UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

SISTEMA DE SOPORTE PARA CONTROL DE INVENTARIOS MEDIANTE RFID.

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

CARLOS ALFREDO COLLAO VILCHES

PROFESOR GUÍA:
SR. JOSÉ MIGUEL PIQUER GARDNER

MIEMBROS DE LA COMISIÓN:
SR. PABLO BARCELÓ BAEZA
SR. PATRICIO NELLEF INOSTROZA FAJARDIN

SANTIAGO DE CHILE
ENERO 2008

Resumen

Esta memoria se basa en el desarrollo de un sistema de control de circulación de volúmenes dentro de la biblioteca de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile. Este desarrollo se enmarca en la generación de nuevas herramientas para expandir la funcionalidad de los equipos adquiridos por la Facultad para mejorar la gestión bibliotecaria, proyecto que se ha estado implementando en la Facultad desde hace casi 2 años.

Este proyecto, al implementar el modelo de manejo del sistema de circulación de volúmenes, genera una plataforma de control para el portal de circulación de volúmenes de biblioteca, pero que a su vez puede ser extensible a otras aplicaciones bibliotecarias que requieran comunicación usando módulos RFID.

La complejidad de ésta memoria estaba en poder integrar todas las piezas del control, incluyendo el controlar la nueva tecnología e integrarla con los sistemas actuales en funcionamiento en biblioteca, sin interrumpir los servicios ni reemplazar sistemas para poner en marcha ésta plataforma.

La solución desarrollada logra cumplir todas éstas exigencias, las que son posibles de ser extendidas para mejorar la completitud de los servicios. Todo en el sentido de generar una mejor administración de los recursos de biblioteca, tanto en el control de espacio de colección general así como en controlar la cantidad de volúmenes presentes.

Además, este desarrollo puede ser usado para crear nuevas aplicaciones, que desarrollen soluciones para el manejo, gestión y control dentro de la biblioteca misma o como base para desarrollos de similares características en otras áreas.

Agradezco a mis padres, por su apoyo incondicional durante toda mi vida. Y a Tina, Sebastian y Klara por incentivarme en estos importantes años.

Tabla de Contenidos

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

En este trabajo de memoria se desarrolló un sistema que registra y controla el flujo de los libros de la biblioteca de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, mediante equipos y etiquetas electrónicas usando tecnología de Identificación mediante Radio Frecuencia[1], en la zona de colección general de la biblioteca.

Usando esta tecnología podemos identificar como elementos únicos, a cada elemento que tenga presente una etiqueta de identificación adosada[2], de forma transparente y en la mayoría de los casos, en libre circulación.

Es así como la biblioteca de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile ha estado implantando en sus libros estos TAGs RFID, para el monitoreo y control de préstamo de estos.

Esta tecnología ofrece muchas ventajas frente a otros sistemas de monitoreo e inventario, como los sistemas basados en ID numéricos o mediante códigos de barras, ya que podemos:

[1] RFID, por sus siglas en Inglés
[2] más conocido como TAG RFID

- Saber la posición de lo que se está monitoreando.
- Analizar los procesos con mayor velocidad, al hacerlos en cada segundo y sin detener las cadenas de procesos.
- Controlar de formas más avanzadas, al poder tener los registros o procesos actualizados en línea.

En la actualidad, los sistemas de la biblioteca están siendo administrados con códigos de barras numéricos y únicos, adosados a la tapa trasera de cada volúmen existente.

Esto hace que los volúmenes con que cuenta la colección general de la biblioteca de la Facultad sean fácilmente individualizables, pero no se logra una gestión completa de la información de, por ejemplo, el flujo interno de los volúmenes dentro del área de lectura de la biblioteca, sino que solo podemos identificar cuando salen de esta.

Así, con este sistema que monitorea volúmenes que son retirados del área de colección general, podemos saber cuáles son los libros que son más solicitados por los alumnos y reorganizar la cantidad de volúmenes existentes.

Además, el sistema completo puede ser extensible tanto para nuevas aplicaciones, ya sea en funcionamiento en la biblioteca misma, pero principalmente tener un marco de trabajo que sea integrable a otras áreas en los cuales los sistemas de análisis, monitoreo y control de materiales sea aplicable, es decir, cualquier ambiente en el que se necesita mantener un inventario “inteligente".

1.1. Motivación

Muchas son las instituciones o empresas que deben almacenar y controlar grandes volúmenes de materiales, por ejemplo:

- El monitoreo de prendas de vestir y calzado, desde su creación en la fábrica, monitoreo de intermediarios, hasta el control del detalle de venta.

- El control de vida útil de herramientas en procesos productivos, tanto para la empresa productiva como para el monitoreo de reemplazo por parte de los proveedores.
- La gestión de uso, préstamo y renovación de libros en biblioteca.

La gestión de todos estos materiales presenta múltiples desafíos al momento de, por ejemplo, conocer dónde se encuentra el material, tanto a nivel de bodegaje como dentro de la cadena productiva. O bien, conocer quién tiene la responsibilidad del material y dónde este se encuentra.

Todos estos procesos tienen un punto primordial en común, que está en la necesidad de mantenerlos ubicados en todo momento.

Hoy en día, el control se realiza mediante control humano apoyado por sistemas de códigos de barras (Fig. **??**), los cuales poseen múltiples desventajas,como:

- Tener que ubicar un lugar específico y visible dónde localizar con el haz lector el código de barras para cada uno de los productos,
- depender que la persona que está a cargo del inventario, haga su trabajo de forma efectiva, o
- en el caso de sistemas más automatizados, de que el código de producto se encuentre exactamente en la zona de transporte en la cual está el lector de códigos de barras.

Figura 1.1: Lectura de código de barras numéricos

Tampoco poseemos suficiente confianza que los materiales estén donde se supone que deberían estar, puesto que al poseer un inmenso volúmen de materiales, contabilizar todos los volúmenes de una biblioteca por ejemplo, es una tarea que no se logra sino en muchos días de revisión, y ni aún así se tiene una certeza absoluta de lo contabilizado.

Así, un modelo tradicional de control de inventarios depende en el mejor de los casos, del ingreso y control de los materiales mediante lectura de códigos de barras con personal especializado ya sea para entregar, gestionar o ingresar inventarios, o en el peor de los casos, funcionarios contando cuántos materiales existen y digitando a mano los datos. (Fig. **??**).

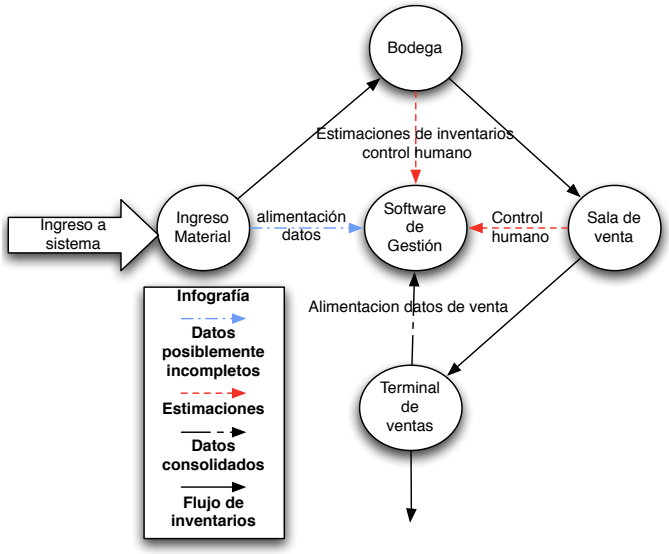

Figura 1.2: Sistema general de flujo de datos en procesos de negocios

En cambio, una de las grandes ventajas de los procesos controlados con sistemas de radio frecuencia, es que no necesita que el TAG esté en una zona visible del material a monitorear, pudiendo estar en zonas no visibles. Además, dependiendo de la tecnología, es posible rastrear con el mismo equipo a decenas de TAGs por segundo, sin tener que depender necesariamente de personal calificado (Fig. **??**), y ofreciendo además ventajas tales como:

- Controlar en tiempo real, dónde y en qué estado de operación se encuentran los materiales, por ejemplo en control de los equipos o repuestos en faenas industriales.
- Determinar si un material está en el lugar en el que debería estar, para saber si el equipamiento está realmente siendo utilizado en la faena asignada.
- Controlar masivamente el tráfico de elementos entre empresas - como sería el control de mercadería traspasada desde un proveedor hacia una tienda.
- Controlar el tráfico de materias entre unidades en la empresa - para saber, por ejemplo, exactamente cuáles artículos han pasado de bodega a sala de ventas, identificando el tamaño, modelo, color, etc., del artículo en cuestión.

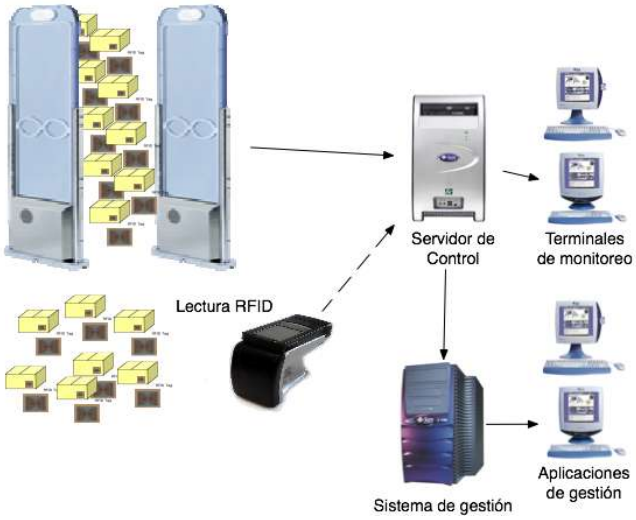

Figura 1.3: Modelo general uso de sistemas RFID

Con estos datos en los registros históricos de los sistemas de gestión de la organización se pueden generar nuevas cadenas de análisis, por ejemplo:

- Controlar el uso correcto de los materiales
- Maximizar la vida útil de los equipos
- Programar mantenciones dados ciertos patrones predictivos de operación
- Controlar las bajas abruptas de inventarios o sobre stock de productos
- Mejorar las curvas de estimación de productos más solicitados, pudiendo optimizar el uso de espacio tanto en bodegas como en salas de ventas.

Es por esto que los sistemas RFID, que determinan el posicionamiento de los materiales con individualización, precisión y factor de temporalidad, sean tan valiosos para la gestión de los procesos productivos, ya que podemos monitorear grandes volúmenes de materiales durante toda cadena productiva o comercial minimizando la intervención humana (Fig. **??**).

Con todos estos antecedentes, se desarrolló un sistema que interactúa con los equipos de Radio Frecuencia y los sistemas de control, gestión y préstamo ya existentes para la biblioteca,entregando

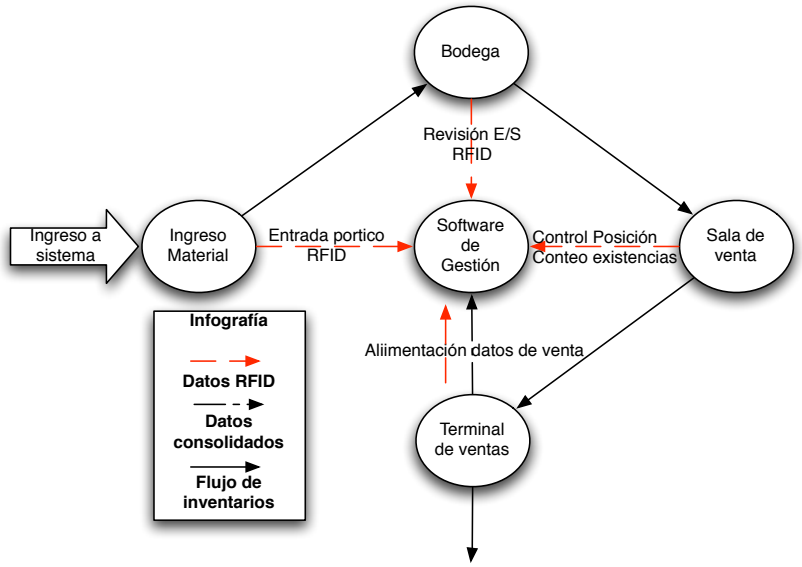

Figura 1.4: Sistema de flujo de datos a software de gestión basado en modelo RFID

nuevas herramientas para una mejor y más completa administración del Catálogo de Colección General disponible a público, con un modelo extensible al manejo y control de otros procesos.

1.2. Objetivos

1.2.1. Objetivo general

El objetivo general del trabajo del alumno corresponde a:

Estudiar los sistemas bibliotecarios, la tecnología RFID y analizar e implementar un sistema base de gestión y control de volúmenes en circulación en la sala de lectura de colección general, para la biblioteca de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.

1.2.2. Objetivos específicos

- Estudiar el funcionamiento de los servicios bibliotecarios de la Universidad de Chile (SISIB), cómo se distribuye la información entre sus diversas bibliotecas y cómo se sincroniza con los sistemas de Casa Central.
- Analizar y comprender el funcionamiento del actual sistema de control bibliotecario, logrando entender cómo se manejan los distintos factores para el préstamo de un volúmen. En particular se desea saber, cómo funcionan los sistemas de préstamo de volúmenes, devolución de volúmenes y manejo de usuarios.
- Implementar el protocolo de comunicación con los sistemas bibliotecarios, logrando una interacción de control de lectura de datos, préstamo y devolución mediante esta implementación.
- Analizar el comportamiento de movimiento los volúmenes de consulta en sala, para desarrollar un método de análisis de flujo de libros en lectura en sala.
- Estudiar los tipos de tecnologías RFID, tanto en sus distintos tipos de operación (tags activos, tags pasivos), sus frecuencias de operación (125 - 135 khz, 13.56 Mhz, 900 Mhz, 2.4 Ghz), y sus principales protocolos de operación.
- Estudiar las metodologías de uso de RFID, dónde es interesante o necesario usar la tecnología, cuáles son sus ventajas en general como herramienta por sobre otras alternativas ya implantadas.
- En particular dentro del punto anterior, comprender el funcionamiento de los sistemas de control de la empresa Ceyon Technology, desarrolladora del hardware a utilizar en la implementación de la memoria.
- Implementar el protocolo de comunicación con los sistemas de manejo RFID, logrando recuperar la información almacenada en las etiquetas de Radio Frecuencia.
- Desarrollar una plataforma de comunicación entre los sistemas RFID y los sistemas de biblioteca, que pueda ser extensible a otros desarrollos.
- Coordinar la información que se recolecte con los dispositivos RFID, para obtener información útil del flujo de volúmenes de préstamo y devolución, así como para analizar la circulación de volúmenes de consulta en sala.

