

PROTOTIPO DE APLICACIÓN WEB QUE PERMITE EL MONITOREO DE LA
UBICACIÓN Y FLUJO DE VISITANTES DENTRO DE UN EDIFICIO, UTILIZANDO
IDENTIFICACIÓN POR RADIO FRECUENCIA (RFID)

ANGEL LEONARDO MERCHAN MONROY

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA

2013

PROTOTIPO DE APLICACIÓN WEB QUE PERMITE EL MONITOREO DE LA
UBICACIÓN Y FLUJO DE VISITANTES DENTRO DE UN EDIFICIO, UTILIZANDO
IDENTIFICACIÓN POR RADIO FRECUENCIA (RFID)

ANGEL LEONARDO MERCHAN MONROY

Trabajo de grado para optar al título de
Ingeniería de Sistemas

Director:

MSC. FERNANDO ROJAS MORALES

Codirector

MSC(c). IRENE LIZETH MANOTAS GUTIÉRREZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA

2013

A mi hermosa familia, por estar siempre a mi lado a pesar de todo; a mi Mamá Celina por enseñarme e inculcarme la paciencia y amor por lo que hago; a mi Papá Aristides por enseñarme que a pesar de lo difícil de cualquier situación, siempre hay que seguir adelante con entusiasmo y buenas energías para lograr el éxito; a mi hermano Sergio y a mi hermanita Noelia por darle esa alegría y diversión a mi vida, que aunque no lo demuestro mucho, me encanta.

A mis amigos y hermanos Arley y Jesús, por todos esos momentos compartidos durante los años, por siempre estar ahí cuando los necesité y por ponerle siempre una cuota de cordura y alegría a la vida.

A Juli por todo su apoyo y amor durante estos años, siempre estuviste presente en el momento justo, gracias por compartir los malos, buenos y excelentes momentos todo este tiempo, gracias por entender y compartir mis locuras.

A la Familia Avila Vargas, por ser mi familia adoptiva en Bucaramanga, por estar siempre pendientes de mi, por su apoyo, consejos y alegría.

A mi amiga Cindy, por aguantar mis locuras, por apoyarme en las buenas y en las malas, por sus consejos en momentos difíciles y por su alegría pesar de lo complicado de la situación.

A mis amigos de carrera, Laura, Leidy, Betty, Gina y Mario, porque con su perseverancia, constancia y convicción siempre fueron un ejemplo a seguir; gracias por su amistad, apoyo y alegría en todo momento.

Angel Leonardo Merchan

AGRADECIMIENTOS

Por todo el apoyo recibido durante el proceso de desarrollo de este proyecto, quiero agradecer:

Al profesor FERNANDO ROJAS MORALES, director de mi proyecto, por la confianza brindada, por los consejos, ideas, conocimientos y paciencia que compartió conmigo durante todo este proceso.

A la Ingeniera IRENE LIZETH MANOTAS, codirectora del mi proyecto, por el tiempo y apoyo brindado, por los conocimientos compartidos en el área de desarrollo de software.

Al Ingeniero HUGO ARMANDO GUALDRÓN, por su amistad, consejos y apoyo incondicional durante el desarrollo del proyecto, podría decirse que fue como mi segundo codirector.

A toda mi familia por estar siempre presentes, por acompañarme en los buenos y malos momentos, por el apoyo incondicional y por el amor que siempre me han dado.

A todos mis amigos y compañeros de carrera por compartir esta experiencia y estar siempre presentes en las buenas y en las malas.

Al Grupo de Investigación en Ingeniería Biomédica GIIB, por todos los conocimientos, apoyo y amistad.

RESUMEN

Título: PROTOTIPO DE APLICACIÓN WEB QUE PERMITE EL MONITOREO DE LA UBICACIÓN Y FLUJO DE VISITANTES DENTRO DE UN EDIFICIO, UTILIZANDO IDENTIFICACIÓN POR RADIO FRECUENCIA (RFID)*.

Autor: Angel Leonardo Merchan Monroy**.

Palabras Claves: Tecnología de Identificación por Radio Frecuencia (RFID), rastreo y monitoreo de personas, emulación de dispositivos, plataforma de desarrollo Rifidi, Java Message Service (JMS).

Descripción: Este proyecto presenta el desarrollo de un prototipo de aplicación web, que basada en la tecnología de identificación por radiofrecuencia, permite llevar a cabo el monitoreo de la ubicación y flujo de visitantes dentro de un edificio.

El desarrollo de la aplicación es el punto de partida para el desarrollo de nuevas herramientas software basadas en la Tecnología de Identificación por Radio Frecuencia dentro de la Universidad Industrial de Santander. Esta tecnología se presenta como una solución eficiente en los procesos de identificación, monitoreo y control de objetos dentro de las organizaciones.

Para llevar a cabo el desarrollo de la aplicación se utilizó la plataforma de código abierto Rifidi que permite llevar a cabo la emulación de los dispositivos electrónicos necesarios para el funcionamiento del sistema, de igual forma permite la realización de pruebas de funcionamiento del sistema mediante la simulación de un flujo de personas a través del edificio.

La aplicación desarrollada basa su lógica de funcionamiento en las reglas del flujo de personas definidas para un edificio específico, el Centro de Tecnologías de Información y Comunicación (CENTIC). Dichas reglas son explicadas claramente en este documento y fueron definidas por el autor del proyecto, de acuerdo a la visión general del funcionamiento del edificio.

Las pruebas realizadas a la aplicación contemplan la verificación de la emulación de los dispositivos utilizados, la correcta conexión de estos dispositivos con el servidor encargado de procesar la información y el adecuado filtrado y envío de información del servidor a la base de datos y a la web para su presentación.

*Trabajo de Grado

**Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director: MSC. Fernando Rojas Morales. Codirector: MSC. Irene Lizeth Manotas Gutiérrez

ABSTRACT

Title: PROTOTYPE OF WEB APPLICATION THAT ALLOWS MONITORING THE LOCATION AND FLOW OF VISITORS IN A BUILDING, USING RADIO FREQUENCY IDENTIFICATION (RFID) *.

Author: Angel Leonardo Merchán Monroy**.

Keywords: Radio Frequency Identification Technology (RFID), Tracking and monitoring of people, device emulation, Rifidi development platform, Java Message Service (JMS).

Description: This project the development of a prototype web application, which based on RFID technology, allow to carry out the monitoring of the location and flow of visitors within a building.

This application development is the starting point for the development of new software tools based on Radio Frequency Identification technology within the Industrial University of Santander. This technology takes more strength every day and is presented as an efficient solution in the processes of identifying, monitoring and control of objects within organizations.

To carry out the development of the application is used Rifidi open source platform. This platform allows to make the emulation of electronic devices required to operate the system, in the same way allows performance testing of the system by creating a flow of people through the building.

The application developed, bases it's operating logic on rules the flow people rules defined for an specific building, the Center of Information and Communication Technologies (CENTIC). These rules are clearly explained later in this document and were defined by the bill's author, according to the overview of the operation of the building.

The testing of the application includes the verification of the emulation of the devices used, the correct connection of these devices to the server responsible for processing information and the appropriate filtering and sending information from the server to the database and it's persentation on the web.

* Degree work

**Faculty of Mechanical Engineering Physic. School of Systems Engineering.

Director: MSC. Fernando Rojas Morales. Codirector: MSC. Irene Lizeth Manotas Gutiérrez

CONTENIDO

INTRODUCCIÓN.....	12
CAPITULO 1	14
1. DESCRIPCIÓN DEL PROYECTO	14
1.1 OBJETIVOS.....	14
1.1.1. Objetivo general:.....	14
1.1.2. Objetivos Específicos:	14
1.2. JUSTIFICACIÓN Y DEFINICIÓN DEL PROBLEMA	15
CAPITULO 2	17
2. MARCO DE REFERENCIA.....	17
2.1. ANTECEDENTES	17
2.2. MARCO TEÓRICO	19
2.2.1. Identificación por Radio Frecuencia (RFID).....	19
2.2.1.1. ¿Qué es un sistema RFID?.....	19
2.2.1.2. ¿Cómo funciona?	20
2.2.2. RIFIDI	23
2.2.2.1. ¿Qué es Rifidi?.....	23
2.3. PROCESAMIENTO DE EVENTOS COMPLEJOS (CEP).....	28
2.4. MOTOR DE PROCESAMIENTO DE EVENTOS COMPLEJOS “ESPER”	30
2.6. JAVA MESSAGE SERVICE (JMS).....	33
2.7. APLICACIONES DIRIGIDAS POR EVENTOS	34
CAPITULO 3	37
3. PROTOTIPO WEB QUE PERMITE EL RASTREO DE PERSONAS DENTRO DE UN EDIFICIO – GIIBPT (GIIB People Tracker)	37
3.2. DEFINICIÓN Y PRESENTACIÓN DEL ESPACIO FÍSICO	39
3.1.2 Organización del espacio físico	41
Figura 10. Distribución de las áreas de monitoreo en el edificio	41
3.3. POSICIONAMIENTO DE LOS LECTORES DENTRO DEL EDIFICIO.....	42
3.4. DEFINICIÓN DE LAS REGLAS Y EVENTOS PARA EL FLUJO DE PERSONA	44
3.5. CONFIGURACIÓN DEL SERVIDOR – RIFIDI EDGE SERVER.....	47

3.5.1.	Creación de una configuración de lectores dentro del servidor.....	47
3.5.2.	Creación de una sesión de comunicación para los lectores	48
3.5.3.	Creación y configuración de un comando para la lectura de las etiquetas RFDI.....	48
3.5.4.	Asignación y ejecución del comando Alien_Poll_1 a cada lector	49
3.5.5.	Inicio de sesión para los dispositivos de lectura RFID.....	50
3.6.	CONFIGURACIÓN Y DESPLIEGUE DE LOS LECTORES EN EL TAG STREAMER	51
3.7.	PROGRAMACIÓN DE LAS REGLAS DE FILTRADO DE INFORMACIÓN EN ESPER – EPL.....	55
3.8.	ADMINISTRACIÓN DE LAS CONSULTAS EN ESPER Y ENVÍO DE LA INFORMACIÓN AL SERVIDOR WEB	58
3.9.	SERVIDOR WEB Y APLICACIÓN GIIBPT	59
3.9.1.	Paquete com.giibpt.rifidi: Comunicación ente servidores y validación de flujo de visitantes 60	
3.9.2.	Paquete com.giibpt.model: Paquete de gestión de base de datos	62
3.9.3.	Paquete com.giibpt.model.entities: Definición del modelo y arquitectura de la base de datos 64	
3.9.4.	Paquete com.giibpt: Presentación de la información en la página Web.....	66
3.9.4.1.	Funcionalidad de rastreo	68
3.9.4.2.	Consulta de historiales de eventos en el sistema	70
3.9.4.3.	Consulta de historiales de alertas en el sistema.....	71
3.9.4.4.	Administración de visitantes y asignación de etiquetas RFID.....	73
	CAPITULO 4	75
4.	VERIFICACIÓN DEL DESEMPEÑO Y CORRECTO FUNCIONAMIENTO DEL PROTOTIPO SOFTWARE 75	
	CAPITULO 5	77
5.	CONCLUSIONES	77
	CAPITULO 6	80
6.	RECOMENDACIONES	80
	Referencias.....	81
	BIBLIOGRAFÍA.....	82

LISTA DE FIGURAS

	Pág.
Figura 1. Lector RFID	21
Figura 2. Etiqueta RFID.....	22
Figura 3. Arquitectura de un sistema RFID.....	22
Figura 4. Componentes de Rifidi	24
Figura 5. Conjunto de aplicaciones de Rifidi Toolkit.....	24
Figura 6. Diagrama de Arquitectura de Edge Server	26
Figura 7. Arquitectura y flujo de datos en Esper	31
Figura 8. Arquitectura General del sistema GIIBPT	38
Figura 9. Distribución del espacio físico para el primer piso del CENTIC.....	40
Figura 10. Distribución de las áreas de monitoreo en el edificio.....	41
Figura 11. Distribución de los lectores RFID dentro del edificio	43
Figura 12. Interfaz gráfica del Tag Streamer de Rifidi que permite la creación de lectores.	51
Figura 13. Interfaz Gráfica que permite la configuración de los lectores RFID.....	52
Figura 14. Interfaz gráfica del Tag Streamer con lectores y etiquetas creadas y la distribución de los lectores en el área del edificio.....	53
Figura 15. Método createWindow para la creación de ventanas en Esper.....	55
Figura 16. Asignación de lectores RFID a la ventana correspondiente (imagen parcial) ...	56
Figura 17. Método para la eliminación de eventos dentro de una ventana	57
Figura 18. Estructura de las consultas EPL que obtienen la información de cada ventana.	57
Figura 19. Estructura del método send()	58
Figura 20. Diagrama de clases paquete com.giibpt.rifidi	61
Figura 21. Diagrama de clases del paquete com.giibpt.model	63
Figura 22. Modelo Entidad/Relación de la base de datos del sistema GIIBPT	64
Figura 23. Diagrama de clase del paquete com.giibpt.model.entities	65
Figura 24. Diagrama de clases del paquete com.giibpt	66
Figura 25. Vista inicial de la aplicación web.....	68
Figura 26. Vista del módulo de rastreo en la aplicación Web	69
Figura 27. Interfaces de consulta de eventos basado en ID de Visitante y en Área	70
Figura 28. Vista de la interfaz de consulta de Alertas	72
Figura 29. Vista de la interfaz de administración de Visitantes.....	73

INTRODUCCIÓN

Aunque a simple vista podría pensarse que la tecnología de Identificación por Radio Frecuencia (RFID) es una tecnología relativamente nueva, tuvo sus inicios durante la segunda guerra mundial basándose en la utilización de radares que transmitían una señal y dispositivos receptores instalados en los aviones de combate que modificaban y reflejaban esta señal permitiéndoles de esta forma identificarse como amigos.

Actualmente, aunque no nos demos cuenta, vivimos rodeados de estos dispositivos de identificación por radiofrecuencia. La tecnología RFID es usada por gran cantidad de aplicaciones dirigidas principalmente a prevenir el robo de automóviles y mercancía, al control de pasajeros en estaciones de transporte, rastreo de libros en bibliotecas y control de paquetes durante la distribución en una cadena de suministro o durante un proceso de manufactura. (1)

Dentro de la Universidad Industrial de Santander son muy pocos los proyectos de desarrollo de software que utilizan la Tecnología de Identificación por Radio frecuencia como base fundamental de su funcionalidad. Por esta razón y debido a la gran acogida que ha teniendo esta tecnología en muchos sectores de la economía, se lleva a cabo el desarrollo de este proyecto, con el fin de aportar información útil a la comunidad académica acerca de esta tecnología y de la plataforma de desarrollo Rifidi utilizada en el proceso de codificación y pruebas del sistema.

RFID funciona básicamente mediante el envío de una señal de radio frecuencia por parte de una fuente; la cual es recibida, modificada y retransmitida o reflejada

por un aparato receptor, de vuelta hacia la fuente con el fin de informar la presencia de éste y suministrar información relevante del objeto rastreado, como un código de identificación, nombre del objeto, y cualquier otro tipo de información que pueda llegar a ser importante para el sistema.

El funcionamiento de la rfid depende principalmente de 2 capas, hardware y software.

En la capa de hardware se encuentran dos tipos de dispositivos, los lectores y los receptores o tags. Los lectores son los dispositivos encargados de generar y posteriormente recibir de vuelta una señal de radio reflejada por los dispositivos receptores. Los receptores o tags, son dispositivos electrónicos sensibles a las frecuencias de onda generadas por los lectores y que se encargan de reflejar y en ocasiones de transmitir información adicional hacia el lector.

En la capa software se encuentra el sistema de control de los dispositivos hardware, este se encarga básicamente de modelar reglas del negocio establecidas por la organización con el fin de condicionar el comportamiento de los lectores y filtrar de forma adecuada la información transmitida, ayudando así al correcto desarrollo de un proceso específico dentro de una organización. El sistema monitorea, analiza, notifica y almacena todos los eventos que ocurren relacionados con los tags y las reglas del negocio definidas previamente por la organización.

La utilización de esta tecnología agiliza en gran medida procesos de control y monitoreo dentro de una organización. La automatización de estos procesos reduce considerablemente los costos y aumentan la productividad de la organización.

CAPITULO 1

1. DESCRIPCIÓN DEL PROYECTO

1.1 OBJETIVOS

1.1.1. Objetivo general:

Desarrollar un prototipo de aplicación web que permita el monitoreo de la ubicación y flujo de los visitantes dentro de un edificio, basándose en la tecnología RFID y la plataforma de código abierto Rifidi.

1.1.2. Objetivos Específicos:

- Analizar y definir claramente las reglas del negocio para el flujo de personas dentro de un edificio para su posterior aplicación en el desarrollo del prototipo software.
- Realizar el análisis, diseño e implementación de un prototipo de aplicación web para el monitoreo de la posición de personas dentro de un edificio, basándose en las reglas definidas.
- Realizar la emulación de todos los dispositivos electrónicos de tecnología RFID necesarios para el funcionamiento del sistema, mediante la utilización de la plataforma de código abierto Rifidi.
- Verificar el desempeño y correcto funcionamiento del software desarrollado en cuanto al cumplimiento de la lógica y reglas del negocio definidas y a la presentación de la información al usuario de la aplicación en tiempo aproximadamente real.

1.2. JUSTIFICACIÓN Y DEFINICIÓN DEL PROBLEMA

El monitoreo de la posición de visitantes y el control de acceso de los mismos a las diferentes instancias de un edificio, siempre ha sido uno de los aspectos importantes a tener en cuenta dentro de los protocolos de seguridad de muchas empresas, principalmente bancos, museos, centros de investigación y laboratorios.

Para este propósito la plataforma de desarrollo de código abierto Rifidi, ofrece un excelente conjunto de herramientas que permite el diseño, modelado, simulación y desarrollo de aplicaciones software basadas en la tecnología RFID. Esta plataforma es actualmente líder en la industria de herramientas de prototipado para aplicaciones que utilizan la tecnología RFID. (2)

Utilizando la plataforma de software Rifidi, es posible desarrollar una aplicación software basada en la tecnología RFID, sin necesidad de tener contacto real con los dispositivos hardware que esto implicaría y que además, sería un costo innecesario para el desarrollo de un prototipo.

