los evaluadores hacia el servicio testado.

* En la sección “B.7. Evaluation tests” del “anexo B. Analysis” del presente documento se puede ver el test de usabilidad realizado a los usuarios finales.

5. CONCLUSIONES

Finalizado el desarrollo del presente proyecto se puede concluir que la valoración final es muy positiva. Los objetivos funcionales propuestos al inicio del proyecto se han cumplido de la forma deseada. La funcionalidad buscada, el hecho de facilitar al usuario final un marco que le permita la interacción con todo tipo de dispositivos NFC, ha sido implementada, testada y evaluada como positiva. Cabe destacar este hecho puesto que, personas implicadas directamente en el proyecto, mencionaron al inicio del mismo que sería “normal” no encontrar una solución al problema planteado, dentro del tiempo establecido hasta la conclusión del proyecto.

Por otra parte, los tiempos dedicados a cada una de las fases, y las horas finales invertidas, son las adecuadas para un proyecto software. Éstas se podían haber mejorado en etapas iniciales, puesto que costó entender el funcionamiento de ciertos aspectos de la tecnología NFC, pero aún con ello, el resultado es positivo. Hay que tener en cuenta que el alumno se enfrentaba por primera vez al uso de NFC, a su estudio y comprensión, y puede que este hecho alargara ligeramente la fase inicial de estudio previo y conocimiento de las tecnologías a utilizar.

Con todo ello, se ha conseguido solventar el vacío encontrado en la tecnología NFC en lo que se refiere a la interacción con etiquetas NFC que requieren un servicio específico para su lectura, por parte de un dispositivo móvil con soporte para NFC. Éstas, salvando las limitaciones detalladas al inicio del documento, podrán ser leídas bajo todos los conceptos.

Con la instalación de una única aplicación, como es `ApplicationDiscoverer`, el usuario podrá acceder a múltiples servicios requeridos por ciertas etiquetas, para que éstas puedan interaccionar con el dispositivo móvil. Este hecho aporta grandes ventajas de cara al usuario final, abriendo un gran abanico de posibilidades en el mundo de las interacciones entre dispositivos NFC.

Como opinión personal, debo destacar que he quedado muy satisfecha con el trabajo realizado, con el proceso llevado a cabo para obtenerlo y los resultados finales obtenidos. He conocido a fondo las capacidades y características de una nueva tecnología, lo cual valoro como favorable y muy interesante. Otro punto destacable, en lo personal, es el haber desarrollado el proyecto, de forma favorable, en un país extranjero, de habla sueca e inglesa. Por último, concluir con el grado de satisfacción que supuso conocer la noticia de que el presente proyecto va a ser incluido en una plataforma de mayor alcance, y será utilizado por la empresa en la que ha sido desarrollado. Ello ha supuesto un perfecto punto final para la conclusión del proyecto.



REFERENCIAS BIBLIOGRÁFICAS

[1] Introducción a J2ME.

<http://www.todosymbian.com/secartprint36.html>

[2] Paquete javax.microedition.contactless.

<http://mobilezoo.biz/jsr/257/javax/microedition/contactless/package-summary.html>

[3] Sun Microsystems. FAQs. How can a MIDlet be launched automatically?

<http://developers.sun.com/mobility/midp/questions/pushregistry/>

[4] Sun Microsystems. The MIDP 2.0 Push Registry.

<http://developers.sun.com/mobility/midp/articles/pushreg/>

[5] Library of congress. Codes for the representation of names of languages.

http://www.loc.gov/standards/iso639-2/php/code_list.php

[6] Sun Microsystems The Java servlet API.

<http://java.sun.com/developer/onlineTraining/Servlets/Fundamentals/servlets.html>

[7] Sun Microsystems. FAQs. What are the defined J2ME system property names?

<http://developers.sun.com/mobility/midp/questions/properties/index.html>

[8] Sun Microsystems. Parking XML in J2ME.

<http://developers.sun.com/mobility/midp/articles/parsingxml/>

[9] Esidea Grupo Tecnológico. Alto nivel – bajo nivel.

http://www.esidea.com/index.asp?pagina_e=noticias&mas=15

[10] Nokia Forum. Design and User Experience Library

[11] NFC Forum

<http://www.nfc-forum.org/home>

[12] John W. Muchow, “Core J2ME. Technology and MIDP”, 2002.

[13] Diferentes consultas:

<http://www.wikipedia.org>

[14] Consultas sobre J2ME y dispositivos móviles.

<http://www.forum.nokia.com>

[15] Java Community Process.

<http://jcp.org/aboutJava/communityprocess/edr/jsr257/index.html>

[16] NFC Forum. Technical specifications.

<http://www.nfc-forum.org/specs/>

[17] The Apache Software Foundation

<http://tomcat.apache.org/>

[18] Sun Microsystems. MySQL.

<http://www.mysql.com/>

[19] kXML 2.

<http://kxml.sourceforge.net/kxml2/>

[20] Nokia. Developer Discussion Boards.

<http://kxml.sourceforge.net/kxml2/>

[21] J2ME PushRegistry. Activación automática de MIDlets.

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=pushRegistry>

[22] Collin Mulliner, "Attacking NFC Mobile Phones", May 2008.

mulliner.org/nfc/feed/collin_mulliner_eusecwest08_attacking_nfc_phones.pdf

[23] ABI Research. NFC.

[www.abiresearch.com/research/1003525-Near+Field+Communication+\(NFC\)](http://www.abiresearch.com/research/1003525-Near+Field+Communication+(NFC))

[24] Java Community Process. The Java API for RESTful Web Services

<http://jcp.org/aboutJava/communityprocess/final/jsr311/index.html>

[25] Universidad de Deusto. Near Field Communication.

<http://www.morelab.deusto.es/images/talks/NFC.ppt>

[26] Innovision, creators of RFID systems. NFC Forum Standards and Tags, Abril 2007.

<http://www.enamax.net/wima/pics/File/confs/Heikki%20Huomo.pdf>

[27] Nokia. Touch a Phone, Touch a Friend: Using RFID and Visual Tags with JSR 257

<http://docs.huihoo.com/javaone/2006/JAVA%20ME/ts-3789.pdf>