1.3. Metodología

La metodología utilizada en el desarrollo del trabajo de título, para lograr los objetivos definidos es:

- **Estado del arte**: Estudio de la situación actual de la tecnología RFID, la solución dada por el proveedor de equipos de RFID, y el ambiente de operación actual de la biblioteca de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile y cómo interactuar con los sistemas centrales bibliotecarios SISIB.
- **Identificación de la solución**: Análisis de los posibles casos de implementación equivalentes existentes para sistemas bibliotecarios y porqué se eligió esta solución.
- **Descripción de la solución**: La forma en que se pueden integrar los sistemas centrales de SISIB con los sistemas provistos por el proveedor tecnológico.
- **Desarrollo de la solución**: Lograr integrar todas las propuestas investigadas respecto a los sistemas, para lograr la mejor forma en que se puede generar el framework para hacer seguimiento de consulta de libros.
- **Implementación de la solución**: La definición de las variables anteriormente desarrolladas, para poder revisar que el desarrollo realmente se ajusta a los parámetros esperados de operación y control de los sistemas bibliotecarios.
- **Evaluación y validación**: Revisión de los resultados obtenidos luego de la implementación para poder analizar cómo han reaccionado los sistemas y cuáles son las etapas que pueden ser escalables en un siguiente desarrollo.

Capítulo 2

Estado del arte

Los sistemas de identificación mediante radio frecuencia, (RFID), existen desde hace varias décadas, principalmente para control de animales, sistemas de transacciones bancarias (como en el caso de SmartCards) y como llaves de control para datos sensitivos [**?**, **?**].

Sin embargo, estos sistemas han ganado momentum, primero durante los 1990s generando impulso para aplicaciones que soportan RFID[**?**] y en los últimos años en los sistemas de cadenas productivos cuando las empresas de Retail Europeas comenzaron a experimentar con tags pasivos incluidos en los productos para consumidores finales, y en Norteamérica tanto el Departamento de Defensa como Walmart comenzaron a usar tags pasivos para el monitoreo de sus envíos.[**?**]

Así, han comenzado a generarse mútiples iniciativas en estandarizar [**?**] los esfuerzos de Radio Frecuencia, debido principalmente a las potencialidades que este tipo de sistemas presuponen [**?**], tanto en monitoreo, rastreo y capacidad tecnológica que muestran este tipo de tecnologías. Además de la ostensible baja en los costos que tanto tags pasivos, lectores y escritores de información invitando al mercado a adoptar esta tecnología. [**?**, **?**, **?**, **?**]

Como describen Phillips, Karygiannis y Huhn [**?**], "No es una tarea fácil diseñar, implementar y optimizar un sistema complejo RFID ", puesto que debemos tener en cuenta la velocidad en que un tag pasa, la cantidad de tags por segundo están pasando por determinado lector, cuál es la potencia a la cual los tags son capaces de responder, si son tags pasivos o activos, requerimientos de seguridad

y privacidad, etc.

La implementación de este tipo de sistemas además deben tener en cuenta el tipo de industria en la cual se está trabajando. Por ejemplo, si necesitamos monitorear equipos en industrias mineras, debemos tener en consideración que es un trabajo el cual tanto los lectores como los tags se someten a condiciones extremas de funcionamiento, con temperaturas de hasta -40° Celcius, con nieve o agua a la cual se pueden ver expuestos los tags y lectores, con lo cual necesitaríamos equipos que cumplan con las normas [**?**] para ello, o como describen Phillips, Karygiannis y Huhn [**?**] en el caso del manejo y control de ganado mediante implantes sub-cutáneos.

Además, se debe ser muy cauteloso al implantar este tipo de tecnologías dentro de una organización, puesto que el personal que va a ser "monitoreado" con esta tecnología puede sentirse amenazada en su confianza debido al permanente control que esta tecnología conlleva, como lo ilustra Phillips, Karygiannis y Huhn [**?**].

Por útimo debemos tener en consideración las variables de integración con las aplicaciones de software existentes, puesto que no puede ser implantado un sistema que no logre una interacción con los sistemas que el personal está acostumbrado, sino que deben dársele, de preferencia, las mismas herramientas, pero con estas nuevas funcionalidades incorporadas como una ventaja y no como una carga, principalmente por la curva de adaptación a este nuevo tipo de control de gestión que esto conlleva.

2.1. Factores para la elección de tecnología en RFID

La tecnología RFID, en estos años de evolución ha logrado especializarse en distintas tecnologías enfocadas a distintos mercados. Gran parte de las variables que influyen en la decisión de uso de una tecnología o arquitectura RFID en particular, depende principalmente de los siguientes factores:

- **Las frecuencias disponibles en los espectros abiertos de frecuencias de operación de los países.**

  En primera instancia, las distintas implementaciones no deben generar distorsiones en las

operaciones de otros equipos o tecnologías. Por ello, los países tienen rangos de operación destinados a uso civil, en que los sistemas RFID usan para operar.

Todas estas bandas civiles de operación deben cumplir con los modos de operación en el país en que se encuentren y sus permisos respectivos.

- **El precio de la implementación.**

  Como fue mencionado, uno de los factores principales para la implementación de esta tecnología en los últimos años es la baja en los precios de los TAGs.

  El costo asociado a la compra de miles (o millones) de TAGs era hasta ahora el tope principal al momento de desarrollar una implementación, en especial si comparámos los costos que supone el hecho de solamente imprimir una etiqueta de códigos de barras.

  Es por esto que una de las principales barreras de obstáculo para decidir acerca de la tecnología a implementar esté dado por el precio de retorno de la inversión.

- **La reusabilidad de los TAGs.**

  En algunos casos es importante saber si podemos reusar o no los TAGs, para estimar los costos de implementación de un sistema RFID.

- **La seguridad de los datos.**

  Este requerimiento proviene de las más recientes necesidades en Radio Frecuencia, puesto que estamos relacionando las implementaciones en Radio frecuencia con algoritmos de encriptación de los datos, punto indispensable en sistemas transaccionales. Por ejemplo, con problemas relacionados con la confiabilidad de sistemas de monedero electrónico, en el cual la capacidad de evitar la clonación de un TAG es relevante.

- **Los volúmenes de manejo o la capacidad de lecturas por segundo.**

  Necesitamos saber si podemos y/o debemos leer múltiples TAGs por segundo, o si tenemos sólo una lectura secuencial de datos.

- **Las condiciones de humedad/temperatura/golpes.**

  En ambientes industriales o mineros de operación, donde la operación de las antenas está dada por zonas de mucho polvo, humedad, altura (en M.S.N.M.[1]), presión, debemos calcular la forma de operar con los distintos equipos, y conseguir equipos apropiados para la operación.[2][**?**]

- **La interacción del TAG mismo con el ambiente.**

  Saber si el TAG debe ser resistente al agua (para lavanderías), si puede ser instalado sobre metales (en equipamiento técnico o hardware), si puede ser leído bajo el agua, o en el caso de animales, subcutáneo.

[1] Metros Sobre el Nivel del Mar

[2] definidos por los estándares de operación IP –International Protection–

- **La distancia a la cual podemos leer el TAG mismo.**

  Este es un factor primordial, dependiendo de la potencia de la antena del lector, de la frecuencia de operación, de la forma y la zona de instalación del TAG. Además, debemos tener en consideración que un lector, si está configurado con demasiada potencia, puede eventualmente, no leer lo que está demasiado cerca, pero terminar leyendo items que están mucho más lejos que lo que se debería.

  Es por esto que uno de los puntos principales dentro de una implementación está dada por la calibración correcta de los equipos [**?**, **?**].

- **El estándar definido para un sector en particular.**

  La principal consideración en la adquisición de cierta tecnología por sobre otra en la implantación de un sistema RFID está dada por los estándares asociados al área en la cual se va a trabajar.

  En el caso de nuestro sistema de bibliotecas, el sistema está basado en el estándar para bibliotecas coordinado por la NISO, organismo Norteamericano para la organización de estándares.

2.2. Esquema general de operación de un sistema RFID

La implantación de un sistema RFID típicamente consta de 3 elementos principales, los TAGs, los sistemas lectores (Fig. **??**) y el sistema de almacenamiento y procesamiento de la información. Toda la capacidad de interacción con la información puede estar alojada en mayor o menor medida en cada una de estas tres áreas.

Así, típicamente uno puede tener una gran cantidad de TAGs esparcidos por toda la zona, los cuales pueden ser sólo códigos de identificación y en el que los lectores van coordinando la organización de ubicación geográfica de los TAGs, para luego ser almacenados en un servidor que almacena toda esta información.

Otro esquema posible es en el que los TAGs mantienen registros de las últimas transacciones realizadas, por lo que la validación de la información está dada por la sincronía de la información desde la base de datos con los datos almacenados en la tarjeta. Así, la antena validadora es usada como un “gatekeeper”.

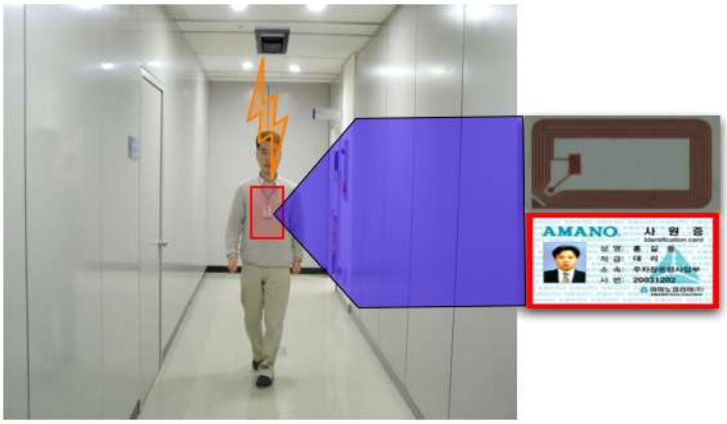

Figura 2.1: Interacción de un sistema lector con un TAG

Este gatekeeper avisa si el TAG que está pasando por esa zona es válido o no.

Esos son sólo algunos ejemplos de cómo se puede realizar la interacción entre los distintos componentes de la cadena de comunicación de datos vía RFID, pero lo importante del caso en un esquema de control mediante RFID es saber identificar cómo queremos interactuar con los elementos a ser identificados, y a partir de ello, especificar dónde necesitamos posicionar la interacción y procesamiento de los datos entregados por los TAGs.

2.3. Arquitectura general de un TAG

Un TAG puede ser diseccionado en 3 áreas principales, la antena, el chip y la protección que posee.

La antena, sirve para dos propósitos básicos, primero, para recibir la radiación que logra que el circuito entero se energice, logrando con esto que el chip sea “activado” y pueda realizar la función de procesamiento (Fig. **??**). Por otra parte, la misma antena es la que, luego de activarse el TAG, transmite la información hacia los lectores.

El CHIP es la zona donde se puede almacenar la información, manejar los datos almacenados y realizar encriptación de los datos, si es necesario. En TAGs más avanzados es posible procesar los datos de control de cada uno de los sensores incluidos en el TAG.

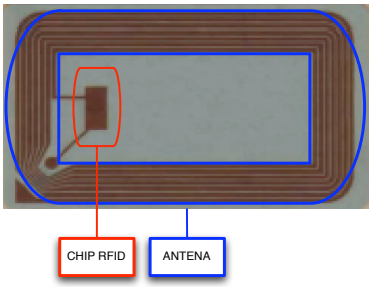

Figura 2.2: Descripción de un TAG RFID

La cobertura, es el revestimiento que se puede imprimir a un TAG. Puede ser una película impresa en una tarjeta identificadora como se ve en la figura **??**, o revestimientos más útiles dentro del proceso productivo, como por ejemplo:

- Revestimientos de protección contra agua.
- Plásticos para poder adosar TAGs en equipamiento metálico.
- Coberturas plásticas para llevar el TAG como un llavero identificador de pacientes en un hospital.

2.4. TAGs activos versus TAGs pasivos

Uno de los puntos de mayor relevancia en la implementación de una solución RFID se encuentra en definir si es necesario contar con TAGs pasivos o TAGs activos y cuál es la principal diferencia entre ellos. Ambos tipos pueden presentar grandes ventajas como desventajas en su operación por lo que es necesario saber sus características más importantes.

Figura 2.3: Ejemplo de cobertura de un TAG RFID

2.4.1. TAGs activos

Corresponden a los TAGs que contienen una fuente de energía propia. Esto puede ser tanto por tener una batería incorporada, o bien estar conectado a alguna fuente de energía(Fig. **??**).

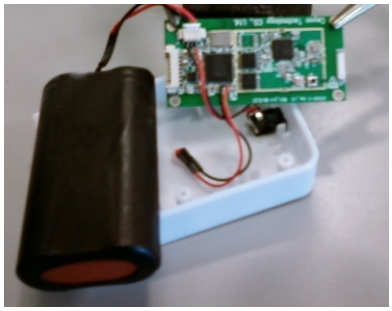

Figura 2.4: TAG activo

La principal ventaja de este mecanismo es su alcance, puesto que al tener energía propia, puede transmitir la información a una distancia superior que la tecnología de tags pasivos, usando además una frecuencia mayor que las tecnologías pasivas (funcionando a 2.4 Ghz). Esto redunda en una mayor distancia de lectura de datos pudiendo colocar las antenas lectoras en distancias mayores a las registradas por frecuencias más bajas.