El gran avance de la tecnología en las últimas décadas ha dado un gran impulso a la utilización de esta tecnología en diferentes campos de la industria. Aprovechando estos avances y más específicamente las herramientas existentes para el desarrollo de aplicaciones software basadas en RFID, es importante sentar precedente y comenzar con el estudio de esta tecnología y el desarrollo de este tipo de aplicaciones analizando los alcances que podría llegar a tener dentro del contexto de la universidad Industrial de Santander.

En la Universidad Industrial de Santander existen edificios, cuyas características físicas y sobre todo el tipo de actividades realizadas dentro, permitirían el

planteamiento y desarrollo de un sistema basado en RFID, que sirva como apoyo al desarrollo de dichas actividades; este es el caso del edificio CENTIC, en donde se presenta una gran afluencia de visitantes, hay una gran cantidad de dispositivos electrónicos y de computo y sobre todo existen áreas de acceso restringido a visitantes y que necesitan estricto control de acceso.

Para este proyecto, el monitoreo de personas dentro de un edificio se hace con fines netamente experimentales, buscando desarrollar conocimientos y experiencia en la creación de aplicaciones basadas en tecnología RFID.

La creación de un prototipo basado en esta tecnología abre las puertas del conocimiento en un campo no explorado dentro de la Universidad y que podría ser fuente de grandes proyectos en el futuro.

CAPITULO 2

2. MARCO DE REFERENCIA

2.1. ANTECEDENTES

Por lo general se dice que las raíces de la tecnología de identificación por radiofrecuencia, se remontan a la Segunda Guerra Mundial. Los Alemanes, Japoneses, Americanos y Británicos, utilizaban un radar (el cual había sido descubierto en 1935 por el físico Escocés Sir Robert Alexander Watson-Watt) para alertar del acercamiento de aviones cuando estos se encontraban aún a kilómetros de distancia. El problema era que no había forma de identificar si los aviones eran enemigos o si eran pilotos del propio país que regresaban de una misión.

Los Alemanes descubrieron que si los pilotos hacían girar sus aviones al regresar a la base, esto podía cambiar la señal de radio reflejada. Este rudimentario método alertaba a los encargados del radar de que se trataba de aviones alemanes y no de aviones enemigos (esto fue en esencia, el primer sistema RFID pasivo). El funcionamiento de RFID se basa en ese mismo concepto, se envía una señal a un transpondedor, que se activa y devuelve la señal (sistema pasivo) o emite una señal (sistema activo).

Los avances en el radar y en el sistema de comunicación por radio frecuencia continuaron durante los años 1950 y 1960. Los científicos y académicos en Estados Unidos, Europa y Japón hicieron una investigación y presentaron artículos

explicando cómo la energía de radio frecuencia puede ser usada para identificar objetos a distancia. Las empresas comenzaron a comercializar usando sistemas anti-robo que utilizan onda de radio para determinar si un artículo había sido pagado o no. Las etiquetas de vigilancia electrónica para artículos, que todavía se utilizan en los empaques de hoy en día, tienen una etiqueta de 1 bit. Este bit está encendido o apagado. Si alguien paga el producto, el bit está apagado y la persona puede salir del almacén. Pero si la persona no paga y trata de salir de la tienda, los lectores de la puerta pueden detectar la etiqueta y hacer sonar una alarma.

La primera patente registrada por los avances en RFID, fue para Mario W. Cardullo, quién patentó una etiqueta de RFID con memoria reescribible, el 23 de enero de 1973.

Ese mismo año, Charles Watson, un empresario de California recibió una patente para un transpondedor pasivo utilizado para abrir una puerta sin la ayuda de una llave. Una tarjeta con un transpondedor integrado, envía una señal a un lector cerca de la puerta. Cuando el lector detecta un número de identidad válido, almacenado dentro de la etiqueta RFID, el lector abre la puerta.

En Febrero de 2005, fue originalmente concebida la plataforma de desarrollo Rifidi. Un grupo de desarrolladores, estaba intentando implementar un sistema RFID para un cliente importante. El equipo quedó completamente aturdido por la complejidad de la tarea, y solamente estaba utilizando 10 lectores. A pesar de que contaban con un equipo muy completo y una gran cantidad de expertos en RFID y J2EE, estaban teniendo problemas con el desarrollo y prueba de la aplicación. Al terminar este proyecto, el equipo se puso en contacto con varias compañías de software y expertos en RFID, todos coincidieron en que una herramienta de simulación sería útil para realizar las pruebas. Desafortunadamente ninguno de ellos estaba dispuesto en desarrollar una aplicación de este tipo. Este problema se

hizo crítico cuando el equipo se enfrentó con otro cliente que les hacía las mismas peticiones. Dado que no había solución a la vista, decidieron hacer algo al respecto y comenzaron oficialmente el desarrollo de Rífid en marzo de 2006, con un pequeño pero dedicado equipo de desarrolladores a tiempo completo y parcial y un grupo de analistas de RFID.

2.2. MARCO TEÓRICO

2.2.1. Identificación por Radio Frecuencia (RFID)

RFID es una tecnología que se ha venido posicionando en muchas empresas y organizaciones gracias a los grandes avances tecnológicos y a la reducción de costos que esto implica. Es una tecnología que ofrece facilidades para su utilización en distintos tipos de aplicaciones, dependiendo del tipo de empresa, de las necesidades que esta tenga, y sobre todo a las reglas del negocio manejadas por la organización.

2.2.1.1. ¿Qué es un sistema RFID?

RFID significan Radio Frequency Identification, en español identificación por radio frecuencia. Se trata de un sistema de almacenamiento y recuperación de datos remotos que emplea dispositivos electrónicos denominados etiquetas RFID o tags y lectores RFID que se comunican mediante ondas de radiofrecuencia.

“RFID es sin duda un paso más hacia delante para las tecnologías de identificación automática que cada día se convierte más en la mejor alternativa al código de barras.” (3)

2.2.1.2. ¿Cómo funciona?

El funcionamiento del sistema a grandes rasgos es bastante sencillo, el lector RFID está conectado a una antena que envía una serie de ondas de radiofrecuencia al tag, que son captadas por la micro-antena de este. Dichas ondas activan el microchip, el cual a través de la micro-antena y mediante ondas de radiofrecuencia, transmite a la antena del lector la información que tenga almacenada en su memoria. Finalmente el lector recibe la información que tiene el tag y la envía a una base de datos en la que previamente se han registrado las características del producto o puede procesar dicha información según convenga a cada aplicación. (4)

Existen dos tipos de etiquetas RFID, activas y pasivas.

La etiqueta pasiva no posee alimentación eléctrica, por lo cual pueden resultar muy pequeñas. La señal que le llega de los lectores induce una corriente eléctrica pequeña y suficiente para operar el circuito integrado CMOS del tag, de forma que puede generar y transmitir una respuesta, esta puede ser de cualquier tipo, no sólo un código identificador. Los tags pasivos pueden tener distancias de uso práctico comprendidas entre 10 cm y llegadas hasta unos pocos metros, según la frecuencia de funcionamiento y el diseño y tamaño de la antena. (5)

En el caso de un sistema RFID pasivo, la etiqueta o tag se activa cuando pasa a través del campo de radio frecuencia generado por el lector, la etiqueta envía la información solicitada, esta información es detectada por la antena del lector. El lector capta la información y se la envía al middleware o subsistema de procesamiento de datos.

Un sistema RFID consta de tres componentes:

- **Lector de RFID:** el lector está compuesto por una antena, un transceptor y un decodificador. Por medio de la antena el lector envía información codificada a través de ondas de radio frecuencia. Esta información es recibida por una etiqueta RFID, la cual devuelve una nueva señal al lector como respuesta. Cuando éste capta la señal de una etiqueta, extrae la información y se la pasa al subsistema de procesamiento de datos. Estas señales podrían transmitir energía en caso de que se trate de leer una etiqueta pasiva. En algunos casos es posible escribir información en las etiquetas por medio de la señal enviada por el lector a estas; esto depende de si la tecnología utilizada tanto en el lector como en la etiqueta lo permiten (Ver Figura 1).

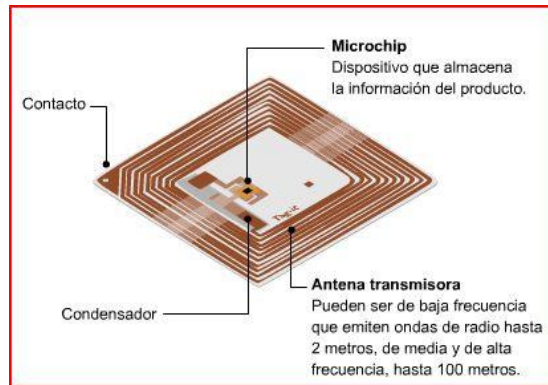
Figura 1. Lector RFID



Fuente: Alien Technology - Products - RFID Readers. [En línea] [Citado el: 15 de 04 de 2010.] <http://www.alientechnology.com/readers/alr9800.php>.

- **Etiqueta RFID, Transpondedor o Tag:** está compuesta por una antena, un transductor de radio y un microchip (Figura 2). El propósito de la antena es permitir al chip transmitir la información de identificación de la etiqueta. El chip es cada vez más pequeño e imperceptible para el ojo humano, posee una memoria interna para almacenar el número de identificación y en algunos casos datos adicionales, dicha memoria depende del modelo del dispositivo.

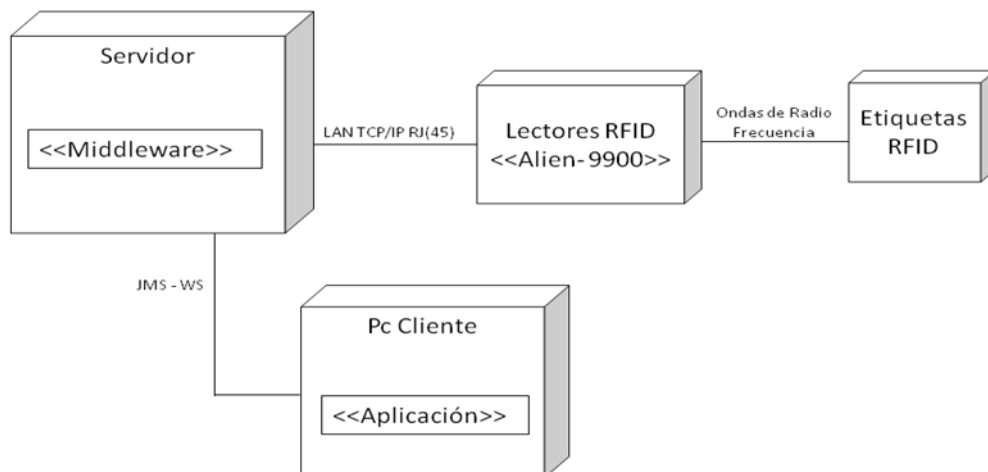
Figura 2. Etiqueta RFID



Fuente: “Estado del Arte en Tecnologías RFID”. Gotor, Eva. Madrid : Escuela Universitaria de Informática – Universidad Politécnica de Madrid, 2009.

- Subsistema de procesamiento de datos o Middleware: proporciona los medios de proceso y almacenamiento de datos. Se trata del software que reside en un servidor, entre el lector y las aplicaciones empresariales. Filtra datos y permite solo pasar la información útil hacia dichas aplicaciones (Ver Figura 3). Algunos pueden también gestionar la red de lectores.

Figura 3. Arquitectura de un sistema RFID



Fuente: autor del proyecto

2.2.2. RIFIDI

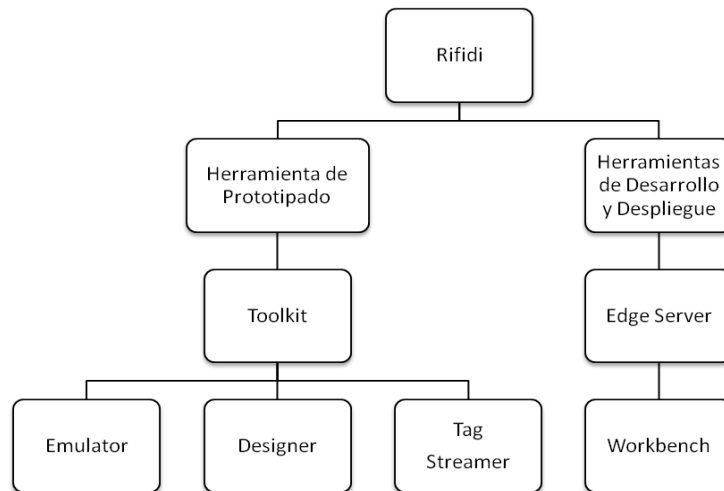
Con el fin de facilitar el estudio y desarrollo de aplicaciones software basadas en tecnologías RFID, fue creada en el año 2006 la plataforma de desarrollo, código abierto, “Rifidi”. Esta plataforma está basada en Java, por lo que el desarrollo de aplicaciones, goza de todas las características y ventajas que este lenguaje ofrece a la hora del desarrollo e implementación de software.

2.2.2.1. ¿Qué es Rifidi?

Rifidi es una completa plataforma middleware para la construcción de todas las facetas de una aplicación RFID. Rifidi hace posible llevar a cabo una rápida creación de prototipos y un despliegue del producto sobre una plataforma código abierto y de alto rendimiento. (2)

La plataforma Rifidi está compuesta por dos conjuntos distintos de aplicaciones (Figura 4.), uno enfocado al prototipado y el otro enfocado al desarrollo y despliegue de aplicaciones. Estos dos conjuntos de herramientas son, Rifidi Toolkit y Rifidi Edge.

Figura 4. Componentes de Rifidi

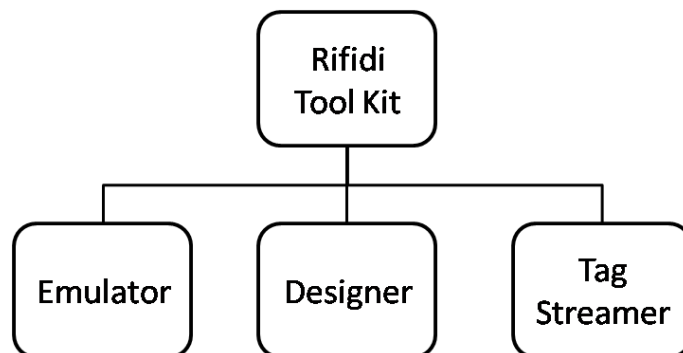


Fuente: Autor del proyecto

- **Rifidi Toolkit:**

Es un ambiente de desarrollo integrado o IDE (por sus siglas en inglés) que apunta al prototipado y simulación de hardware RFID y eventos en un ambiente RFID. Con Toolkit es posible construir una infraestructura RFID virtual con lectores y tags definidos por software y eventos RFID que se comportan exactamente como en la vida real. (7)

Figura 5. Conjunto de aplicaciones de Rifidi Toolkit



Fuente: Autor del proyecto

Dentro de este conjunto de aplicaciones se encuentran:

Emulador: es la herramienta principal de desarrollo de Rifidi Toolkit. Permite la habilidad de emular lectores y tag junto con sus protocolos de comunicación y provee acceso al hardware. Incorpora un motor de scripting para la rápida creación de tags y lectores.

Designer: es una herramienta de presentación construida a la medida sobre el motor de emulación, que permite crear escenarios 3D con elementos gráficos como lectores, paquetes, etc. y animaciones donde los tags pueden ser leídos por lectores RFID emulados. (7)

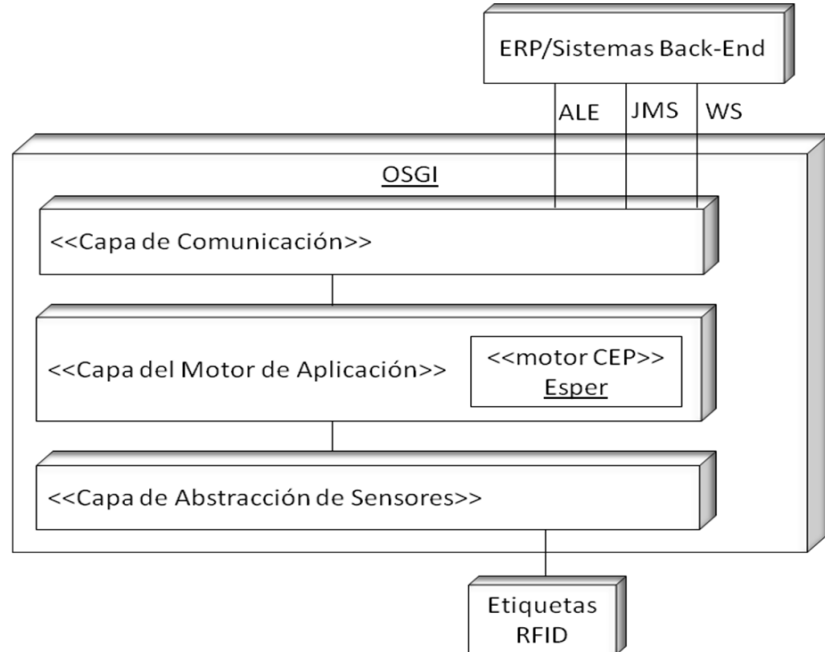
Tag Streamer: es una herramienta que permite generar un gran número de lectores y de etiquetas virtuales permitiendo hacer pruebas al sistema RFID.

- **Rifidi Edge:**

Es una robusta plataforma middleware para desarrollar y desplegar aplicaciones RFID. Rifidi Edge se ocupa de las complejidades de la integración con dispositivos RFID y otros dispositivos y proporciona un motor de reglas de alto rendimiento llamado Esper, que ha sido personalizado para el procesamiento de datos RFID. Construido sobre una ligera plataforma servidor llamada Equinox OSGI, Rifidi brinda un poderoso ambiente de desarrollo de aplicaciones para aplicaciones basadas en RFID.