Otra eventual ventaja que este tipo de TAGs poseen, es que pueden actuar capturando datos o interactuando con el medio en momentos en que no se tiene una antena cerca, por lo que puede servir para registrar una mayor cantidad de datos.

La principal desventaja que poseen, es que los TAGs son muchas veces más caros que los TAGs pasivos, además de tener que cambiar su fuente de energía cada cierta cantidad de tiempo.

2.4.2. TAGs pasivos

Hasta ahora, los TAGs pasivos han sido los principales impulsores de esta tecnología. Esto, principalmente por el bajo costo que hoy por hoy tiene un TAG pasivo.

Esta tecnología, que abarca desde las frecuencias de 125 [Khz] hasta los 900 [Mhz], se basan en que sólo "funcionan" cuando reciben potencia desde un emisor de radiación. Esto hace que la antena capture la suficiente energía para "excitar" al chip, haciendo que el sistema por una parte almacene información en el chip del TAG, y logre transmitir sus códigos almacenados a la antena que está excitando a ese TAG.

Su principal característica visible viene dada por el tipo de antena. Éstos deben su forma principalmente al área que deben abarcar para funcionar en su frecuencia caraterística y a una distancia máxima en particular.

Dentro de las características no visibles está la diferencia en la capacidad de almacenamiento que cada frecuencia puede manejar, dada la cantidad de energía a ser enviada en cada ciclo.

Figura 2.5: TAG 900 [Mhz]

Así por ejemplo, en un TAG EPC Clase 1 de 900 [Mhz] (Fig. **??**), podemos almacenar 96 bits, con rangos de operación de hasta 1 metro, mientras que en un TAG ISO 15693, de 13.56 [Mhz]

(Fig. **??**), podemos tener 128 bytes de uso, con una distancia de 30 cms., cada uno con sus distintas alternativas de seguridad y anti-colisión.

Capítulo 3

Selección de la solución RFID

Los sistemas bibliotecarios en general, ya tienen definido un estándar para el funcionamiento con RFID. Este estándar está coordinado por la NISO, en el grupo de trabajo en aplicaciones bibliotecarias de RFID de ALA [**?**].

Las metas de este grupo de trabajo NISO son:

- Identificar estándares que hagan a las aplicaciones de tecnologías RFID lo más efectivas posibles para su comunidad.
- Identificar estándares relacionados donde el grupo de trabajo pueda ser un aliado.
- Identificar puntos donde RFID pueda ser integrado con otros grupos de trabajo NISO.

Además los tipos de organizaciones que participan en el grupo de trabajo de ALA son:

- Empresas de manufactura de hardware RFID.
- Empresas proveedoras de soluciones RFID (tanto de software como de integración).

- Usuarios de bibliotecas RFID.
- El personal encargado de las bibliotecas.

Así, el punto inicial para el estándar de bibliotecas, el cual está basado esta implementación, es de sistemas RFID operando en la frecuencia de 13.56 MHz, basado en el estándar ISO 15693.

Los puntos principales para adoptar este rango de frecuencia de operación son:

- Medidas de privacidad.
- Soporte de capacidades funcionales.
- Eficiencia y rendimiento.
- Costo total de propiedad.

3.1. Principales proveedores Tecnológicos de RFID para Bibliotecas

3.1.1. 3M

Es el principal impulsor de tecnologías RFID en bibliotecas y el más respetado. Es una empresa multinacional de US$18.000.000.000, que crea, manufactura y vende más de 50.000 productos en más de 200 países a nivel mundial. La empresa ofrece soluciones a empresas, consumidores, mercados de transporte, telecomunicaciones, salud y mercados industriales.

Dentro de sus soluciones en esta área, se cuentan soluciones completas de Hardware y Software [**?**] para sistemas de extensión de aplicaciones bibliotecarias y autopréstamo, desarrollando sistemas de generación de etiquetas RFID, sistemas de detección de robos, accesorios de revisión de circulación, etc (Fig. **??**).

Figura 3.1: Distintas soluciones RFID 3M

En particular, al ser el impulsor del protocolo 3M SIP2, presente en los sistemas Sirsi Unicorn –como el implementado en los sistemas centrales de la biblioteca de la Univedrsidad de Chile–, las soluciones RFID están completamente desarrolladas e integradas para lograr una excelente interacción con estos sistemas bibliotecarios.

3.1.2. Sentry Technology Corporation

Empresa Norteamericana especializada en sistemas de manejo y control de seguridad [**?**] . Esta empresa posee sistemas de monitoreo de cámaras, sistemas de registro y control de accesso y sistemas de manejo bibliotecario.

El principal motivo de análisis de esta empresa en particular, con respecto a esta memoria, es que fue la empresa seleccionada por casa central para el desarrollo e implementación del sistema piloto de biblioteca RFID en la Facultad de Filosofía de esta casa de estudios.

El sistema cuenta con sistemas de control de Inventarios en estanterías, sistemas de autopréstamo con control de RFID, códigos de barras y desmagnetización, todo en uno, basado en el mismo protocolo ISO 15693 de operación.

3.2. El Proveedor seleccionado por la Facultad de Ciencias Físicas y Matemáticas

La empresa elegida para entregar la solución de hardware RFID es la Empresa Coreana Ceyon Technologies Inc. [**?**], empresa especializada en el desarrollo de hardware de RFID y una de las principales impulsoras de esta tecnología en el mercado Coreano.

Dentro de sus principales clientes cuentan Empresas como HP, LG, Samsung, SK Telecom, HYNIX semiconductor y la Agencia de Computarización Nacional en Corea (NCA), entre otros.

El proveedor fue previamente seleccionado por las autoridades de la Facultad de Ciencias Físicas y Matemáticas por recomendación del Prof. Javier Ruiz del Solar, experto en materias de Radio Frecuencia del Departamento de Ingeniería Eléctrica de esta misma casa de estudios.

Los motivos para seleccionar esta empresa en particular, van de la mano con la cooperación de la Facultad con el Gobierno Coreano, dentro del Proyecto ITCC Chile-Korea, marco en el cual, la facultad logró obtener los mejores precios asociados dentro de las implementaciones, además de conseguir equipamiento desarrollado especialmente para sus intereses y necesidades.

Figura 3.2: Equipos RFID desarrollados para la Universidad de Chile

La relación con Ceyon Technologies no está pensada sólo como un proveedor convencional de soluciones tecnológicas, sino que como una asociado en investigación de nuevas tecnologías en este ámbito, punto sustancial dentro de los intereses de I+D de la Facultad.

Dentro de los equipos desarrollados por Ceyon Technologies para esta implementación se cuentan con sistemas de inventarios en estanterías, sistemas para las estaciones de trabajo de los funcionarios de biblioteca, portales para monitoreo de circulación de volúmenes dentro de la biblioteca y kits para el desarrollo de soluciones RFID (Fig. **??**).

Capítulo 4

Descripción de la solución RFID en biblioteca

Los siguientes son los requerimientos que quisiéramos que nuestra solución satisfaciera:

- Interactuar con los sistemas existentes.
- Trabajen con la tecnología existente, como es la tarjeta de biblioteca de los alumnos con código de barras.
- Los códigos de barras de los libros sean fácilmente mapeables a RFID.
- La tecnología entregue nuevas herramientas de control para los volúmenes de la biblioteca.
- Permitan escalabilidad del software a nuevas implementaciones por desarrollar.

4.1. Sistema bibliotecario Universidad de Chile

El sistema de control de biblioteca está implementado por la solución dada por la corporación Sirsi[**?**], compañía dedicada a generar soluciones de automatización completa para sistemas bi-

bliotecarios, incluyendo el manejo de préstamo, devolución, administración de clientes, compra de volúmenes, recuperación de texto, entre otros servicios.

Este sistema está implementado como solución integral desde casa central, siendo una solución que por el lado del control de volúmenes, el control de usuarios, el manejo de la compra de nuevos volúmenes, mantiene una base de datos completa para el manejo de una biblioteca de gran tamaño, como es la de la Universidad de Chile. A la fecha "cuenta con 55 bibliotecas distribuidas entre sus catorce Facultades, 3 Institutos Interdisciplinarios y Servicios Centrales, más de 15.000 suscritos, y su colección impresa que asciende a más de 2.247.000 volúmenes e incluye todo tipo de materiales: libros, revistas, apuntes, tesis, mapas, videos, fotografías, audiovisuales, partituras, manuscritos, entre otros"[**?**].

El sistema Sirsi Unicorn también logra intregrar distintas unidades de información de la misma biblioteca, logrando que por una parte, cada una de las Facultades e Institutos que poseen bibliotecas sean tratadas como unidades individuales, pero la comunicación y control de los volúmenes que fluyen entre las distintas bibliotecas sean tratadas de forma transparente, logrando que los préstamos y devoluciones en distintas bibliotecas sean catalogados en un estado correcto, por lo que uno no tiene problemas en, por ejemplo, pedir remotamente un libro que está en la biblioteca de un campus distinto, o devolver un libro en un campus distinto al que el volumen pertenece originalmente.

El sistema además mantiene una arquitectura flexible y abierta, la cual está desarrollada en arquitectura cliente servidor n-tier, con arquitectura y APIs abiertos para nuevos desarrollos, de características similares a las implementadas por empresas como Oracle o SAP para sus modelos de negocios (Fig. **??**).

De la información disponible en SISIB de Universidad de Chile acerca de Sirsi Unicorn [**?**]: "La arquitectura completamente abierta de Unicorn permite a los desarrolladores de Sirsi incorporar nuevas funcionalidades a los módulos ya existentes o añadir al sistema módulos y clientes completamente nuevos – todo ello es posible sin tener que realizar un rediseño extensivo del servidor Unicorn. Unicorn también proporciona interfaces de programación de la aplicación muy bien definidos para cada módulo. Estas APIs permiten a los usuarios de Unicorn personalizar el software para ajustarse a sus requisitos específicos, así como para facilitar la integración de Unicorn con otros sistemas."

El punto principal para poder lograr la interacción es que el sistema Unicorn está "atado" a estándares.

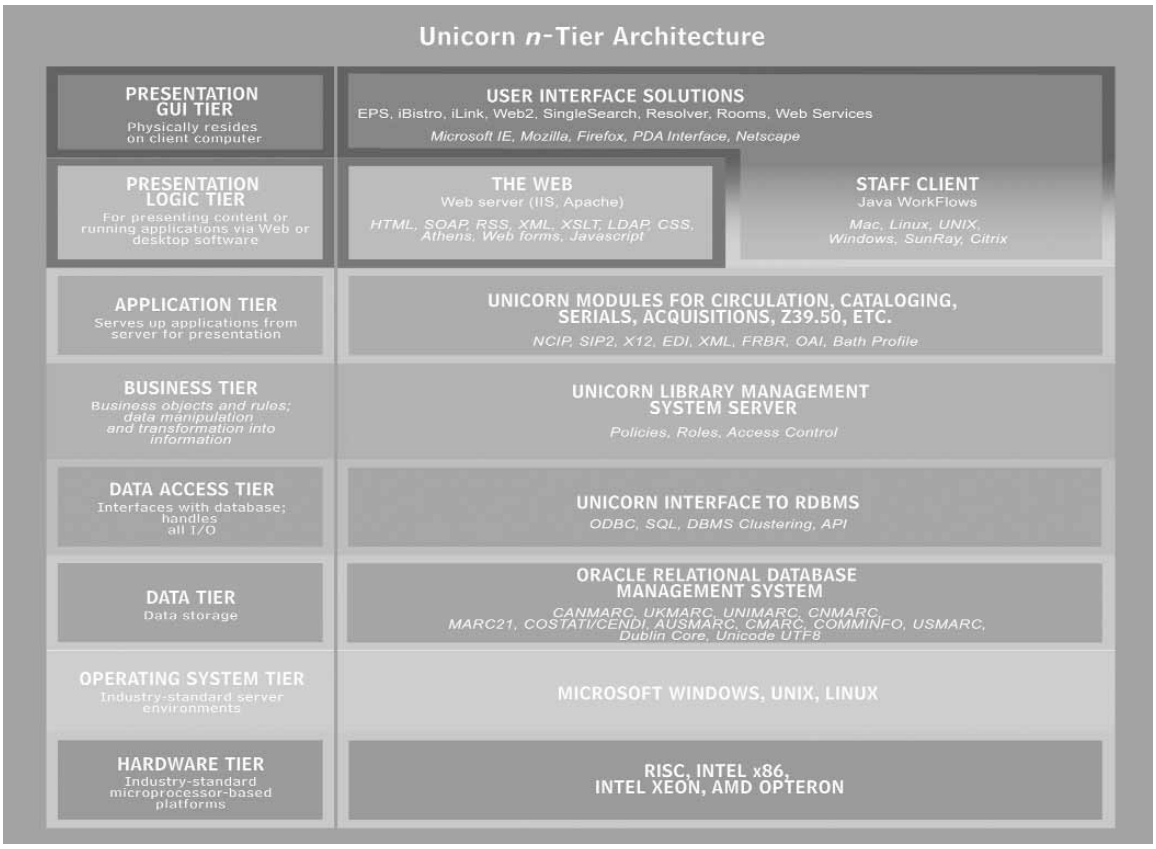

Figura 4.1: Arquitectura cliente/servidor “n-tier” de Unicorn

Todas las capas de interacción cliente/servidor, están desarrollados bajo la idea de la extensibilidad, desde el nivel más bajo, la arquitectura de hardware del sistema, hasta las API’s de interfaz cliente.