2.2.2.2. Arquitectura de Edge Server

Figura 6. Diagrama de Arquitectura de Edge Server



Fuente: Tomado y adaptado de Rifidi.org. [En línea]

http://rifidi.org/documentation_edgeserver_architecture.html

Dentro de la arquitectura de Edge Server se encuentran 3 capas; la capa de abstracción de sensores, la capa del motor de la aplicación y la capa de comunicación.

- **Capa de abstracción de sensores**

El propósito de Edge Server es conectar un cierto tipo de sensores (por ejemplo, lectores RFID, dispositivos móviles) y coleccionar la información proveniente de ellos. Este proceso de conexión y recolección de la información se lleva a cabo en la capa de abstracción de sensores. El Edge Server está diseñado de tal forma que

pueda recoger información de varios tipos (activa, pasiva, etc.) proveniente de distintos dispositivos.

- **Capa de motor de aplicación**

Para la gran mayoría de las aplicaciones, no es deseable ni necesario guardar todos los eventos producidos por los sensores. Muchos sensores pueden enviar 1000 eventos por segundo, de los cuales, un gran número serán eventos repetidos. Rifi Edge Server usa un procesador de eventos complejos llamado Esper. Este permite crear consultas usando una sintaxis similar a SQL. El procesamiento de eventos complejos, CEP (por sus siglas en Ingles) es un paradigma de visualización de datos como eventos fugaces (una secuencia de eventos) y de identificación de los eventos significativos (por ejemplo, las reglas de negocio) mediante reglas.

El propósito de la capa de motor aplicación, es permitir a los desarrolladores crear la mayor parte de la funcionalidad de las aplicaciones RFID utilizando consultas Esper.

- **Capa de Comunicación**

Luego de que los datos han sido procesados, estos probablemente necesitarán ser entregados a otro tipo de aplicaciones o sistemas dependientes. Por ejemplo dicha información puede ser almacenada en una base de datos. Edge Server tiene varios conectores que pueden ser utilizados como JMS y Web Services (a través del marco de trabajo de Spring). Sin embargo ya que esto depende de la aplicación, es posible crear una nueva conexión (por ejemplo una conexión TCP/IP Socket) si la aplicación lo necesita.

- **OSGI**

Es un sistema de módulos dinámicos para Java. Ofrece varias piezas de funcionalidad tales como:

1. Una unidad de despliegue denominada paquete, que es un JAR normal con alguna información extra en el manifiesto.
2. Un tiempo de ejecución ligero que permite llevar a cabo un control al ciclo de vida del paquete. En otras palabras, permite iniciar, detener y actualizar los paquetes en tiempo de ejecución.
3. Las dependencias pueden ser declaradas explícitamente en su manifiesto como “dependencias de paquete”.
4. Las clases son accesibles a otros paquetes si sus paquetes contenedores han sido expuestos. Esto permite a los desarrolladores esconder las clases que deben ser estrictamente internas.
5. El tiempo de ejecución proporciona un servicio de registro, un repositorio de todo el sistema para ofrecer mayor funcionalidad.

Edge Server es simplemente una colección de paquetes que corren en un ambiente OSGI (Equinox). La funcionalidad entre los paquetes es compartida por servicios que están disponibles en el registro de servicio. Ya que los paquetes pueden ser detenidos mientras el resto del servicio sigue funcionando, se pueden actualizar las partes no esenciales del servidor sin reiniciarlo.

2.3. PROCESAMIENTO DE EVENTOS COMPLEJOS (CEP)

Actualmente las empresas son mucho más complejas. Diferentes procesos tienen lugar en el mundo y muchos de esos procesos son manejados a través de los

sistemas empresariales de tecnologías de la información. Estos sistemas están distribuidos a lo largo de grandes empresas y generan muchos eventos que fluyen a través de sus sistemas empresariales. Dichos eventos a su vez alimentan otras aplicaciones o servicios que generan nuevos eventos. Se puede hablar verdaderamente de una nube de eventos que cuelga dentro de la empresa. (9)

El CEP puede ser utilizado para varios propósitos, dentro de los que se destacan principalmente, la Arquitectura Dirigida por Eventos (EDA) y el Monitoreo de las Actividades de Negocio (BAM).

La Arquitectura Dirigida por Eventos (EDA) es una infraestructura software que por naturaleza es libremente acoplada. La idea principal detrás de EDA, es que un gran sistema software consiste de muchos y pequeños componentes que tienen sus propias funciones. La comunicación entre los componentes se hace usando eventos. Un evento puede ser visto como una notificación que le dice a otros componentes que cierto trabajo o proceso está hecho. Ya que los eventos son muy importantes dentro de la arquitectura dirigida por eventos, también lo es el manejo y direccionamiento de estos. El CEP es una adición muy importante a la EDA, ya que está puede detectar situaciones complejas en tiempo real.

El Monitoreo de las Actividades del Negocio (BAM) es una herramienta de soporte que permite la profundización en el rendimiento del negocio y ayuda a encontrar posibles cuellos de botella. CEP es una adición muy importante dentro de BAM, ya que puede detectar situaciones complejas que ocurren dentro de una empresa grande, y por lo tanto puede ayudar al BAM a reportar información aún más compleja.

2.4. MOTOR DE PROCESAMIENTO DE EVENTOS COMPLEJOS “ESPER”

Esper es un motor de Procesamiento de Flujo de Eventos (ESP) y correlación de eventos (CEP, Procesamiento de Eventos Complejos). Está dirigido a Arquitecturas Dirigidas por Eventos (EDA), Esper es capaz de activar acciones específicas, escritas como Plain Old Java Objects (POJO), cuando las condiciones de los eventos tienen lugar en el flujo de eventos. Esper se diseñó para un alto volumen de correlación de eventos, donde hay millones de eventos entrantes al sistema y se hace imposible el almacenamiento de todos ellos para después consultarlos utilizando una arquitectura de bases de datos clásica. (10)

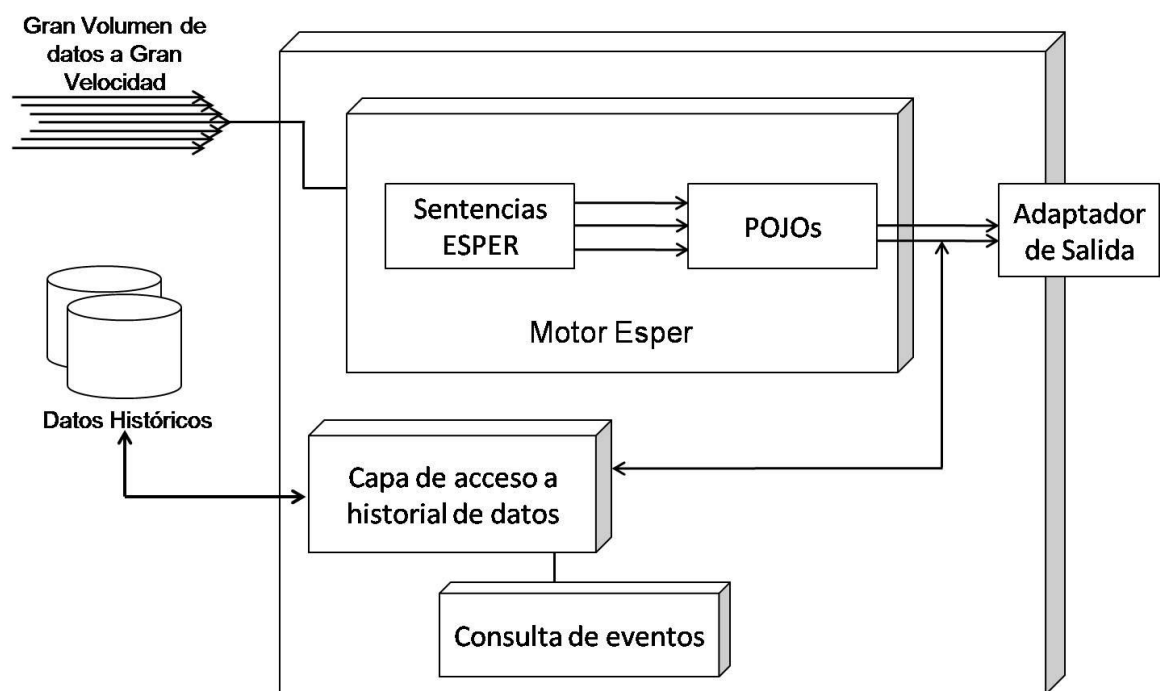
El Lenguaje de Procesamiento de Eventos (EPL) permite expresar las condiciones y la correlación de eventos, posiblemente abarcando ventanas de tiempo, esto minimiza el esfuerzo requerido por el desarrollador para configurar un sistema que pueda reaccionar a situaciones complejas.

Esper es un núcleo ligero escrito en Java, que es completamente integrable dentro de cualquier proceso Java, servidores de aplicaciones JEE o basados en Java Enterprise Service Bus. Permite el desarrollo rápido de aplicaciones que procesan grandes volúmenes de mensajes o eventos entrantes.

Las bases de datos relacionales o sistemas basados en mensajes como JMS hacen realmente difícil lidiar con los datos temporales y consultas en tiempo real; de hecho, las bases de datos requieren consultas específicas para devolver los datos significativos y no son adecuadas para enviar datos a medida que cambian. JMS requiere que el desarrollador implemente la lógica y la agregación temporal por sí mismo. Por el contrario, el motor de procesamiento Esper, proporciona una abstracción más alta e inteligente, y puede ser pensado como una base de datos al revés: en lugar de almacenar los datos y ejecutar las consultas con dichos

datos, Esper posee aplicaciones para almacenar y ejecutar consultas a través de los datos. La respuesta de Esper es en tiempo real y sucede cuando las condiciones de ocurrencia de los eventos, coinciden con la consulta definida, por lo tanto, el modelo de ejecución es continuo y no solo cuando se presenta una consulta. La Figura 7 presenta la arquitectura del motor de procesamiento complejos esper.

Figura 7. Arquitectura y flujo de datos en Esper



Fuente: Tomado y adaptado de Esper - Complex Event Processing. *Tutorial*. [En línea] [Citado el: 18 de 03 de 2010.] <http://esper.codehaus.org/tutorials/tutorial/tutorial.html>.

En Esper, el EPL permite registrar consultas en el motor. Una clase oyente (que es básicamente una POJO) será llamada por el motor, cuando las condiciones del EPL coincidan con el flujo de eventos entrante. El EPL permite expresar condiciones complejas de coincidencia que incluyen ventanas de temporales,

unión con diferentes flujos de eventos, así como filtrado, agregación y clasificación de los datos.

2.5. LENGUAJE DE PROCESAMIENTO DE EVENTOS (EPL)

El módulo procesador de eventos completos, puede ser dividido en los siguientes componentes funcionales: Presentación de eventos, modelo de procesamiento, interfaz de programación y especificación del lenguaje.

Los eventos son presentados como POJOs, siguiendo las convenciones de JavaBeans. Las propiedades de los eventos son expuestas a través de métodos get sobre los POJO. Cuando es posible, los resultados de la ejecución de las declaraciones EPL son también retornadas como POJOs. Sin embargo, hay ocasiones en que eventos sin tipo son retornados, como cuando los flujos de eventos se unen. En ese caso, una instancia de la colección Map es retornada.

Los eventos POJO son instancias de objetos que exponen las propiedades de los eventos a través de métodos de estilo JavaBeans. Las clases de eventos o las interfaces no tienen que ser plenamente compatibles con las especificaciones de JavaBeans, sin embargo para que el motor EPL obtenga las propiedades de los eventos, se requiere que los métodos get de JavaBeans estén presentes.

El modelo de procesamiento EPL es continuo: los resultados se obtienen tan pronto como los eventos entrantes son recibidos y cumplen con las restricciones de las declaraciones EPL.

Los eventos entrantes pueden ser procesados ya sea a través de ventanas deslizables o de ventanas por lotes. Las ventanas deslizables procesan los eventos gradualmente, moviendo la ventana sobre los datos en incrementos sencillos; mientras que las ventanas por lotes procesan los eventos moviendo la

ventana sobre los datos en trozos discretos. El tamaño de la ventana puede ser definido por el máximo número de eventos contenidos por la máxima cantidad de tiempo que se mantienen los eventos.

El lenguaje procesador de eventos es como un lenguaje SQL con cláusulas SELECT, FROM, WHERE, GROUP BY, HAVING y ORDER BY. Los flujos reemplazan a las tablas como las fuentes de los datos, con los eventos reemplazando las filas como unidad básica de información.

2.6. JAVA MESSAGE SERVICE (JMS)

La mensajería es un método de comunicación entre componentes software o aplicaciones. Un sistema de mensajería es una habilidad peer-to-peer: Un cliente de mensajería pueden enviar y recibir mensajes de cualquier otro cliente. Cada cliente se conecta a agentes de mensajería que ofrecen la facilidad de crear, enviar, recibir y leer mensajes.

La mensajería permite comunicación distribuida que es débilmente acoplada. Un componente envía un mensaje al destino, y el receptor puede recuperar el mensaje del destino. Sin embargo, el remitente y el receptor no tienen que estar disponibles al mismo tiempo para comunicarse. De hecho el remitente no necesita saber nada acerca del receptor; tampoco el receptor necesita saber nada acerca del remitente. El remitente y el destinatario solo necesitan saber el formato y el destino.

Java Message Service es un API de Java que permite a las aplicaciones crear, enviar, recibir y leer mensajes. Diseñada por Sun y varias empresas asociadas, JMS API define un conjunto común de interfaces y semánticas asociadas que

permiten escribir programas en el lenguaje Java para comunicarse con otras implementaciones de mensajería.

Este API minimiza el conjunto de conceptos que un programador debe aprender para usar los productos de mensajería, pero brinda suficientes características para soportar sofisticadas aplicaciones de mensajería. De igual forma, se esfuerza por maximizar la portabilidad de las aplicaciones JMS a través de los proveedores del servicio en el mismo dominio de mensajería. Este API no solamente permite comunicación débilmente acoplada, también permite comunicación: (11)

Asincrónica: Un proveedor de JMS puede entregar mensajes a un cliente a medida que llegan; un cliente no tiene que solicitar mensajes con el fin de recibirlos.

Confiable: JMS puede garantizar que un mensaje es entregado una y solo una vez. Están disponibles bajos niveles de confiabilidad en caso de que algunas aplicaciones puedan permitirse el perder mensajes o recibir mensajes duplicados.

2.7. APLICACIONES DIRIGIDAS POR EVENTOS

Si el funcionamiento de la aplicación desarrollada está basado en la recepción y análisis de eventos, podría decirse que es una aplicación dirigida por eventos.

Un evento, llamado también suceso, es algo que transcurre durante un periodo de tiempo muy breve, como por ejemplo pulsar un botón del mouse, el despegue de un avión, etc. Aunque un evento no tiene un tiempo de duración determinado, se asocia a algo que se produce muy rápidamente y que produce una serie de efectos. Por ejemplo, al pulsar el botón del televisor, éste se enciende y aparece el

canal predefinido o se apaga, según el caso. Un evento puede preceder o seguir a otro, o los dos eventos pueden o no estar relacionados. (12)

Los sistemas o aplicaciones dirigidas por eventos se rigen por eventos generados externamente. El término evento en este contexto no sólo significa una señal binaria. Puede ser una señal dentro de un rango de valores o una entrada de un comando desde un menú. La diferencia entre un evento y una entrada simple es que la aparición del evento está fuera del proceso de control del proceso que maneja ese evento. (13)

Durante el proceso de desarrollo de aplicaciones dirigidas por eventos, se recomienda seguir los siguientes pasos (10):

1. Definir la misión de la aplicación, analizando el dominio del negocio y definiendo la situación a ser detectada o la información a ser reportada.
2. Definir los requisitos de desempeño, específicamente rendimiento y latencia.
3. Identificar de donde vienen los eventos.
4. Identificar el formato de eventos más sencillo y el contenido de los eventos que es aplicable a su dominio.
5. Diseño de las relaciones de eventos que conducen a eventos complejos.
6. Instrumentar las fuentes de eventos.
7. Diseñar como quiere presentar los eventos: Clases Java, Maps o como eventos XML.

8. Definir las declaraciones EPL para el patrón de procesamiento.
9. Usar una herramienta de simulación de eventos para probar las situaciones a ser detectadas o para generar la carga.
10. Probar contra las necesidades de desempeño: el rendimiento y la latencia de la aplicación en el medio ambiente real.

CAPITULO 3

3. PROTOTIPO WEB QUE PERMITE EL RASTREO DE PERSONAS DENTRO DE UN EDIFICIO – GIIBPT (GIIB People Tracker)

Este capítulo hace una descripción detallada de Sistema GIIBPT, presentando su arquitectura general, configuración de los servidores y de las herramientas de emulación de los dispositivos, organización de espacio físico, programación y despliegue de la aplicación.

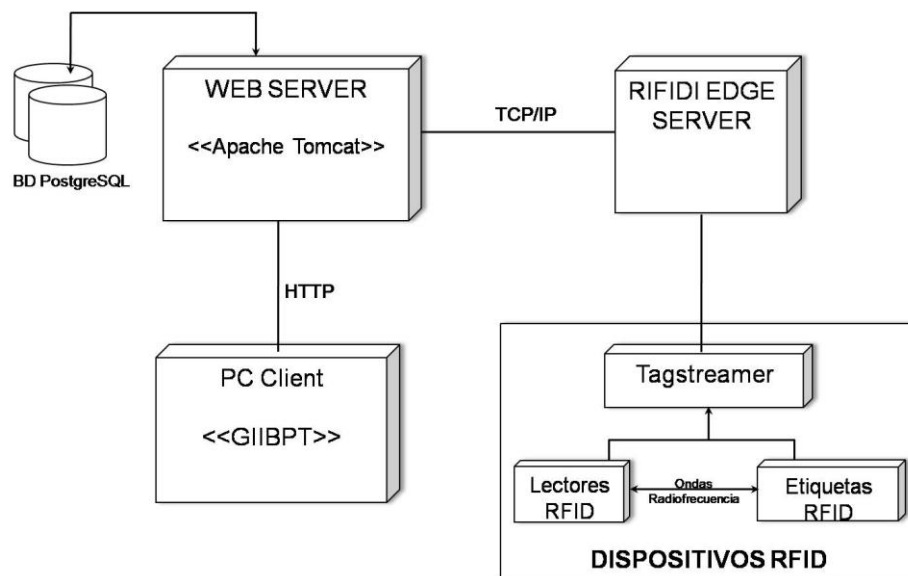
3.1. ARQUITECTURA GENERAL DEL SISTEMA GIIBPT

GIIBPT es una aplicación web que permite el rastreo de personas dentro de un edificio, apoyándose en la utilización de dispositivos de identificación por radiofrecuencia. Basado en la teoría anteriormente expuesta, GIIBPT podría considerarse además como una aplicación dirigida por eventos.

A grandes rasgos la arquitectura del sistema (Figura 8) está formada por dos servidores, uno de los cuáles se encarga de facilitar la conexión con los dispositivos de Identificación por Radiofrecuencia y del filtrado de la información recibida de estos, para llevar a cabo su almacenamiento, presentación y consulta; el otro servidor se encarga de la parte web de la aplicación, permitiendo la interacción del usuario final con el sistema. En conjunto con el servidor web, se cuenta con una base de datos que almacena toda la información relevante para el rastreo de personas dentro del edificio; esta información puede ser consultada por el usuario utilizando los diferentes módulos de la interfaz de usuario.

Otro de los componentes del sistema son los dispositivos de Identificación por Radiofrecuencia, que en este proyecto serán emulados mediante una de las herramientas que ofrece la plataforma de desarrollo Rifidi y que recibe el nombre de TagStreamer. Esta herramienta permite emular tanto los dispositivos de lectura como las etiquetas que sirven de identificación a cada una de las personas del sistema. TagSteamer permite configurar y utilizar los dispositivos RFID tal como si fueran dispositivos reales.

Figura 8. Arquitectura General del sistema GIIBPT



Fuente: Autor del proyecto

El flujo de información dentro del sistema da inicio en los dispositivos de lectura RFID, quienes en un principio transmiten ondas de radiofrecuencia en un área determinada y están alertas al arribo de información de etiquetas que se encuentran dentro de su área de cobertura. Toda la información captada por los lectores es transmitida al servidor de Rifidi, donde es filtrada mediante el motor de

procesamiento de eventos complejos Esper, basándose en las reglas de flujo de personas definidas por el desarrollador.

La información filtrada en el servidor es transmitida mediante sockets hacia el servidor web, el cual se encarga de almacenarla en una base de datos y presentarla de forma ordenada al usuario final del sistema mediante la web creada para GIIBPT.

3.2. DEFINICIÓN Y PRESENTACIÓN DEL ESPACIO FÍSICO

El espacio físico tomado como guía para el planteamiento y desarrollo del sistema GIIBPT, corresponde al primero piso del Centro de Tecnologías de Información y Comunicación (CENTIC) de la Universidad Industrial de Santander. La distribución física de dicho espacio se obtuvo a partir de la observación directa del lugar. Esta se presenta en la Figura 9.

Dicho espacio físico comprende: Sala de espera o lobby, oficinas del personal administrativos del edificio, seis aulas o salas de trabajo, dos baños, y un área de acceso restringido donde se encuentran la salas de control y de seguridad del edificio y están instalados algunos servidores de la Universidad.

Cada una de las áreas mencionadas anteriormente está claramente definida dentro de la arquitectura del edificio y se encuentran separadas unas de otras. El acceso a cada una de las áreas se realiza mediante la utilización de una tarjeta RFID obtenida en la recepción del edificio, la cual permite visitar lugares específicos, dependiendo de las necesidades del visitante y de los permisos concedidos a este.

Figura 9. Distribución del espacio físico para el primer piso del CENTIC



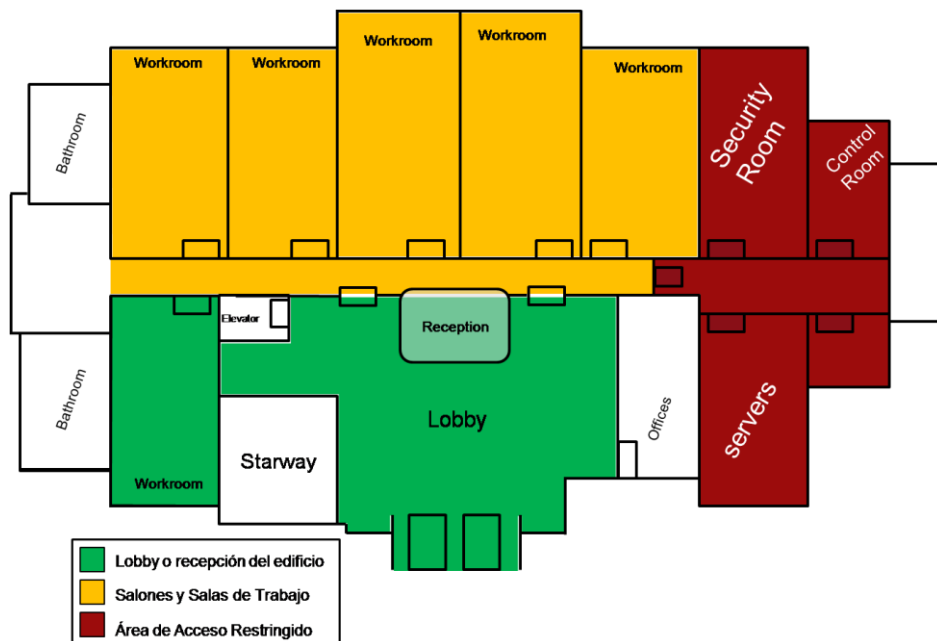
Fuente: Autor del proyecto.

Las tarjetas de identificación por radiofrecuencia utilizadas dentro del funcionamiento actual del edificio y que permiten a los visitantes tener acceso a las diferentes áreas, no corresponden a las utilizadas dentro del sistema GIIBPT, debido a que dichas tarjetas únicamente funcionan como llaves a los diferentes lugares del edificio; no se encargan de llevar a cabo una identificación detallada de las personas que las usan y mucho menos son utilizadas para realizar un rastreo y monitoreo de la posición actual de los visitantes. Las tarjetas utilizadas por GIIBPT son de identificación única, por lo tanto una tarjeta RFID pertenece exclusivamente a un visitante dentro del edificio, bien sea administrativo, profesor o estudiante.

3.1.2 Organización del espacio físico

Tomando como base la distribución física del edificio presentada anteriormente, se definen tres grandes áreas (Figura 10); lobby o recepción del edificio, área de Salones de trabajo y área con acceso restringido. El planteamiento de estas tres áreas, permite definir más claramente las reglas de flujo de acuerdo al acceso permitido a los visitantes del edificio. No es lo mismo que una persona se esté movilizándose en el área del lobby del edificio a que lo esté haciendo en el área de servidores y cuartos de control; el grado de seguridad y de control requerido para cada área es diferente. Esta distribución permite a su vez definir y agrupar los diferentes eventos presentados en el sistema de acuerdo al área del edificio en que una persona se encuentre.

Figura 10. Distribución de las áreas de monitoreo en el edificio



Fuente: Autor del proyecto

La recepción o lobby, es un área que se encuentra inmediatamente después de ingresar al edificio, es un área de acceso libre, y es paso obligado para cualquier visitante, sin importar si es estudiante, profesor o administrativo.

El área de salones o “workrooms”, agrupa todos los salones de clase, laboratorios y auditorios que hay en el edificio. El acceso a estas áreas está permitido a estudiantes, profesores y administrativos siempre y cuando se haga un registro previo en recepción.

Las áreas restringidas o “restrictedAreas”, son áreas a las cuáles solamente tiene acceso personal autorizado, básicamente administrativos del edificio; en esta área se encuentran salas de vigilancia, salas de servidores y aires acondicionados.

Como se observa en la Figura 10, hay lugares dentro del edificio en las cuales no se está llevado a cabo el monitoreo del flujo de personas. Estas áreas corresponden a las oficinas de los administradores del edificio y al área de descanso y baños; la exclusión de estas áreas dentro del proceso de monitoreo se debe, respectivamente, a que son áreas dentro de las cuáles el flujo de personas es reducido, son áreas de acceso libre y con el fin de ofrecer algo de privacidad a los visitantes del edificio.

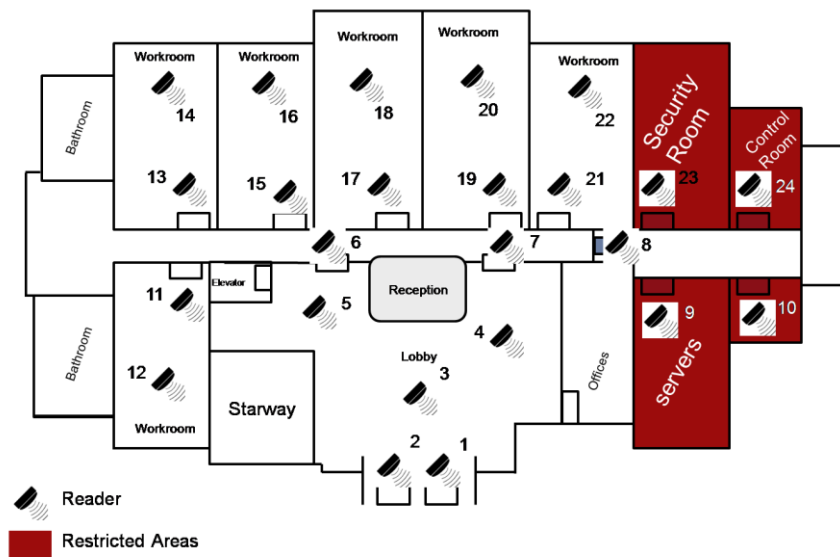
3.3. POSICIONAMIENTO DE LOS LECTORES DENTRO DEL EDIFICIO

Un aspecto importante dentro el sistema es la distribución de los dispositivos de tecnología RFID a lo largo del edificio con el fin de garantizar la mayor cobertura posible y un monitoreo permanente de la ubicación de los visitantes dentro cada área. Para esto es necesario ubicar lectores RFID en los puntos de acceso y salida del edificio, al igual que en los puntos de acceso y salida de las diferentes áreas y salas de trabajo. La Figura 11 presenta la distribución de los lectores RFID

a lo largo de las áreas en las cuales se llevará a cabo el monitoreo del flujo de personas.

Además de la ubicación de lectores RFID en los puntos de ingreso y salida de las diferentes áreas, se disponen en puntos intermedios entre ellas y puntos internos de las mismas, buscando cubrir todas las posibles rutas de tránsito de los visitantes.

Figura 11. Distribución de los lectores RFID dentro del edificio



Fuente: Autor del proyecto

La numeración de los lectores presentada en la Figura 11 se utilizará posteriormente para su clara identificación y para la presentación más detallada de ciertas características y configuraciones dentro del sistema.

3.4. DEFINICIÓN DE LAS REGLAS Y EVENTOS PARA EL FLUJO DE PERSONA

Las reglas de flujo de personas dentro del edificio son la base del correcto funcionamiento del sistema, a partir de ellas se hace el filtrado de los eventos recibidos por el sistema. Las reglas para el flujo de personas fueron redactadas por el autor del proyecto basándose en la observación del lugar y del flujo de visitantes presentes en este.

Sabiendo de antemano que la aplicación desarrollada es una aplicación dirigida por eventos, se hace necesario plantear los tipos de eventos y las reglas bajo las cuales son analizados para llevar a cabo el monitoreo de visitantes dentro del edificio.

Como el objetivo principal de la aplicación es permitir un monitoreo de la ubicación y del flujo de personas dentro del edificio, se definen todos los posibles flujos o rutas por las cuales podría una persona transitar a lo largo del edificio. El análisis y validación de las rutas, se llevará a cabo mediante la identificación del lector en el cuál se encuentra ubicada una persona en un momento específico y el lector inmediatamente anterior por el cuál fue registrada.

Los eventos y las condiciones bajo las cuáles estos se desencadena y se registran en el sistema se definen a continuación:

- **Entrada al edificio o inicio de flujo:** el inicio del flujo de visitantes da comienzo cuando una etiqueta es registrada por el lector #1 ubicado en la puerta de acceso al edificio, una vez se da comienzo al flujo para una etiqueta, ésta no puede pasar nuevamente por este lector, debido que esa puerta del edificio está definida únicamente como puerta de acceso, no de

salida; en caso de que esto llegara a suceder se activaría una alerta de flujo.

- **Salida del edificio o fin de flujo:** se da cuando una etiqueta es registrada por el lector #2, ubicada en la puerta de salida del edificio. Una vez se finaliza el flujo para una etiqueta ésta no puede volver a ser registrada por lectores dentro del edificio, en tal caso se generaría un error de flujo. Para evitar el error de flujo sería necesario que la persona volviera a la puerta de ingreso del edificio y diera inicio nuevamente a flujo dentro del edificio.
- **Llegada a un lector:** cada vez que una etiqueta es leída por un dispositivo RFID, el servidor envía el evento al sistema y éste queda registrado en la base de datos. Todos los eventos de llegada a un lector son registrados en la base de datos sin importar si la lectura de esa etiqueta desencadenó algún otro tipo de evento.
- **Salida de un lector:** este evento se desencadena dentro del sistema una vez una etiqueta deja de ser registrada durante 2 o más segundos por el lector en el cual se encontraba. El registro en base de datos tanto de los eventos de llegada como de los de salida de los lectores, permitirá realizar el control de flujo de visitantes y hacer consultas cronológicas de las mismas en caso de que sea requerido.
- **Acceso a un área restringida:** la ocurrencia de este evento se da cada vez que una etiqueta es registrada por lector que se encuentra en un área de acceso restringido. Los lectores que activarían el envío de este evento son los lectores número 8, 9, 10, 23 y 24.

- **Error de flujo:** permiten identificar posibles errores en la lectura de las etiquetas o posibles saltos y violaciones de seguridad del edificio. Un error de flujo se produce cuándo la tupla entre el lector actual y el lector anterior no está dentro las posibles combinaciones permitidas para el flujo. A diferencia de los eventos anteriores, existen muchas más posibilidades para el desencadenamiento de este evento. A continuación se presenta la Tabla1 con todas las posibilidades bajo las cuales el flujo de personas es correcto, en caso de que se registre en el sistema una combinación que no se encuentra dentro de las planteadas en la Tabla 1, se desencadena el evento de error de flujo.

Tabla 1. Lista de condiciones que definen los posibles flujos de personas dentro del edificio

# Lector Actual	# Lector Anterior	# Lector Actual	# Lector Anterior
1	No hay Lector Anterior	13	6, 11, 13, 14 15
2	1, 3, 4, 5	14	13, 14
3	1, 3, 5, 4	15	6, 11, 13, 15, 16
4	1, 3, 4, 5, 7	16	15, 16
5	1, 3, 4, 5, 6	17	6, 7, 17, 18
6	5, 6, 7, 11, 13, 15, 17	18	17, 18
7	4, 6, 7, 8, 17, 19, 21	19	7,17, 19, 20
8	7, 8, 9, 10, 21, 23, 24	20	19, 20
9	8, 9, 10, 23, 24	21	7, 8, 21, 22
10	8,9, 10,23,24	22	21, 22
11	6, 11, 12, 13, 15	23	8, 9, 10, 23, 24
12	11, 12	24	8, 9, 10, 23, 24

Fuente: Autor del proyecto

3.5. CONFIGURACIÓN DEL SERVIDOR – RIFIDI EDGE SERVER

El servidor es el encargado de llevar a cabo la tarea de integración de los dispositivos de lectura y las etiquetas RFID, además proporciona el motor de reglas llamado Esper, mediante el cual se lleva a cabo el filtrado de la información registrada.

La configuración se lleva a cabo mediante la consola del servidor mediante comandos definidos específicamente para el manejo de información RFID.

A continuación se presenta el proceso de configuración del servidor y se explican detalladamente los comandos utilizados.

3.5.1. Creación de una configuración de lectores dentro del servidor

El servidor ofrece varias referencias o marcas de dispositivos RFID que se pueden utilizar dentro de la configuración y funcionamiento del sistema, cada referencia ofrece diferentes prestaciones de acuerdo al objetivo final de la aplicación que se está desarrollando. La referencia escogida para el proyecto es AlienALR9800 (6). Esta referencia de lector es una de las más actuales que existen, permite conexión con el servidor mediante puerto serial o mediante LAN TCP/IP (RJ45), incluye un software para desarrolladores y sus rangos de lectura están dentro de los más amplios (3,5 – 4,5 metros).

Los principales comandos utilizados para la creación de los lectores en el servidor son los siguientes:

readertypes: Hace una lista de todos los tipos de lectores disponibles en el sistema. El que se utiliza en el proyecto es el Alien9800.

createreader Alien9800: Este comando se encarga de crear una nueva instancia para la configuración de un lector tipo Alien9800 y define las propiedades por defecto (la IP y el puerto de comunicación: 127.0.0.1:20000) además asigna una identificación para dicho lector (Alien_1), esta identificación se va autogenerando a medida que se van creando más lectores.

Cada uno de los lectores creados en el sistema necesitan un puerto diferente de comunicación, por esto luego de la creación del primer lector, es necesario dar un parámetro más al comando createreader, que contenga el puerto asignado a dicho lector: `createreader Alien9800 Port 20001`; esta instrucción crearía un nuevo lector en el servidor cuya IP y puerto de comunicación son respectivamente 127.0.0.1:20001.

readers: permite hacer una lista de los lectores que han sido creados en el servidor.

3.5.2. Creación de una sesión de comunicación para los lectores

createsession Alien_1: crea una sesión de comunicación que será usada para conectar el lector RFID (emulado mediante el Tag Streamer) con el servidor. "Createsession" recibe como parámetro el nombre de cada uno de los lectores creados; es necesario crear una sesión para cada lector.