4.2. El hardware utilizado

Por el lado del Hardware, se logró que Ceyon Technologies generara equipos que atacaron algunos de estos requerimientos de manera efectiva, como los siguientes:

4.2.1. Equipo escritor de TAGs

Es el equipo que lee un código de barras y automáticamente traspasa ese código de barras al TAG RFID (Fig. **??**).

Figura 4.2: Equipo escritor de TAG RFID

Funciona de modo discreto, es decir, para poder transferir el valor del código de barras al TAG RFID, debemos activar el hardware mediante una secuencia de botones.

El sistema funciona de modo stand-alone, es decir, no necesita estar adosado a ningún otro equipamiento.

4.2.2. Equipo Workstation

Este equipo, al ser conectado a la estación de trabajo del personal de biblioteca, lee fácilmente el TAG del volúmen que está sobre él (Fig. **??**).

Figura 4.3: Accesorio para estación de trabajo

Este equipo funciona de modo discreto, para no interferir con otros posibles volúmenes que se encuentren en la cercanía de la estación de trabajo. Eso sí, puede leer múltiples volúmenes al momento de activarlo.

4.2.3. Portal de circulación

Además, Ceyon Technologies desarrolló un equipo portal que, al pasar por esta zona lee todos los TAGs que circulan por allí (Fig. **??**).

Figura 4.4: Portal de circulación

Este equipo funciona de modo continuo, es decir, en todo momento está leyendo lo que pasa alrededor de zona, transfiriendo todos los datos que este equipo captura a un computador conectado al portal mediante comunicación serial.

4.2.4. Equipo para control de inventario

Ceyon Technologies[**?**] también desarrolló un equipo portátil que puede leer los TAGs RFID en las estanterías, para poder realizar control de inventario (Fig. **??**).

Este equipo logra revisar si los volúmenes se encuentran en la zona correcta del inventario, de acuerdo al valor que está escrito en el lomo del libro. Este equipo funciona con una PDA que se comunica con el hardware, por lo que las aplicaciones que posee para comunicarse este equipo son desarrollos para *Windows Mobile*.

Figura 4.5: Equipo catálogo de estanterías

Para la comunicación con todos estos equipos, Ceyon ha desarrollado y provee de un conjunto de DLL's para *Microsoft Windows XP* capaces de interactuar con el software, una serie de librerías escritas en Microsoft Visual C++ para la interacción con estos, además de entregar los protocolos de comunicación de acceso a estos equipos.

4.3. Primera solución para la implementación

Como primera idea de solución se analizaron los programas cliente de Sirsi Unicorn, para ver cuál podría ser la solución integrando los equipos a los mismos programas de la Empresa Unicorn.

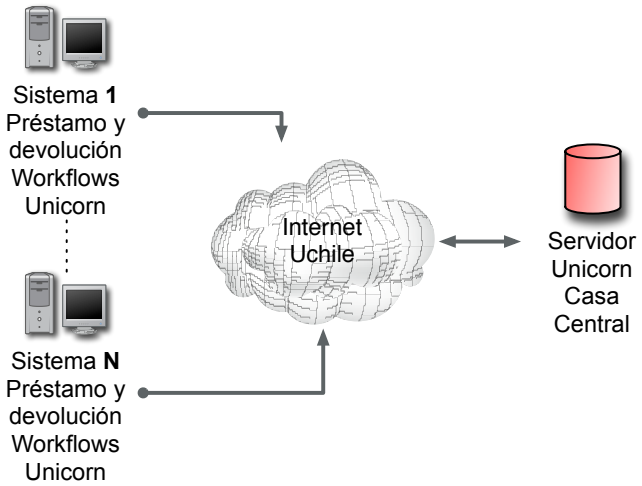

Figura 4.6: Modelo actual de operación

El sistema Sirsi Unicorn, desarrollado en Java, permite agregarle plug-ins para interacción con

módulos RFID conectados a la estación de trabajo (Fig. **??**) pero lamentablemente, para poder desarrollar estos hay que firmar un contrato con la empresa misma y las gestiones desarrolladas por el profesor Ruiz del Solar no prosperaron.

Además, se determinó que el implementar esta solución no ofrece la flexibilidad necesaria para interactuar con todo el resto de los equipos, impidiendo implementar nuevas funcionalidades y limitando la utilidad de los sistemas RFID sólo al funcionamiento con la estación de trabajo del personal de biblioteca.

4.4. Primera Implementación

La primera implementación de software utilizada por biblioteca, fue desarrollada por un grupo de Ingenieros eléctricos, liderados por el Profesor Javier Ruiz del Solar, usando DLL's y librerías[1] entregadas por Ceyon. Este desarrollo permitió generar un botón, el cual activa la comunicación con el equipo conectado a la estación de trabajo y logra escribir lo leído en un campo de ingreso de texto.

Así, el personal de biblioteca tiene que:

- Leer el código de alumno, usando la pistola de código de barras.
- Activar el cuadro de texto dónde incluir el código del libro, haciendo click en el campo de texto (el campo sombreado en la figura **??**) .
- Colocar el libro sobre el equipo lector.
- Presionar el botón "Leer TAG"para leer el código asociado.

Como se muestra en la figura **??**.

[1] para funcionar en Microsoft Windows XP y Visual C++

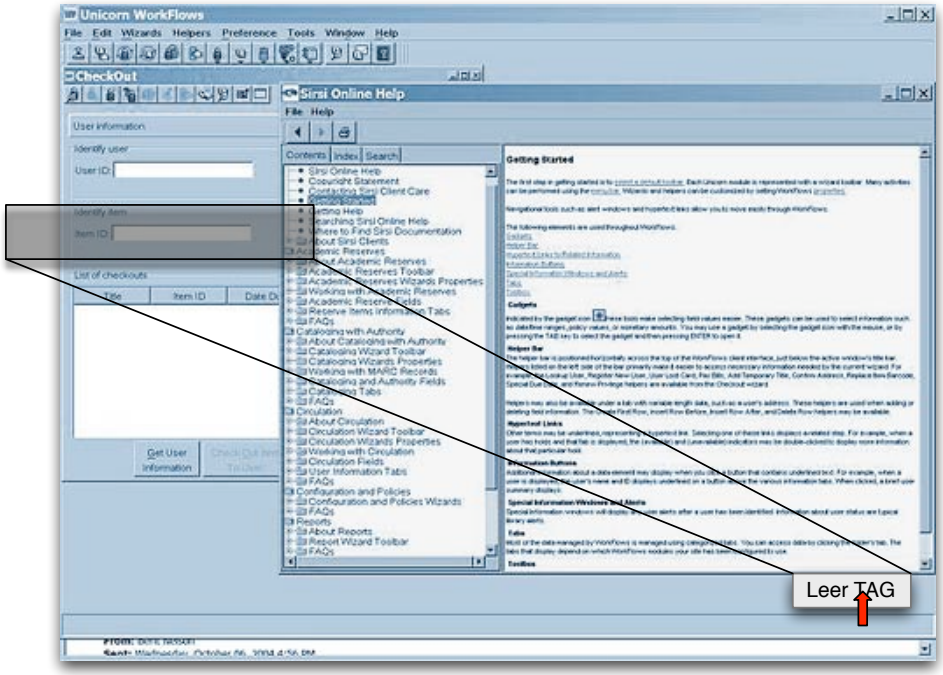

Figura 4.7: Primera implementación

De este modo, se lograba integrar los sistemas RFID instalados en los volúmenes de biblioteca para su uso en el ambiente de trabajo de la Biblioteca.

Este desarrollo fue pensado para obtener un recurso rápido para controlar los equipos de Ceyon, principalmente para evaluar la operabilidad de los equipos mencionados anteriormente.

El primer problema generado al utilizar esta implementación, es que las DLL's de comunicación desarrolladas por Ceyon, al poco tiempo de estar funcionando, ocupaban el 100 % de la memoria de las estaciones de trabajo del personal incluso con las versiones más recientes de las DLL's, debiendo reiniciar los equipos continuamente.

Además, la posibilidad de escalabilidad de esta solución es baja, puesto que sólo inserta información recuperada desde el TAG en un campo de texto, utilizando la funcionalidad básica del sistema RFID, sin entregar valor agregado alguno. Por último, al no tener el respaldo de Sirsi, no es posible generar el plug-in apropiado para comunicarse con la aplicación Unicorn directamente.

4.5. Solución propuesta por esta memoria

En esta memoria se decidió por buscar opciones alternativas para conectarse al sistema Sirsi Unicorn. Esto, porque la opción de certificación Sirsi Unicorn, que incluye la documentación necesaria del desarrollo de plug-ins para las aplicaciones Unicorn, no fue posible de concretar. Por esta razón, el desarrollo de esta memoria comienza todos los desarrollos desde cero.

Bajo este alero, se comenzó a investigar cuáles podrían ser las formas de consultar los datos del sistema Unicorn y se encontró la factibilidad de comunicación mediante el procotolo de comunicación 3M SIP2 [**?**], presente y habilitado en la plataforma Unicorn de Biblioteca.

El sistema de funcionamiento ideal para un sistema bibliotecario, es una plataforma de hardware y software que cuenta con múltiple equipamiento, como:

- Computadores que tengan lectores RFID directamente conectados a sus puertas seriales.
- PDA's con módulos de Radio Frecuencia y red inalámbrica incorporada.
- Equipos de RFID que realicen la transmisión de sus datos y recepción de comandos mediante puertas Ethernet incorporadas.

Todos estos equipos pudiendo tener comunicación de distinto nivel con Unicorn, ya sea con necesidad de conectarse directamente a este, comunicándose a gateways de comunicación, o equipos que actúen como terminales tontos y que sea el middleware el que realice todas las operaciones de proceso de datos RFID como la comunicación con Unicorn.

En el caso de esta memoria, el modelo deseable para la interacción de Unicorn con los equipos de RFID de biblioteca, se muestra en la figura **??**

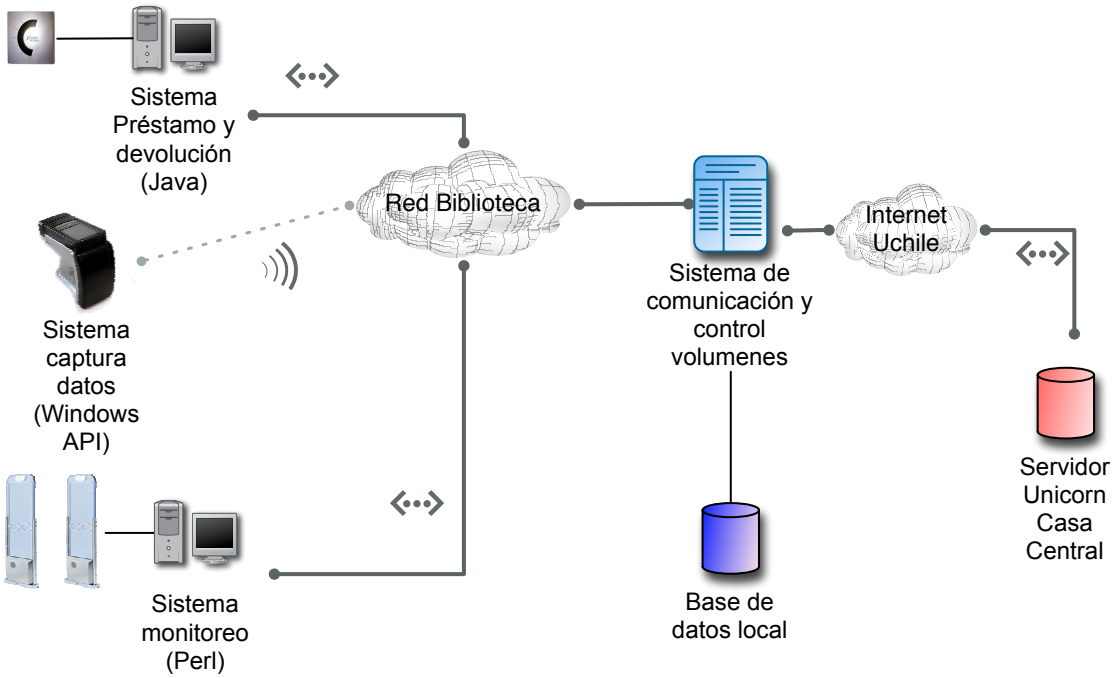

Figura 4.8: Modelo de solución ideal

Al generar un servidor central de comunicación con el servidor Unicorn, que además interactúe con una base de datos local, podemos mantener los registros de eventos que ocurren dentro de la circulación de los distintos volúmenes de la biblioteca.

Este servidor debe ser capaz de comunicarse mediante el protocolo SIP2, tanto al generar comandos hacia Unicorn como para interpretar sus respuestas.

Al crear este servidor, los clientes no necesitan interactuar directamente con Unicorn, sólo les basta leer el TAG RFID, enviarle el requerimiento correspondiente a este módulo, y de acuerdo a las distintas acciones y códigos que reciba, depende la acción que ejecute, ya sea para modificar o leer datos en el sistema de Biblioteca Central, o para almacenar datos específicos en la base de datos local.

En el caso de nuestra implementación, es necesario mantener el sistema de comunicación con Unicorn de forma fluída. Esto porque la comunicación con Unicorn debe ser con respuestas lo suficientemente rápidas como para capturar múltiples ráfagas de consultas por segundo.