3.5.3. Creación y configuración de un comando para la lectura de las etiquetas RFDI

De una forma parecida a la que se crean los lectores, es necesario crear un comando que se encargue de ordenar a los dispositivos de lectura que hagan

chequeo de la existencia o no de etiquetas dentro de su área de cobertura. La instrucción para crear dicho comando es:

```
createcommand ALien_Poll
```

Una vez creado el comando, el sistema le asigna un ID, **ALien_Poll_1**. Este comando queda guardado en el servidor y será asignado más adelante a cada uno de los lectores.

Dentro de las características físicas y de funcionamiento de los lectores RFID, estos cuentan con más de una antena que se encarga de enviar y recibir señales de radiofrecuencia. En el caso de los lectores tipo 'Alien', estos cuentan con cuatro antenas. Por defecto, el comando 'Alien_Poll_1' solamente hace uso de una de las antenas que tiene disponible el lector RFID, por lo tanto es necesario cambiar la configuración del comando, con el fin de obtener funcionalidad de las cuatro antenas con las que se cuenta. Esta configuración se realiza mediante la siguiente línea de comandos:

```
setproperties ALien_Poll_1 AntennaSequence 0,1,2,3
```

3.5.4. Asignación y ejecución del comando ALien_Poll_1 a cada lector

La asignación y ejecución del comando ALien_Poll_1 a cada uno de los lectores del sistema, se realiza mediante el siguiente comando:

```
executecommand ALien_1 1 ALien_Poll_1 1000
```

El primer parámetro del comando indica el nombre del lector al cuál se está asignando el comando, el segundo parámetro indica el número de la sesión en que se está ejecutando. El tercer parámetro indica el ID del comando que se está asignando y ejecutando y por último, se define la frecuencia con que cada lector revisará si existen etiquetas o no en su área de cobertura; este tiempo está en

milisegundos. Este comando puede ser asignado a todos los lectores, así que solamente es necesaria la creación de uno solo.

Como se presentó anteriormente, la cantidad total de lectores de que constará el sistema es veinticuatro, por lo tanto el proceso anterior debe ser llevado a cabo para cada uno de los lectores. Una vez finalizada dicha configuración en el servidor se guardan cambios mediante el comando **save**.

3.5.5. Inicio de sesión para los dispositivos de lectura RFID

Con los lectores previamente configurados es necesario iniciar la sesión de comunicación para cada uno de ellos. Este inicio de sesión activa la comunicación TCP/IP entre los lectores emulados y el servidor. El comando para inicio de sesión de los lectores es el siguiente:

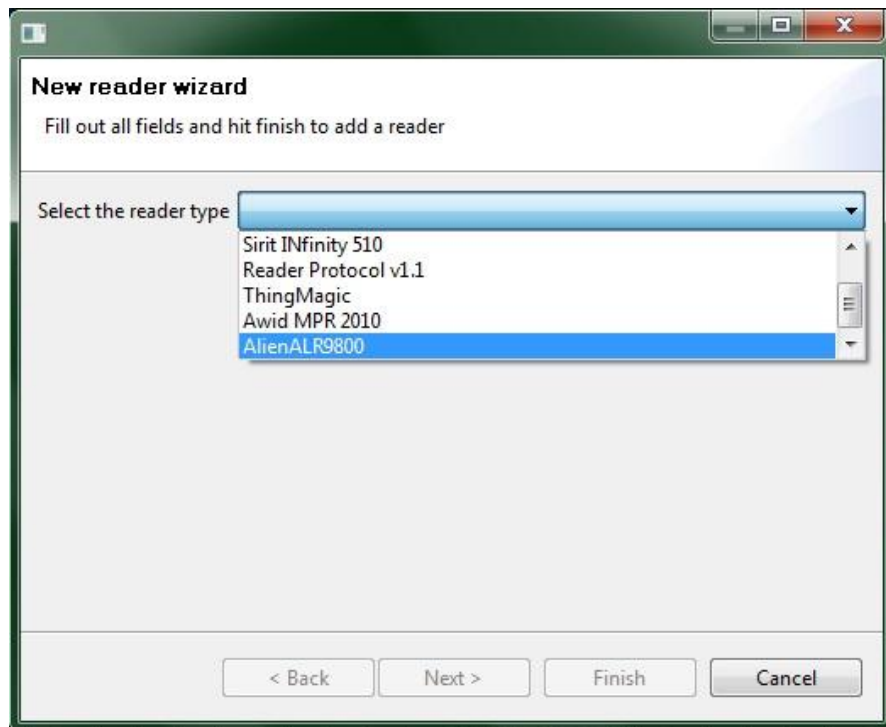
```
startsession Alien_1 1
```

El primer parámetro de este comando es el ID del lector del cual se desea iniciar sesión y el segundo parámetro es el número de la sesión que fue creada anteriormente.

3.6. CONFIGURACIÓN Y DESPLIEGUE DE LOS LECTORES EN EL TAG STREAMER

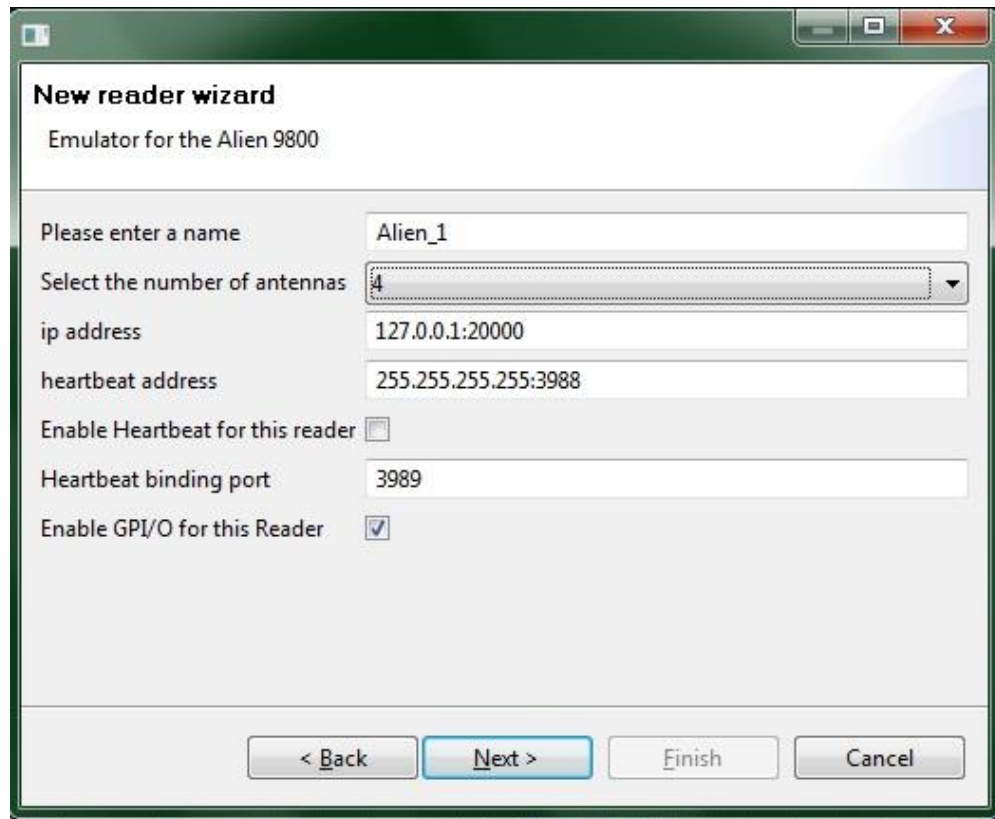
Una vez realizada la creación de las configuraciones para los lectores dentro del servidor, se realiza la configuración física de los dispositivos que se encargan de hacer el envío y recepción de las ondas de radiofrecuencia. Los dispositivos utilizados en este proyecto serán emulados mediante la utilización de una de las herramientas con que cuenta Rifidi, el tag streamer. La configuración y despliegue de dichos dispositivos es un poco más intuitiva comparada con la configuración del servidor, puesto que el tag streamer cuenta con una interfaz gráfica amigable, por medio de la cual se lleva a cabo el proceso; no es necesario utilizar consola. Esta interfaz se presenta en la Figura 12.

Figura 12. Interfaz gráfica del Tag Streamer de Rifidi que permite la creación de lectores



Fuente: Pantallazo tomado durante el proceso de desarrollo del proyecto

Figura 13. Interfaz Gráfica que permite la configuración de los lectores RFID

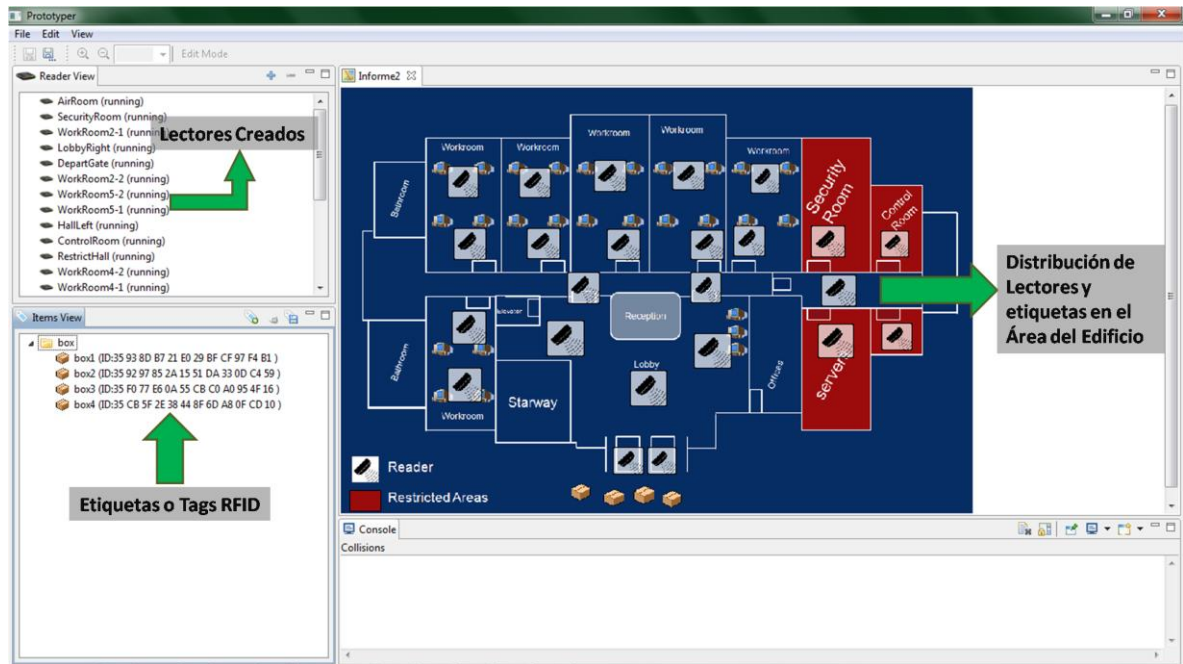


Fuente: pantallazo tomado durante el proceso de desarrollo del proyecto

Las configuraciones realizadas anteriormente en la parte del servidor deben coincidir con las realizadas en el Tag Streamer; especialmente todos los puertos, pues es mediante estos que se comunican los lectores con el servidor.

Una vez creados los lectores se hace el despliegue e inicio de los lectores sobre el área en la cual se va a llevar a cabo el rastreo. Tag Streamer utilizar una imagen en 2D del área en el cual se instalarán los lectores, permitiendo de esta forma tener una panorámica general de la distribución de los mismos (Figura14).

Figura 14. Interfaz gráfica del Tag Streamer con lectores y etiquetas creadas y la distribución de los lectores en el área del edificio.



Fuente: pantallazo tomado durante el proceso de desarrollo del proyecto

Tag Streamer permite también la creación de múltiples etiquetas o tags RFID mediante las cuáles serán identificados los visitantes del edificio. A cada etiqueta creada le corresponde un código único de identificación denominado EPC (Electronic Product Code), consistente en un número hexadecimal de 24 caracteres.

Una vez finalizada la creación, configuración y despliegue de los lectores y de haber creado también las etiquetas, se encienden los lectores dando click derecho en cada uno de ellos y seleccionando la opción *Start Reader*. Para comenzar a hacer la emulación del flujo de personas por los lectores del edificio se desactiva la opción *Edite Mode* que aparece en la interfaz principal de Tag Streamer; de esta forma es posible hacer el arrastre de uno o más lectores a vez por los lectores RFID desplegados anteriormente.

A continuación se presenta en la Tabla 2 un resumen con las configuraciones definitivas de los lectores Alien9800 del sistema GIIBPT.

Tabla 2. Tabla resumen con la configuración de cada uno de los lectores RFID utilizados en el sistema.

CONFIGURACIÓN DE LECTORES RFID							
Número del lector	Nombre del Lector	ID del Lector	Dirección IP	Número del lector	Nombre del Lector	ID del Lector	Dirección IP
1	AccessGate	Alien_1	127.0.0.1:20000	13	WorkRoom2-1	Alien_13	127.0.0.1:20012
2	DepartGate	Alien_2	127.0.0.1:20001	14	WorkRoom2-2	Alien_14	127.0.0.1:20013
3	LobbyCenter	Alien_3	127.0.0.1:20002	15	WorkRoom3-1	Alien_15	127.0.0.1:20014
4	LobbyRight	Alien_4	127.0.0.1:20003	16	WorkRoom3-2	Alien_16	127.0.0.1:20015
5	LobbyLeft	Alien_5	127.0.0.1:20004	17	WorkRoom4-1	Alien_17	127.0.0.1:20016
6	HallLeft	Alien_6	127.0.0.1:20005	18	WorkRoom4-2	Alien_18	127.0.0.1:20017
7	HallCenter	Alien_7	127.0.0.1:20006	19	WorkRoom5-1	Alien_19	127.0.0.1:20018
8	RestrictHall	Alien_8	127.0.0.1:20007	20	WorkRoom5-2	Alien_20	127.0.0.1:20019
9	ServersRoom	Alien_9	127.0.0.1:20008	21	WorkRoom6-1	Alien_21	127.0.0.1:20020
10	AirRoom	Alien_10	127.0.0.1:20009	22	WorkRoom6-2	Alien_22	127.0.0.1:20021
11	WorkRoom1-1	Alien_11	127.0.0.1:20010	23	SecurityRoom	Alien_23	127.0.0.1:20022
12	WorkRoom1-2	Alien_12	127.0.0.1:20011	24	ControlRoom	Alien_24	127.0.0.1:20023

Fuente: Autor del proyecto

3.7. PROGRAMACIÓN DE LAS REGLAS DE FILTRADO DE INFORMACIÓN EN ESPER – EPL

La información procedente de los dispositivos RFID pasa por un proceso de filtrado antes de ser enviada al servidor web para ser almacenada y presentada al usuario final. Este filtrado se lleva a cabo en el servidor de Rifidi, mediante una serie de reglas y restricciones programadas en el motor de procesamiento de eventos complejos Esper. Las reglas y restricciones funcionan como una red, tomando únicamente los eventos que le interesan al usuario, evitando de esta manera saturación o colapso del sistema por ingreso de información no necesaria o redundante.

Para comenzar el filtrado es necesario crear ventanas que se encarguen de recibir los eventos RFID. El código de dicho método se presenta en la Figura 15.

Figura 15. Método createWindow para la creación de ventanas en Esper

```
public void createWindow(String nombreVentana) {
    statements.add(esperService.getProvider().getEPAdministrator().createEPL(
        "create window "+nombreVentana+".std:firstunique(tag_ID, reader_ID, antenna)"+
        "(reader_ID String, antenna int, tag_ID String, timestamp Long)");
}
```

Fuente: Autor del proyecto

Todas las ventanas utilizadas por Esper para el funcionamiento del sistema son creadas mediante el método createWindow. Estas ventanas almacenan el primero y único evento reportado por los lectores RFID, distinguiendo cada uno de los eventos por el código de identificación de la etiqueta RFID (tag_ID) que porta el usuario y por el lector y la antena de este que registró la etiqueta (reader_ID y antenna). La información almacenada por cada una de las ventanas, corresponde al ID del lector, al ID de la antena, el ID de la etiqueta que fue registrada y la fecha y hora de ocurrencia del evento.

Con base en la distribución del espacio físico planteado en la sección 3.1.2. donde se definen tres áreas generales para el rastreo de personas, se realiza la creación de tres ventanas para el almacenamiento de los eventos registrados por el los dispositivos de identificación por radiofrecuencia. Cada una de estas ventanas se encarga de almacenar los eventos de un área determinada. Las ventanas creadas corresponden a las áreas de recepción o lobby, salones o salas de trabajo y al área de acceso restringido. Es necesario asignar a cada una de las ventanas los lectores de los cuales almacenará los eventos; la asignación llevada a cabo se presenta en la Figura 16.

Figura 16. Asignación de lectores RFID a la ventana correspondiente (imagen parcial)

```
statements.add(esperService.getProvider().getEPAdministrator().createEPL(
    "on ReadCycle[select * from tags]" +
    "insert into LobbyEdificio select readerID as reader_ID, antennaID as antenna, cast(tag.epc?, String) as tag_ID, " +
    " timestamp as timestamp where readerID = 'Alien_1'" +
    "insert into LobbyEdificio select readerID as reader_ID, antennaID as antenna, cast(tag.epc?, String) as tag_ID, " +
    " timestamp as timestamp where readerID = 'Alien_2'" +
    "insert into LobbyEdificio select readerID as reader_ID, antennaID as antenna, cast(tag.epc?, String) as tag_ID, " +
    " timestamp as timestamp where readerID = 'Alien_3'" +
    "insert into LobbyEdificio select readerID as reader_ID, anten naID as antenna, cast(tag.epc?, String) as tag_ID, " +
    " timestamp as timestamp where readerID = 'Alien_4'" +
    "insert into LobbyEdificio select readerID as reader_ID, antennaID as antenna, cast(tag.epc?, String) as tag_ID, " +
    " timestamp as timestamp where readerID = 'Alien_5'" +
    "insert into WorkRooms select readerID as reader_ID, antennaID as antenna, cast(tag.epc?, String) as tag_ID, " +
    " timestamp as timestamp where readerID = 'Alien_6'" +
    "insert into WorkRooms select readerID as reader_ID, antennaID as antenna, cast(tag.epc?, String) as tag_ID, " +
    " timestamp as timestamp where readerID = 'Alien_7'" +
    "insert into RestrictedAreas select readerID as reader_ID, antennaID as antenna, cast(tag.epc?, String) as tag_ID, " +
    " timestamp as timestamp where readerID = 'Alien_8'" +
    "insert into RestrictedAreas select readerID as reader_ID, antennaID as antenna, cast(tag.epc?, String) as tag_ID, " +
    " timestamp as timestamp where readerID = 'Alien_9'" +
```

Fuente: Autor del proyecto

Teniendo en cuenta que en las ventanas se registran eventos únicos, cuyos parámetros son el código de identificación de la etiqueta (tag_ID) y el lector y la antena correspondiente al lector (reader_ID y antenna), es necesario hacer un limpiado de los eventos almacenados, esto con el fin de evitar la omisión de eventos que suceden algún tiempo después y que podrían contener los mismos parámetros que un evento anterior. Para esto se crea un método de eliminación de eventos, que se encarga de eliminar los eventos de una ventana basándose en la ausencia de actividad de una etiqueta por más de dos segundos en un lector;

entonces si un lector deja de registrar una etiqueta por más de dos segundos, los eventos relacionados a ese lector y a esa etiqueta se eliminan de la ventana, permitiendo el registro de futuros eventos que contengan las mismas propiedades. El código que realiza dichas funciones se encuentra en la Figura 17.

Figura 17. Método para la eliminación de eventos dentro de una ventana

```
public void removeEvents(String nombreVentana){
    statements.add(esperService.getProvider().getEPAdministrator().createEPL(
        "on pattern [every tag="+nombreVentana+" ->" +
        "(timer:interval(2 sec) and not "+nombreVentana+"(tag_ID = tag.tag_ID))]" +
        "delete from "+nombreVentana+" where tag_ID = tag.tag_ID"));
}
```

Fuente: Autor del proyecto

Para finalizar el proceso de captura y filtrado de la información mediante Esper hace falta crear las consultas EPL que tomarán la información de cada uno de los eventos almacenados en las ventanas, permitiendo su envío al servidor web que se encargará de almacenarlos y presentarlos al usuario del sistema mediante la web. La sintaxis del código utilizado para esto se presenta en la Figura 18.

Figura 18. Estructura de las consultas EPL que obtienen la información de cada ventana

```
EPStatement buscoTagsLobby = esperService.getProvider().getEPAdministrator().
    createEPL("select istream * from LobbyEdificio");

EPStatement buscoTagsWorkRooms = esperService.getProvider().getEPAdministrator().
    createEPL("select istream * from WorkRooms");

EPStatement buscoTagsRestrictedAreas = esperService.getProvider().getEPAdministrator().
    createEPL("select istream * from RestrictedAreas");
```

Fuente: Autor del proyecto

Las consultas EPL se encargan de tomar toda la información disponible de los eventos, incluyendo la fecha de ocurrencia del evento y el tipo de evento con que es registrado, que puede ser: llegada de la etiqueta al lector o partida de la etiqueta del lector.

3.8. ADMINISTRACIÓN DE LAS CONSULTAS EN ESPER Y ENVÍO DE LA INFORMACIÓN AL SERVIDOR WEB

Para administrar los eventos filtrados por el EPL, se utilizan escuchadores (listeners) que recibirán los eventos de las consultas asignadas a cada ventana y los separan en eventos de llegada y eventos de salida de los lectores. Debido a que la administración de los eventos se lleva a cabo en tres escuchadores diferentes, es posible hacer la asignación de otros tipos de eventos en esta parte del proceso de análisis de la información. Es posible identificar los eventos de inicio y finalización del flujo de una persona dentro del edificio; esto cuando se presenten eventos registrados por los lectores Alien_1 y Alien_2 respectivamente para cada uno de los tipos de eventos. También se pueden identificar las alertas de seguridad dentro del edificio, al presentarse eventos procedentes del escuchador asignado al área de acceso restringido.

Una vez recibidos y clasificados los eventos por parte de los listener, se procede al envío de la información al servidor web mediante sockets utilizando el puerto 2445. De estos se encarga el método send(), cuya estructura se muestra en la Figura 19.

Figura 19. Estructura del método send()

```
private void send(String zona, String etiqueta, String lector, String tiempo, String tipoEvento){
    try{

        StringBuilder builder = new StringBuilder();
        String espacio = " ";
        builder.append(zona)
                .append(espacio)
                .append(etiqueta)
                .append(espacio)
                .append(lector)
                .append(espacio)
                .append(tiempo)
                .append(espacio)
                .append(tipoEvento)//1. llegada al lector || 2. Salida del Lector || 3. Entrada al edificio
                .append("\n"); //4. Salida del Edificio ||5. Acceso no Autorizado || 6. Error de Flijo

        output.write(builder.toString().getBytes());
        output.flush();
    }catch(Exception e){
        e.printStackTrace();
    }
}
```

Fuente: Autor del proyecto

El método send() a su vez define el formato de envío de la información; este contiene el área en la cual fue detectado el evento, la ID de la etiqueta, la ID del lector, la fecha y hora de la ocurrencia del evento, y por último el tipo de evento identificado. Todos estos parámetros son enviados, y recibidos por el servidor web en forma de caracteres y allí se lleva a cabo el parsing del evento para obtener por separado cada uno de los parámetros y proceder a su presentación en la web y al almacenamiento en la base de datos.

3.9. SERVIDOR WEB Y APLICACIÓN GIIBPT

Dentro de la documentación y tutoriales enfocados al desarrollo de aplicaciones con Rifidi, se plantea la configuración del servidor web y el desarrollo de la aplicación web dentro del mismo proyecto en el cual se lleva a cabo la configuración del servidor de Rifidi y donde se desarrollaron las consultas Esper encargadas de administrar los eventos provenientes de los dispositivos RFID; infortunadamente durante el proceso de desarrollo de la aplicación se comprobó que no era posible la inserción y utilización de nuevos documentos, archivos de todo tipo o plugins que hicieran parte de la aplicación web o que ayudaran en el proceso de desarrollo del sistema, generando como consecuencia, problemas y demoras en las fases de prueba y desarrollo del sistema.

Como solución a este inconveniente se hizo la configuración del servidor y el desarrollo de la aplicación web en un IDE independiente del utilizado para la parte de identificación por radiofrecuencia.

El IDE utilizado fue Eclipse, siendo este uno de los entornos más estables y configurable para el desarrollo de todo tipo de aplicaciones software. Adicionalmente al entorno de desarrollo se le agregó el framework de desarrollo Vaadin, que permite la creación de aplicaciones web, mediante la utilización de

código Java, haciendo la abstracción del código HTML del código Java y utilizando además componentes prefabricados y altamente configurables incluidos en el framework.

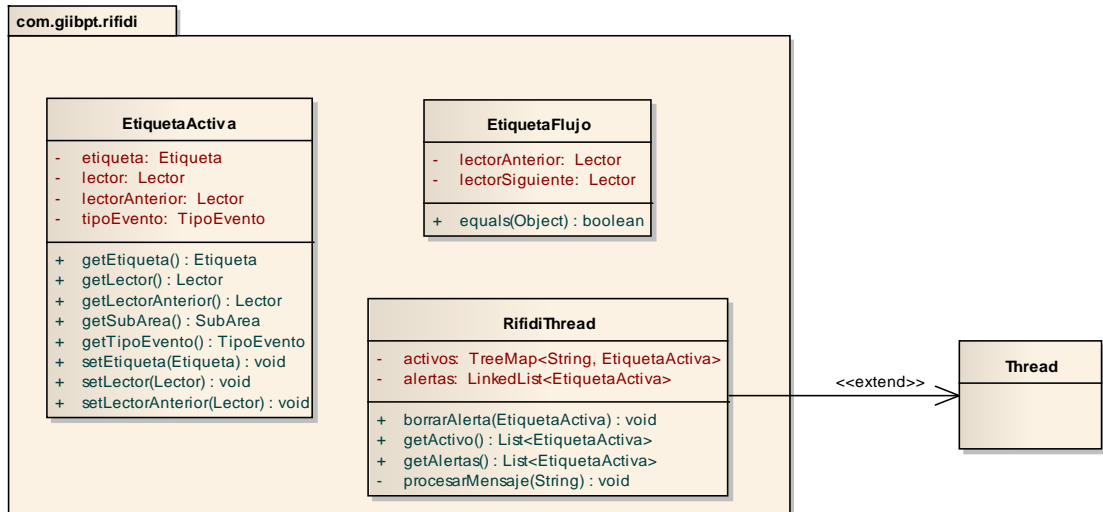
Desde el punto de vista del desarrollo y de la arquitectura del sistema, la aplicación web está dividida en cuatro paquetes principales; `com.giibpt`, `com.giibpt.rifidi`, `com.giibpt.model` y `com.giibpt.model.entities`. A continuación se explica cada uno de los paquetes y las funcionalidades que estos ofrece al sistema de monitoreo y rastreo de personas Giibpt.

3.9.1. Paquete `com.giibpt.rifidi`: Comunicación ente servidores y validación de flujo de visitantes

En la Figura 20 se presenta el diagrama de clases del paquete `com.giibpt.rifidi`. Este paquete está formado por cuatro clases y se encarga de la captura y administración de toda la información proveniente del servidor de Rifidi.

La clase `RifidiThread` ofrece varias de las principales funcionalidades al sistema. En un principio se encarga de proporcionar, mantener y administrar el canal de comunicación entre el servidor web y el servidor de Rifidi por medio de sockets utilizando hilos.

Figura 20. Diagrama de clases paquete com.giibpt.rifidi



Fuente: Autor del proyecto

Una vez establecida la comunicación entre los servidores, esta clase recibe la información de los mensajes enviados por el servidor de Rifidi y mediante un método creado para el procesamiento del mensaje, se hace el parsing y se separan cada unos de los datos que componen el evento, que son: etiqueta, lector, fecha del evento y el tipo de evento.

Con la información del evento lista para su análisis, se procede a registrar mediante la clase EtiquetaActiva, en qué lector se encuentra en un determinado momento una etiqueta y en cuál era el lector inmediatamente anterior a la posición actual. Estos dos datos se utilizan posteriormente para la verificación del flujo de visitantes dentro del edificio y para la creación y asignación de nuevos eventos, en caso de que las condiciones específicas presentadas en ese momento en el sistema así lo requieran.

En este punto es importante hacer énfasis y algunas aclaraciones acerca de la asignación de los tipos de eventos. Teniendo presentes los tipos de eventos que existen y que se mencionan en el numeral 3.4, una aclaración importante tiene

que ver con que existen condiciones específicas dentro los parámetros de entrada al sistema que pueden coincidir con uno o más tipos de eventos. Esto se da principalmente debido al hecho de que hay eventos que son muy generales y que no requieren de muchas condiciones para su cumplimiento, este es el caso de eventos como “Llegada a un lector”; este evento se registra en el sistema siempre que alguna etiqueta sea leída, siendo posible que en ese mismo momento se estén cumpliendo las condiciones para que se lleve a cabo la creación de otro evento debido a esa misma lectura, como por ejemplo “Acceso a un área restringida” o “Error de flujo”.

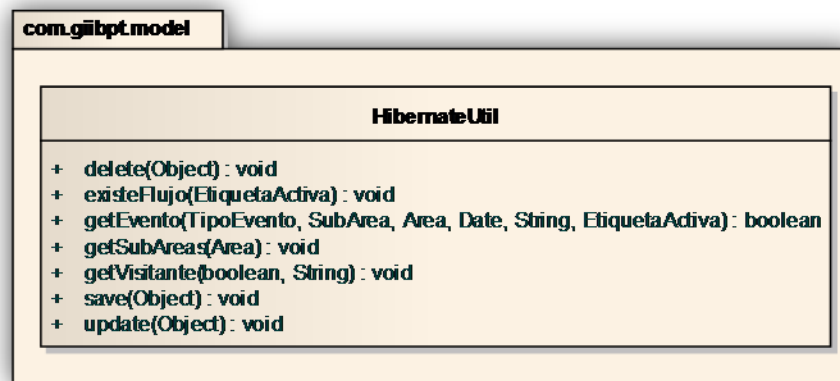
Continuando con el tema de asignación de tipos de eventos, es necesario aclarar que no todos los tipos de eventos se asignan y envían desde el servidor de Rifidi; hay eventos cuya validación y asignación se lleva a cabo en el servidor web, precisamente mediante la clase RifidiThread. Una vez realizado el parsing y la obtención de los datos de cada evento, se procede a hacer la validación de flujo de personas a través de los lectores, basándose en las posibles combinaciones presentadas anteriormente en la Tabla 1 correspondiente al numeral 3.4. Dichas combinaciones están almacenadas en la base de datos del sistema y cada vez que un evento ingresa al servidor web se hace la comparación de las propiedades del evento con las posibles tuplas entre lector anterior y lector actual que definen el correcto flujo de personas dentro del edificio. La organización y arquitectura de la base de datos se presentará más adelante el numeral 3.9.3. El proceso de validación de flujo se lleva a cabo en conjunto con otra clase llamada HibernateUtil del paquete com.giibpt.model.

3.9.2. Paquete com.giibpt.model: Paquete de gestión de base de datos

Este paquete está conformado por una sola clase llamada HibernateUtil, como se muestra en la Figura 21. Ofrece la funcionalidad de gestionar la base de datos, permitiendo la creación, eliminación y actualización de registros dentro de ésta.

Permite también la consulta de la información que se encuentra almacenada. Por ejemplo, la clase cuenta con el método `getEvento`, que devuelve la información de los eventos, basándose en unas condiciones especificadas, como la identificación de la persona, el lector en el cual se lleva a cabo el evento o la fecha. El método *existeFlujo* que permite verificar que exista un flujo de información dentro del sistema revisando la existencia del registro del lector actual y el lector anterior para cada etiqueta. La verificación de la existencia de flujo y la obtención de la información de los eventos es utilizada por la clase `RifidiThread` para cumplir con su tarea de validación de flujo de visitantes y con la creación de nuevos eventos reportando alertas de error de flujo en caso de que las condiciones así lo requieran.

Figura 21. Diagrama de clases del paquete `com.giibpt.model`

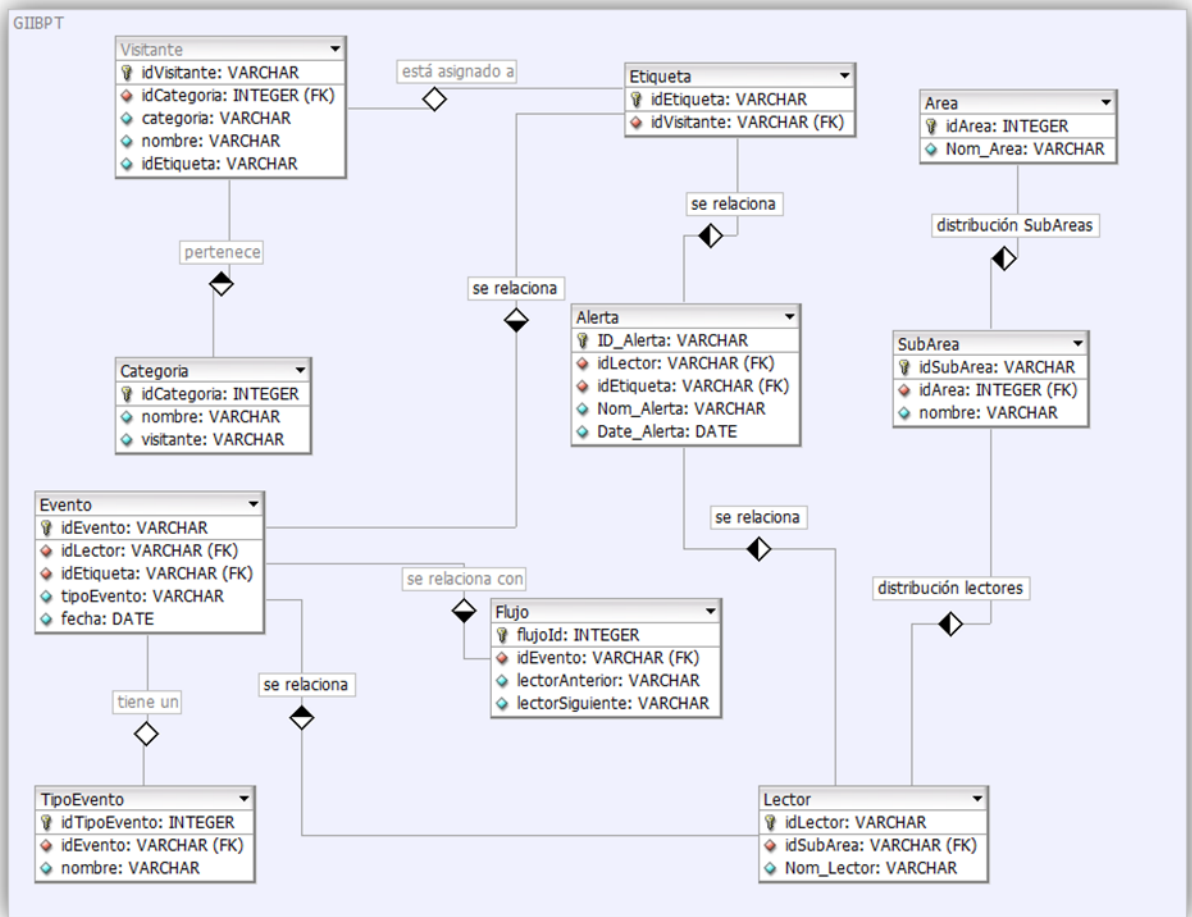


Fuente: Autor del proyecto

3.9.3. Paquete com.giibpt.model.entities: Definición del modelo y arquitectura de la base de datos

La base de datos del sistema GIIBPT está montada sobre el manejador de base de datos PostgreSQL y su administración y consulta se lleva a cabo mediante mapeo utilizando clases Java que utilizan la herramienta Hibernate para Eclipse. En la Figura 22 se presenta el modelo de Entidad/Relación de la base de datos del sistema GIIBPT.