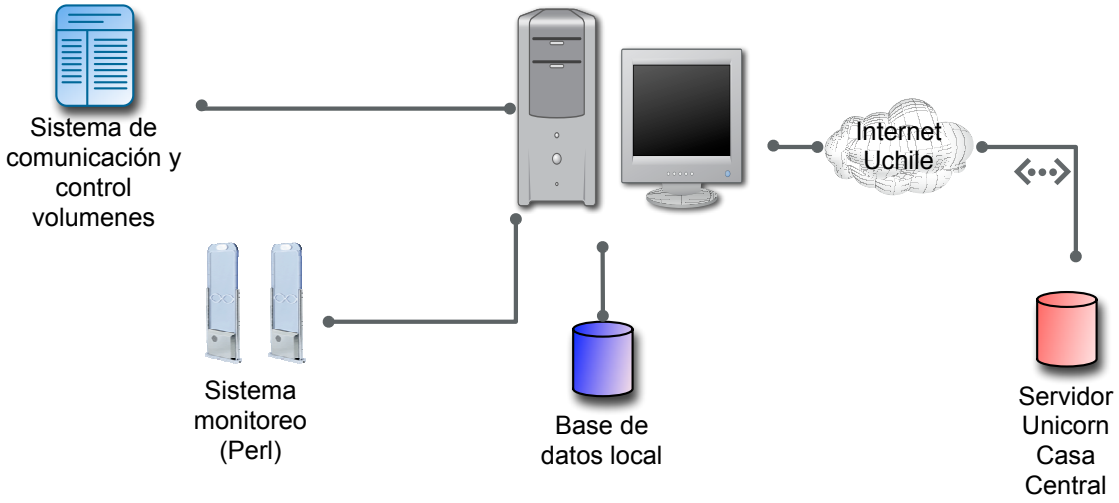

Figura 4.9: Modelo implementación de esta memoria

Para ello, podemos usar el mismo modelo, aunque usando un servidor dedicado sólo para esa aplicación que interactúe de la misma forma con Unicorn (Fig. **??**), siendo este un servidor dedicado que administre todos los recursos del sistema a implementar. Esta implementación entrega una base para desarrollar la solución completa que administra las consultas de múltiples equipos como el expuesto en la figura **??**.

Mediante comunicaciones vía sockets, podemos enviar mensajes al sistema central Sirsi Unicorn^{TM}, de manera de poder rescatar información desde Unicorn.

Para mantener los sistemas abiertos en la comunicación con los equipos RFID sin depender de un sistema operativo, se decidió reimplementar los protocolos de comunicación de los equipos RFID, sin usar las DLLs de *Microsoft Windows* ni las librerías *Microsoft Visual C++*, y pensar en un lenguaje de programación alternativo.

La primera opción fue implementar la solución completa en Java, debido a su portabilidad, multiplicidad de plataformas, vasto conocimiento de uso, etc. Lamentablemente, las opciones para desarrollar en Java, estaban atados a tratar de comunicarse a una puerta serial mediante Java, lo cual obliga a hacer las bibliotecas externas en JNI y volver a quedar atado a *Microsoft Windows*.

La segunda opción era portar las librerías de comunicación serial ya desarrolladas por Ceyon en Win32 hacia Linux, pero sólo lográbamos portar de un sistema a otro y terminando con el mismo problema de estar atado a un sistema operativo en particular.

Por este motivo se analizó la posibilidad de uso del lenguaje Perl, en el cual fue posible encontrar información básica para el desarrollo de la comunicación con SIP2. Además, la comunicación a la puerta serial desde este lenguaje de programación varía muy poco entre los distintos sistemas operativos, por lo que el funcionamiento entre distintas plataformas de operación podría darse sin mayores problemas.

Por último, para el sistema de despliegue de los datos así como con la base de datos local, también se pensó en alternativas que fueran más allá que *Microsoft Windows, Internet Information Server* y *SQL Server* o *LAMP* [2], sino que buscar una alternativa que pudiese ser mas independiente del sistema operativo, como lo es *Ruby on Rails con MySql*, el cual puede ser ejecutado transparentemente en la mayoría de los Sistemas Operativos de la actualidad.

[2] Linux, Apache, Mysql y PHP

Capítulo 5

Protocolos de Comunicación

Para que en este trabajo de memoria se pudiera realizar la implementación de forma completa, es necesario entender los protocolos de comunicación con los sistemas de la Universidad de Chile, así como ser capaz de interactuar con los sistemas RFID de Ceyon Technologies.

En esta sección se detalla la forma en que se integran los protocolos de comunicación hacia ambas fuentes de datos, cómo fue posible disectar la información útil, y cómo se manejan los mensajes de error de estos.

5.1. 3M SIP 2

El desarrollo se ha centrado en uno de los protocolos de comunicación con el sistema Unicorn de Sirsi, que es el Protocolo 3M Standard Interchange Protocol version 2 (SIP2)[**?**], – no confundir con el protocolo IETF de Voz sobre IP SIP 2 (Session Initiation Protocol), que es el acronismo más conocido hoy –.

Este protocolo de comunicación permite una excelente interacción con el sistema Unicorn, con

todos los protocolos de comunicación abiertos a la comunidad, con los cuales podemos generar todos los comandos para préstamo y devolución, integrando los desarrollos de RFID para trabajar con el sistema Unicorn de forma transparente.

5.1.1. Descripción del protocolo SIP 2

Para poder entender el protocolo de comunicación SIP2, debemos tener una idea de cómo conectarse mediante sockets. En el caso particular de SIP2 el protocolo es muy similar al implementado en una comunicación serial estándar, con un header de control, un login de informacion, el dato a consultar, y un set de bytes de chequeo de errores. La respuesta a las consultas enviadas también vienen dadas por un comando de la misma naturaleza, con un encabezado correspondiente a la respuesta, la información de respuesta y los bytes de checksum para corroborar que la información llegó completa.Todo esto transmitido a un puerto definido en el servidor remoto Unicorn.

A modo de ejemplo podemos dar un escenario típico de comunicación entre el cliente y el servidor:

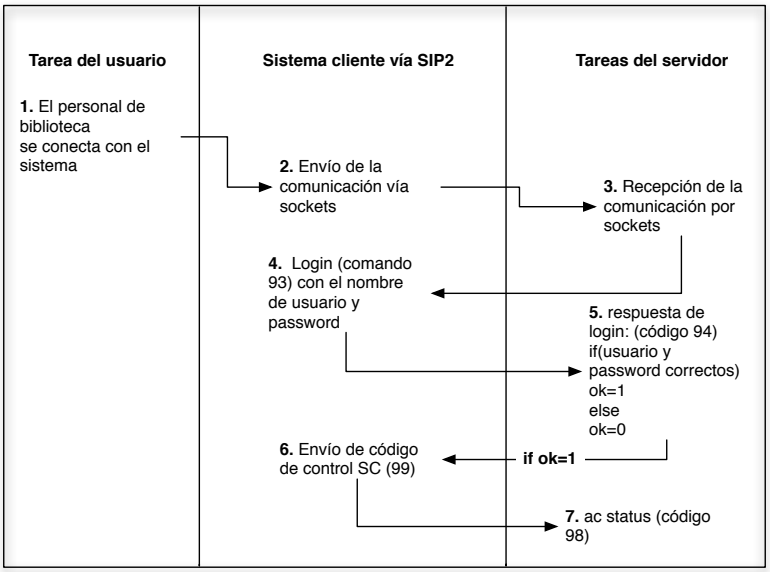

Figura 5.1: Comienzo de la comunicación mediante sockets

Acá se describen los pasos en los cuales se genera la comunicación inicial con Unicorn, describiendo cada uno de los pasos involucrados para la conexión.

Para entender cómo se produce la comunicación podemos detallar el comando enviado en el paso 4 de la figura **??**:

9300CN<LoginUserID>|CO<LoginPassword>|CP<LocationCode>|AY5AZEC78<CR>

La descripción del ejemplo nos entrega los comandos a enviar hacia Unicorn:

93	:	Login(Valor = 00)
CN	:	Identificación del usuario
CO	:	Contraseña del usuario del sistema
CP	:	Ubicación de la estación de trabajo (ID de la biblioteca)
AY	:	Secuencia (valor = 5)
AZ	:	Checksum (EC78) (fig.**??**)
		(Carriage Return)

Y la respuesta del servidor luego de haber recibido este comando, paso 5 de la figura **??** :

941AY3AZFDFA<CR>

Descripción de la respuesta al paso 5 de la figura **??**:

94	:	Respuesta de Login (valor = 1; O.K.)
AY	:	Código de secuencia (valor = 3)
AZ	:	Checksum (FDFA) (fig.**??**)
		(Carriage Return)

Para calcular el checksum, 3M provee una implementación de ejemplo con la cual podemos tanto generar el código correcto para la generación del patrón a enviar al servidor, como para verificar que el checksum que viene en los últimos caracteres de la respuesta corresponde a la información recibida.

```
/*********************************************************/
/* compute_checksum   Computes a Standard Protocol checksum
 * for a string passed as a parameter. The string is assumed
 * to include all characters of the message up to and
 * including the checksum's field ID characters.
 * Returns a pointer to an ASCII representation
 * of the checksum.
 *
 *    message
 *     NULL-terminated message on which to compute checksum
 */
/*********************************************************/
char * compute_checksum (char * message)
{
static char ascii_checksum[5];
unsigned int checksum = 0;
int i = 0;
while (message[i] != '\0')
{
checksum += (unsigned) message[i];
i++;
}
checksum = -(checksum & 0xFFFF);
sprintf (ascii_checksum, "%4.4X", checksum);
return (ascii_checksum);
}
```

Figura 5.2: Código de generación de Checksum

Con esto se logra generar el patrón completo para ser implementado en el lenguaje con que queremos desarrollar esta memoria.

5.1.2. Códigos de autorización de Usuarios y volúmenes

Para hacer una transacción – ya sea préstamo o devolución –, debemos conocer la estructura de identificación tanto del volúmen el cual deseamos hacer la consulta, como la estructura del número identificador del suscriptor al cual deseamos realizar el préstamo o la devolución del libro.

El libro cuenta con un identificador único de volumen consistente en un número de 14 dígitos, que en la estructura actual está codificado en un código de barras en la parte inferior de la tapa trasera del volúmen (Fig. **??**).

Figura 5.3: Ejemplo de código de barras presente en los libros

Éste código nos relaciona en la base de datos Unicorn con los datos del volúmen en cuestión, ya sea para actualizar ésos datos, como para recuperar esta información, por ejemplo:

- El nombre del volúmen.
- La biblioteca donde se ubica.
- El tipo de colección en la que se encuentra.
- El tipo de material.
- La ubicación dentro de las estanterías, mediante el código de localización bibliotecaria.
- El estado en que se encuentra el libro (Prestado, disponible, en reparación, en tránsito, etc.).

Éstos datos son recopilados desde Unicorn usando la estructura recientemente explicada. Así, por ejemplo, si se desea recuperar la Información de un volúmen en particular, debemos usar la estructura siguiente:

1720070704 190029AO|AB35601006031219|AC|AY1AZF519

Donde:

17	:	Código que indica revisón de datos de un Volúmen
20070704 190029	:	Fecha y hora de la consulta
AO	:	<ID Institución>(Valor = *null*)
AB	:	Transporta el código del libro (Valor = 35601006031219)
AC	:	Password (Valor = *null*)
AY	:	Código de secuencia (Valor=1)
AZ	:	Checksum (F519) (fig. **??**)
		(Carriage Return)

Lo que debe responder el servidor, en caso que se esté conectado y se le entregue un código de libro válido es:

1810000120070704 184757AB35601006031219|AJExecutive report : Prepared for: The Intitute[1] of Internal Auditors|BG<BIBLIO ID>|CK000|AQC_GENERAL| APINTRANSIT|CRBOOK|CS657.450285 Ex31 1977|CTCF20|AY1AZC832

La respuesta del servidor está dada por:

18	:	Código que indica respuesta a revisón de datos de un Volúmen
100001	:	Características del volumen en cuestión
20070704 184757	:	Fecha y hora de la respuesta
AB	:	Código del Volúmen consultado
AJ	:	Nombre del Volúmen consultado
BG	:	Nombre de la biblioteca que consulta
CK	:	Tipo de Volúmen (Libro, video, revista, etc.)
AQ	:	Ubicación permanente
AP	:	Estado del Volumen (valor= *Intransit*)
CR	:	Tipo de impreso (valor = BOOK)
CS	:	Información del Lomo del libro (valor = *657.450285 Ex31 1977*)
CT	:	Nombre de biblioteca que lo posee
AY	:	Código de secuencia (Valor=1)
AZ	:	Checksum (C832) (fig. **??**)
		(Carriage Return)

Para el caso del usuario, también poseemos un identificador único de 14 dígitos, generado por Unicorn para mantener la compatibilidad de los códigos generados por el sistema mismo.

En el caso de revisar el estatus de un usuario, debemos mandar el código para revisión del usuario, el cual como respuesta nos debería dar datos como:

- El nombre del usuario.
- Su direccion
- Teléfono
- Status (bloqueado, permitido para pedir libros, suspendido)

Entre otros datos.

Así, por ejemplo

6300120070704 192404 AO|AA05601001064369|AY1AZF445

Donde:

63	:	Código que indica revisión de datos de un usuario
001	:	Tipo de usuario
20070704 192404	:	Fecha y hora de la consulta
AO	:	<ID Institución>(Valor = *null*)
AA	:	Transporta el código del usuario (Valor = 05601001064369)
AY	:	Código de secuencia (Valor=1)
AZ	:	Checksum (F445) (fig. **??**)
		(Carriage Return)

La respuesta del servidor de este código de consulta SIP2 (código 64) devuelve la siguiente información:

```
64YYYY 00020070704 191126000000010001000000000000AO|AA05601001064369
|AEKIM, JONG HYUN|BZ0003|CA0001|CB0010|BLY|BV|BDBOMBERO NU?EZ 356
|BF7320950|PA20071231 235900|PD19830101|PCALUM_CF20|PEESTUDIANTE
|PFMASCULINO|AFUsuario BLOQUEADO|AY1AZC0A4
```

De este código es posible rescatar la información necesaria para el control del usuario.