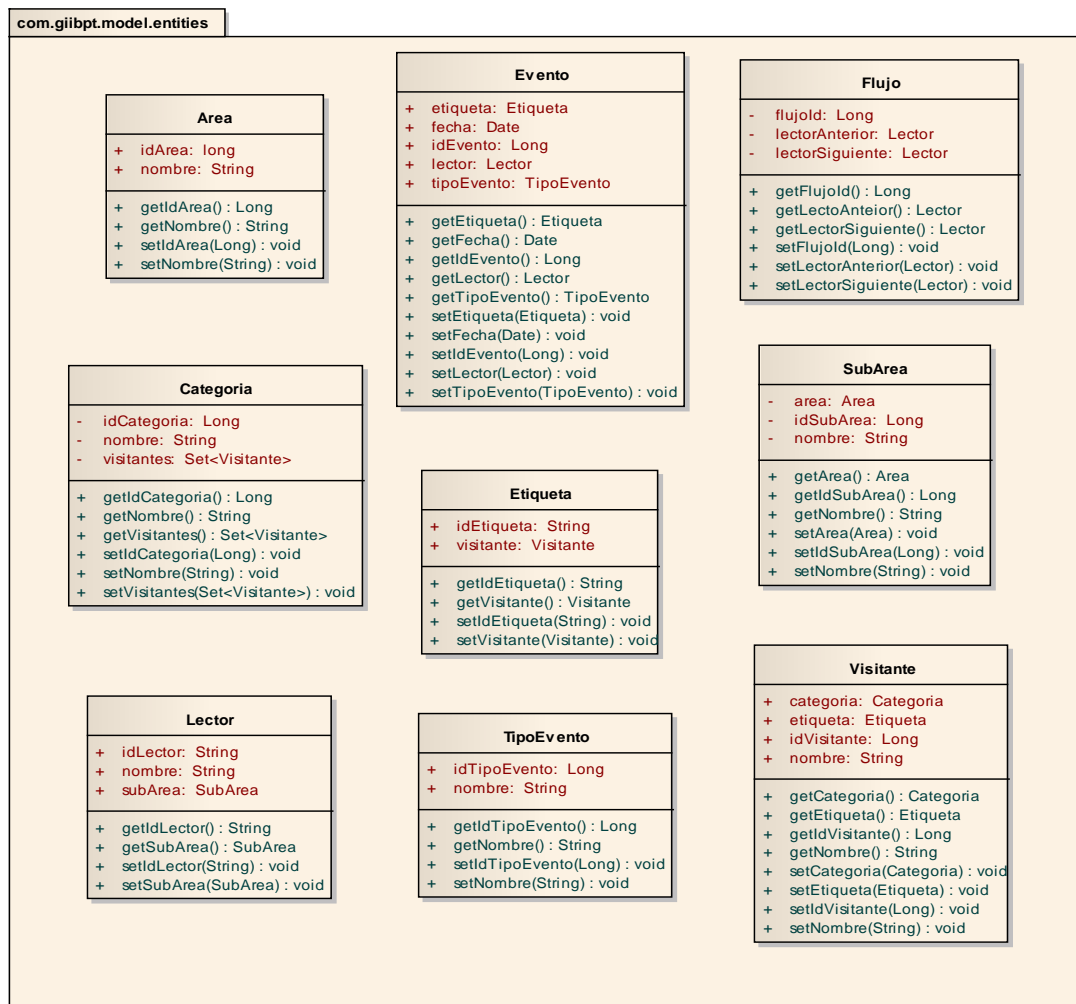
Figura 22. Modelo Entidad/Relación de la base de datos del sistema GIIBPT



Fuente: Autor del proyecto

El modelo E/R presentado en la Figura 22 fue utilizado como referencia para la creación de la base de datos utilizando Hibernate. En este proceso de creación de la base de datos, las tablas son tomadas como clases, dentro de las cuales se definen los atributos, se definen las llaves primarias y foráneas y la cardinalidad de las relaciones entre las diferentes tablas. La Figura 23 presenta el diagrama de clases del paquete com.giibpt.model.entities en el cual está programada toda la base de datos.

Figura 23. Diagrama de clase del paquete com.giibpt.model.entities



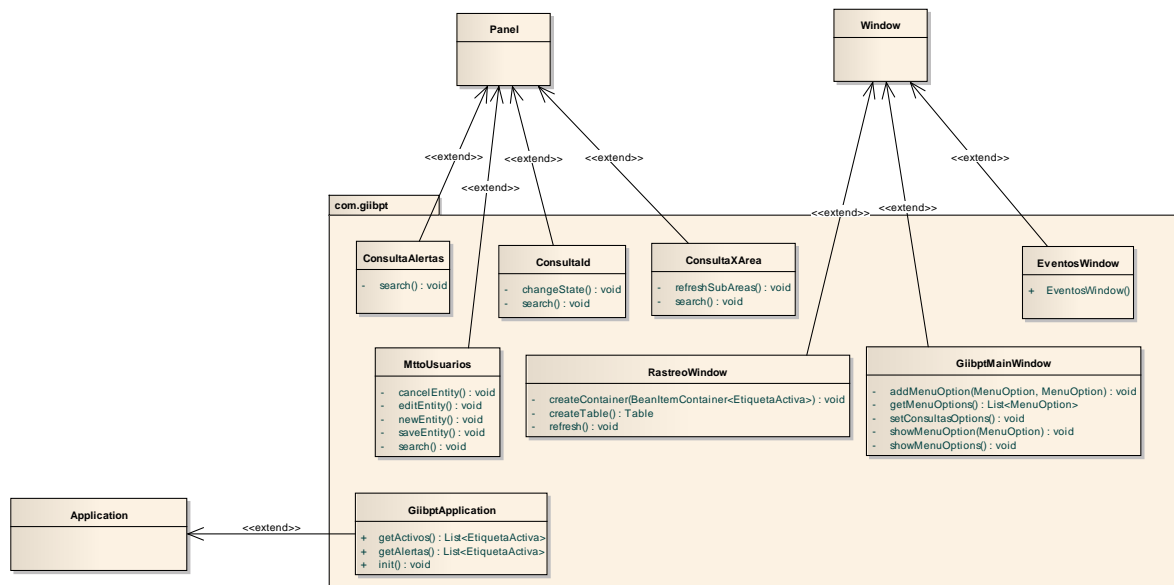
Fuente: Autor del proyecto

La creación y manipulación de la base de datos mediante mapeo, facilita en gran medida la creación de consultas y la presentación de la información en la página web del sistema. En el diagrama de clases del paquete se puede observar que cada una de las clases cuenta con sus métodos get y set, encargados de la extracción y asignación de la información a cada uno de los objetos de las clases, que en últimas son los registros en la base de datos del sistema.

3.9.4. Paquete com.giibpt: Presentación de la información en la página Web

El paso final de toda la información procesada por el sistema GIIBPT tiene lugar en la página web del usuario. Es mediante esta web que se puede apreciar el rastreo y monitoreo de personas dentro del edificio, ofreciendo una interfaz gráfica al usuario. La arquitectura utilizada para la creación y para darle las funcionalidades necesarias a la web, se presenta en la Figura 24, que contiene el diagrama de clase.

Figura 24. Diagrama de clases del paquete com.giibpt – Pagina web



Fuente: Autor del proyecto

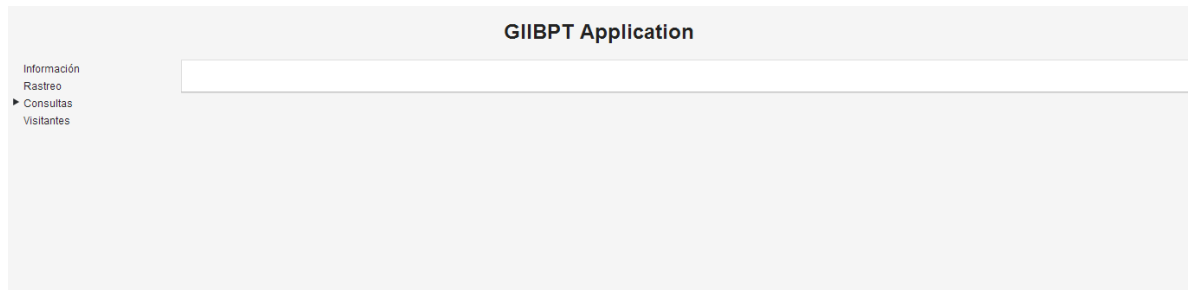
En la arquitectura de la web presentada anteriormente, cada una de las clases se encarga de un módulo de presentación de información u ofrece funcionalidades a alguno de ellos. Para empezar es importante recordar que la parte web de la aplicación fue desarrollada mediante Vaadin, por lo tanto la clase principal de la aplicación debe extender de la clase *Application* con el fin de poder utilizar las funcionalidades de Vaadin como plugin de desarrollo web.

En este caso la clase principal de la aplicación es *GiibptApplication*; como se dijo anteriormente esta clase extiende de la clase *Application* y se encarga principalmente de dos cosas. Primero, es la responsable de iniciar el hilo que recibe la información del servidor de Rifi. Y segundo, se encarga de definir la ventana principal de la aplicación web, donde se podrán ir agregando los diferentes componentes de la página, de tal forma que el usuario final de la aplicación vaya navegando por los diferentes módulos de consulta que ofrece la página web.

La primera ventana o ventana principal, que aparece al iniciar y sincronizar los servidores es la correspondiente a la ejecutada por la clase *GiibptMainWindow* inicialmente sobre esta ventana está cargado el menú de la aplicación web y sobre esta se van presentando las diferentes funcionalidades de acuerdo a las actividades que desee hacer el usuario final.

La Figura 25 presenta una vista de esta ventana principal en la cual se observa el menú con las diferentes funcionalidades del sistema GIBPT, rastreo, consultas, y visitantes.

Figura 25. Vista inicial de la aplicación web – Menú de navegación



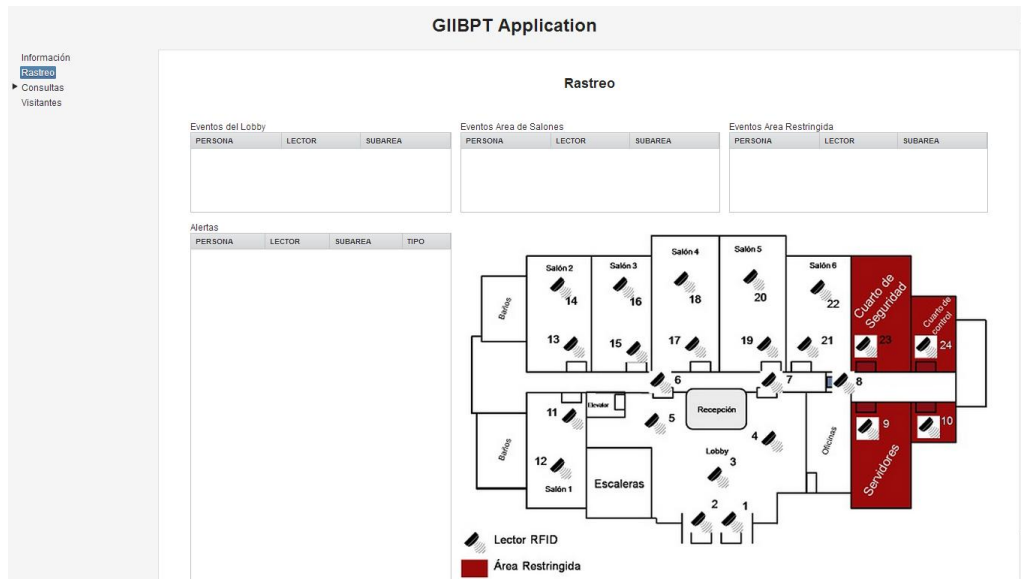
Fuente: Autor del proyecto

Cada una de las opciones del menú de navegación ofrece al usuario final del sistema una funcionalidad diferente en la página web. A continuación se explicarán cada una de las funcionalidades ofrecidas por el sistema y se presentarán pantallazos de cada una de ellas.

3.9.4.1. Funcionalidad de rastreo

Esta funcionalidad del sistema permite observar en tiempo casi real, el desplazamiento de una persona dentro del edificio, basado en las lecturas realizadas por los dispositivos RFID. Dicho desplazamiento se puede observar en esta vista mediante la ubicación de los diferentes eventos dentro de los diferentes campos definidos para cada una de las áreas del edificio. La Figura 26 presenta la vista de rastreo con los campos donde aparecen cada uno de los eventos. Cada uno de los eventos presentado en la vista tiene como información el nombre de la persona a la cual corresponde dicho evento, el lector que registró a esa persona y la subárea en la cual se encuentra dicho lector.

Figura 26. Vista del módulo de rastreo en la aplicación Web



Fuente: Autor del Proyecto

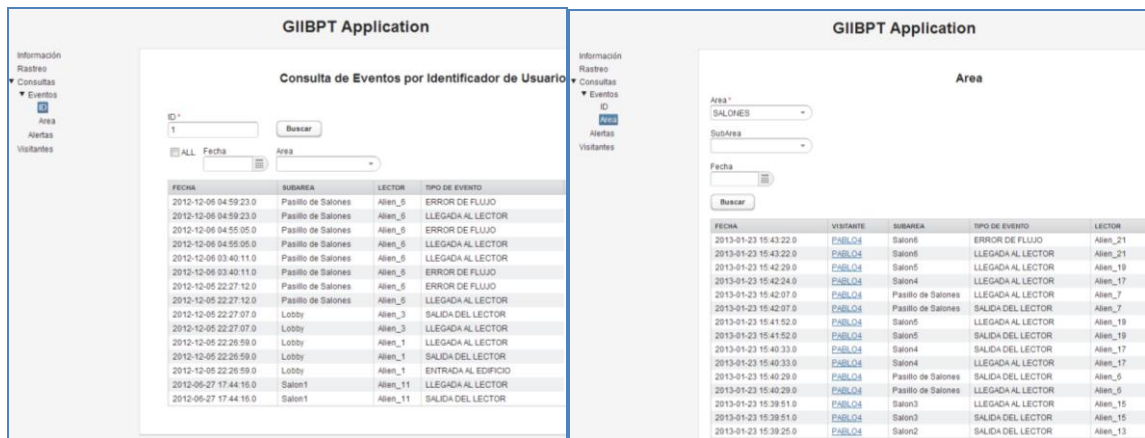
A medida que se van dando las apariciones y el registro de eventos dentro del sistema, en los tres campos superiores del área de rastreo, van apareciendo los eventos de acuerdo al lugar dentro del edificio en que se encuentre una persona. En dichos campos solamente aparecerá un evento por persona a la vez, dicho evento corresponde al último lugar en que haya sido registrada esa persona dentro del edificio y se irá actualizando de acuerdo al los movimientos que haga dentro del edificio; basados en esto, es posible entonces que los eventos presentados en la vista de rastreo para una persona determinada, estén cambiando de un espacio a otro dentro de las áreas de presentación. El área de presentación de los eventos a la izquierda, corresponde al área del lobby del edificio, el área del centro presenta los eventos que están sucediendo en los salones, el área de la derecha muestra los eventos sucedidos en el los lugares del edificio con acceso restringido y finalmente el espacio de presentación de eventos que se encuentra en la parte inferior izquierda del módulo de rastreo, corresponde a la presentación de todas las alertas ocurridas durante el flujo de personas del

edificio. En este campo de presentación de alertas es posible que se encuentre más de un evento relacionado a una misma persona, esto con el fin de mantener al tanto al usuario del sistema de todos los ingresos en áreas restringidas o de posibles errores en el registro del flujo dentro del edificio.

3.9.4.2. Consulta de historiales de eventos en el sistema

Cada uno de los eventos que ingresan al sistema, son almacenados en la base de datos y pueden ser consultados por el usuario final, en caso de que se requiera hacer consultas de eventos pasados. La consulta de historiales del sistema permite hacer búsqueda de eventos o de alertas. En la Figura 27 se muestra el árbol de opciones de consulta desplegado y los módulos de consulta de eventos.

Figura 27. Interfaces de consulta de eventos basado en ID de Visitante y en Área



Fuente: Autor del proyecto

El sistema permite hacer consulta de eventos de dos formas diferentes, dependiendo de las necesidades del usuario final del sistema.

La primera forma de consulta, permite presentar en pantalla todos los eventos ocurridos en el sistema correspondientes a un visitante en específico, es por esto que el parámetro de búsqueda ID es obligatorio para poder llevar a cabo la

consulta. En caso de que este campo esté vacío, el sistema hará una alerta y solicitará nuevamente el ingreso de la identificación de un visitante. Dentro de esta forma de consulta también es posible definir parámetros de búsqueda como la fecha de ocurrencia del evento y el área en el cuál se llevó a cabo el evento, restringiendo de esta forma el volumen de eventos presentado al usuario final y facilitando la búsqueda de eventos específicos.

La segunda forma de hacer consulta de eventos tiene como objetivo principal mostrar todos los eventos ocurridos en un área específica del edificio, sin importar a quién estén relacionados esos eventos. Como parámetros adicionales dentro de la consulta es posible especificar una subárea; reduciendo de esta forma el rango geográfico de la consulta dentro del edificio; y una fecha específica de ocurrencia de los eventos.

3.9.4.3. Consulta de historiales de alertas en el sistema

Partiendo del principio de que todas alertas ocurridas en el sistema son consideradas como eventos, éstas también son almacenadas en la base de datos y es posible realizar una consulta sobre ellas. El porqué de la utilización de una vista especial para este tipo de eventos es simplemente por la importancia que podrían llegar a tener dichos eventos en un sistema implantado, bajo condiciones y políticas de funcionamiento y seguridad reales. Complementado lo planteado en el numeral 3.4., las alertas identificadas y registradas dentro del sistema corresponden a eventos de dos tipos: Alertas de acceso a áreas restringidas y errores de flujo. Estos dos tipos de eventos pueden otorgar información importante acerca de los diferentes controles de seguridad utilizados por una empresa, basándose en la cantidad de accesos a áreas no restringidas registrados en un determinado periodo de tiempo. También puede servir de apoyo en el proceso de mantenimiento y calibración de los dispositivos lectores RFID, al ofrecer un

registro de los posibles errores de flujo ocurridos en el sistema, debidos posiblemente a fallos en la calibración de los equipos o por la existencia de puntos ciegos de lectura en el edificio a causa de la mala ubicación de las antenas.

Figura 28. Vista de la interfaz de consulta de Alertas

FECHA	TIPO DE EVENTO	VISITANTE	SUBAREA	LECTOR
2012-12-06 04:59:23.0	ERROR DE FLUJO	ANGEL	Pasillo de Salo	Alien_6
2012-12-06 04:55:05.0	ERROR DE FLUJO	ANGEL	Pasillo de Salo	Alien_6
2012-12-06 03:40:11.0	ERROR DE FLUJO	ANGEL	Pasillo de Salo	Alien_6
2012-12-05 22:27:12.0	ERROR DE FLUJO	ANGEL	Pasillo de Salo	Alien_6
2012-06-26 13:31:19.0	ERROR DE FLUJO	ANGEL	Salon6	Alien_21
2012-06-26 09:37:10.0	ERROR DE FLUJO	ANGEL	Pasillo de Salo	Alien_6
2012-06-26 09:36:21.0	ERROR DE FLUJO	ANGEL	Pasillo de Salo	Alien_6
2012-06-13 15:30:27.0	ERROR DE FLUJO	ANGEL	Salon4	Alien_17

Fuente: Autor del proyecto

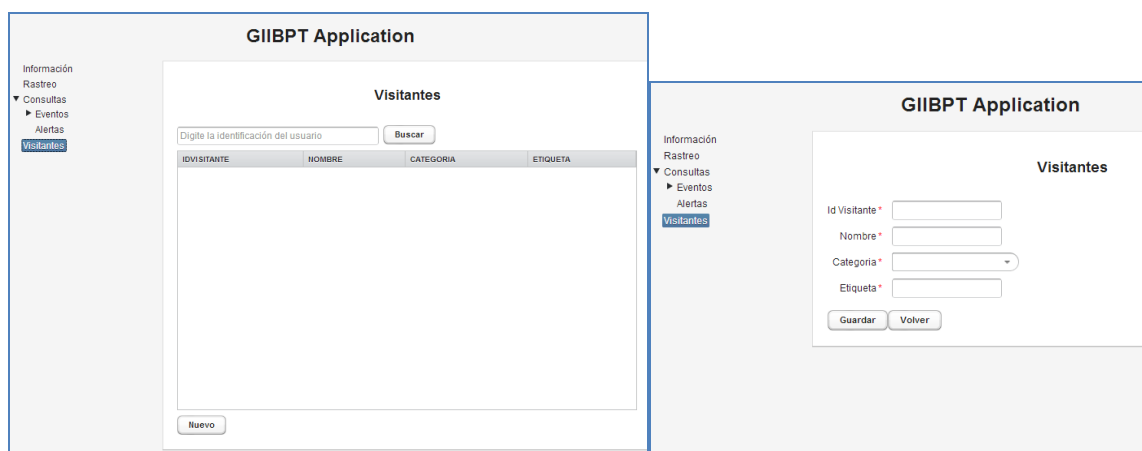
La Figura 28, presenta la vista de la interfaz de consulta para las alertas ocurridas en el sistema. Dicha interfaz permite hacer una combinación de diferentes parámetros con el fin de identificar eventos específicos. Es posible combinar el tipo de evento con la identificación del visitante al que está asignado dicho evento, con el área en la que se da la ocurrencia del evento y por último con la fecha en que este evento sucedió. Para realizar consulta de alertas no es necesario definir ningún parámetro en específico, en caso de no definir alguno, sistema presentará todas las alertas almacenadas en la base de datos, organizándolas cronológicamente de la más reciente a la más antigua.

3.9.4.4. Administración de visitantes y asignación de etiquetas RFID

Es importante que el sistema GIBPT, ofrezca la posibilidad de consultar los existentes y agregar nuevos visitantes en la base de datos y con esto asignar una etiqueta RFID diferente a cada uno de ellos, esto permite hacer frente al muy seguro ingreso de nuevos visitantes de un edificio, en este caso en particular el CENTIC, al cual cada semestre ingresan nuevos estudiantes, profesores y administrativos.

La administración de visitantes se lleva a cabo mediante la interfaz de visitantes que se muestra en la Figura 29.

Figura 29. Vista de la interfaz de administración de Visitantes



Fuente: Autor del proyecto

En primera instancia, esta interfaz permite hacer una consulta de visitantes almacenado en la base de datos por medio del número de identificación de cada uno de ellos. Es posible también listar todos los visitantes registrados en la base de datos, simplemente dejando vacío el campo para la identificación y procediendo a buscar. Una vez obtenidos los resultados de la búsqueda, es posible hacer click sobre el o los resultados obtenidos y proceder a hacer una edición a la información

del visitante, dado el caso por ejemplo que el visitante pierda su etiqueta de identificación RFID y sea necesario asignarle una nueva. También es posible hacer un cambio de categoría de visitante, en caso de ser requerido; aunque una situación de este tipo es menos probable.

Como se mencionó anteriormente, esta interfaz permite hacer el registro de nuevos visitantes en el sistema. Para registrar un nuevo visitante en el sistema se solicita información básica del mismo, como número de identificación, nombre y categoría a la que pertenece. Por último es indispensable asignar al nuevo visitante el código de una etiqueta RFID mediante la cual se realizará el rastreo en el sistema. El código de cada etiqueta es único, y está formado por un número hexadecimal de 24 caracteres. Para obtener el código de una nueva etiqueta es necesario hacer la creación de dicho dispositivo en el Tag Streamer, de la forma en que se presenta en el numeral 3.6 y hacer la copia manual del código de la etiqueta que aparece en la Figura 14 e ingresarlo en campo correspondiente a Etiqueta. Este proceso de creación y asignación de una etiqueta nueva a un visitante, correspondería en un sistema implantado, a la compra y registro de un lote de etiquetas y a la asignación paulatina de dichas etiquetas a los visitantes, el sistema GIBPT, no puede por sí mismo crear el código de identificación de las etiquetas RFID, pues ese código depende exclusivamente de la etiqueta física como tal y del fabricante del dispositivo.

CAPITULO 4

4. VERIFICACIÓN DEL DESEMPEÑO Y CORRECTO FUNCIONAMIENTO DEL PROTOTIPO SOFTWARE

Dentro de los objetivos del proyecto se plantea la verificación del desempeño y correcto funcionamiento del prototipo software desarrollado en cuanto a cumplimiento de la lógica y reglas del negocio definidas y a la presentación de la información al usuario final de la aplicación.

Con el fin de dar cumplimiento a este objetivo, se realizaron diferentes pruebas durante el proceso de desarrollo del sistema. Por medio de estas pruebas se buscó comprobar el correcto funcionamiento del sistema, la navegación entre los diferentes módulos de la aplicación y la integridad de los datos almacenados.

Se definieron situaciones específicas entre los lectores y las etiquetas RFID, bajo las cuales se esperaban salidas puntuales en la interfaz de usuario, de acuerdo a las reglas de flujo de visitantes y a las reglas para las alertas de seguridad y de errores de flujo.

La integridad de los datos almacenados en base de datos se comprobó mediante la comparación de los resultados esperados en cada situación con las consultas diseñadas para el funcionamiento de la aplicación y con consultas realizadas directamente en la base de datos, sin tener como intermedio la aplicación Web desarrollada.

Estas pruebas buscaban validar el correcto funcionamiento de los siguientes aspectos del prototipo software:

Tabla 3. Pruebas realizadas y validación de las mismas.

<ul style="list-style-type: none"> • Conexión entre los dispositivos RFID de lectura emulados mediante Tag Streamer y el servidor de Rifidi. 	✓
<ul style="list-style-type: none"> • Registro efectivo de una etiqueta o de un grupo de etiquetas emuladas, por parte de cada uno de los lectores del sistema y envío de esa información al servidor de Rifidi para su procesamiento. 	✓
<ul style="list-style-type: none"> • Filtrado de eventos enviados por los lectores omitiendo repeticiones, evitando de esta forma saturación del sistema por flujo de información innecesaria y redundante. 	✓
<ul style="list-style-type: none"> • Conexión entre el servidor de Rifidi y el servidor Web y envío de la información completa y correcta hacia este segundo servidor. 	✓
<ul style="list-style-type: none"> • Recepción, parcing y almacenamiento de la información entrante al servidor Web. 	✓
<ul style="list-style-type: none"> • Aplicación de las reglas del negocio para el flujo de personas dentro del edificio a cada uno de los eventos registrados por el sistema GIIBPT. 	✓
<ul style="list-style-type: none"> • Presentación de la información registrada por los lectores RFID cada una de las interfaces Web del sistema. 	✓
<ul style="list-style-type: none"> • Consulta, edición y creación de usuarios en el sistema. 	✓

Fuente: Autor del proyecto

CAPITULO 5

5. CONCLUSIONES

- Aunque el concepto de monitoreo y rastreo de la posición de personas dentro de un edificio, podría llegar a ser considerado un tanto extremista y en cierta forma una violación a la privacidad, es importante reconocer que los niveles y protocolos de control de acceso en diferentes instituciones, ha venido aumentando debido a las facilidades ofrecidas por las nuevas tecnologías y a la necesidad de llevar un registro detallado de todo lo que ocurre dentro de las instalaciones, garantizando de esta forma el desarrollo normal y seguro de las actividades dentro del edificio.
- La finalización exitosa del desarrollo de la aplicación GIIBPT ofrece pautas importantes que apoyan el desarrollo de futuras aplicaciones basadas en la Tecnología de Identificación por Radio Frecuencia. Estas pautas incluyen la configuración del servidor encargado de la administración de los dispositivos RFID, definición de sentencias para filtrar la información provenientes de esos dispositivos, definición de las reglas para el monitoreo y rastreo de los objetos; y presentación de la información al usuario final.
- La creación de módulos independientes para cada una de las funcionalidades que ofrece la aplicación, facilitó y agilizó el desarrollo software, permitió la identificación y corrección rápida de errores, además ofrece facilidades al momento de hacer modificaciones, cambios o adiciones de nuevas características al sistema.

- La correcta definición de las reglas del negocio para el flujo de personas dentro del edificio, permite tener una idea general y clara de las funcionalidades y consideraciones a tener en cuenta durante el proceso de diseño y desarrollo de prototipo software, dado que son estas reglas las que definen el correcto funcionamiento del sistema en cuanto al monitoreo de la posición y flujo de visitantes dentro del edificio.
- El proceso de emulación de dispositivos RFID llevado a cabo en este proyecto hizo posible la realización de forma rápida y eficiente, de las pruebas necesarias para validar el buen funcionamiento de sistema, a pesar de no contar con la disponibilidad física de los dispositivos.
- La emulación de dispositivos RFID ofrece una muy buena idea del comportamiento real de estos componentes hardware en la implementación e implantación de un sistema real; esto hace posible tener en cuenta opciones de configuración importantes como la frecuencia de lectura de los dispositivos, la posición de las antenas de cada lector dentro del edificio y la separación entre ellos.
- Sin lugar a dudas, Rifidi posee un conjunto de herramientas excelentes para llevar a cabo el desarrollo y prueba de sistemas relacionados con la tecnología RFID, permitiendo llevar a cabo la emulación, conexión y administración de los dispositivos físicos de una manera rápida; siendo la emulación de dichos dispositivos una aproximación bastante cercana al comportamiento de los lectores RFID reales.
- La investigación sobre nuevas tecnologías y la apropiación de las mismas para el desarrollo de este proyecto, forman parte de un gran aporte no solo

a nivel personal sino también institucional, pues abre puertas en nuevas áreas de investigación y desarrollo, siendo este uno de los fines principales de la Universidad Industrial de Santander y más puntualmente del Grupo de Investigación en Ingeniería Biomédica, que se ha destacado siempre por la innovación en sus áreas de investigación y en la utilización de tecnologías de punta para el desarrollo de nuevos proyectos.

CAPITULO 6

6. RECOMENDACIONES

La finalización de este proyecto es simplemente el principio de un proceso de investigación y de aplicación de nuevas tecnologías que aún tiene muchos caminos por recorrer, a continuación se plantean algunas recomendaciones para el desarrollo de nuevos proyectos relacionados con el tema.

- La utilización de la plataforma Rifidi fue de gran ayuda durante el proceso de desarrollo del proyecto, por tal razón es importante hacer una investigación mucho más profunda de todas sus aplicaciones, de las herramientas que esta ofrece y de los posibles cambios y personalizaciones que podrían hacerse sobre esta con el fin de obtener un entorno de desarrollo mucho más completo y funcional.
- Aunque la emulación de los dispositivos RFID es de gran ayuda para llevar a cabo el desarrollo de un prototipo software funcional, es indispensable dar un paso al desarrollo en ambientes reales de implantación, utilizando dispositivos Hardware reales con el fin de afianzar conocimientos y adquirir experiencia en el manejo de dicha tecnología. Las aplicaciones desarrolladas pueden en un principio estar basadas en situaciones tales que solo sea requerido uno o dos lectores RFID, teniendo en cuenta el elevado costo de dichos dispositivos, el cual puede oscilar entre los \$150US y los \$2500US, dependiendo de varios factores como su rango de lectura y métodos de conexión con el servidor.

Referencias

- 1. *The History of RFID*. **Land, Jeremy**. s.l. : IEEE POTENTIALS, 2005.
- 2. Rifidi | Software Defined RFID. [En línea] [Citado el: 16 de Marzo de 2010.] <http://rifidi.org/index.html>.
- 3. *Tesis: "Estado del Arte en Tecnologías RFID"*. **Gotor, Eva**. Madrid : Escuela Universitaria de Informática – Universidad Politécnica de Madrid, 2009.
- 4. *Trabajo Final de Carrera: "Diseño y construcción del sistema RFID para un expositor inteligente"*. **Payá, José**. s.l. : Universidad Politécnica de Cataluña, 2008.
- 5. *Automatización de inventarios, prestamos y devolucion de libros en biblioteca por medio de tecnología RFID*. **Argüello Diaza, Yvette Carolina y Espinosa Peralta, Ricardo**. 2009.
- 6. Alien Technology - Products - RFID Readers. [En línea] [Citado el: 15 de 04 de 2010.] <http://www.alientechnology.com/readers/alr9800.php>.
- 7. *RFID Emulation in Rifidi Environment*. **Palazzi, Claudio E., Ceriali, Alessandro y Dal Monte, Marco**. Padova, Italia : Universita degli Studi di Padova.
- 8. Rifidi.org. [En línea] http://rifidi.org/documentation_edgeserver_architecture.html.
- 9. *Master Thesis Computer Science: "Complex Event Processing"*. **Dekkers, Paul**. s.l. : Radbound University Nijmegen, 2007.
- 10. Esper - Complex Event Processing. *Tutorial*. [En línea] [Citado el: 18 de 03 de 2010.] <http://esper.codehaus.org/tutorials/tutorial/tutorial.html>.
- 11. **Haase, Kim**. <http://www.oracle.com/>. [En línea] http://download.oracle.com/javaee/1.3/jms/tutorial/jms_tutorial-1_3_1.pdf.
- 12. **Amo, Fernando y Martinez, Loïc**. *Introducción a la Ingeniería del Software - Modelos de Desarrollo de Programas*. s.l. : Delta Publicaciones, 2005.
- 13. **Sommerville, Ian**. *Ingeniería del Software*. s.l. : Pearson, 2005. 84-7829-074-5.

BIBLIOGRAFÍA

- **Amo, Fernando y Martinez, Loïc.** *Introducción a la Ingeniería del Software - Modelos de Desarrollo de Programas.* s.l. : Delta Publicaciones, 2005.
- **Argüello Diaz, Yvette Carolina y Espinosa Peralta, Ricardo.** *Automatización de inventarios, prestamos y devolucion de libros en biblioteca por medio de tecnología RFID.* 2009.
- **Dekkers, Paul.** s.l. *Master Thesis Computer Science: "Complex Event Processing"* : Radbound University Nijmegen, 2007.
- **Gotor, Eva.** *Tesis: "Estado del Arte en Tecnologías RFID"*. Madrid : Escuela Universitaria de Informática – Universidad Politécnica de Madrid, 2009.
- **Haase, Kim.** <http://www.oracle.com/>. [En línea]
http://download.oracle.com/javaee/1.3/jms/tutorial/jms_tutorial-1_3_1.pdf.
- **Land, Jeremy.** s.l. *The History of RFID: IEEE POTENTIALS*, 2005.
- **Palazzi, Claudio E., Ceriali, Alessandro y Dal Monte, Marco.** *RFID Emulation in Rifidi Environment.* Padova, Italia : Universita degli Studi di Padova.
- **Payá, José.** s.l. *Trabajo Final de Carrera: "Diseño y construcción del sistema RFID para un expositor inteligente"*: Universidad Politécnica de Cataluña, 2008.
- **Sommerville, Ian.** *Ingeniería del Software.* s.l. : Pearson, 2005. 84-7829-074-5.
- Alien Technology - Products - RFID Readers. [En línea] [Citado el: 15 de 04 de 2010.]
<http://www.alientechnology.com/readers/alr9800.php>.
- Esper - Complex Event Processing. *Tutorial.* [En línea] [Citado el: 18 de 03 de 2010.]
<http://esper.codehaus.org/tutorials/tutorial/tutorial.html>.
- Rifidi | Software Defined RFID. [En línea] [Citado el: 16 de Marzo de 2010.]
<http://rifidi.org/index.html>.
- Rifidi.org. [En línea] http://rifidi.org/documentation_edgeserver_architecture.html.