Con todos estos datos, más los datos básicos de comunicación de usuario, password y ubicación, que por razones de seguridad no incluiremos aquí, podemos generar las consultas necesarias para el manejo básico para préstamo y devolución de libros.

5.2. Procotolo de Comunicación con los lectores RFID

Todos los equipos RFID adquiridos por la Biblioteca de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile se comunican por el puerto serial, incluyendo el portal de lectura de circulación de volúmenes, por lo que esta memoria se basa en la comunicación con los equipos por este medio.

5.2.1. Descripción del protocolo de comunicación

El protocolo de comunicación es un protocolo de transmisión para conexiones punto a punto entre el lector CEYON y el PC. Este es un protocolo asíncrono con un tamaño de datos de 8 bits. El radio de baudios es variable, pero viene configurado por omisión a 38400 bps. Los caracteres transferidos incluyen 1 bit de partida sin paridad y un bit de parada (38400, N/8/1). Para poder comenzar la comunicación, tanto el receptor como el transmisor deben estar listos [**?**].

Definición de formato y caracteres

Caracter	**Valor**	**Descripción**
STX	0x02	Comienzo de texto
ETX	0x03	Fin de Texto
DLE	0x10	Data Link Escape

Cuadro 5.1: Caracteres básicos de la comunicación

Al final de la transmisión, el sistema que envía el mensaje transmite DLE, ETX como se describen en el Cuadro **??**. DLE es necesario para distinguir entre un carácter de control y el primer byte de información. Si es transmitido un 0x10 dentro de un bloque de información este es transmitido dos veces, para distinguirlo del carácter de control DLE. Además este DLE Agregado no es incluído en el cálculo del CRC (Fig. **??**).

Nombre	PAQUETE [1:n]				
Orden	1	2	3 . . . i-2	i-1	i
Significado	Inicio		Mensaje [1:j]	Final	
Bytes	DLE	STX		DLE	ETX
	0x10	0x02		0x10	0x03

Cuadro 5.2: Estructura de un paquete enviado por un Lector Ceyon

En el cuadro **??** se puede ver la estructura de un paquete enviado por un lector Ceyon al reaccionar a un mensaje recibido. El paquete tiene esta misma estructura si el servidor le envía un mensaje, o si un TAG ha pasado por la zona de alcance de la antena de este. Podemos ver que el paquete está formado por los controles de entrada (DLE+STX), el mensaje enviado y el fin de paquete (DLE+EXT).

Formato de información por campo

Nombre	Mensaje [1:j]		
Orden	Información[1:k]	j-1	j
Significado		CRC16	

Cuadro 5.3: Detalle del mensaje incluido

Dentro del mensaje que viene en el paquete, podemos desprender 2 zonas, la Información que viene en el paquete y su CRC de verificación del dato recibido.

Nombre	Información [1:k]				
Orden	1	2	3	4	5 . . . k
Significado	tipo	id	cmd	largo	data[1:k-4]

Cuadro 5.4: Información incluida con el mensaje

El CRC16 debe ser creado con la información [1 . . . k] del cuadro **??**, usando el algoritmo que se detalla en la función expresada en la figura **??**. El detalle del Cuadro **??** que posee el paquete está definido como:

- **tipo:** Entrega el tipo de información a ser enviado, el cual la información viene dada por 1 byte.
- **id:** Entrega el id del Lector CEYON, usando 1 byte para ello
- **cmd:** Comando a ser enviado al lector, 1 byte.
- **largo:** Largo del dato enviado en 1 byte
- **data:** Bytes de datos a recibir/transmitir (parámetros de comando), el largo en bytes depende del valor entregado en el byte anterior.

El intercambio de información entre el Equipo control y el lector Ceyón está dada por el control entre el comando enviado desde el host al lector y su respectiva respuesta, como en la figura (Fig.**??**).

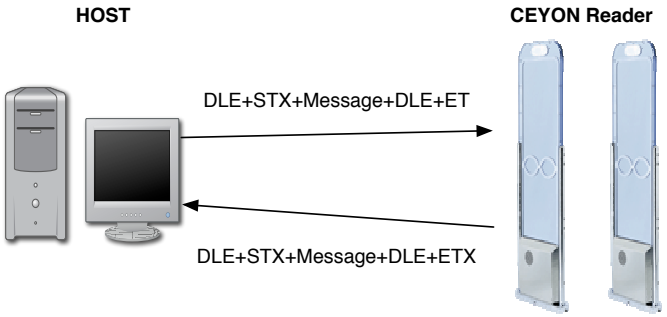

Figura 5.4: Procedimiento de comunicación Host - Reader

Comunicación hacia el lector

Para graficar de mejor forma la comunicación entre el lector y el host, es analizando un comando enviado desde el Host hacia el lector CEYON y la respuesta de parte del lector CEYON.

10 02 01 01 01 02 0B 80 7B DB 10 03

Con la información entregada en la sección de descripción del protocolo de comunicación RFID, se puede entender el mensaje enviado al lector como en el Cuadro **??**.

10	DLE	Data Link Escape	0x10	(10)
02	STX	Comienzo de texto	002x02	(02)
01	Type	Aviso de mensaje desde Host a Reader	0x01	(01)
01	ID	ID del Lector CEYON	0x01	(01)
01	CMD	Set Configuration	0x01	(01)
02	Largo	2 bytes	0x02	(02)
0B 80	Data	Envio del comando	0x0B 0x80	(11 140)
7B DB	CRC	CRC para el mensaje en particular	0x7B 0xDB	(12 219)
10	DLE	Data Link Escape	0x10	(10)
03	ET	Fin de texto	0x03	(03)

Cuadro 5.5: Descripción del mensaje enviado desde el Host al Lector Ceyon

En este caso en particular, se está informando al lector que realice la operación "Set Configuration", con la configuración de la dirección de memoria $0x0B$ y como valor para este registro el $0x80$, que significa "Activar sonido" (Enable buzzer).

La respuesta de Lector Ceyon al Host está dada por:

10 02 10 10 01 01 01 00 76 EF 10 03

Así, haciendo el mismo análisis para la respuesta entregada por el Lector al Host en el cuadro **??**

10	DLE	Data Link Escape	0x10	(10)
02	STX	Comienzo de texto	0x02	(02)
10	Type	Aviso de mensaje desde Reader a Host	0x10	(10)
10	ID	ID del Lector CEYON a enviar el mensaje	0x01	(10)
01	CMD	Set Configuration	0x01	(01)
01	Largo	1 byte	0x01	(01)
01 00	Data	Envio de la respuesta del comando	0x01 0x00	(01 00)
76 EF	CRC	CRC para el mensaje en particular	0x76 0xEF	(118 239)
10	DLE	Data Link Escape	0x10	(10)
03	ET	Fin de texto	0x03	(03)

Cuadro 5.6: Descripción del mensaje enviado desde el Lector Ceyon al Host

La respuesta del lector

El lector Ceyon responde bajo el mismo protocolo, por lo que es posible saber si es el lector por el tercer byte del paquete, que vendrá con un $0x10$ (a diferencia del mensaje desde el host que es $0x01$), y en el cuarto byte de paquete enviado por el lector, que entrega el valor identificador con el cual tenemos configurado el lector en sí mismo, en caso de mantener una cantidad considerable de lecotres conectados al mismo host.

Luego se obtiene el comando al que el lector está obedeciendo, o qué tipo de respuesta está

dando, – en caso que el lector esté en modo de lectura continua– y por último el largo del mensaje, el mensaje mismo y el CRC-16, de la misma manera explicada en la sección anterior.

Generación del CRC-16

Para generar correctamente el CRC de control de comunicación entre el host y el Lector, Ceyon entrega las funciones [**?**] (en C++), necesarias para la creación de este.Para ello, primero Ceyon nos provee con una tabla para generacion del CRC-16 mediante la tabla CCITT (Fig. **??**).

Este arreglo de 256 elementos, cada uno de ellos formado por 2 bytes, entregados por la función de la figura **??**, es usado por la función de creación del CRC de 2-bytes en el xor respectivo con el valor a ser creado en la función de la figura **??**

Validación de un CRC-16 recibido

Para revisar la validez de la información enviada en un cierto paquete de datos desde un lector Ceyon, es posible utilizar otra función entregada por los Protocolos de Acceso de Ceyon, disponible en la figura **??**.

Con este algoritmo de la figura **??** podemos comprobar si el mensaje recibido viene completo, al poder corroborar la validez de los datos recibidos de desde el lector Ceyon.

```
unsigned short CCITT_CRC16Table[256]={
0x0000,0x1021,0x2042,0x3063,0x4084,0x50A5,0x60C6,0x70E7,
0x8108,0x9129,0xA14A,0xB16B,0xC18C,0xD1AD,0xE1CE,0xF1EF,
0x1231,0x0210,0x3273,0x2252,0x52B5,0x4294,0x72F7,0x62D6,
0x9339,0x8318,0xB37B,0xA35A,0xD3BD,0xC39C,0xF3FF,0xE3DE,
0x2462,0x3443,0x0420,0x1401,0x64E6,0x74C7,0x44A4,0x5485,
0xA56A,0xB54B,0x8528,0x9509,0xE5EE,0xF5CF,0xC5AC,0xD58D,
0x3653,0x2672,0x1611,0x0630,0x76D7,0x66F6,0x5695,0x46B4,
0xB75B,0xA77A,0x9719,0x8738,0xF7DF,0xE7FE,0xD79D,0xC7BC,
0x48C4,0x58E5,0x6886,0x78A7,0x0840,0x1861,0x2802,0x3823,
0xC9CC,0xD9ED,0xE98E,0xF9AF,0x8948,0x9969,0xA90A,0xB92B,
0x5AF5,0x4AD4,0x7AB7,0x6A96,0x1A71,0x0A50,0x3A33,0x2A12,
0xDBFD,0xCBDC,0xFBBF,0xEB9E,0x9B79,0x8B58,0xBB3B,0xAB1A,
0x6CA6,0x7C87,0x4CE4,0x5CC5,0x2C22,0x3C03,0x0C60,0x1C41,
0xEDAE,0xFD8F,0xCDEC,0xDDCD,0xAD2A,0xBD0B,0x8D68,0x9D49,
0x7E97,0x6EB6,0x5ED5,0x4EF4,0x3E13,0x2E32,0x1E51,0x0E70,
0xFF9F,0xEFBE,0xDFDD,0xCFFC,0xBF1B,0xAF3A,0x9F59,0x8F78,
0x9188,0x81A9,0xB1CA,0xA1EB,0xD10C,0xC12D,0xF14E,0xE16F,
0x1080,0x00A1,0x30C2,0x20E3,0x5004,0x4025,0x7046,0x6067,
0x83B9,0x9398,0xA3FB,0xB3DA,0xC33D,0xD31C,0xE37F,0xF35E,
0x02B1,0x1290,0x22F3,0x32D2,0x4235,0x5214,0x6277,0x7256,
0xB5EA,0xA5CB,0x95A8,0x8589,0xF56E,0xE54F,0xD52C,0xC50D,
0x34E2,0x24C3,0x14A0,0x0481,0x7466,0x6447,0x5424,0x4405,
0xA7DB,0xB7FA,0x8799,0x97B8,0xE75F,0xF77E,0xC71D,0xD73C,
0x26D3,0x36F2,0x0691,0x16B0,0x6657,0x7676,0x4615,0x5634,
0xD94C,0xC96D,0xF90E,0xE92F,0x99C8,0x89E9,0xB98A,0xA9AB,
0x5844,0x4865,0x7806,0x6827,0x18C0,0x08E1,0x3882,0x28A3,
0xCB7D,0xDB5C,0xEB3F,0xFB1E,0x8BF9,0x9BD8,0xABBB,0xBB9A,
0x4A75,0x5A54,0x6A37,0x7A16,0x0AF1,0x1AD0,0x2AB3,0x3A92,
0xFD2E,0xED0F,0xDD6C,0xCD4D,0xBDAA,0xAD8B,0x9DE8,0x8DC9,
0x7C26,0x6C07,0x5C64,0x4C45,0x3CA2,0x2C83,0x1CE0,0x0CC1,
0xEF1F,0xFF3E,0xCF5D,0xDF7C,0xAF9B,0xBFBA,0x8FD9,0x9FF8,
0x6E17,0x7E36,0x4E55,0x5E74,0x2E93,0x3EB2,0x0ED1,0x1EF0
   };
```

Figura 5.5: Tabla CCIT

```
/*-----------------------------------------------------*/
/*  Inputed String make to 2-Byte CRC                  */
/*  return value : None                                */
/*-----------------------------------------------------*/
void CCIT_Make_CRC(unsigned char *comm_buf,
unsigned int comm_length)
{
    unsigned  short  com_val, accum, i;
    accum = 0x00;
    for  (i=0; i<(unsigned  short)comm_length; i++)
    {
       com_val = (accum >> 8) ^
                            (unsigned char)(comm_buf[i]);
       accum = (accum << 8) ^
              CCITT_CRC16Table[com_val];
    }
    comm_buf[comm_length]=
          (unsigned char)((accum>>8)&0x00ff);
    comm_buf[comm_length+1]=
          (unsigned char)(accum & 0x00ff);
}
```

Figura 5.6: Función para generar el CRC de 2 bytes.

```
/* Check Strings crc, and return result */
/* (true:crc ok, false:crc error) */
bool CCIT_Check_CRC(unsigned char *comm_buf,
                    unsigned int comm_length )
{
  unsigned  short com_val, accum, i;
  accum = 0x00;

  for  (i=0; i<(unsigned  short)comm_length; i++)
  {
    com_val = (accum >> 8) ^ (unsigned char)(comm_buf[i]);
    accum = (accum << 8) ^ CCITT_CRC16Table[com_val];
   }

  com_val = (accum >> 8) & 0x00ff;
  accum &= 0x00ff;

  if ((unsigned char)com_val == comm_buf[comm_length])
  {
    if ((unsigned char)accum == comm_buf[comm_length+1])
    {
      return (true);
    }
    else
    {
      return (false);
    }
  }
  else
  {
       return (false);
   }
}
```

Figura 5.7: Chequeo de CRC con un string dado

Capítulo 6

Implementación

Para la implementación de esta memoria, al comenzar el desarrollo de plataforma desde cero, se debió tener en cuenta un desarrollo de arquitectura completa, es decir, pensar en cómo puede evolucionar el hardware, el software y las aplicaciones a ser desarrolladas como extensión a esta arquitectura de monitoreo y control. Y como fue mencionado anteriormente, sin tener que depender necesariamente de la arquitectura de hardware en la que opera el servidor implementado o del sistema operativo a utilizar.

6.1. Arquitectura de Hardware usada

En el desarrollo de esta memoria, se utilizó un lector RFID continuo de Ceyon Technologies, un computador *Apple Powerbook 12'* con la siguientes características (fig **??**):

- Procesador IBM PowerPC de 1,5 GHz.
- 512 MB en Memoria RAM.
- Adaptador genérico USB a Serial.

Figura 6.1: Arquitectura de hardware utilizada

La arquitectura final en uso del sistema es el portal Ceyon de Circulación, con un computador Intel genérico conectado mediante la puerta serial.

6.2. Arquitectura de Software usada

Para el desarrollo del sistema, se utilizaron las siguientes características de software:

- Apple Mac OS X 10.4
- fink 0.8.6
- Perl 5.8.6, con los siguientes módulos extras:
    - Device::SerialPort,
    - IO:Socket
    - Net::Telnet

    - DBI::Mysql
    - Time::Timestamp
    - DateTime::Timezone
- Mysql 5.0.24a
- Ruby 1.8.6
- On Rails 1.2
- minicom 2.1

Si bien *Mac OS X* es un sistema propietario, todas las características de software listadas pueden ser encontradas para prácticamente cualquier sistema operativo moderno, con mínimas diferencias. En linux por ejemplo, es posible encontrar exactamente el mismo software[1] por lo que al instalar Perl, Mysql, Ruby y Ruby on Rails en las versiones correspondientes, este sistema funciona sin dificultades.

En *Microsoft Windows* sólo debe hacerse un pequeño cambio, y es que en la librería de Perl para conexión con la puerta serial éste posee pequeñas diferencias. Así, en vez de instalar Device::SerialPort, es necesario reemplazarlo por Win32::SerialPort[2] haciendo el cambio correspondiente en el servidor de control del lector RFID.

6.3. Desarrollo del sistema

Se generaron 3 áreas principales del desarrollo, el módulo de captura de datos, la base de datos local y la interfaz de revisión de pasos por el pórtico.

Todos estos procesos son indispensables en la implementación del sistema completo, registrando correctamente los pasos por el pórtico, almacenando la información sin duplicidades, para luego poder desplegar esa información para el uso de los datos por el personal encargado.

[1] a excepción de fink, al ser este el motor de administración para las aplicaciones open source en *Mac OS X*

[2] Device::SerialPort es una adaptación en *nix de Win32::SerialPort, por lo que funcionan de forma similar

6.3.1. Servidor de Captura de datos

El servidor de captura de datos es un proceso *deamon* que cumple con las siguientes características:

- Captura los datos desde los lectores
- Se conecta con la base de datos para control de equipo
- Se conecta con Unicorn, en caso de ser necesario
- Coordina los pasos por el pórtico para que no existan pasos duplicados.
- Registra los pasos verdaderos a través del pórtico.

El servidor está implementado en *Perl*, usando varias fuentes de librerías estándar CPAN. Este servidor es el encargado de controlar tanto la validez de la información capturada desde un TAG, revisando si se encuentra la información contenida en la base de Datos que corresponda y almacena los datos en la base de datos local.

Para explicar mejor este módulo se puede ver el diagrama de procesos de la figura **??**

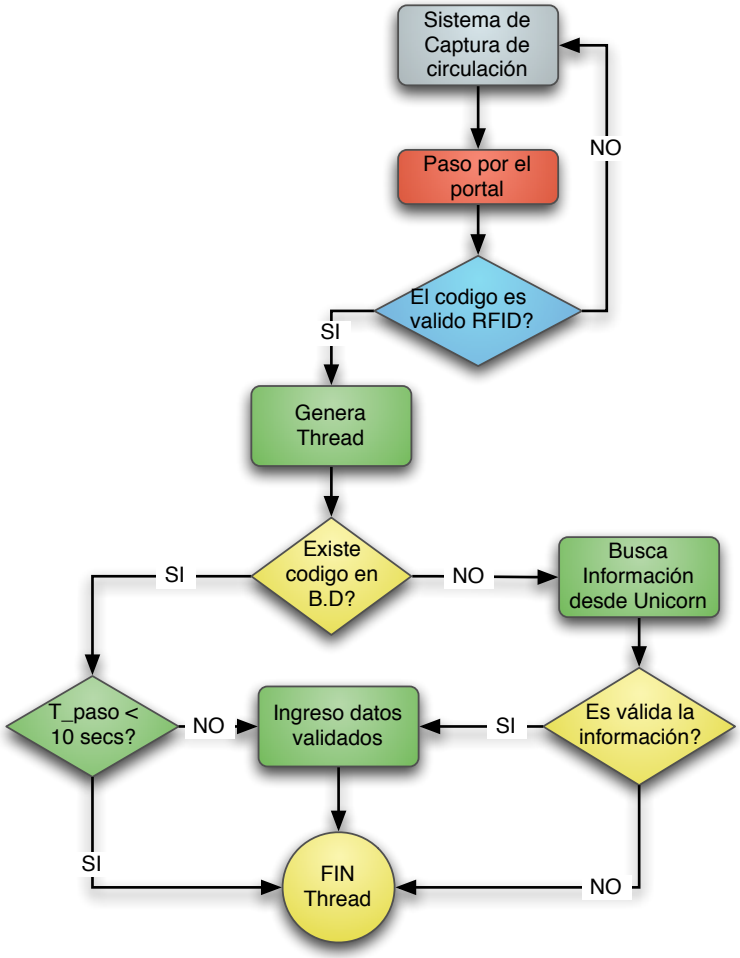

Figura 6.2: Diagrama de procesos del Módulo de Captura de datos

Captura de la información de los TAGs

Lo primero que se debe registrar al momento que un TAG pase por el portal de circulación es poder determinar si se genera una respuesta válida. Para ello, se debe ver si la respuesta que entrega el portal es una secuencia que sea coherente con los patrones que entrega un lector Ceyon, explicado en el Capítulo 5.

Así se tiene un *daemon* (fig. **??**) que está corrriendo permanentemente, a la espera que un volúmen con identificación RFID pase por el portal. Al momento que el TAG pasa por el portal, se

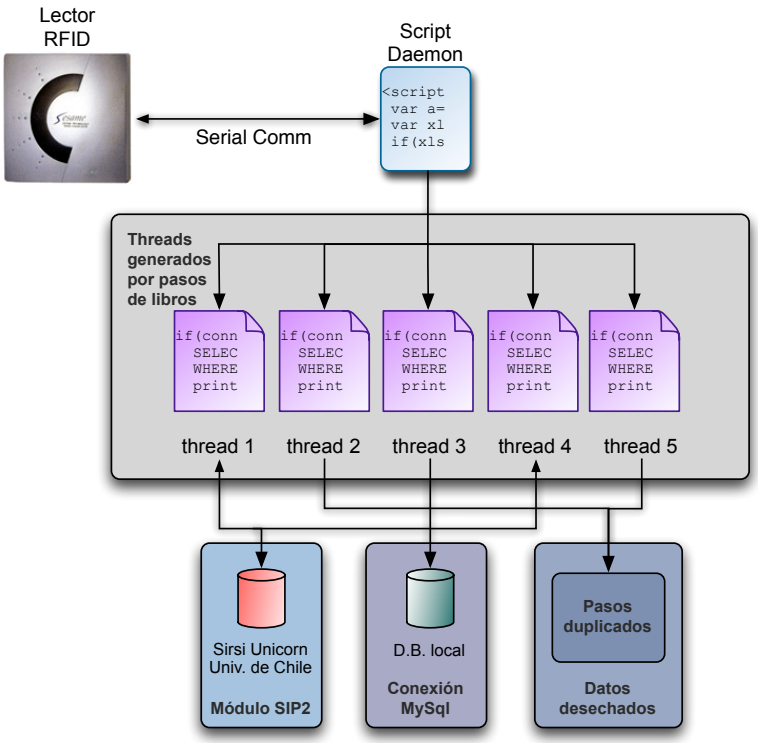

Figura 6.3: Modelo de servidor conectado a un lector RFID

analiza la validez de la información. Si no es válido, la información es completamente desechada y se continúa a la espera de un paso válido. Si el código que pasa es validado, el código recibido es derivado a un thread, para poder liberar el proceso y poder capturar otros TAGs. Si este paso no se realiza, se corre el riesgo de que el proceso central quede esperando alguna respuesta de otros procesos, como conexiones a base de datos, momento en el cual se pueden perder otros TAGs que pasen por el portal hasta que llegue un Timeout, o el sistema completo quede congelado a la espera de una respuesta que nunca llegue.

Revisión de los datos recibidos

La primera tarea del thread recién abierto (fig. **??**) , es revisar si el código capturado por el portal de paso ya se encuentra en la Base de Datos.

Este paso es realizado por dos razones:

- Para reducir la cantidad de conexiones hacia Unicorn, las cuales son mas lentas que las conexiones a la base de datos presente en servidores locales.
- El segundo punto, y el más importante, es para revisar si el dato es una captura "falsa", lo cual explicaremos a continuación.

6.3.2. Eliminación de duplicados

En este punto se revisa si el item que ha sido encontrado en la base de datos local, no sea “mas joven” que 10 segundos, puesto que a veces, el pórtico puede leer más de una vez el mismo libro mientras se encuentra pasando por este. Así, cada thread abierto tratará de almacenar el dato, pero con esta revisión se evita que se produzcan datos inflados producto de la "sobre sensibildad"del pórtico.Al recibir un dato que ya ha sido almacenado dentro de esos 10 segundos, el dato es desechado, y el thread muere.

Captura de Datos Unicorn

Si el código buscado no se encontraba en la base de datos local – que implica que el libro está pasando por primera vez a través del pórtico – se debe recuperar la información haciendo llamadas a Unicorn. Estas conexiones son lentas, por lo que durante la implementación se decidió reducir su cantidad de conexiones. Es por esto, que se ideó este esquema de recuperar la información sólo la primera vez, y el resto de las veces, contar con la información ya guardada en la base de datos local. Si el código retorna información no válida –es decir, no se encuentra el nombre del volúmen–, la información es desechada y el thread muere. Si la información es válida,se procede al ingreso de los datos a la base de datos local.

Almacenamiento de los datos

Si los datos capturados por el lector logran sortear todos las revisiones explicadas anteriormente, es posible almacenarlos. Así, se ingresa en la base de datos local :

- El código y el Unix Timestamp de paso del volúmen, en el caso que la información del volúmen ya exista en la base de datos,
- O el código de volumen, la información recuperada desde Unicorn y el Unix Timestamp si ha sido necesario rescatar información desde Unicorn.

6.3.3. El modelo de Base de Datos

El modelo de almacén de los datos, se trató de mantener lo más simple posible, creando una base de datos ligera, para capturar todos los datos en caso que circulen muchos volúmenes en un pequeño período de tiempo (Fig. **??**).

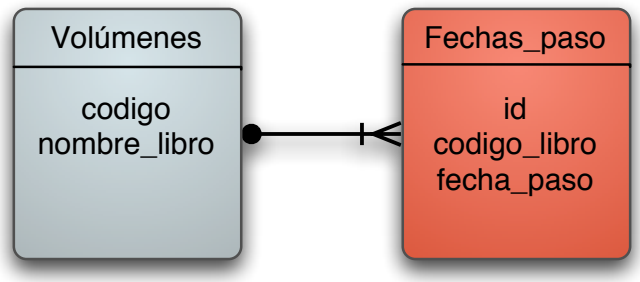

Figura 6.4: Modelo Entidad Relación

Se crearon sólo 2 tablas, una en la que se almacena la información relativa al volúmen – el código del libro y su nombre–, extraída desde Unicorn, y otra tabla que mantiene un índice de pasos de cada código de libro con su Unix Timestamp. Con esto es posible rastrear los libros con su respectiva fecha de paso, y lograr un despliegue coherente de estos. Además, está abierta la estructura para la generación de nuevos campos y tablas, al momento de agregarle mayor funcionalidad a las nuevas vistas generadas por la interfaz, como a las mejoras en la funcionalidad de los equipos.

6.3.4. Interfaz de revisión de datos

A nivel de la interfaz de usuario para la revisión de datos, la decisión de desarrollo está dada por pensar en métodos en que el personal de biblioteca interactúe con el sistema mismo, de manera simple y sin ambiguedades.

Toda la información debe ser desplegada de un modo simple, preciso y que permita nuevas funcionalidades con los datos en caso que sea necesario. Por todas estas razones, se decidió que la plataforma de revisión de historiales de paso fuese desarrollado en interfaz Web, específicamente, en una plataforma de desarrollo que esté basado en el diseño MVC (**M**odelo - **V**ista - **C**ontrolador).

En un sistema de MVC podemos mantener separados los modelos de datos de la interfaz de usuarios, para que los cambios en la interfaz de usuarios no afecten el manejo de los datos, y los datos puedan ser reorganizados sin modificar en demasía la interfaz de usuario. Con este modelo podemos separar los accesos de datos y la lógica del control de la presentación de los datos y su interacción con los usuarios, introduciendo un componente intermedio, que es el controlador (Fig. **??**).

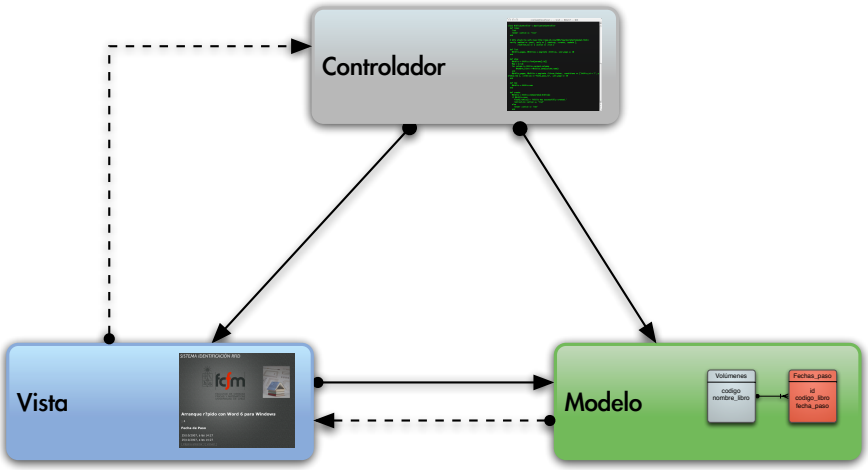

Figura 6.5: Modelo-Vista-Controlador

Por este motivo, se decidió realizar el desarrollo en Ruby on Rails, por ser la plataforma más popular entre la oferta de plataformas MVC's basados en Web de hoy.

Ruby on Rails promete aumentar la velocidad y facilidad en la creación de sitios web basados en Bases de Datos, entregando esqueletos previamente armados (scaffolding) para la creación de nuevos sitios. (Fig. **??**).

Figura 6.6: Página de inicio del sistema de monitoreo de volúmenes -Ruby on Rails-

La idea fundamental en la que Ruby on Rails se basa es en ordenar la información en solo un lugar, sin ambiguedades. Con esta idea en mente fueron desarrolladas las tablas de datos en el sistema, de modo que por convención de los nombres dentro de la aplicación misma se definan los nombres de las bases de datos, sin necesidad de definir configuraciones.

El sistema permite desplegar de manera eficaz los datos que se van almacenando en el sistema, desde el módulo de captura.

La idea central de esta área del sistema es en ser una herramienta de despliegue de los datos recibidos automáticamente, por lo que no se debería permitir agregar ni modificar datos desde este medio. Sin embargo, la posibilidad de hacer esto está habilitada, sólo para referencias prácticas del desarrollo (Fig. **??**).

El despliegue de la información de pasos de un volúmen en particular por el portal se puede ver al hacer click en el botón "Listar".

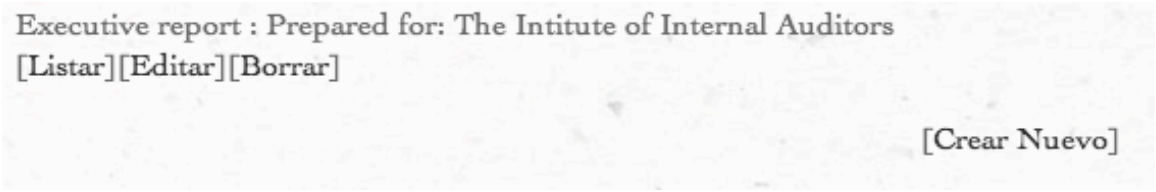

Figura 6.7: Detalle de opciones de un volúmen en particular

Allí se pueden ver los movimentos que ha tenido este libro a través del portal de circulación, ordenados por fecha y paginados de forma tal que no sea una molestia para la revisión de los datos en pantalla (Fig. **??**).

Figura 6.8: Página que lista todos los pasos por el portal de un libro en particular.

Otra vista de los datos, es poder revisar el paso de todos los libros que han pasado por el portal. En esta ventana es posible revisar el código de cada volumen, y su fecha de paso, desde el paso más reciente hasta el más antiguo, también con un orden de paginación similar al punto anterior (Fig. **??**).

Con esto se logra generar una interfaz visual que permite realizar a los encargados de biblioteca las operaciones más importantes para el control de circulación frente del portal RFID .

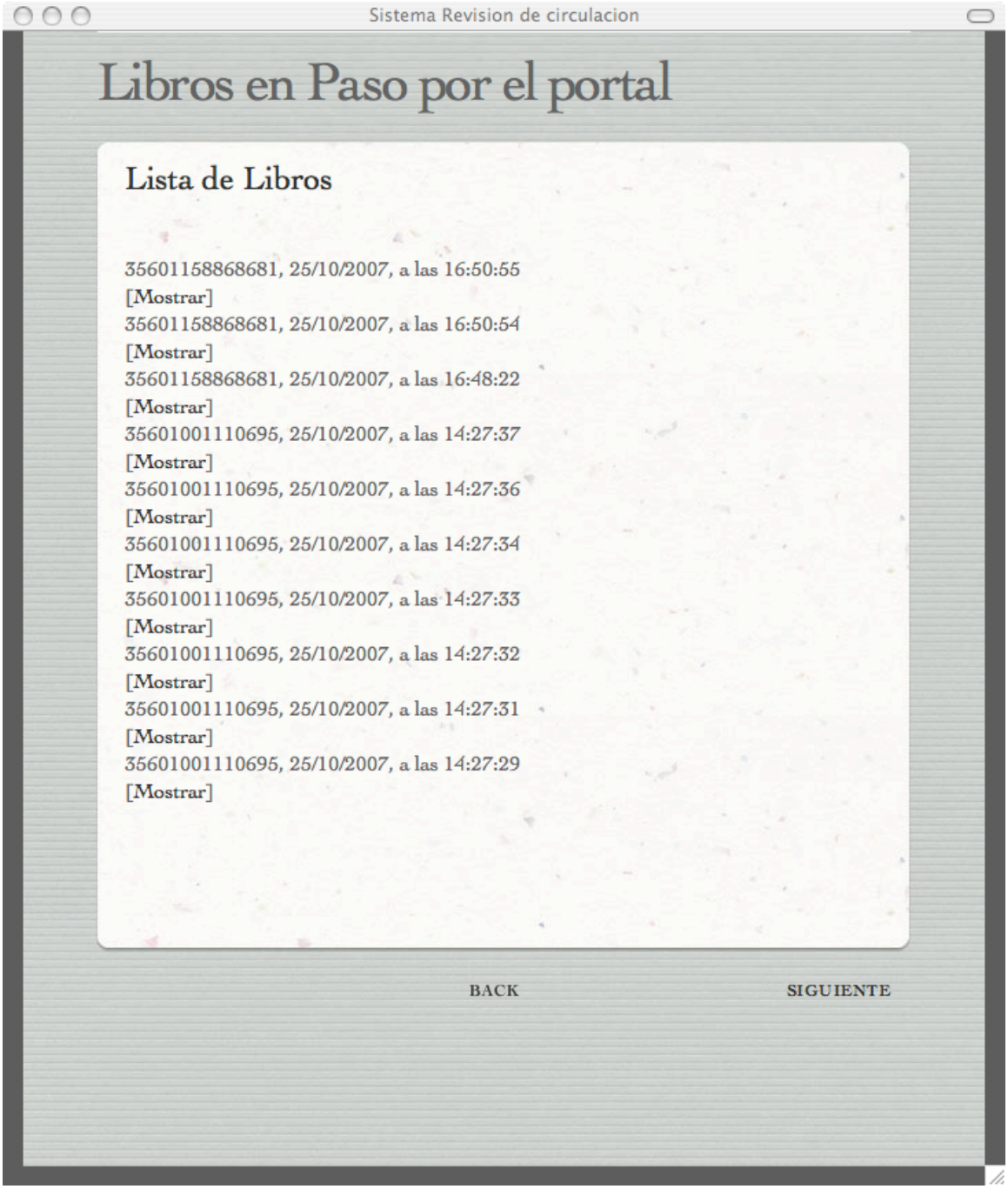

Figura 6.9: Página que lista todos los pasos por el portal.

Capítulo 7

Conclusiones

A través del presente trabajo se desarrolló un sistema capaz de mantener la información de circulación de volúmenes desde la colección general de la Biblioteca, basados en modelos que puedan ser extensibles a nuevas aplicaciones.

Debido a esto, se optó por el diseño de una arquitectura de trabajo capaz de integrar nuevas aplicaciones, extendiendo el alcance de lo desarrollado para aplicaciones futuras, tanto para el mismo ambiente RFID en biblioteca como para otros ambientes donde RFID sea un componente para la integración. La forma de desarrollo se hizo pensando en todos los desarrollos bajo estándares de software libre, que dan la posibilidad a terceros de ejecutar, distribuir, estudiar, cambiar o mejorar lo realizado por esta memoria.

Con esta memoria se logra identificar individualmente el paso de volúmenes que circulan a través del portal RFID en la zona de libros de la colección general registrando la fecha y hora en la cual el libro pasó por el portal, almacenando esta información en un servidor dedicado para ello.

Como trabajo futuro de esta implementación, está el desarrollar mejores herramientas de gestión misma de los datos, como estadísticas complejas, modelos de predicción de uso de recursos y manejos de tiempos de volúmenes fuera de la zona de colección general.

La validación del funcionamiento sistema en producción aún no ha sido posible, puesto que el portal aún no ha sido colocado en su sitio porque hay que realizar toda una modificación en estructura física de la entrada a la zona de colección general. Por esto aún no es posible evaluar si el sistema necesita nuevos ajustes en el modelo o en la plataforma utilizada.

A pesar de lo anterior, es posible pensar que tanto Mysql como los módulos de Ruby y Perl están diseñados de forma que no sea problema el manejar una gran cantidad de consultas concurrentes o expandir el modelo a nuevas funcionalidades, teniendo en consideración que es uno de los objetivos principales de esta memoria.

Así, si bien esta memoria abarca una parte de la funcionalidad de todo el potencial de RFID y su aplicabilidad, se sientan bases para poder pensar en nuevos desarrollos futuros, siendo posible extender la funcionalidad de la arquitectura para ello. En este sentido es posible pensar como desarrollos que pueden extender de manera simple esta arquitectura, el de un sistema de devolución de libros del tipo "Caja Buzón", que pueda validar automáticamente la devolución en el sistema de control bibliotecario, entregando un recibo de la devolución realizada. Otra aplicación a desarrollar podría ser un sistema de monitoreo usando múltiples portales, en los cuales es posible rastrear la posición y el sentido de movimiento de los libros por toda la biblioteca.

Otra área de desarrollo, que si bien no está en el alcance del sistema mismo, pero es un punto necesario a desarrollar, es el módulo de comunicación con los lectores de Ceyon mediante una conexión *telnet*.

La nueva generación de lectores poseen dos puertas de comunicación, la puerta serial estándar y una puerta telnet. Ambas puertas hablan el mismo protocolo, pero la ventaja que da la comunicación vía *telnet* es que se puede consultar remotamente, y no es necesario tener un computador conectado al equipo. Con esto, es posible generar implementaciones de mayor escala, posibilitando la centralizacion de las consultas RFID. Como los lectores usados en biblioteca sólo poseen puerta serial, no fue posible realizar los módulos *telnet*, pero si se implementa este tipo de comunicación, es posible escalar la solución completa de manejo de RFID.

Bibliografía

[1] 3M. Información sistemas rfid bibliotecarios 3m. In *http://www.3m.com/uk/library/*.

[2] 3M. 3m standard interchange protocol v2.00. In *3M^{TM} SelfCheck^{TM} system SIP2 Interface Developer's Guide*, 1999.

[3] Jürgen Bohn and Friedemann Mattern. Super-Distributed RFID Tag Infrastructures. In Panos Markopoulos, Berry Eggen, Emile Aarts, and James Crowley, editors, *Proc. 2nd European Symposium on Ambient Intelligence (EUSAI 2004)*, number 3295 in LNCS, pages 1–12, Eindhoven, The Netherlands, November 2004. Springer-Verlag.

[4] Sirsi Dynix. Página web. In *http://www.sirsidynix.com/*.

[5] Christian Floerkemeier. *Infrastructure Support for RFID Systems*. PhD thesis, ETH Zurich, Zurich, Switzerland, July 2006.

[6] Christian Floerkemeier. Transmission control scheme for fast rfid object identification. In *Second IEEE PerCom Workshop on Pervasive Wireless Networking (PWN06)*, March 2006.

[7] Dave Harrold. Comparing enclosure standards. In *http://www.controleng.com/article/CA188226.html?pubdate=3 %2F1 %2F1999*, 1999.

[8] AIM Inc. Rfid standards. In *http://www.aimglobal.org/standards/rfidstds/RFIDStandard.asp*, 2000-2001.

[9] Ceyon Technology Inc. Página web principal. In *http://www.ceyon.net/*.

[10] National Information Standards Organization (NISO). American libraries association, comité para estándares rfid. In *http://www.niso.org/committees/RFID/RFID_comm.html*.

[11] Ted Phillips, Tom Karygiannis, and Rick Huhn. Security standards for the rfid market. In *IEEE Security and Privacy*, volume 03, pages 85–89, Nov/Dec 2005.

[12] Sanjay E. Sarma. Towards the five-cent tag. In *Technical Report MIT-AUTOID-WH-006 MIT Auto-ID Center*, 2001.

[13] SISIB. Información general de biblioteca. In *http://www.sisib.uchile.cl/*.

[14] Ceyon Technology. Ceyon reader access protocols, version 2.0. October 2006.

[15] Sentry Technology. Página web principal. In *http://www.sentrytechnology.com/*.

[16] J.R. Tuttle. Traditional and emerging technologies and applications in the radio frequency identification (rfid) industry. In *IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, page 5–8, June 1997.